

# VendorConnect – Phase 5: Apex Programming Documentation

---

Project: VendorConnect – Vendor & Purchase Order Management

Phase: 5 – Apex Programming

Author: [Your Name]

Date: [Submission Date]

## Objective

Handle complex logic using Apex in Salesforce.

Automate tasks such as:

- Prevent overlapping active POs for the same vendor
- Calculate vendor performance reports
- Track unique vendors
- Recalculate vendor ratings
- Auto-close expired POs

## Components Implemented

Component	Description	Files / Classes
Trigger on Purchase_Order__c	Prevent overlapping active POs	PurchaseOrderTrigger.trigger, PurchaseOrderTriggerHandler.cls
Apex Class – VendorPerformanceService	Calculates vendor performance, tracks unique vendors	VendorPerformanceService.cls
Collections	Track unique vendors using Set and Map	Implemented in VendorPerformanceService
SOQL & SOSL	Fetch completed PO counts, total amounts	Aggregate SOQL in VendorPerformanceService
Exception Handling	Prevent runtime errors in Apex logic	try-catch blocks in triggers/classes
Test Classes	Ensure 75%+ coverage	TestVendorPerformanceService.cls, TestPurchaseOrderTrigger.cls

## Implementation Steps

### Step 1: Trigger – Prevent Overlapping POs

Objective: Ensure a vendor does not have multiple active POs in Submitted or Approved status.

Logic:

1. Trigger fires before insert and before update on Purchase\_Order\_\_c.
2. Collect vendor IDs from incoming POs.
3. Query existing active POs for those vendors.
4. If overlapping POs found, throw an error.

Screenshot: Trigger setup in Salesforce and Developer Console code view.

## Step 2: Vendor Performance – SOQL & SOSL

Objective: Fetch vendor performance for completed POs.

Example SOQL:

```
List<AggregateResult> vendorPerf = [  
    SELECT Vendor__c, COUNT(Id) totalPOs, SUM(Amount__c)  
    totalAmount  
    FROM Purchase_Order__c  
    WHERE Status__c = 'Completed'  
    GROUP BY Vendor__c  
];
```

---

Screenshot: Developer Console Query Editor results and Debug log showing output.

## Step 3: Collections – Track Unique Vendors

Objective: Track vendors with active POs using Set.

Example Code:

```
Set<Id> uniqueVendors = new Set<Id>();  
for(Purchase_Order__c po : [SELECT Vendor__c FROM  
Purchase_Order__c WHERE Status__c='Submitted']) {  
    uniqueVendors.add(po.Vendor__c);  
}  
System.debug('Unique Vendors with active POs: ' + uniqueVendors);
```

---

Screenshot: Debug log showing unique vendor IDs.

## Step 4: Apex Class – VendorPerformanceService

Objective: Combine logic for collections, SOQL, and reporting.

Screenshot: Apex Class code in Developer Console, Execute Anonymous execution, Debug log.

## Step 5: Test Classes

Objective: Ensure ≥75% code coverage.

Screenshot: Test class code, Apex Test Execution page, Code Coverage report, Debug log.

## Step 6: Asynchronous Apex (Optional)

Batch Apex: Recalculate vendor ratings monthly

Queueable Apex: Send bulk notifications

Scheduled Apex: Auto-close expired POs

Screenshot: Scheduled/Batch Apex setup and execution logs.

## Step 7: Exception Handling

Objective: Prevent runtime errors using try-catch blocks.

Screenshot: Code snippet and debug logs showing handled exceptions.

## Deliverables

- Fully working Apex classes & triggers
- Test classes with  $\geq 75\%$  coverage
- Reports for vendor performance and unique active vendors
- Debug logs showing successful execution

## Suggested Screenshots to Include

Section	Screenshot
Trigger	Developer Console / Trigger code view
SOQL Query	Query Editor result
Collections	Debug log showing unique vendor IDs
Apex Class	VendorPerformanceService class code
Execute Anonymous	Execution of trackVendorPerformance()
Test Class	Test class code
Test Execution	Run All Tests page
Code Coverage	Coverage report showing $\geq 75\%$
Debug Logs	Vendor performance calculation output
Asynchronous Apex	Scheduled/Batch Apex setup and job execution

## Screenshots

Setup

Home

Object Manager

apex

▼ Email

▼ Custom Code

▼ Apex Classes

▼ Apex Settings

▼ Apex Test Execution

▼ Apex Test History

▼ Apex Triggers

▼ Environments

▼ Jobs

▼ Apex Flex Queue

▼ Apex Jobs

Didn't find what you're looking for?  
Try using Global Search.

Apex Classes

Apex Code is an object oriented programming language that allows developers to develop on-demand business applications on the Lightning Platform.

Percent of Apex Used: 0.19%

You are currently using 8,718 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Estimate your organization's code coverage

Compile all classes

View: All Create New View

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Other All

Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	CloseExpendPOs		64.0	Active	446	Qm Pote, 9/22/2025, 8:19 AM	<input type="checkbox"/>
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	PurchaseOrderTriggerHandler		64.0	Active	1,044	Qm Pote, 9/22/2025, 7:52 AM	<input type="checkbox"/>
<a href="#">Edit</a>   <a href="#">Del</a>	TestPurchaseOrderTrigger		64.0	Active	393	Qm Pote, 9/22/2025, 8:56 AM	<input type="checkbox"/>
<a href="#">Edit</a>   <a href="#">Del</a>	TestVendorPerformanceService		64.0	Active	1,664	Qm Pote, 9/22/2025, 9:12 AM	<input type="checkbox"/>
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	VendorNotificationQueueable		64.0	Active	368	Qm Pote, 9/22/2025, 8:16 AM	<input type="checkbox"/>
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	VendorPerformanceService		64.0	Active	591	Qm Pote, 9/22/2025, 8:01 AM	<input type="checkbox"/>
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	VendorRatingBatch		64.0	Active	933	Qm Pote, 9/22/2025, 8:15 AM	<input type="checkbox"/>
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	VendorTracker		64.0	Active	682	Qm Pote, 9/22/2025, 8:08 AM	<input type="checkbox"/>
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	VendorUIController		64.0	Active	4,396	Qm Pote, 9/23/2025, 1:49 AM	<input type="checkbox"/>

<https://org1am-3da1f75533-dev-ed.develop.lightning.force.com/lightning/setup/ApexClasses/home>

Setup

Home

Object Manager

apex

▼ Email

▼ Custom Code

▼ Apex Classes

▼ Apex Settings

▼ Apex Test Execution

▼ Apex Test History

▼ Apex Triggers

▼ Environments

▼ Jobs

▼ Apex Flex Queue

▼ Apex Jobs

Didn't find what you're looking for?  
Try using Global Search.

Apex Triggers

This page allows you to view and modify all the triggers in your organization. To create a new trigger, navigate to the appropriate sObject triggers page.

Percent of Apex Used: 0.19%

You are currently using 8,718 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Compile all triggers

View: All Create New View

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Other All

Action	Name	Namespace Prefix	sObject Type	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
<a href="#">Edit</a>   <a href="#">Del</a>	PurchaseOrderTrigger		Purchase_Order	64.0	Active	258	Qm Pote, 9/22/2025, 7:52 AM	<input type="checkbox"/>

<https://org1am-3da1f75533-dev-ed.develop.lightning.force.com/lightning/setup/ApexTriggers/home>