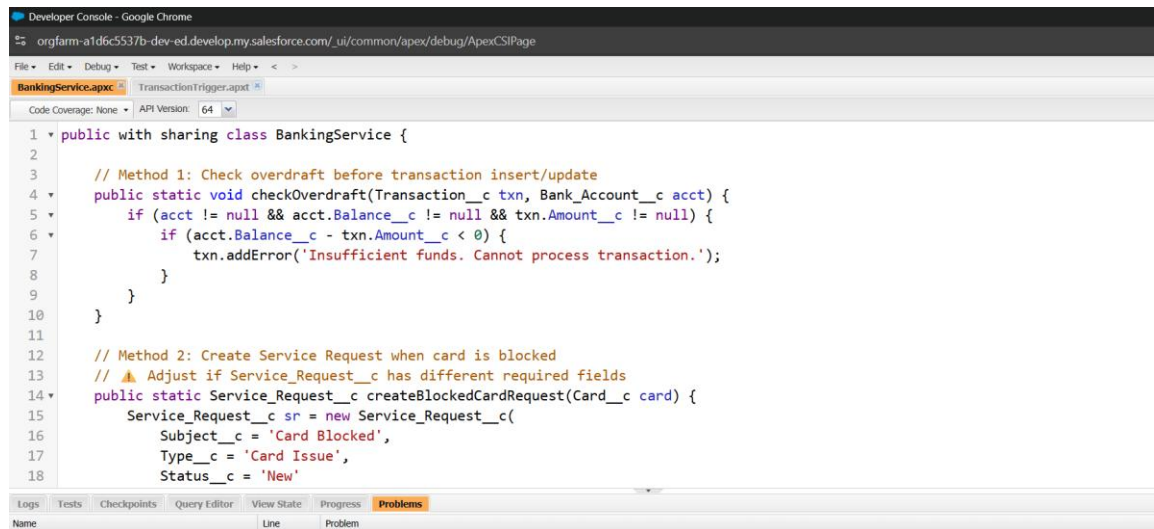


Digital Banking CRM - Apex Programming

This phase implements Apex programming concepts for the Digital Banking CRM project. It includes Apex classes, triggers, batch processes, scheduled jobs, SOQL and SOSL queries, and test classes to ensure automation and reliability.

1. Apex Classes

A utility class `BankingNotificationService` was created to handle automatic alerts. For example, when a `Transaction` fails or a `Loan` crosses a due date, the class inserts `ServiceRequest__c` records or sends alerts to the customer.

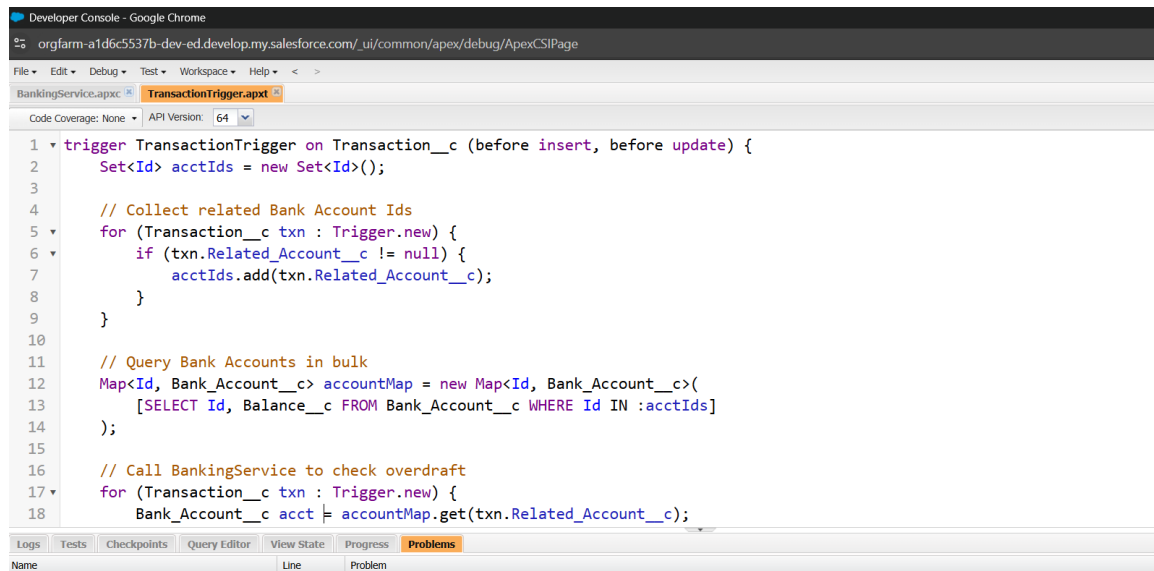


```
1 public with sharing class BankingService {
2
3     // Method 1: Check overdraft before transaction insert/update
4     public static void checkOverdraft(Transaction__c txn, Bank_Account__c acct) {
5         if (acct != null && acct.Balance__c != null && txn.Amount__c != null) {
6             if (acct.Balance__c - txn.Amount__c < 0) {
7                 txn.addError('Insufficient funds. Cannot process transaction.');
```

