Salesforce Project Implementation Phases with Concepts (Admin + Developer)

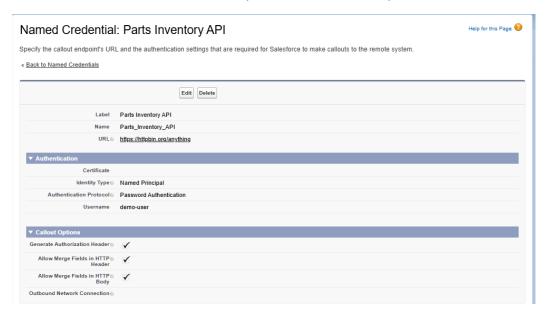## Project Title: Manufacturing After-Sales & Service CRM

Phase 7: Integration & External Access

This phase focuses on connecting the Salesforce CRM with external systems and services to enhance its capabilities and enable seamless data exchange. This includes configuring secure API integrations, enabling real-time event-driven communication, and making callouts to external web services.

Named Credentials

Named Credentials are used to securely store and manage the authentication and endpoint information for external services. This approach simplifies the management of authenticated callouts by separating the endpoint URL and authentication details from the Apex code.
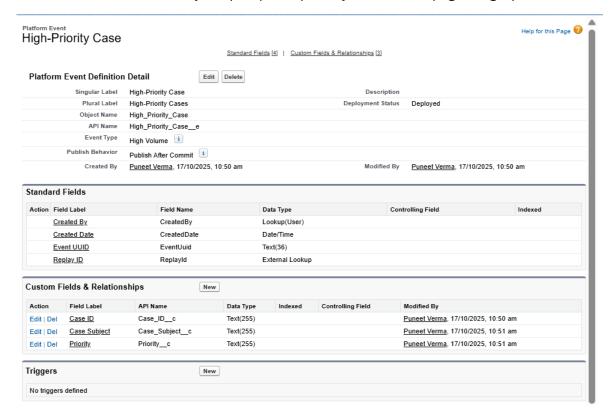
- **Use Case:** To ensure that API callouts from Apex are secure, maintainable, and easy to manage, a Named Credential is used. It separates the endpoint URL and authentication details from the code, eliminating the need for hardcoded URLs or credentials and simplifying administration. For this project, a Named Credential for a mock API service is configured to facilitate development and testing of integration features without depending on a live external system.
- **Implementation Details:**
    - **Label:** Parts Inventory API
    - **Name:** Parts_Inventory_API
    - **URL:** https://api.parts-inventory.example.com/v1
    - **Authentication:** No Auth (for the mock service)

Platform Events

Platform Events are used to enable real-time communication between Salesforce and external systems in an event-driven architecture. This allows for the decoupling of systems and enables asynchronous processing of events.

- **Use Case:** To enable real-time, event-driven communication with external systems, a Platform Event is used. When a critical business event occurs, such as the creation of a high-priority service case, Salesforce can publish an event message. External applications (like a monitoring dashboard or a separate notification service) can subscribe to these events to receive immediate updates and trigger their own processes, creating a decoupled and scalable architecture.
- **Implementation Details:**
    - **Platform Event:** High-Priority Case (High_Priority_Case__e)
    - **Publishing Mechanism:** The event is published using a record-triggered Flow that fires after a Service_Case__c record is created with Priority__c set to 'High'. Alternatively, this logic can be handled in the ServiceCaseTrigger.
    - **Fields:**
        - Case_ID__c (Text): The record ID of the service case.
        - Case_Subject__c (Text): The subject line of the case.
        - Priority__c (Text): The priority of the case (e.g., "High").

Platform Event
High-Priority Case

Help for this Page

Standard Fields [4] | Custom Fields & Relationships [3]

Platform Event Definition Detail   Edit   Delete

Singular Label	High-Priority Case	Description	
Plural Label	High-Priority Cases	Deployment Status	Deployed
Object Name	High_Priority_Case		
API Name	High_Priority_Case__e		
Event Type	High Volume ⓘ		
Publish Behavior	Publish After Commit ⓘ		
Created By	Puneet Verma, 17/10/2025, 10:50 am	Modified By	Puneet Verma, 17/10/2025, 10:50 am

Standard Fields

Action	Field Label	Field Name	Data Type	Controlling Field	Indexed
	Created By	CreatedBy	Lookup(User)		
	Created Date	CreatedDate	Date/Time		
	Event UUID	EventUuid	Text(36)		
	Replay ID	ReplayId	External Lookup		

Custom Fields & Relationships   New

Action	Field Label	API Name	Data Type	Indexed	Controlling Field	Modified By
Edit | Del	Case ID	Case_ID__c	Text(255)			Puneet Verma, 17/10/2025, 10:50 am
Edit | Del	Case Subject	Case_Subject__c	Text(255)			Puneet Verma, 17/10/2025, 10:51 am
Edit | Del	Priority	Priority__c	Text(255)			Puneet Verma, 17/10/2025, 10:51 am

Triggers   New

No triggers defined

Experience Cloud for External Access

While not a traditional integration, the Experience Cloud (formerly Community Cloud) site provides external access to customers, allowing them to interact with the CRM.

- **Use Case:** The customer portal, built with Experience Cloud, allows customers to register products, create and view their service cases, and submit feedback. This provides a self-service channel for customers and reduces the workload on service agents.
- **Implementation Details:**
    - A Customer Community Login license is used for external users.
    - The "Customer Service" template is used as the base for the portal.
    - The Customer Profile is configured with appropriate permissions, and Org-Wide Defaults are set to "Private" for external access to ensure customers can only see their own records.