**CODE WITH ME – PAIR PROGRMMING**

**TOOL ANALYSIS**

Code With Me offers different types of subscriptions that you can choose at the [JetBrains](https://www.jetbrains.com/code-with-me/buy/?fromIDE#personal?billing=yearly) website.

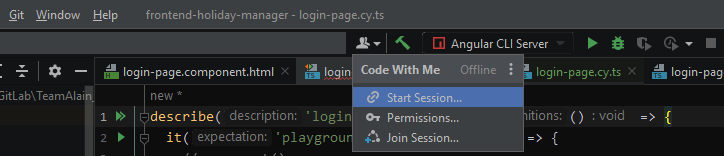
If you have an active IDE subscription (except for DataGrip and Rider), you don't need a separate Code With Me license as it's already included in the IDE's license as a plugin.

If you do not have an active IDE subscription, the following should be employed, in order to activate the license:

1. Get a license from the JetBrains website.
2. From the main menu, select Help | Register
3. In the Licenses dialog, select Code With Me.
4. From the options on the right, select Activate New License.
5. Depending on what IDE you have, select JB Account.

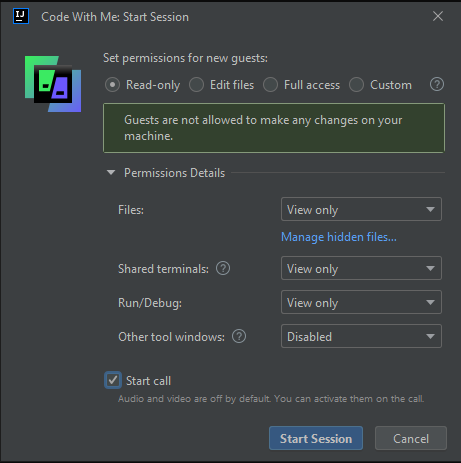
**Initializing a session**

**STEP 1:**



**STEP 2:**

Here, the guests access level may be selected.

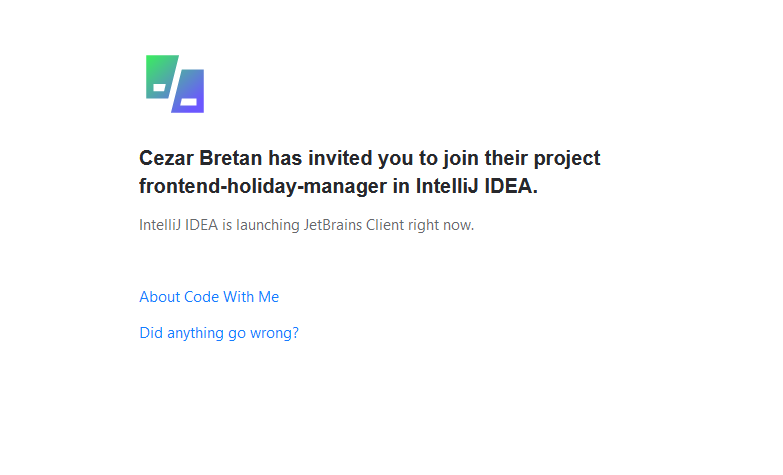
****

**STEP 3:**

The link for the session is generated and must be sent to the guests by the initiator.

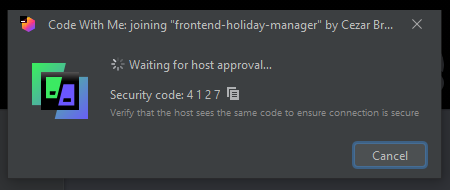
**STEP 4:**

The guests must access the link in order to connect to the session.