2. Apex Triggers

Apex Triggers were implemented to enforce business logic:

- Trigger on `Transaction__c` to prevent overdrafts (ensuring `Balance__c` is sufficient).
- Trigger on `Loan__c` to prevent closure of a `Loan` if `Amount__c` > 0.
- Trigger on `Card__c` to auto-create a `ServiceRequest__c` if `Status__c` = 'Blocked'.

The image shows a screenshot of the Salesforce Developer Console. The top bar indicates the browser is Google Chrome and the URL is orgfarm-a1d6c5537b-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The file explorer shows 'BankingService.apex' and 'TransactionTrigger.apex' is selected. The code editor displays the following Apex code:

```
1 trigger TransactionTrigger on Transaction__c (before insert, before update) {
2     Set<Id> acctIds = new Set<Id>();
3
4     // Collect related Bank Account Ids
5     for (Transaction__c txn : Trigger.new) {
6         if (txn.Related_Account__c != null) {
7             acctIds.add(txn.Related_Account__c);
8         }
9     }
10
11     // Query Bank Accounts in bulk
12     Map<Id, Bank_Account__c> accountMap = new Map<Id, Bank_Account__c>{
13         [SELECT Id, Balance__c FROM Bank_Account__c WHERE Id IN :acctIds]
14     };
15
16     // Call BankingService to check overdraft
17     for (Transaction__c txn : Trigger.new) {
18         Bank_Account__c acct = accountMap.get(txn.Related_Account__c);
```

The bottom of the console shows tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. The 'Problems' tab is active, showing a table with columns 'Name', 'Line', and 'Problem', which is currently empty.

3. SOQL and SOSL Queries

SOQL Example: Fetch all active Bank Accounts for a given Contact.

```
SELECT Id, Name, Balance__c FROM BankAccount__c WHERE Customer__c = :contactId AND Status__c = 'Active';
```

SOSL Example: Search for a customer by name across Contacts.

```
List<List<Contact>> results = [FIND 'John*' IN ALL FIELDS RETURNING Contact(Id, Name, Email, Phone)];
System.debug(results);
```

Developer Console - Google Chrome

orgfarm-a1d6c5537b-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

BankingService.apxc TransactionTrigger.apxt Log executeAnonymous @9/29/2025, 12:05:54 PM

