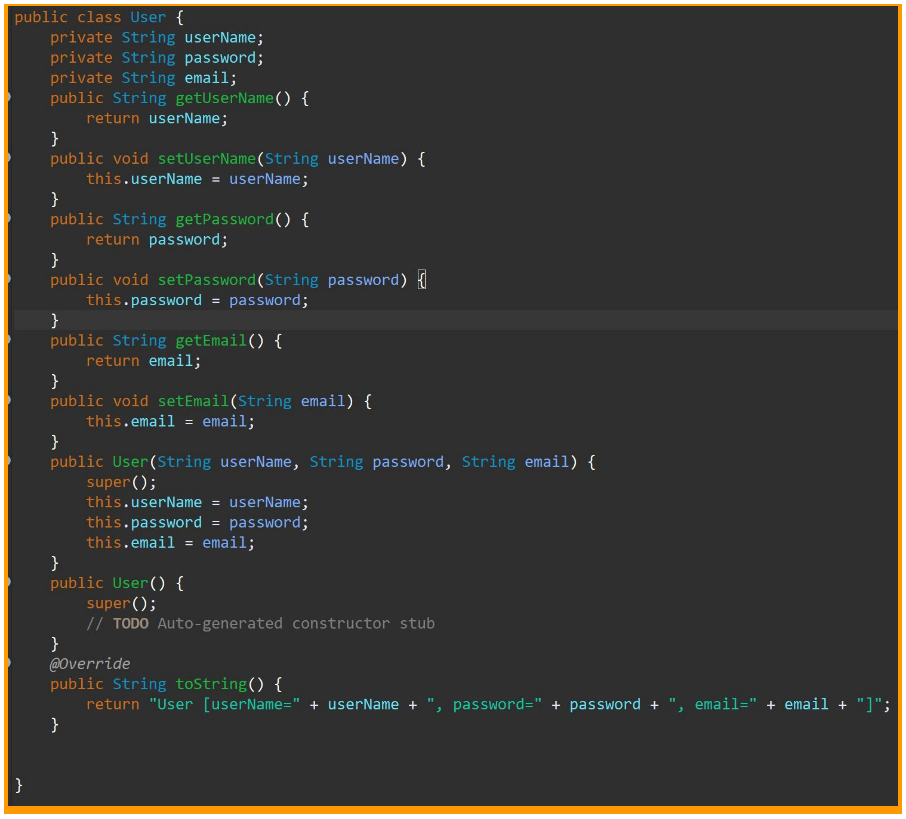
Handling User Authentication

Steps:

1. Create a standalone java application using Maven
2. Create an authentication class
    a. Login
    b. getEmail
    c. getUsername
    d. Logout
3. Create a JUnit test class to create unit tests for the authentication class
    a. testLogin
    b. testWrongUserLogin
    c. testUserAssert
    d. testLogout
4. Run the test class directly as a JUnit and check if all the tests pass

