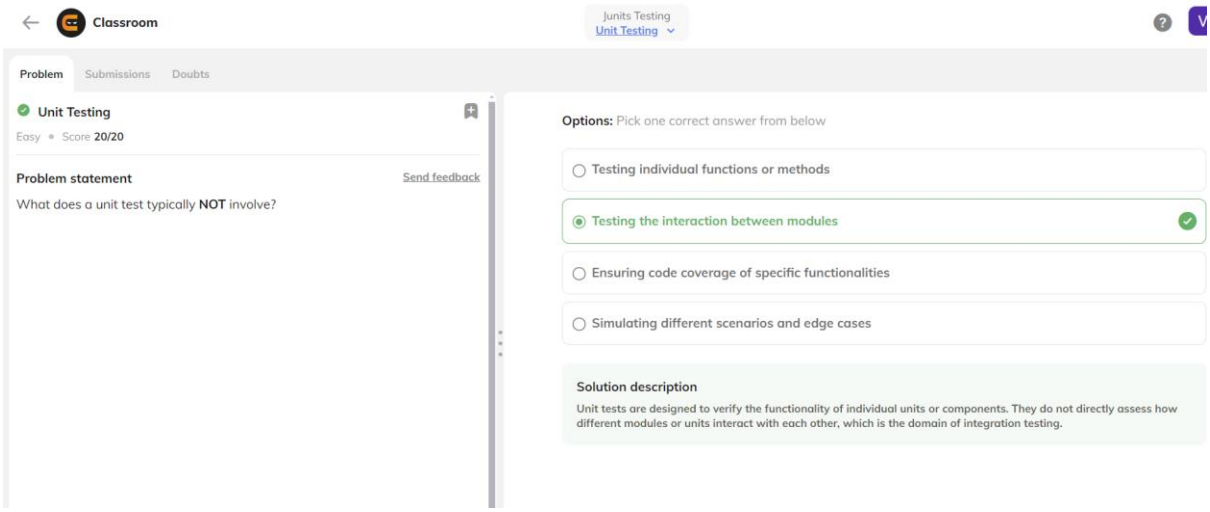
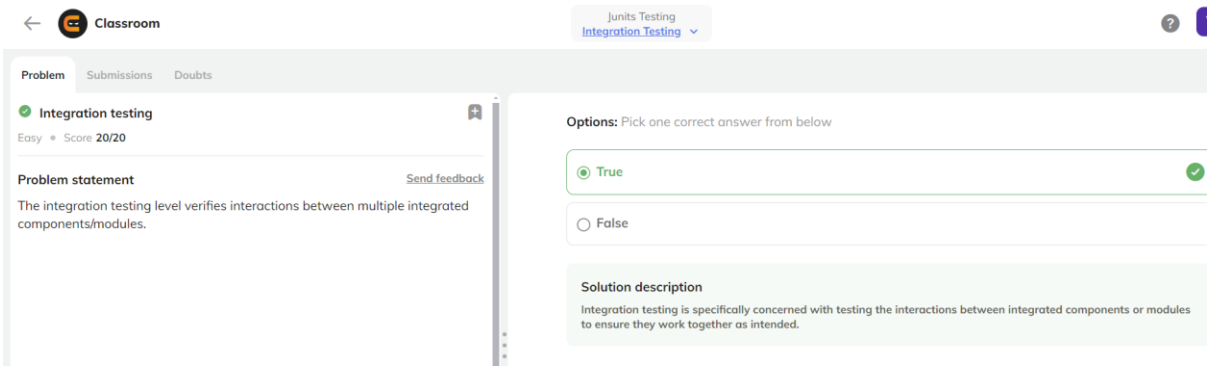
Quiz Questions

