Project Documentation: WhatNext Vision Motors

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

Project Lead: Gabriel Ian C Mustera

I. Project Overview

WhatNext Vision Motors, a pioneering force in the automotive industry, is dedicated to transforming the mobility sector with innovative technology and solutions that prioritize customer needs. The company has embarked on an ambitious **Salesforce CRM project** with the core objective of enhancing the **customer experience** and streamlining its **operational processes**.

This CRM implementation focuses on creating a robust system for managing key assets and relationships, including **Vehicles, Dealers, Customers, Orders, Test Drives, and Service Requests.** The key features implemented address critical business needs: automated assignment of the nearest dealer to a customer, proactive stock validation to prevent out-of-stock orders, and scheduled updates for bulk order statuses to maintain data accuracy.

II. Objectives

The primary goals for building and implementing this Salesforce CRM solution are clearly defined and linked to tangible business value:

- **Enhanced Customer Management:** To provide a seamless and convenient ordering experience by **automatically assigning the nearest dealer** based on the customer's location, thereby reducing friction and improving service efficiency.
- **Streamlined Bookings and Order Accuracy:** To enforce business rules that **prevent the creation of orders for out-of-stock vehicles**, ensuring that order fulfillment expectations are realistic and minimizing customer dissatisfaction.
- **Improved Operational Efficiency (Backend):** To leverage **Batch Apex** for a scheduled process that accurately updates the status of bulk order records based on current stock availability, reducing manual administrative tasks for staff.
- **Effective Communication:** To use **Record-Triggered Flows** to send automated email reminders for scheduled test drives, increasing attendance rates and improving customer engagement.

III. Phase 1: Requirement Analysis&Planning

1. Understanding Business Requirements

The project was necessitated by the need to digitally transform several core operations:

- **Inventory and Dealer Management:** Centralized storage of vehicle details, stock availability, and dealer information.
- **Process Automation:** Automating the complex logic of dealer assignment based on proximity and enforcing real-time stock checks.
- **Scheduled Processing:** Implementing a reliable mechanism to update order statuses in bulk, outside of real-time transactions.
- **Customer Lifecycle Tracking:** Efficiently tracking and managing customer interactions from orders to test drives and service requests.

2. Defining Project Scope and Objectives

The project scope covered the **Salesforce CRM Implementation** across five key areas: Data Modelling, Customization & Configuration, Automation (Flows/Apex), Reporting, and Security, focusing specifically on the objects listed in the data model.

3. Design Data Model and Security Model

3.1. Data Model (Objects&Relationships)

The solution is built around several custom objects and their defined relationships to support the vehicle sales and service lifecycle:

Object Name	Purpose	Key Relationships
Vehicle__c	Stores vehicle details and stock availability.	Related to Orders&Test Drives.
Vehicle_Dealer__c	Stores authorized dealer information.	Related to Orders.
Vehicle_Customer__c	Stores customer details.	Related to Orders, Test Drives,&Service Requests.
Vehicle_Order__c	Tracks vehicle purchases.	Related to Customer&Vehicle.
Vehicle_Test_Drive__c	Tracks test drive bookings.	Related to Customer&Vehicle.

Vehicle_Service_Request__c	Tracks vehicle servicing requests.	Related to Customer&Vehicle.

4. Stakeholders Mapping

- **Project Sponsor:** Provided overall vision and budget approval.
- **Sales Team:** Primary users of the CRM, focused on order creation and management.
- **Dealership Managers:** Users responsible for managing assigned orders and inventory.
- **System Administrator/Developer:** Responsible for implementation, maintenance, and monitoring.
- **End Customers:** Beneficiaries of the streamlined ordering and communication processes.

5. Execution RoadMap

The project followed a standard iterative approach: Requirements Data Model Configuration Automation (Flows&Apex) Testing Documentation Deployment.

IV. Phase 2: Salesforce Development - Backend&Configurations

1. Customization of Objects and Fields

Custom fields were added to the defined objects to meet specific business requirements (e.g., custom picklists, lookups, and currency fields).

