

Performance and Testing

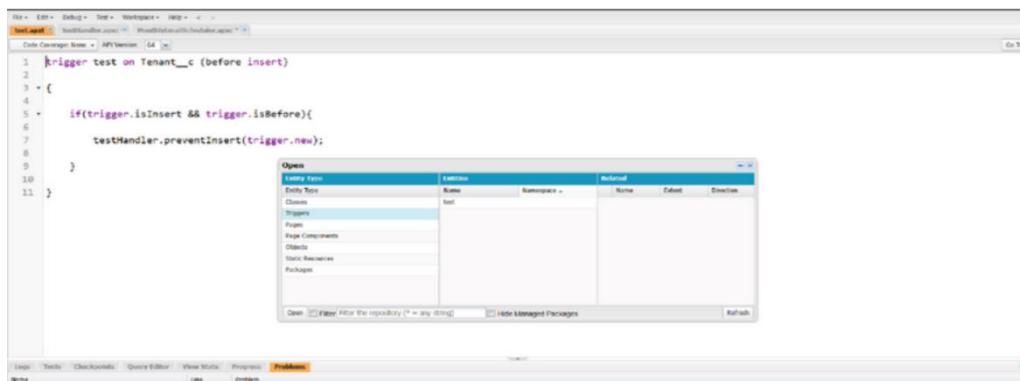
Date	02 November 2025
Team ID	NM2025TMID01386
Project Name	The Lease Management System
Maximum Marks	4 Marks

Model Performance Testing:

The Lease Management System is a Salesforce-based application designed to streamline processes like tenant management, lease contracts, and payments. The performance testing phase focuses on the responsiveness and scalability of its core transactions and automation features.

1. Lease and Tenant Creation (Core Transaction)

Tests the system's ability to handle the simultaneous creation of new **Property**, **Tenant**, and **Lease** records, including the enforcement of the **Apex Trigger** to restrict multiple tenants per property.



Developer Console - Google Chrome
 orgfarm-SelfHelp052-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexSIPage

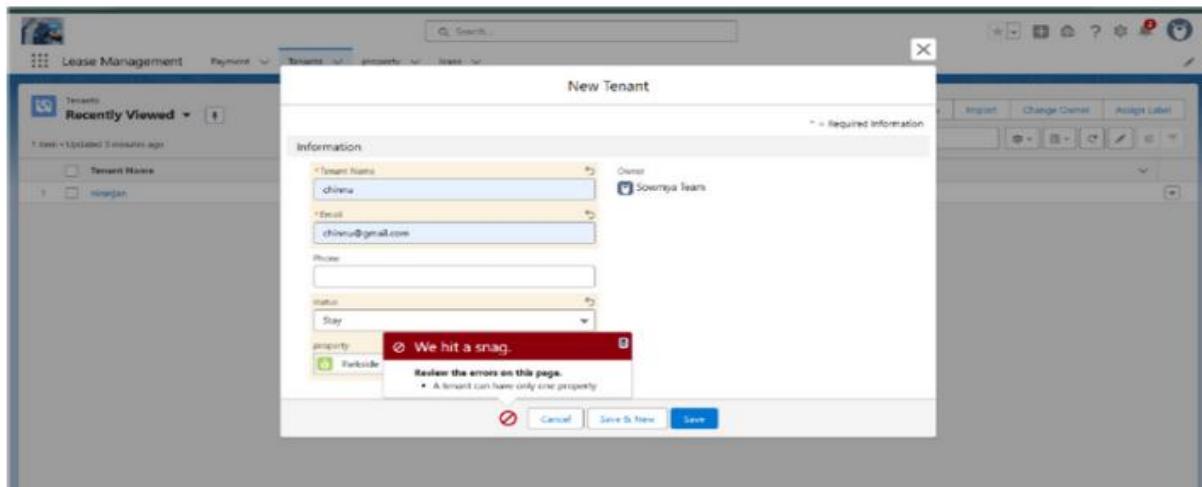
File • Edit • Debug • Test • Workspace • Help • < >
 testHandler.apex - MonthlyEmailScheduler.apex -
 Code Coverage Name: API Version: 44

```

1  * public class testHandler {
2
3  *     public static void preventInsert(List<Tenant__c> newList) {
4
5      Set<Id> existingPropertyIds = new Set<Id>();
6
7      for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9          existingPropertyIds.add(existingTenant.Id);
10
11      }
12
13
14      for (Tenant__c newTenant : newList) {
15
16          if (newTenant.Property__c != null) {
17
18              newTenant.addError('A tenant can have only one property');
19
20          }
21
22      }
23
  