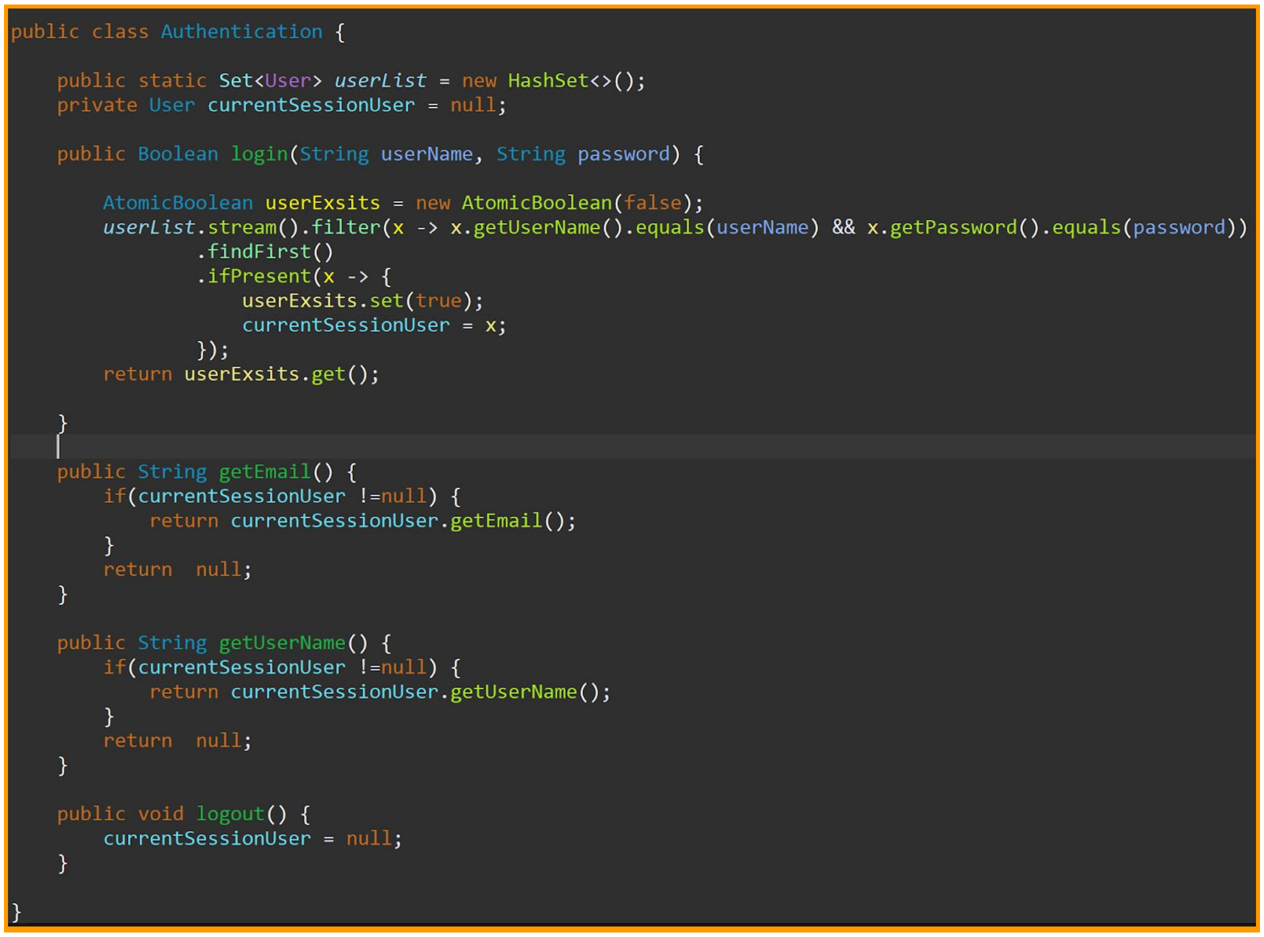
Screenshots:

