PHASE 5: Apex Programming (Developer)

Step 1: Apex Classes & Triggers Implementation

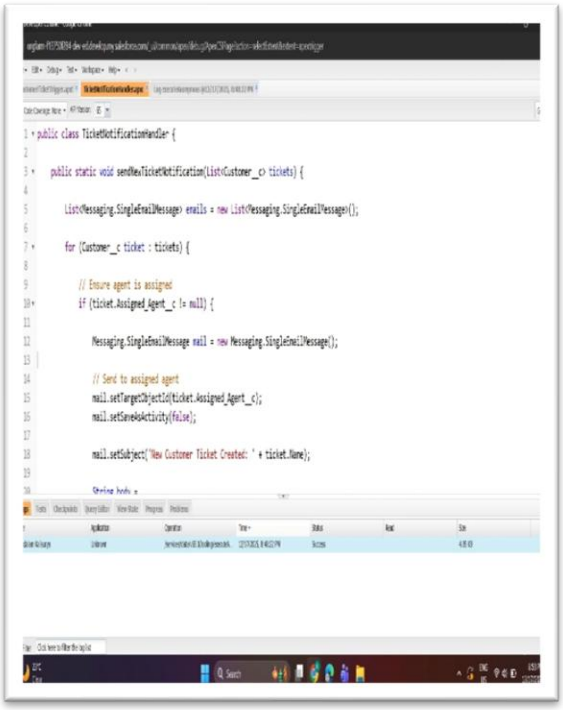
Apex programming was used to handle backend logic that requires programmatic control, specifically for sending email notifications when a new customer ticket is created.

Apex Class: TicketNotificationHandler

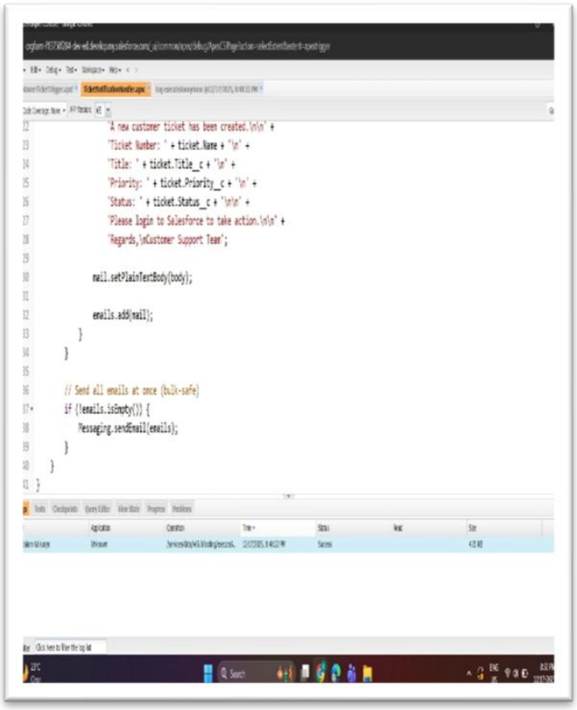
Purpose:
The TicketNotificationHandler Apex class is responsible for sending automated email notifications to the assigned support agent whenever a new Customer Ticket record is created.

Implementation Details:

- The class contains a static method sendNewTicketNotification that accepts a list of Customer__c records.
- It processes tickets in bulk to comply with Salesforce governor limits.
- For each ticket:
    - Verifies that an agent is assigned (Assigned_Agent__c != null).
    - Creates a Messaging.SingleEmailMessage.
    - Sets the recipient dynamically using the assigned agent's User Id.
    - Constructs a plain-text email body containing ticket details.



```
public class TicketNotificationHandler {

    public static void sendNewTicketNotification(List<Customer__c> tickets) {

        List<Messaging.SingleEmailMessage> emails = new List<Messaging.SingleEmailMessage>();

        for (Customer__c ticket : tickets) {

            // Ensure agent is assigned
            if (ticket.Assigned_Agent__c != null) {

                Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();

                // Send to assigned agent
                mail.setTargetObjectId(ticket.Assigned_Agent__c);
                mail.setSaveAsActivity(false);

                mail.setSubject('New Customer Ticket Created: ' + ticket.Name);

                String body =
```



```
                    'A new customer ticket has been created.\n\n' +
                    'Ticket Number: ' + ticket.Name + '\n' +
                    'Title: ' + ticket.Title__c + '\n' +
                    'Priority: ' + ticket.Priority__c + '\n' +
                    'Status: ' + ticket.Status__c + '\n\n' +
                    'Please login to Salesforce to take action.\n\n' +
                    'Regards,\nCustomer Support Team';

                mail.setPlainTextBody(body);

                emails.add(mail);
            }
        }

        // Send all emails at once (bulk-safe)
        if (!emails.isEmpty()) {
            Messaging.sendEmail(emails);
        }
    }
}
```

Email Content Includes:

- Ticket Number
- Title
- Priority
- Status
- Instruction to log in to Salesforce for further action

All email messages are added to a list and sent together using Messaging.sendEmail, ensuring bulk-safe execution.