Object	Key Fields Implemented (Examples)
Vehicle__c	Stock_Quantity__c (Number), Price__c (Currency), Dealer__c (Lookup)
Vehicle_Order__c	Status__c (Picklist: Pending, Confirmed, Cancelled), Vehicle__c (Lookup), Customer__c (Lookup)
Vehicle_Test_Drive__c	Test_Drive_Date__c (Date), Status__c (Picklist: Scheduled, Completed)

2. Automation: Record-Triggered Flows

2.1. Flow 1: Auto-Assignment of Nearest Dealer

- **Description:** A **Record-Triggered Flow** is initiated when a new Vehicle_Order__c record is created and its status is 'Pending'. The flow queries the Vehicle_Dealer__c records to identify and assign the nearest dealer to the new order, automating a critical logistics step.
- **Object:** Vehicle_Order__c
- **Trigger:** A record is created.
- **Entry Condition:** All Conditions Are Met (AND) | Status__c | Equals | Pending
- **Action:** Uses a **Get Records** element and subsequent logic to update the order's Dealer__c field.

2.2. Flow 2: Automated Test Drive Reminder Email

- **Description:** A **Record-Triggered Flow** utilizes a **Scheduled Path** to send an automated email to the customer as a reminder for their upcoming test drive appointment.
- **Object:** Vehicle_Test_Drive__c
- **Trigger:** A record is created or updated.
- **Entry Condition:** All Conditions Are Met (AND) | Status__c | Equals | Scheduled
- **Action: Scheduled Path** is set to run **1 Day Before** the Test_Drive_Date__c. The action is to send an email alert to the associated customer.

3. Apex Classes and Triggers

3.1. Apex Trigger: Stock Validation

- **Description:** An Apex Trigger is implemented on the Vehicle_Order__c object to enforce the business rule that prevents an order from being saved if the selected vehicle is currently **out of stock** (Stock_Quantity__c on Vehicle__c is zero or less). This is done using a **Trigger Handler** named VehicleOrderTriggerHandler for best practices (modularity).
- **Class Name:** VehicleOrderTriggerHandler
- **Purpose:** To validate the stock level of the associated vehicle **before insert** of a new order record, throwing a custom validation error if stock is insufficient.

3.2. Batch Apex Job: Bulk Order Status Update

- **Description:** A **Scheduled Batch Apex** job is developed to periodically check vehicle stock levels and update the Status__c field on all outstanding

Vehicle_Order__c records. If the vehicle is now in stock, the order status is updated from 'Pending' to 'Confirmed'. This ensures the system reflects accurate order fulfillment status daily/periodically.

V. Phase 3: UI/UX Development&Customization

1. Lightning App Setup

- **App Name:** WhatNext Vision Motors
- **Description:** Vision Motors: Shaping the future of mobility with innovation and excellence.
- **Navigation Items:** The relevant custom tabs were added to the navigation bar, including **Vehicle, Dealer, Customer, Vehicle Orders, Test Drives, Reports,** and **Dashboards**.

2. Object Tab Creation

A custom tab was created for each custom object, allowing users easy access and navigation to the respective record home pages.

- **Example:** The **Vehicle** custom tab was created for the Vehicle__c object.

3. Page Layouts and Dynamic Forms

Page layouts for critical objects like Vehicle_Order__c were customized to prioritize key information for sales reps. **Dynamic Forms** were implemented where possible to display fields and sections conditionally, enhancing user experience and data entry efficiency.

4. User Management

Default user access was granted by setting the Lightning App to the **System Administrator** profile and the custom **Sales User** profile.

VI. Phase 4: Data Migration, Testing&Security

1. Data Loading Process

Initial data for Vehicle__c (inventory) and Vehicle_Dealer__c (locations) was loaded using the **Salesforce Data Loader**. This was chosen for its ability to handle larger volumes of initial master data and simplify the mapping of external IDs.

2. Security Configuration

- **Profiles and Permission Sets:** Custom profiles were created/cloned to define base CRUD (Create, Read, Update, Delete) access for objects. Permission Sets were used for granular access rights (e.g., granting the ability to run the Batch Apex job to specific power users).
- **Role Hierarchy:** A Role Hierarchy was established (e.g., CEO Sales Director Sales Manager Sales Rep) to ensure data visibility rules were enforced, allowing managers to view subordinates' records (**Data Sharing**).