****

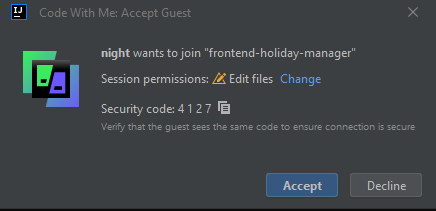
**STEP 5:**

The guests await for the initiator approval

****

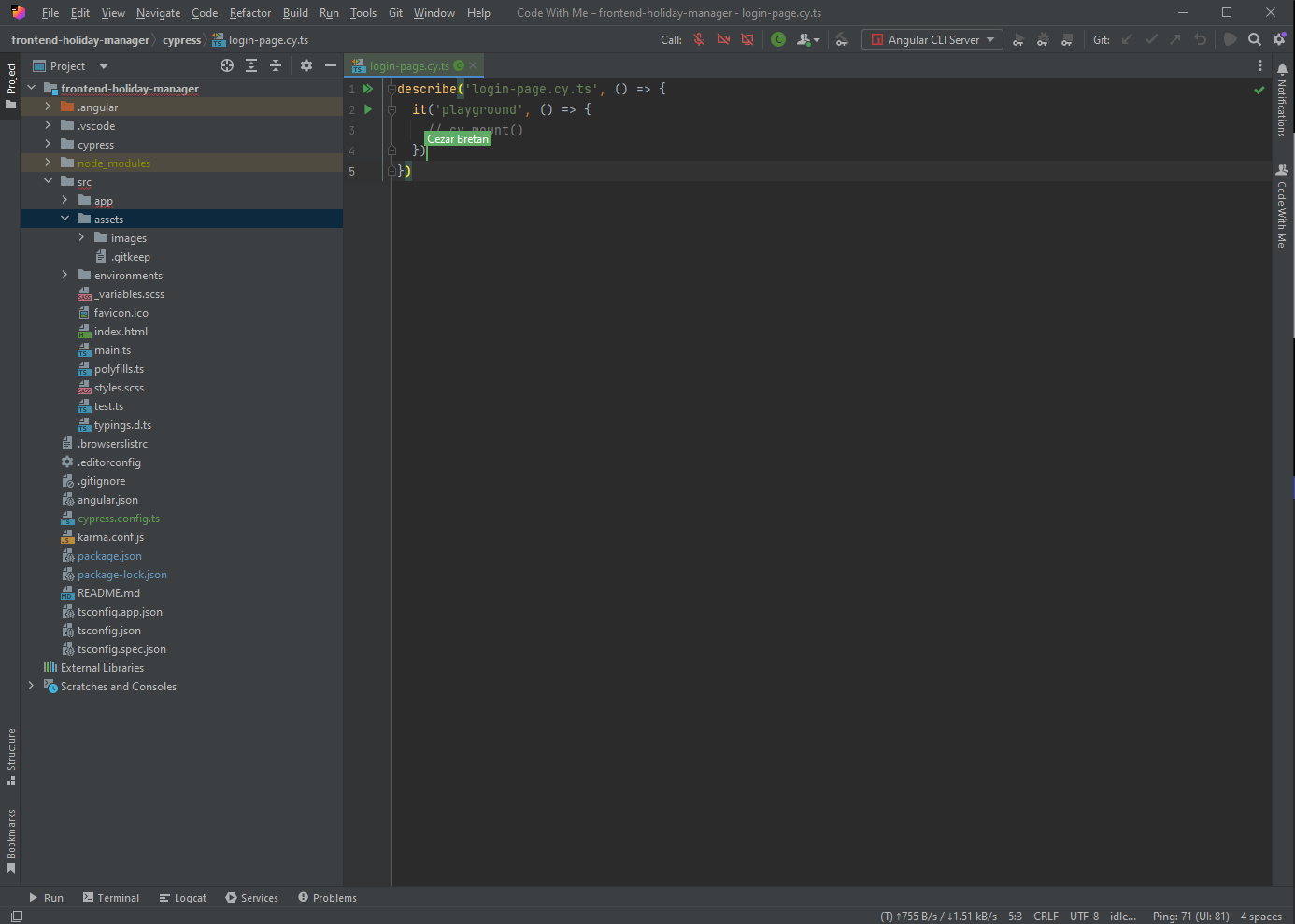
**STEP 6:**

The initiator must accept the requests.

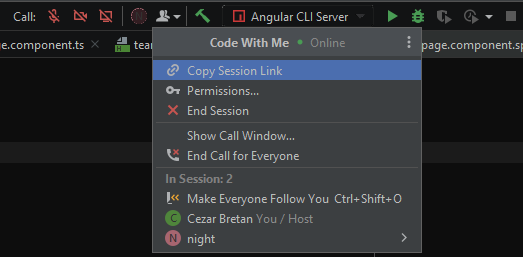
****

**STEP 7:**

After the request is approved by the initiator, the project is loaded for each of the guests.