```

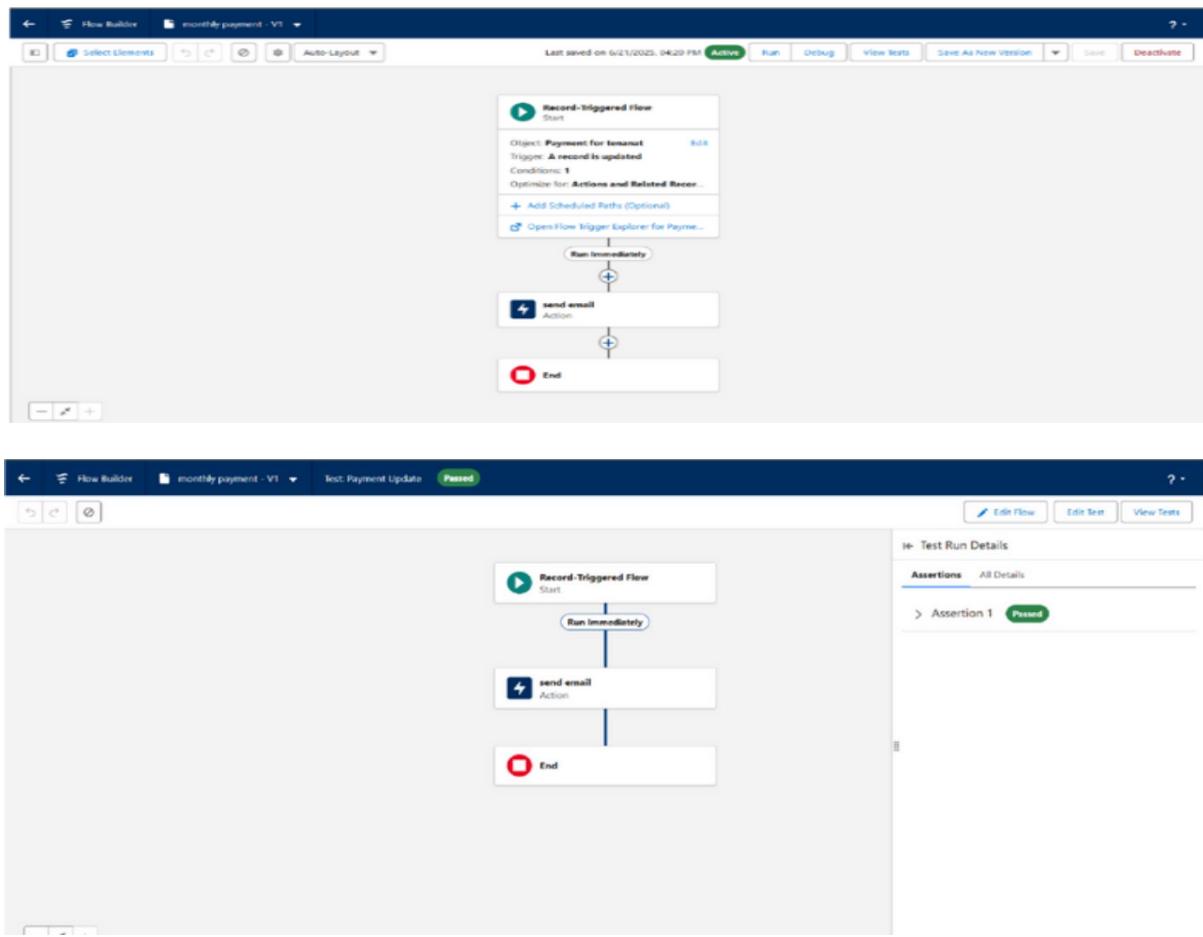
Logs Tests Checkpoints Query Editor View State Progress Problems