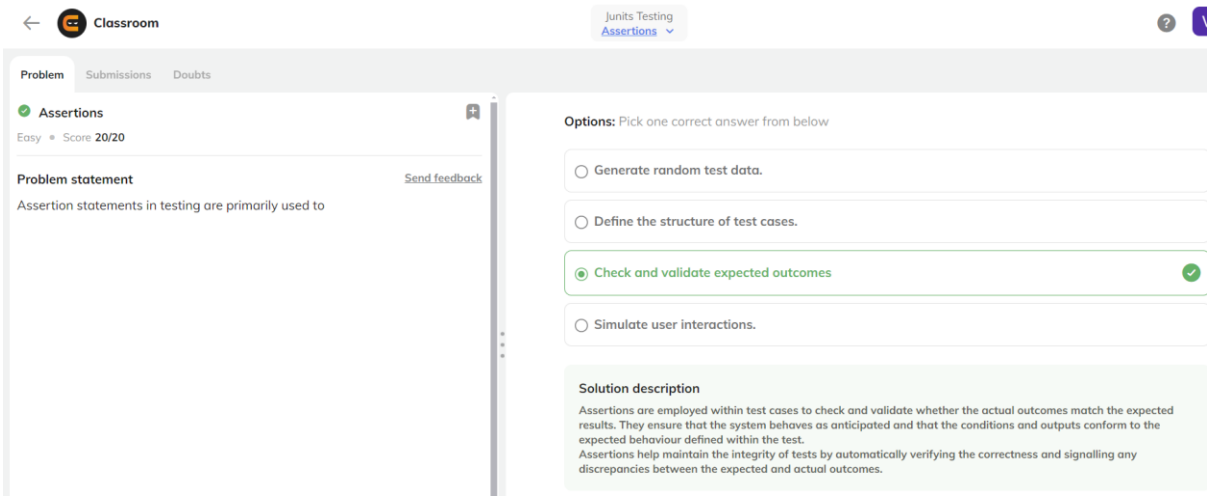
(1) Unit Testing



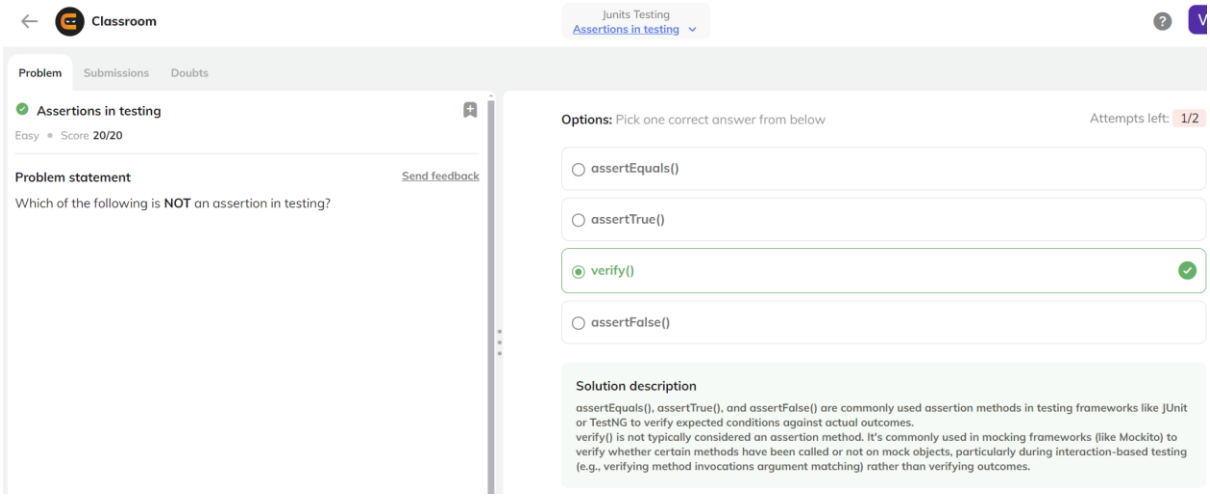
(2) Integration testing



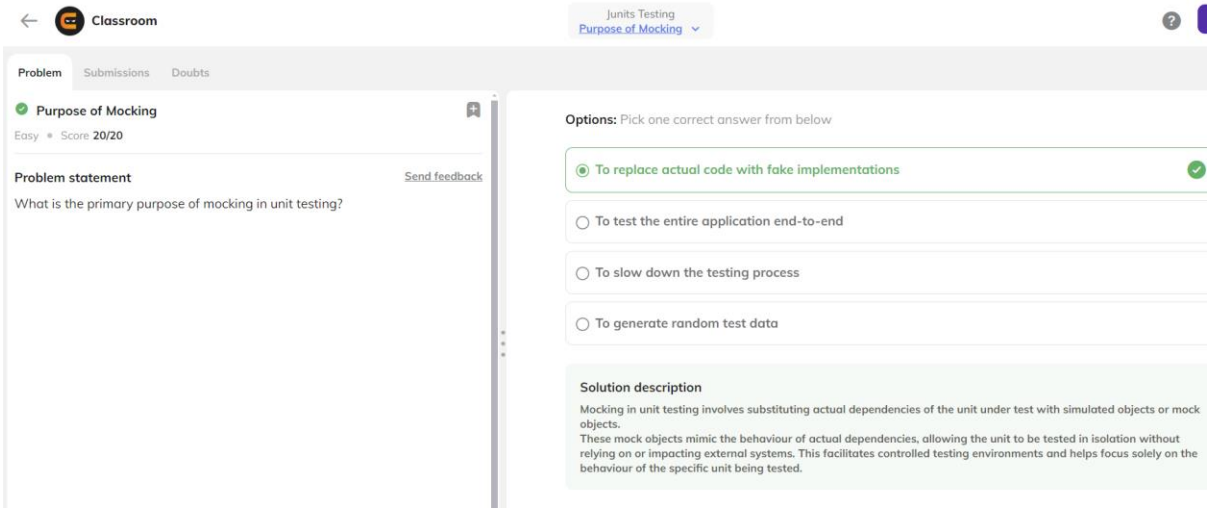
(3) Assertions



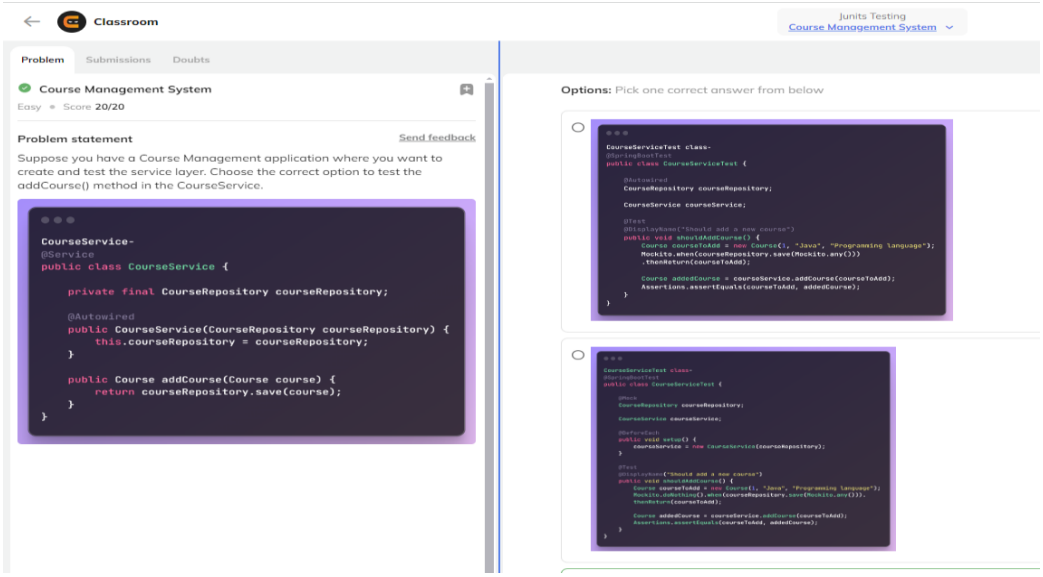
(4) Assertions in testing



(5) Purpose of Mocking



(6) Course Management System



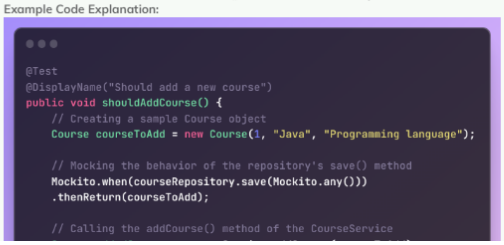
None of the above

Solution description

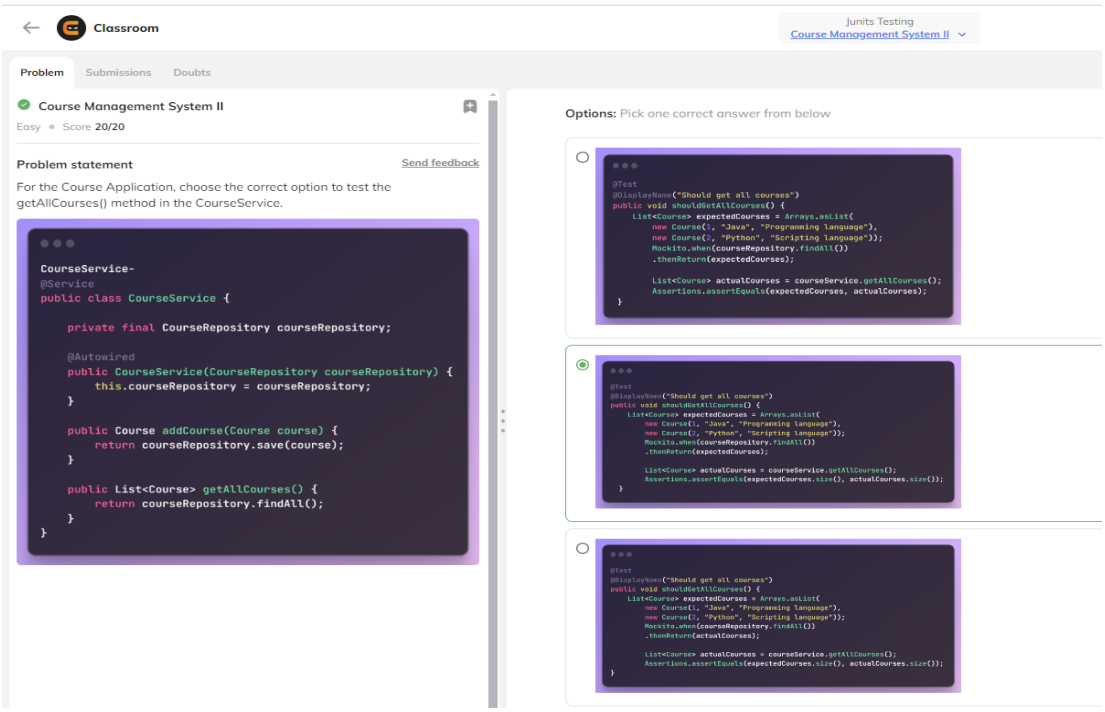
Explanation of the Test:

1. Setup:
   - The test method initializes a CourseService instance and mocks the CourseRepository, simulating the behaviour of the data access layer.
   - A sample Course object (courseToAdd) representing a new course is created with specific attributes (e.g., ID, name, description).
2. Mock Behavior:
   - The behaviour of the CourseRepository is mocked using Mockito's when().thenReturn() construct. It mocks the behaviour of the repository's save() method to simulate adding a course to the database.
3. Execution:
   - The addCourse() method of the CourseService is called with the courseToAdd object as an argument.
4. Assertion:
   - The test verifies whether the addCourse() method returns the expected result. In this case, it expects the addCourse() method to return the same Course object passed to it (courseToAdd).