****

Here, on the initiator screen, all of the guests are being shown.

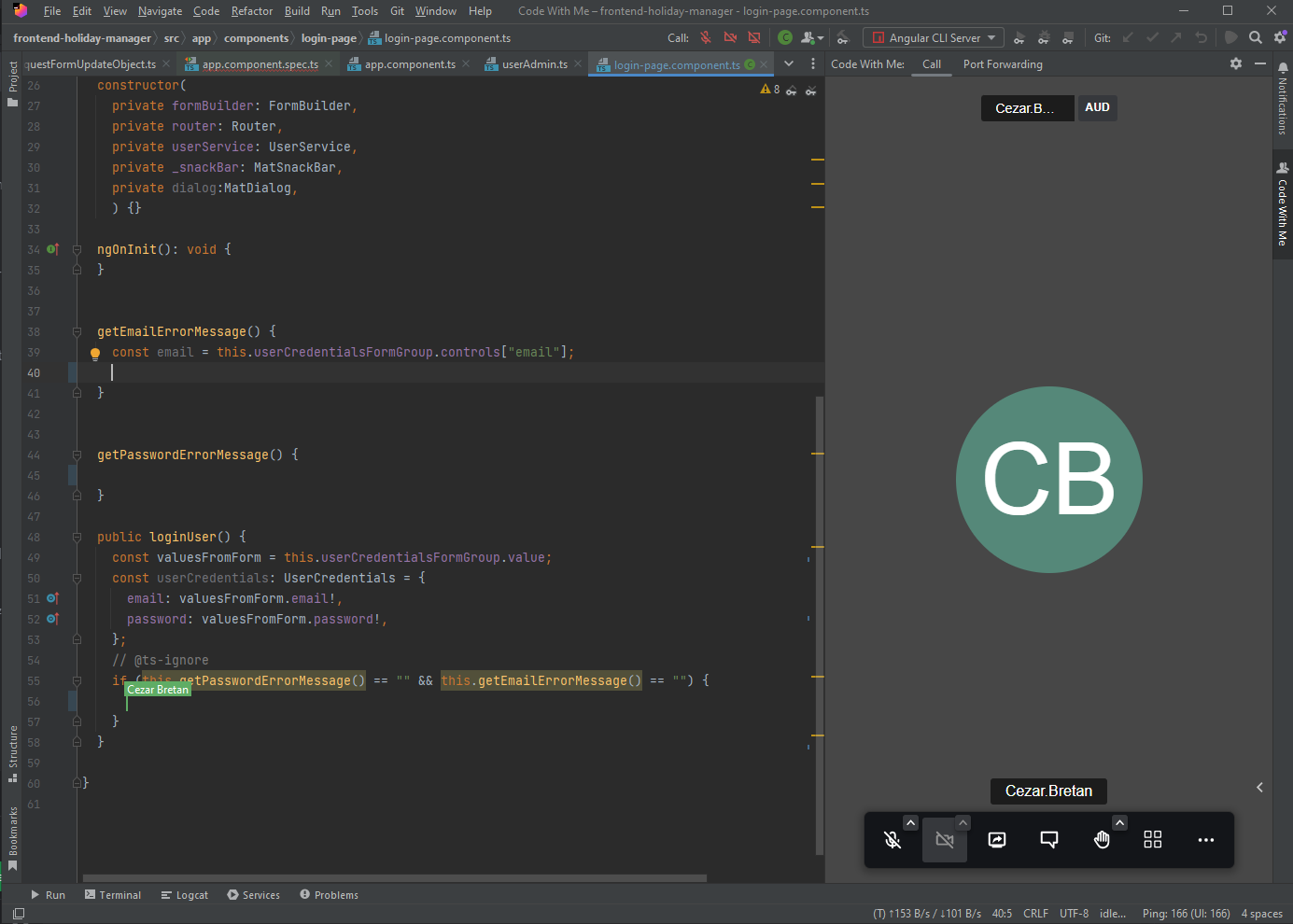
****

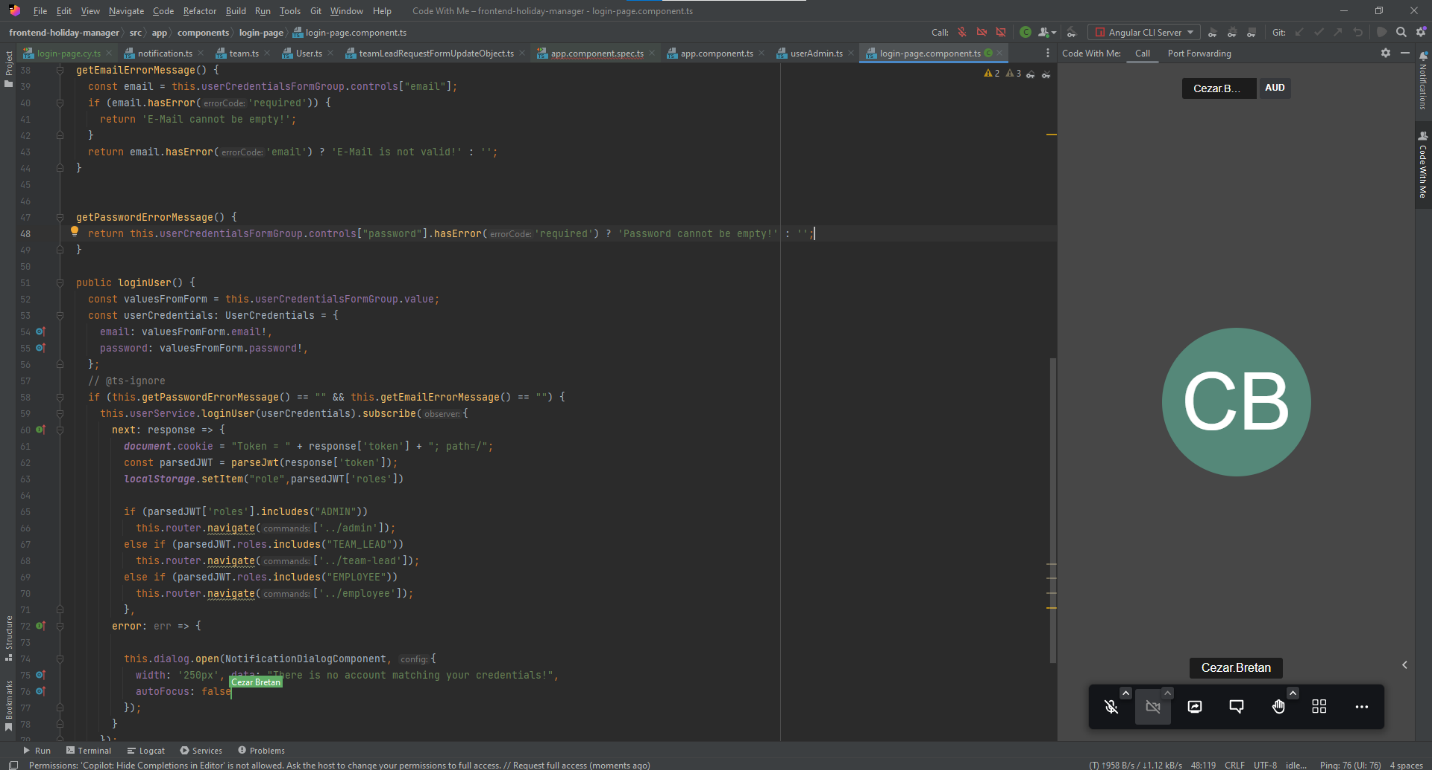
**USAGE IN THREE PROJECTS:**

**PROJECT 1:**

In the first project, we chose to use an unstructured style and not necessarily have some specific rules regarding the way we write and structure the code. In order to be able to use such an approach, all the guests which participate in the same session must be at the same level of programming and possess very close knowledge in the means of the project.

In the first project, we used the Code With Me plugin, such that we can write together the getEmailErrorMessage(), getPasswordErrorMessage() and the login cases if the credentials are valid. We didn’t set any rules, one wrote the getEmailErrorMessage() method and getPasswordErrorMessage() method and the other wrote the login cases.