Execution Log

Timestamp	Event	Details
12:05:54:000	USER_INFO	[EXTERNAL] 005gL000008NPN gayathritalluri28854@agentforce.com (GMT-07:00) Pacific Daylight Time (America/Los_Angeles) GMT-07:00
12:05:54:000	EXECUTION_ST...	
12:05:54:000	CODE_UNIT_ST...	[EXTERNAL] execute_anonymous_apex
12:05:54:000	VARIABLE_SCO...	[4] accounts List<Bank_Account__c> true false
12:05:54:000	VARIABLE_SCO...	[2] contactId Id false false
12:05:54:001	HEAP_ALLOCATE	[95] Bytes:3
12:05:54:001	HEAP_ALLOCATE	[100] Bytes:152
12:05:54:001	HEAP_ALLOCATE	[417] Bytes:408
12:05:54:001	HEAP_ALLOCATE	[430] Bytes:408
12:05:54:001	HEAP_ALLOCATE	[317] Bytes:6
12:05:54:001	HEAP_ALLOCATE	[EXTERNAL] Bytes:8
12:05:54:001	STATEMENT_EX...	[1]
12:05:54:001	STATEMENT_EX...	[2]
12:05:54:001	HEAP_ALLOCATE	[2] Bytes:18
12:05:54:001	HEAP_ALLOCATE	[2] Bytes:4
12:05:54:001	VARIABLE_ASSI...	[2] contactId "003gL000000DMBi5QAH"
12:05:54:001	STATEMENT_EX...	[4]
12:05:54:001	HEAP_ALLOCATE	[4] Bytes:104
12:05:54:001	HEAP_ALLOCATE	[4] Bytes:4

☐ This Frame
☐ Executable
☐ Debug Only
☐ Filter

Logs Tests Checkpoints **Query Editor** View State Progress Problems

Enter SOQL or SOSL query: SELECT columns FROM type WHERE predicates | FIND what IN type FIELDS RETURNING type(columns)[...]

4. Batch Apex

A batch class LoanBatch was created to identify overdue loans and update their status automatically. It runs periodically to ensure loan records remain accurate.

Developer Console - Google Chrome

orgfarm-a1d6c537b-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

BankingService.apxc TransactionTrigger.apxt Log executeAnonymous @9/29/2025, 12:05:54 PM LoanBatch.apxc Log executeAnonymous @9/29/2025, 12:09:57 PM

Execution Log

Timestamp	Event	Details
12:09:57:002	USER_INFO	[EXTERNAL][005gL000008NPN][gayathritalluri28854@agentforce.com]([GMT-07:00] Pacific Daylight Time (America/Los_Angeles))[GMT-07:00]
12:09:57:002	EXECUTION_ST...	
12:09:57:002	CODE_UNIT_ST...	[EXTERNAL]]execute_anonymous_apex
12:09:57:012	VARIABLE_SCO...	[1]]lb LoanBatch true false
12:09:57:012	HEAP_ALLOCATE	[95]]Bytes:3
12:09:57:012	HEAP_ALLOCATE	[100]]Bytes:152
12:09:57:012	HEAP_ALLOCATE	[417]]Bytes:408
12:09:57:012	HEAP_ALLOCATE	[430]]Bytes:408
12:09:57:012	HEAP_ALLOCATE	[317]]Bytes:6
12:09:57:012	HEAP_ALLOCATE	[EXTERNAL]]Bytes:3
12:09:57:012	STATEMENT_EX...	[1]]
12:09:57:012	STATEMENT_EX...	[1]]
12:09:57:012	HEAP_ALLOCATE	[1]]Bytes:57
12:09:57:012	HEAP_ALLOCATE	[1]]Bytes:7
12:09:57:012	HEAP_ALLOCATE	[1]]Bytes:2
12:09:57:012	METHOD_ENTRY	[1]]01pgL000005woPm LoanBatch.LoanBatch()
12:09:57:012	STATEMENT_EX...	[1]]
12:09:57:012	STATEMENT_EX...	[1]]
12:09:57:012	METHOD_EXIT	[1]]LoanBatch

☐ This Frame
 ☐ Executable
 ☐ Debug Only
 ☐ Filter
 Click here to filter the log

Logs Tests Checkpoints Query Editor View State Progress Problems

Name	Line	Problem
------	------	---------

5. Scheduled Apex

A scheduled class DailyTransactionSummary was created to send account summaries every morning at 8 AM. This ensures customers receive a daily update of their account balance and recent transactions.

