

Saving Payment Process Twice

System Impact Summary

Each successful payment invocation results in two database writes, potentially causing duplicate invoice errors and performance degradation.

Description

The `PaymentService.processPayment(orderId, paymentInfo)` method invokes `orderRepository.save(order)` two times per call, but the unit test expects exactly one.

Steps to Reproduce

1. Mock dependencies (`OrderRepository`, `CartService`, etc.) in `PaymentServiceTest`.
2. Call `paymentService.processPayment(orderId, cardInfo)`.
3. Verify repository interactions: `verify(orderRepository, times(1)).save(...)`.

Expected Behavior: `orderRepository.save(...)` should be called exactly once when processing a successful payment.

Actual Behavior: Two invocations occur:

- > at `PaymentService.processPayment(...:line130)`
- > at `PaymentService.processPayment(...:line157)`

The test fails with a **TooManyActualInvocations** error.

Severity: Low – Primarily a test failure; in production two saves may be idempotent but wasteful.

Root Cause

The service implementation likely calls `save()` after setting payment details, then again after generating or emailing the invoice, instead of batching updates in a single save.

Proposed Fix: Modify the service to only call `save()` once after all modifications are applied.