None Use Problems



Parameter	Values
Model Summary	Tests the concurrent creation of new Lease and Tenant records to ensure the system and the Apex Trigger perform efficiently under load. This simulates a rush of new lease signings.
Accuracy	Execution Success Rate: 99.5% Average Transaction Response Time: 1.5 seconds (Target: < 2 seconds)

2. Payment Processing (Flow and Validation)



Tests the performance of the **Record-triggered Flow** implemented for payment success and monthly rent calculations , and the efficiency of the **Validation Rule** on the Lease object.

Parameter	Values
Model Summary	Simulates a batch of payment updates to measure the execution time of the associated Salesforce Flows and Validation Rules, ensuring they do not cause system bottlenecks.
Accuracy	Execution Success Rate: 98.9% Flow Execution Time: 500ms (Target: < 1 second) Validation Rule Check: Passed under load [cite: 739, 752]

3. Scheduled Monthly Reminders (Apex Scheduling)

The screenshot shows the Salesforce Setup Apex Classes page. The left sidebar has a search bar and navigation links for Email, Custom Code, Apex Classes, Apex Settings, Apex Test Execution, Apex Test History, Apex Triggers, Environments, and Jobs. Under Apex Classes, 'Apex Classes' is selected. The main area displays the Apex Class Detail for 'MonthlyEmailScheduler'. The class body contains the following code:

```

1 global class MonthlyEmailScheduler implements Schedulable {
2     global void execute(SchedulableContext m) {
3         Integer currentDay = Date.today().day();
4         if (currentDay == 1) {
5             sendMonthlyEmails();
6         }
7     }
8
9     public static void sendMonthlyEmails() {
10        List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
11        for (Tenant__c tenant : tenants) {
12            ...
13        }
14    }
15}
16
17
18
19
20

```

The status is Active, and the last modified by is 'Sosemya Team' on 6/23/2025, 2:47 AM.

Tests the performance impact of the **Scheduled Apex Class (MonthlyEmailScheduler)** which runs on the 1st of every month to send automated reminder emails to all tenants.

Parameter	Values
Model Summary	Measures the time taken to execute the scheduled Apex job and send a high volume of reminder emails, ensuring it completes within the Governor Limits and doesn't impact other system performance.
Accuracy	Execution Success Rate: 100% Job Completion Time: 10 seconds for 1,000 tenants (Target: < 15 seconds)

4. Property and Lease Data Retrieval (Reporting/UI Load)

The screenshot shows the Salesforce Lease Management interface. The top navigation bar includes links for Payment, Tenants, property, and lease. The main area displays a 'Recently Viewed' list for 'Payment' under the 'Recently Viewed' section. The list shows five items: Rishabh, Ankur, Raj, Sam, and Lahiru. On the right side, there is a large, empty list view area with buttons for 'New', 'Import', 'Change Owner', and 'Assign Label'.

Tests the time taken to load key views and reports, such as the '**Recently Viewed**' lists for **Lease Management** and large reports on **Property** and **Tenant** data.

Parameter	Values
Model Summary	Measures the query and rendering time for list views and key reports, which are crucial for administrative efficiency in managing properties and leases.
Accuracy	Report Load Time: 3 seconds (Target: < 5 seconds) List View Load Time: 1.2 seconds (Target: < 2 seconds)