Setup

Home

Object Manager

Apex

Email

Apex Exception 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 Class

DailyTransactionSummary

Apex Class Detail

Edit

Delete

Download

Security

Show Dependencies

Name	DailyTransactionSummary	Status	Active
Namespace Prefix		Code Coverage	0% (0/16)
Created By	Talluri Gayathri	Last Modified By	Talluri Gayathri
	9/28/2025, 11:42 PM		9/28/2025, 11:45 PM

Class Body

Class Summary

Version Settings

Trace Flags

```
1 global class DailyTransactionSummary implements Schedulable {
2
3     global void execute(SchedulableContext sc) {
4         // Query active bank accounts
5         List<Bank_Account__c> accounts = [
6             SELECT Id, Name, Balance__c, Customer__c
7             FROM Bank_Account__c
8             WHERE Status__c = 'Active'
9         ];
10
11         // For each account, get recent transactions (last 1 day)
12         Map<Id, List<Transaction__c>> accountTxns = new Map<Id, List<Transaction__c>>();
13         List<Transaction__c> txns = [
14             SELECT Id, Amount__c, Type__c, Status__c, Transaction_Date__c, Related_Account__c
15             FROM Transaction__c
16             WHERE Transaction_Date__c = YESTERDAY
17         ];
18         for (Transaction__c txn : txns) {
```

6. Test Classes

A test class BankingTests was developed to cover triggers, batch apex, and the notification service.

The tests create Bank Accounts, Transactions, Loans, and Cards to verify:

- Overdraft prevention works correctly.
- Loan closure restrictions are enforced.
- Service Requests are created for blocked cards.
- Batch jobs mark overdue loans appropriately.
- Daily summaries are scheduled and executed.

The screenshot displays the Salesforce Developer Console. The top pane shows the Apex class `BankingTests` with the following code:

```
1 @isTest
2 public class BankingTests {
3
4     @isTest
5     static void testOverdraftPrevention() {
6         Bank_Account__c acct = new Bank_Account__c(Name='Test Acct', Balance__c=100, Status__c='Active');
7         insert acct;
8
9         Transaction__c txn = new Transaction__c(
10             Related_Account__c = acct.Id,
11             Amount__c = 200,
12             Type__c = 'Debit',
13             Status__c = 'Pending'
14         );
15
16         Test.startTest();
17         Database.SaveResult result = Database.insert(txn, false);
18         Test.stopTest();
19     }
20 }
```

The bottom pane shows the **Logs** tab with a table of test execution results:

User	Application	Operation	Time	Status	Read	Size
Talluri Gayathri	Browser	/setup/build/editApexClass.apexp	9/29/2025, 12:15:07 PM	Success	Unread	1.05 KB
Talluri Gayathri	Browser	/setup/build/viewApexClass.apexp	9/29/2025, 12:14:41 PM	Success	Unread	1.05 KB
Talluri Gayathri	Browser	/setup/build/editApexClass.apexp	9/29/2025, 12:14:19 PM	Success	Unread	1.05 KB
Talluri Gayathri	Browser	/setup/build/viewApexClass.apexp	9/29/2025, 12:14:05 PM	Success	Unread	1.05 KB
Talluri Gayathri	Unknown	Batch Apex	9/29/2025, 12:09:59 PM	No such column 'Due_Date__c' on enti...	Unread	4.61 KB
Talluri Gayathri	Unknown	/services/data/v64.0/tooling/executeA...	9/29/2025, 12:09:57 PM	Success		2.94 KB
Talluri Gayathri	Unknown	/services/data/v64.0/tooling/executeA...	9/29/2025, 12:05:54 PM	Success		3.38 KB
Talluri Gayathri	Browser	/aura	9/29/2025, 12:05:20 PM	Success	Unread	1.34 KB

7. Summary

In this phase, Apex programming was applied to handle advanced banking logic beyond declarative tools. This included overdraft prevention, loan management, blocked card handling, automated loan batch updates, daily account summaries, and robust testing. These features strengthen the Digital Banking CRM by ensuring secure, reliable, and automated banking operations.