Phase-3 Practice Project

Handling User Authentication

Source code

UserAuthenticationTest:

```
import static org.junit.jupiter.api.Assertions.assertFalse; import static
org.junit.jupiter.api.Assertions.assertThrows; import static
org.junit.jupiter.api.Assertions.assertTrue;
import org.junit.jupiter.api.DisplayName;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.function.Executable;
@DisplayName("UserAuthentication Test")
public class UserAuthenticationTest {
@Test
@DisplayName("Test valid username and password")
public void testValidCredentials() {
final UserAuthentication userAuth = new UserAuthentication(); boolean isAuthenticated =
userAuth.authenticate("john", "password"); System.out.println("Test valid username and
password:
isAuthenticated=" + isAuthenticated);
assertTrue(isAuthenticated);
}
@Test
@DisplayName("Test invalid username and password")
public void testInvalidCredentials() {
final UserAuthentication userAuth = new UserAuthentication();
boolean isAuthenticated = userAuth.authenticate("john", "wrong password");
System.out.println("Test invalid username and password: isAuthenticated=" +
isAuthenticated);
assertFalse(isAuthenticated);
}
@Test
@DisplayName("Test null username and password")
public void testNullCredentials() {
final UserAuthentication userAuth = new UserAuthentication();
assertThrows(NullPointerException.class, new Executable() {
public void execute() throws Throwable { userAuth.authenticate(null, null);
}
});
System.out.println("Test null username and password: Exception thrown as expected");
}
}
```

UserAuthentication :

```
public class UserAuthentication {
private static final String VALID_USERNAME = "john";
private static final String VALID_PASSWORD = "password";
public boolean authenticate(String username, String password) {
if (username == null || password == null) {
throw new IllegalArgumentException("Username and password
cannot be null");
}
return username.equals(VALID_USERNAME) && password.equals(VALID_PASSWORD);
}
}
```