Step 1 : User class has username, password, and email



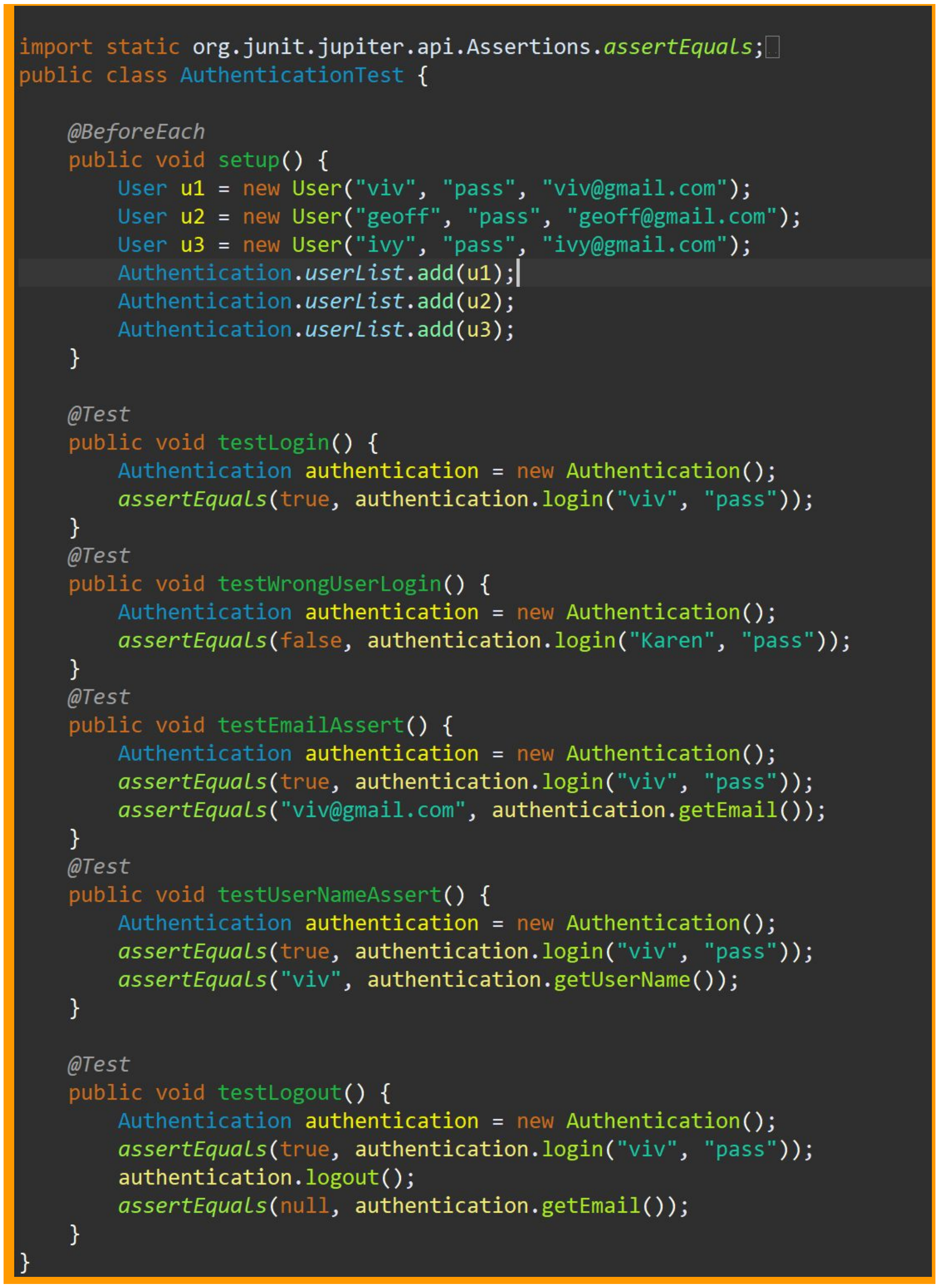
```java
public class User {
    private String userName;
    private String password;
    private String email;
    public String getUserName() {
        return userName;
    }
    public void setUserName(String userName) {
        this.userName = userName;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public User(String userName, String password, String email) {
        super();
        this.userName = userName;
        this.password = password;
        this.email = email;
    }
    public User() {
        super();
        // TODO Auto-generated constructor stub
    }
    @Override
    public String toString() {
        return "User [userName=" + userName + ", password=" + password + ", email=" + email + "]";
    }


}
```

Step2: Authentication class has login , getEmail, getUsername and logout methods related to user authentication.
Verify login: If a user login with the correct username and password, it will filter the userList collection and mark currentUser as true.
Verify email: If currentUser exists, then get user email.
Verify Username: same logic as email
Verify logout: Set currentUser equals to null.