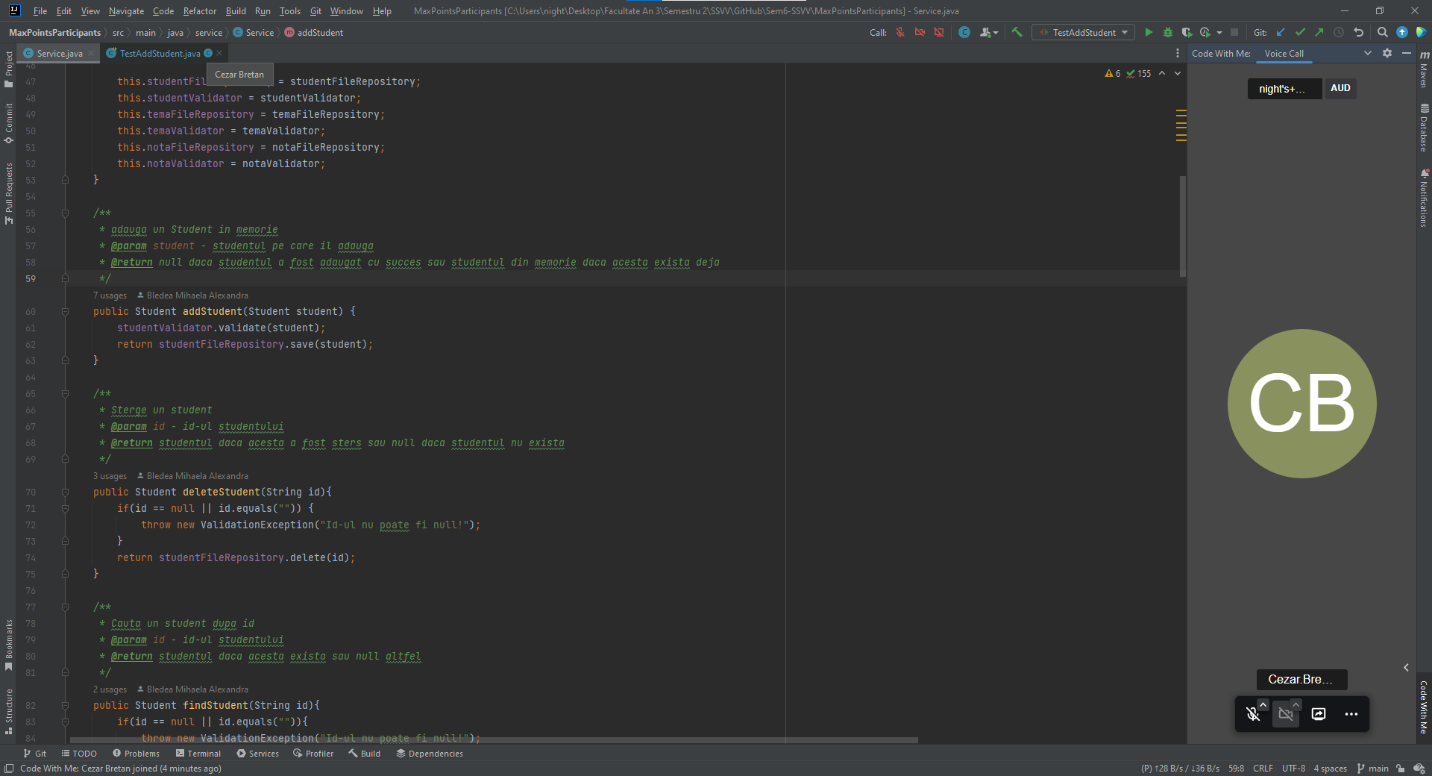
****

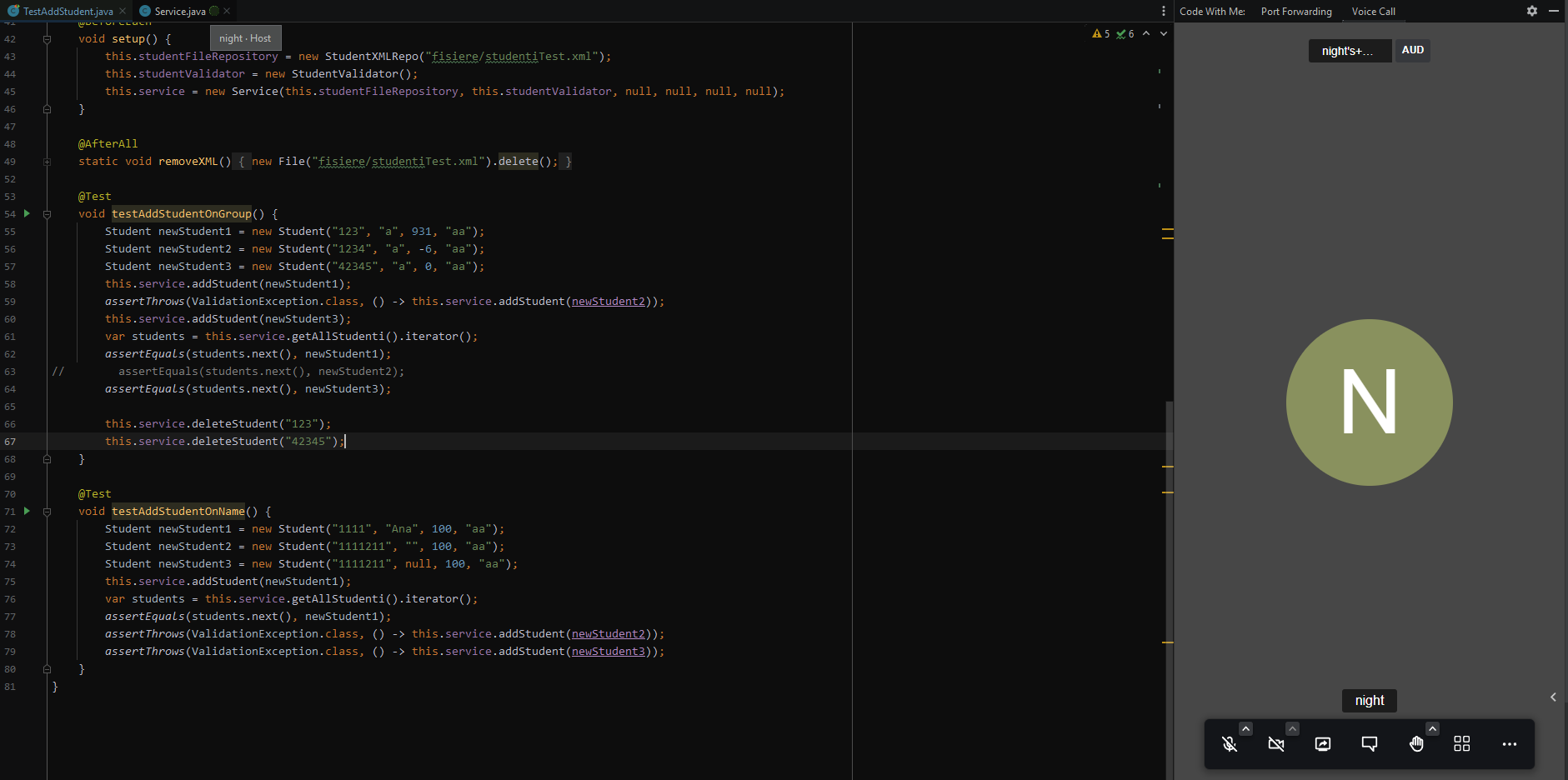
****

**PROJECT 2:**

In the second project, we chose to use the Ping-Pong style. With the ping-pong approach, one developer writes a test, and the other developer makes the test pass. Each person alternates between writing and passing tests. When two developers shift roles regularly, it is unlikely one programmer will control the workflow. This style of pair programming is normally performed in conjunction with [test-driven development](https://www.techtarget.com/searchsoftwarequality/definition/test-driven-development).

Therefore, one writes a functionality and the other tests it. So one of us wrote the addStudent functionality and the other one tested it using two test cases, testAddStudentOnGroup() and testAddStudentOnName().





**PROJECT 3:**

In this project, we chose to use Driver/Navigator style.

The Driver/Navigator approach is a popular pair programming style where one programmer handles the mechanical side like coding, and the other is in control of the strategic or architectural elements like reviewing code. The driver and navigator switch roles often. This style works well for a novice paired with an expert programmer. The navigator role can range from a reserved approach to a tactical hands-on role.

Therefore, with this approach, one of us wrote the class HotelKeeperImplementation and the other one provided feedback regarding the code.

**A screenshot of a computer

Description automatically generated with medium confidence**

**Graphical user interface, text

Description automatically generated**

**Text

Description automatically generated**