**Phase 5: Apex Programming (Developer)**

In this phase, **WhatNext Vision Motos** leverages the power of **Apex** to implement complex business logic, automate backend operations, and enable scalable processes beyond point-and-click configuration. Apex ensures the system is **robust, secure, and tailored** to the automotive and mobility business model.

**Classes & Objects**

* Apex classes represent business logic such as *VehicleServiceManager* or *SubscriptionHandler*.
* Objects (standard & custom) are accessed programmatically to perform CRUD operations on entities like *Vehicle*, *Service Request*, and *Mobility Subscription*.

**Apex Triggers (before/after insert/update/delete)**

* **Before Insert/Update**: Validate *Vehicle VIN* uniqueness before saving.
* **After Insert**: Automatically create *Service Schedule Records* once a new vehicle is registered.
* **After Delete**: Notify account manager when a *Mobility Subscription* is canceled.

**Trigger Design Pattern**

* Centralized trigger framework to avoid recursion and bulk issues.
* Example: A **Handler Class** processes logic for *Service Request Trigger*, ensuring scalability and clean code structure.

**SOQL & SOSL**

* **SOQL**: Fetch vehicle service history (SELECT Id, Status FROM Service\_Request\_\_c WHERE Vehicle\_\_c = :vehicleId).
* **SOSL**: Search across objects for customer inquiries like VIN, phone, or email.

**Collections: List, Set, Map**

* **List**: Store multiple *Service Requests* for batch processing.
* **Set**: Ensure no duplicate *VIN Numbers*.
* **Map**: Link *Account IDs* to corresponding *Vehicles* for faster lookups.

**Control Statements**

* Conditional logic (if-else, switch) to determine service priority.
* Looping structures to process bulk data (e.g., mass updating subscription renewals).

**Batch Apex**

* Processing thousands of records such as *Service History Archival* or *Vehicle Sensor Data Processing*.
* Example: Monthly cleanup of inactive subscription records.

**Queueable Apex**

* Chaining jobs for complex mobility analytics.
* Example: Run predictive analytics on vehicle usage data, then notify customers.

**Scheduled Apex**

* Automating recurring jobs like sending *Subscription Renewal Reminders* or *Monthly Service Reports*.
* Example: Run every 1st of the month at midnight.

**Future Methods**

* For lightweight async processes such as making callouts to IoT systems for *Vehicle Health Monitoring*.
* Example: Send real-time mileage data to external telematics API.

**Exception Handling**

* Try-Catch blocks ensure smooth handling of failures (e.g., missing service record).
* Custom exception classes log issues in *Error Log Object* for proactive fixes.

**Test Classes**

* At least **75% code coverage** to ensure deployment readiness.
* Validate business logic like warranty checks, subscription renewals, and trigger workflows.

**Asynchronous Processing**

* Efficient handling of large volumes of data from connected vehicles.
* Example: Async processing of *IoT sensor logs* for thousands of cars simultaneously.