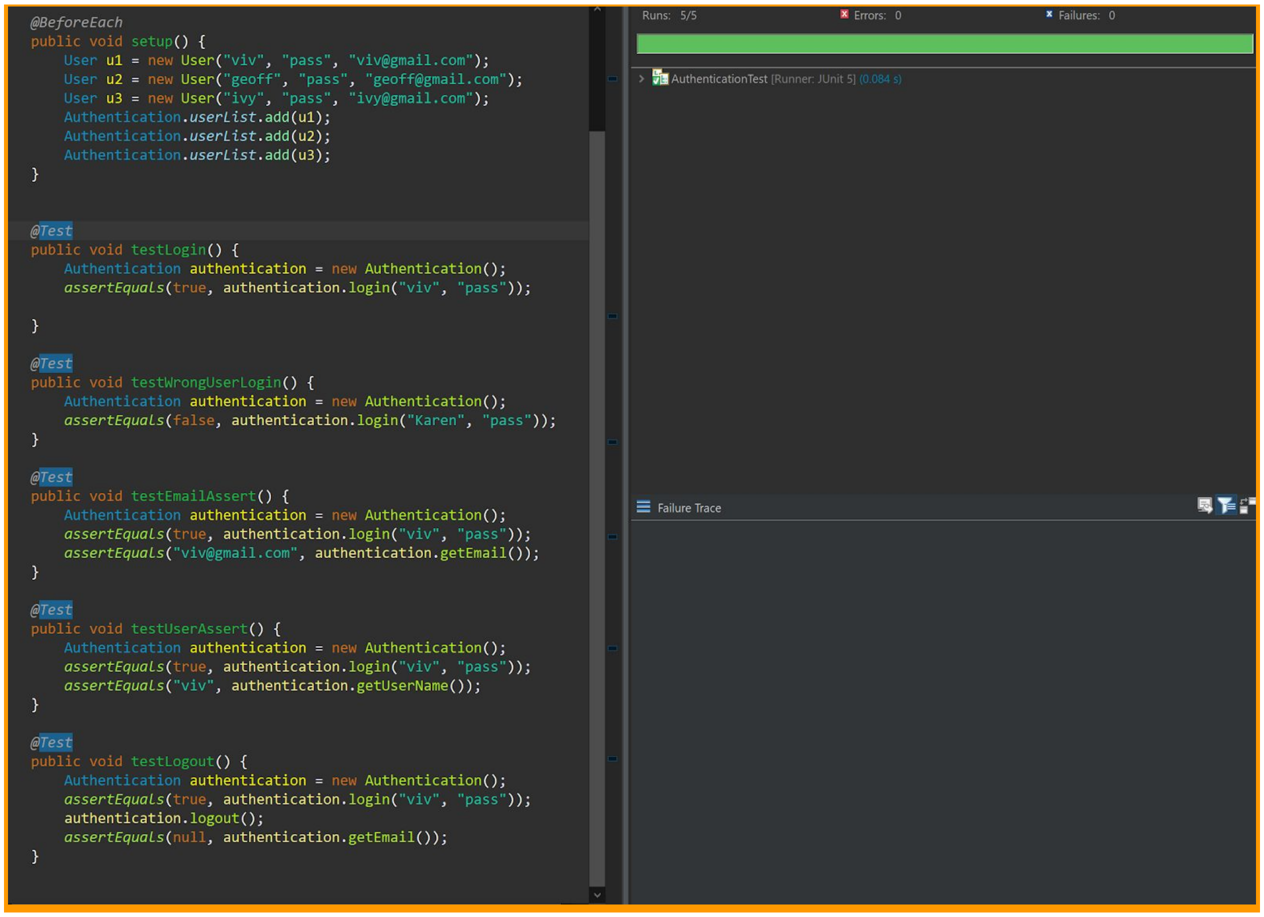
```
public class Authentication {

    public static Set<User> userList = new HashSet<>();
    private User currentSessionUser = null;

    public Boolean login(String userName, String password) {

        AtomicBoolean userExsits = new AtomicBoolean(false);
        userList.stream().filter(x -> x.getUserName().equals(userName) && x.getPassword().equals(password))
                .findFirst()
                .ifPresent(x -> {
                    userExsits.set(true);
                    currentSessionUser = x;
                });
        return userExsits.get();

    }

    public String getEmail() {
        if(currentSessionUser !=null) {
            return currentSessionUser.getEmail();
        }
        return  null;
    }

    public String getUserName() {
        if(currentSessionUser !=null) {
            return currentSessionUser.getUserName();
        }
        return  null;
    }

    public void logout() {
        currentSessionUser = null;
    }

}
```