3. Creation of Test Classes

Dedicated **Apex Test Classes** were written for the VehicleOrderTriggerHandler and the Batch Apex implementation to guarantee **100% code coverage** for the custom business logic and to prevent regressions during future deployments.

4. Preparation of Test Cases (Mandatory)

The following test cases validate the core automated and custom features of the implementation:

Test Case 1: Stock Validation (Apex Trigger)

Feature	Scenario	Input	Expected Output	Screenshot Status
Apex Trigger (Stock Check)	Attempt to create an order for a vehicle with **0** Stock_Quantity__c.	New Vehicle_Order__c created, linked to Vehicle__c where Stock = **0**.	Order creation fails. Apex displays a **validation error message** preventing the insert.	*Screenshot of Error Message Mandatory*

Test Case 2: Auto-Assignment of Nearest Dealer (Flow)

Feature	Scenario	Input	Expected Output	Screenshot Status
Record-Triggered Flow (Dealer Assign)	Create a new order with a customer address	New Vehicle_Order__c created with Status__c	The flow runs successfully, and the Dealer__c lookup field on the Order record	*Screenshot of Order Record with Auto-Populated*

	(implicitly used for proximity) and set Status__c to **Pending**.	= Pending and Customer Lookup populated.	is automatically populated with the **ID of the nearest dealer**.	*Dealer Mandatory*

Test Case 3: Test Drive Reminder (Scheduled Flow Path)

Feature	Scenario	Input	Expected Output	Screenshot Status
Scheduled Flow Path	Create a Vehicle_Test_Drive__c record scheduled for tomorrow.	New Vehicle_Test_Drive__c record created with Status__c = Scheduled and a Test_Drive_Date__c set for +1 day.	The flow creates a **Scheduled Path Interview**. After 24 hours (1 Day Before), an **automated email reminder** is sent to the customer.	*Screenshot of Scheduled Path Interview Log Mandatory*

VII. Phase 5: Deployment, Documentation&Maintenance

1. Deployment Strategy

The project deployment from the sandbox environment to production utilized **Change Sets**. This method was chosen for its simplicity and built-in dependency tracking, ensuring all configuration changes, custom objects, fields, flows, and Apex code were migrated correctly and in the proper sequence.

2. Basic Maintenance and Monitoring

- **Automated Process Monitoring:** The scheduled frequency and execution of the **Batch Apex Job** will be monitored via the **Apex Jobs** and **Scheduled Jobs** pages in Setup.
- **Flow Monitoring:** Flow interview errors will be monitored proactively using the **Flow Interview Logs** to identify and fix any runtime issues.
- **System Integrity:** Periodic checks will be conducted to ensure that data integrity, especially stock levels and order statuses, remains accurate.

3. Documentation of Troubleshooting Approach

The primary troubleshooting approach follows this sequence:

1. **UI/UX Errors (Validation/Formula):** Check Validation Rules and Formula Fields.

2. **Automation Errors (Flows):** Check Flow Interview Logs for the specific record ID.
3. **Code/Trigger Errors (Apex):** Check Debug Logs for exceptions and system failures.
4. **Data Issues:** Validate object relationships and data migration integrity.

VIII. Conclusion

The successful implementation of the Salesforce CRM for WhatNext Vision Motors has delivered a modern, automated platform that directly addresses key business challenges. By automating dealer assignment, enforcing stock validation via Apex, and streamlining status updates with Batch Apex, the project has achieved its objectives of enhancing the customer experience and driving operational excellence. This platform is now a solid foundation for the company's future growth and innovation.

IX. Future Enhancements

The following enhancements are recommended for the next project phase to further leverage the Salesforce platform:

- **LWC Development:** Implement a custom **Lightning Web Component (LWC)** to display a dynamic, map-based interface for visualizing nearest dealer locations on the Order page.
- **AI Integration:** Implement **Einstein Prediction Builder** to forecast vehicle sales demand based on historical order data, assisting dealers with stock replenishment.
- **External Integration:** Integrate with the company's external ERP system to achieve real-time synchronization of **Inventory Stock Levels**, ensuring the most accurate data for the Apex Trigger validation.