Apex Trigger: CustomerTicketTrigger

Purpose:
The trigger ensures that the notification logic is executed automatically when a new customer ticket is created.

Trigger Configuration:

- **Trigger Name:** CustomerTicketTrigger
- **Object:** Customer__c
- **Event:** after insert

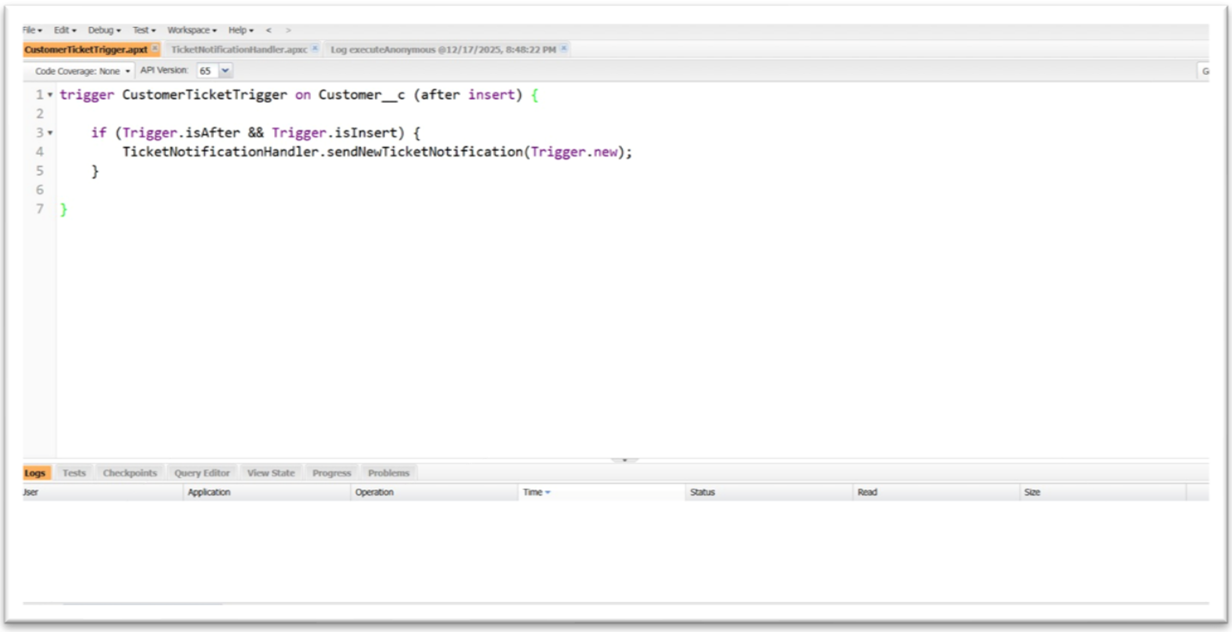
Trigger Logic:

- Executes only in the after insert context.
- Passes newly created ticket records (Trigger.new) to the handler class.
- Delegates all email logic to TicketNotificationHandler, keeping the trigger lightweight.

Trigger Code Behavior:

- Automatically fires when a new ticket is inserted.
- Invokes sendNewTicketNotification method.
- Ensures support agents are notified immediately after ticket creation.

This approach follows Salesforce best practices by separating trigger logic from business logic.


File Edit Debug Test Workspace Help
CustomerTicketTrigger.apxt
TicketNotificationHandler.apxc
Log executeAnonymous @12/17/2025, 8:48:22 PM
Code Coverage: None
API Version: 65
Logs Tests Checkpoints Query Editor View State Progress Problems
User Application Operation Time Status Read Size

```
trigger CustomerTicketTrigger on Customer__c (after insert) {

    if (Trigger.isAfter && Trigger.isInsert) {
        TicketNotificationHandler.sendNewTicketNotification(Trigger.new);
    }

}
```

Step 2: Use of Collections (List, Set, Map)

Purpose:
Collections were used to efficiently handle multiple records, improve performance, and ensure scalability.

Collections Used in the Project:

- **List<Customer__c>**
  - Used to iterate over newly created ticket records passed from the trigger.
- **List<Messaging.SingleEmailMessage>**
  - Stores multiple email messages.
  - Enables sending all notification emails in a single operation (bulk-safe).
- **Set<Id> (Conceptual Usage)**
  - Can be used to track unique ticket or agent IDs if extended in future enhancements.
  - Helps avoid duplicate processing.

Using collections ensures the solution remains compliant with Salesforce governor limits and performs efficiently during bulk operations.

Benefits of Apex Implementation in Phase 5

- Enables real-time notifications beyond declarative automation
- Ensures immediate agent awareness of new tickets
- Follows trigger handler best practices

- Bulk-safe and scalable solution
- Clean separation of responsibilities between trigger and handler class