Step 3: AuthenticationTest class beforeEach will set an userlist with userifomation.
TestCase with corresponding with the methods in Authentication class



```
import static org.junit.jupiter.api.Assertions.assertEquals;
public class AuthenticationTest {

    @BeforeEach
    public void setup() {
        User u1 = new User("viv", "pass", "viv@gmail.com");
        User u2 = new User("geoff", "pass", "geoff@gmail.com");
        User u3 = new User("ivy", "pass", "ivy@gmail.com");
        Authentication.userList.add(u1);
        Authentication.userList.add(u2);
        Authentication.userList.add(u3);
    }

    @Test
    public void testLogin() {
        Authentication authentication = new Authentication();
        assertEquals(true, authentication.login("viv", "pass"));
    }
    @Test
    public void testWrongUserLogin() {
        Authentication authentication = new Authentication();
        assertEquals(false, authentication.login("Karen", "pass"));
    }
    @Test
    public void testEmailAssert() {
        Authentication authentication = new Authentication();
        assertEquals(true, authentication.login("viv", "pass"));
        assertEquals("viv@gmail.com", authentication.getEmail());
    }
    @Test
    public void testUserNameAssert() {
        Authentication authentication = new Authentication();
        assertEquals(true, authentication.login("viv", "pass"));
        assertEquals("viv", authentication.getUserName());
    }

    @Test
    public void testLogout() {
        Authentication authentication = new Authentication();
        assertEquals(true, authentication.login("viv", "pass"));
        authentication.logout();
        assertEquals(null, authentication.getEmail());
    }
}
```

Step 4 Run JUnit test, all 5 tests passed.



```java
    @BeforeEach
    public void setup() {
        User u1 = new User("viv", "pass", "viv@gmail.com");
        User u2 = new User("geoff", "pass", "geoff@gmail.com");
        User u3 = new User("ivy", "pass", "ivy@gmail.com");
        Authentication.userList.add(u1);
        Authentication.userList.add(u2);
        Authentication.userList.add(u3);
    }


    @Test
    public void testLogin() {
        Authentication authentication = new Authentication();
        assertEquals(true, authentication.login("viv", "pass"));

    }

    @Test
    public void testWrongUserLogin() {
        Authentication authentication = new Authentication();
        assertEquals(false, authentication.login("Karen", "pass"));
    }

    @Test
    public void testEmailAssert() {
        Authentication authentication = new Authentication();
        assertEquals(true, authentication.login("viv", "pass"));
        assertEquals("viv@gmail.com", authentication.getEmail());
    }

    @Test
    public void testUserAssert() {
        Authentication authentication = new Authentication();
        assertEquals(true, authentication.login("viv", "pass"));
        assertEquals("viv", authentication.getUserName());
    }

    @Test
    public void testLogout() {
        Authentication authentication = new Authentication();
        assertEquals(true, authentication.login("viv", "pass"));
        authentication.logout();
        assertEquals(null, authentication.getEmail());
    }
```

```
Runs: 5/5        Errors: 0        Failures: 0

> AuthenticationTest [Runner: JUnit 5] (0.084 s)
```

Github link:

https://github.com/chefvivica/Handling-User-Authentication-