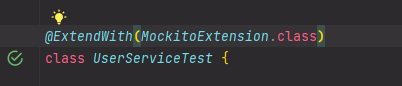
**Mockito:**

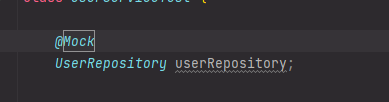
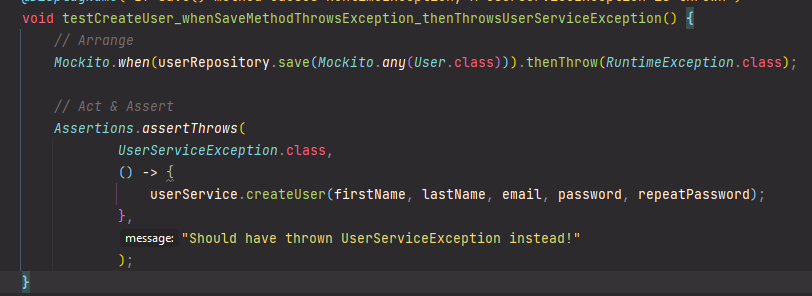
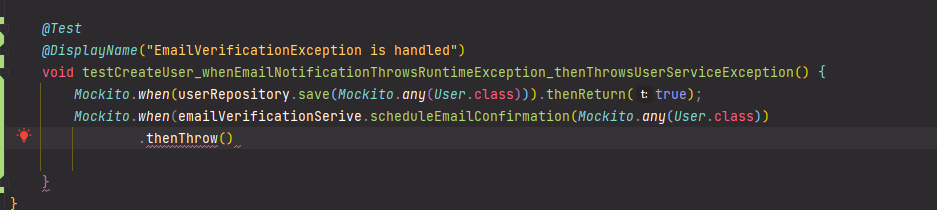
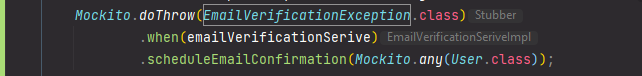
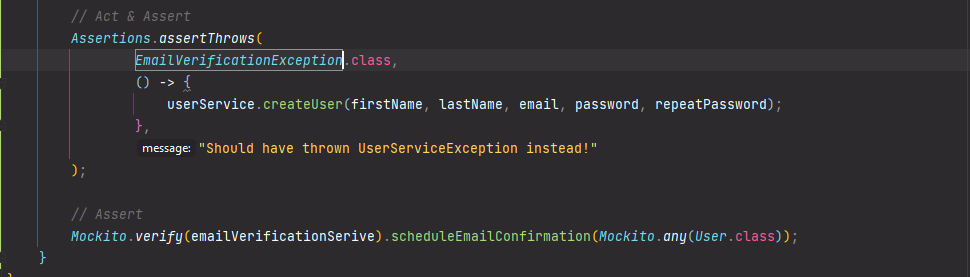
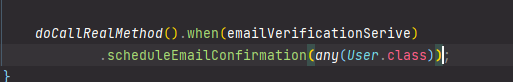
1. **What is Mokito:**

* Là một mã nguồn mở được tạo ra bởi Java. Cho phép chúng ta dễ dàng tạo đối tượng “Test doubles” trong unit test.
* Test doubles là gì:
  + Là 1 Object có thể thay thế cho real object trong unit test
  + Các kiểu của nó:
    - Mock
    - Fake
    - Spy
    - Stub
  + Chúng sẽ được sử dụng để thay thế tạm thời cho object thực
  + Tại sao chúng ta cần phải thay thế real Object?
    - Mock object cho phép ta mock các hành vi của chính method đó
    - Giúp chúng ta có thể thực hiện test method đó 1 cách độc lập mà không cần phụ thuộc vào các bean được khai báo và sử dụng bên trong method.
    - Hay nói cách khác là chúng ta chỉ test code trong 1 method của class đó. Và khi trong method có gọi đến 1 method của class khác, chúng ta không test code của method khác đó. Vì thế chúng ta cần cách biệt method mình đang test với mọi method của class khác
    - Giúp chúng ta thực hiện chính xác các hành động với db mà không cần lưu lại các bản ghi vào db

1. **Các bước để viết Mockito:**

* Đầu tiên, ta cần thêm @ExtendWith(MockitoExtension.class) annotation vào đầu class test của chúng ta. Input bên trong Annotation này sẽ kích hoạt cho chúng ta sử dụng Mockito annotation trong class test



* Sau đó, sẽ khai báo 1 biến là class hoặc interface với @Mock annotation ở bên trên nó
* 
  + Annotation này sẽ tự động tạo mock object để triển khai class hoặc interface đó
* Tiếp theo, chúng ta sẽ để cho Mockito tự gắn bean vào trong real object mà mình muốn test với @InjectMock annotation. Class có inject mock phải là 1 class triển khai từ 1 lớp interface đó
* Sau đó, chúng ta cần phải cho Mockito biết cần phải làm gì khi chúng ta gọi các method mà chúng ta đã khai báo trong interface. Nó sẽ được thực hiện ở phần Arrange.
  + Sử dụng : **Mockito.when**(myClass.myMethod(Mockito.any(myClass.class))).thenReturn(value mà class đó trả về).
  + Với **Mockito.any()**: Sẽ chấp nhận mọi chi tiết của class mà không cần biết giá trị là gì.
  + Với **Mokito.verify**(Mock Object, Mockito.times(Số lần gọi method mà mình muốn so sánh. Default là 1)).Method\_independent(Mockito.any(myclass.class)): Để xem method bên trong hàm mà mình muốn test chạy bao nhiêu lần theo ý mình muốn
  + 
  + Với Mockito.never(): Dùng để test xem method sẽ không được gọi
* Nếu chúng ta muốn test xem method phụ thuộc bên trong sẽ throw ra exception mà chúng ta mong muốn. Thì chúng ta sẽ sử dụng như sau:
  + 
  + Dùng này khiến cho method phụ thuộc bên trong sẽ throw ra exception như ta mong muốn.
  + Sau đó thực hiện test như sau:
    - 
* Nếu method kiểu trả về là **void,** thì chúng ta sẽ làm như sau:
  + Nếu chúng ta sử dụng 2 when() trong cùng 1 method test. Thì nó sẽ không hoạt động cả 2 when() đó. Ví dụ:
    - 
  + Do đó, ta sẽ sử dụng
    - doThrow(Expect\_Exception).When(Mock\_object).Method\_Of\_Mock\_Object.
    - Dòng code trên sẽ giúp t sử dụng được when() cho method khác, và đồng thời set cho method không có kiểu trả về 1 exception. Ví dụ:
    - 
  + Sau đó ta thực hiện:
  + 
  + Với cách sử dụng doNothing() thì method phụ thuộc sẽ không làm gì cả. Chúng ta sẽ dùng nó khi ta muốn method phụ thuộc thực hiện bất cứ hành vi nào và ta không muốn nó được gọi khi method này được thực hiện.
    - 
* Khi ta muốn sử dụng Mock Object nhưng vấn muốn gọi real method thì thực hiện như sau:
  + 
  + doCallRealMethod().when(Mock\_object).method\_of\_mock\_object(Object)