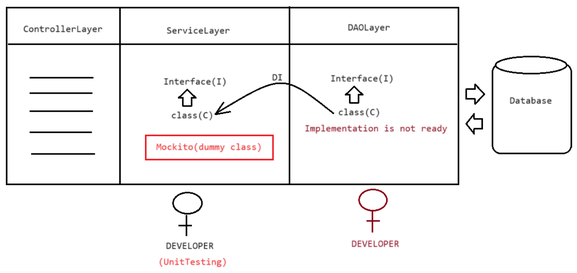
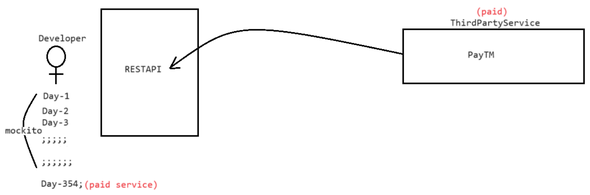
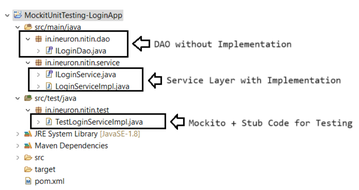
Mockito








```java
@Test
public void testLoginWithValidCredentials() {
     Mockito.when(loginDaoMock.authenticate("root", "root123")).thenReturn(1);
     Assertions.assertTrue(service.login("root", "root123"));
}
```

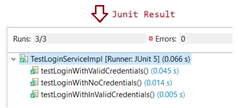
stub code

```java
@Test
public void testLoginWithInValidCredentials() {
     Mockito.when(loginDaoMock.authenticate("root", "root123@")).thenReturn(0);
     Assertions.assertFalse(service.login("root", "root123@"));
}

@Test
public void testLoginWithNoCredentials() {
     Mockito.when(loginDaoMock.authenticate("", "")).thenReturn(0);
     Assertions.assertThrows(IllegalArgumentException.class, () -> service.login("",""));
}
```

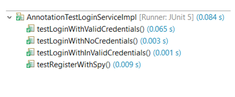
```java
@BeforeAll
public static void setUpOnce() {
    // create a mock object for DAO class having null method Implementation
    loginDaoMock = Mockito.mock(ILoginDao.class);
    System.out.println("Mock class object is :: " + loginDaoMock.getClass().getName());

    // Create Service class object by using Mock object
    loginService = new LoginServiceImpl(loginDaoMock);
}
```

Mock Object

Junit Result




```java
public class AnnotationTestLoginServiceImpl {

    @Mock
    private static ILoginDao loginDaoMock;

    @Spy
    private static ILoginDao loginDaoSpy;

    @InjectMocks
    private static LoginServiceImpl loginService;


    public AnnotationTestLoginServiceImpl() {
        MockitoAnnotations.openMocks(this);
    }
```

Dealing with Mockito Annotations

```java
    @Test
    public void testRegisterWithSpy() {
        ILoginDao loginDaoSpy = Mockito.spy(ILoginDao.class);
        System.out.println("Spy object is :: " + loginDaoSpy.getClass().getName());
        LoginServiceImpl loginService = new LoginServiceImpl(loginDaoSpy);
        System.out.println("AnnotationTestLoginServiceImpl.testRegisterWithSpy() :: "+loginService);

        loginService.registerUser("sachin", "admin");
        loginService.registerUser("dhoni", "visitor");
        loginService.registerUser("kohli", "");

        Mockito.verify(loginDaoSpy, Mockito.times(1)).addUser("sachin", "admin");
        Mockito.verify(loginDaoSpy, Mockito.times(0)).addUser("dhoni", "visitor");
        Mockito.verify(loginDaoSpy, Mockito.never()).addUser("kohli", "");
    }
}
```