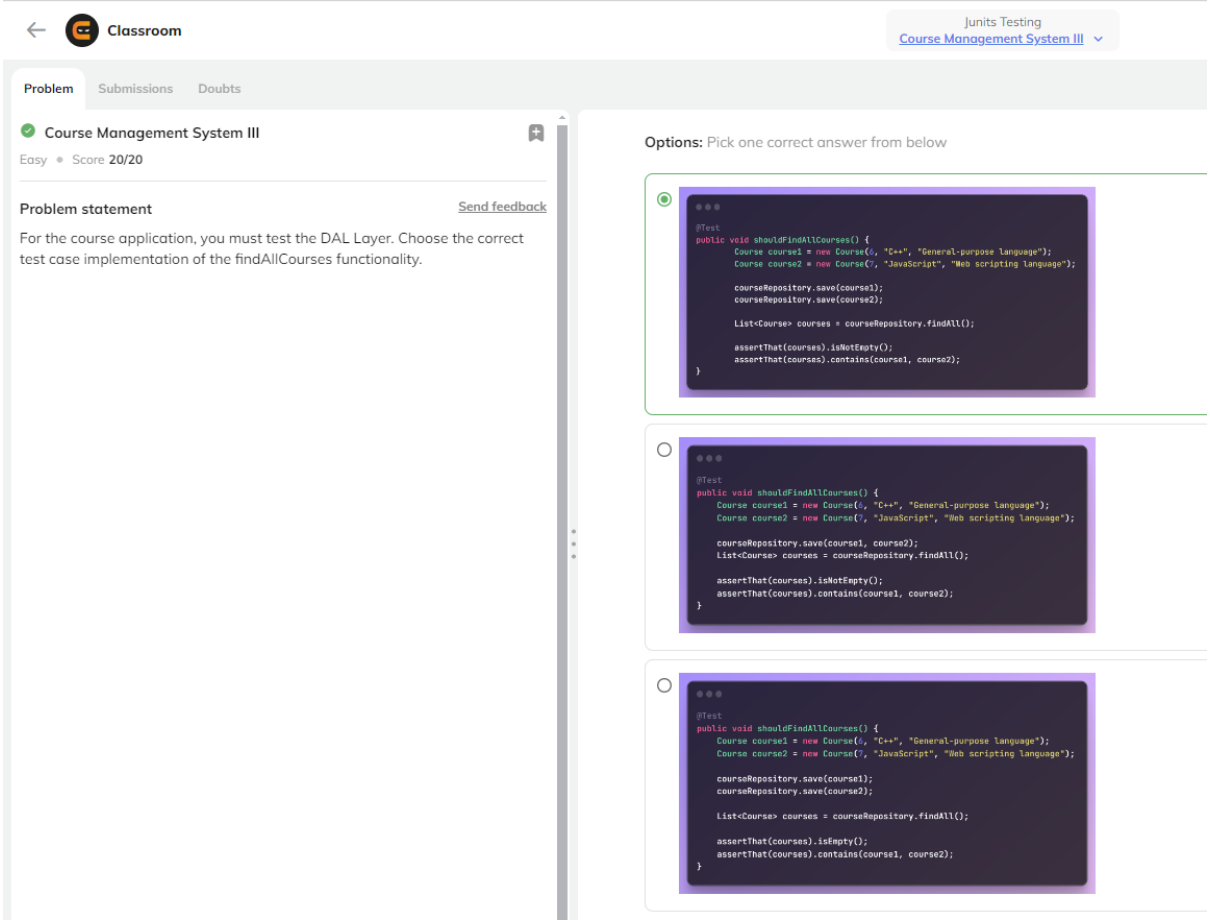
Example Code Explanation:



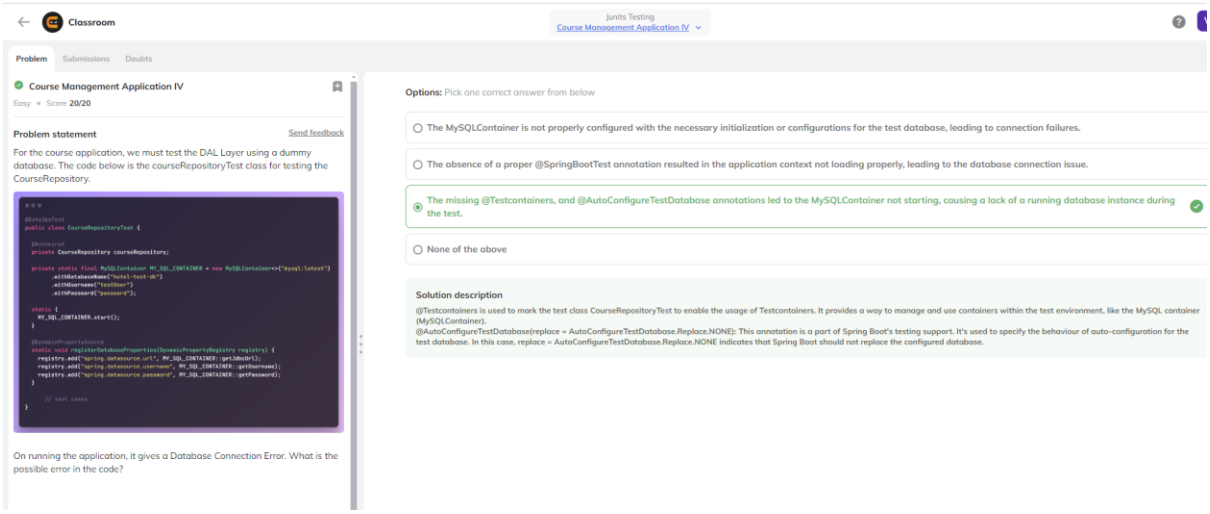
(7) Course Management System II



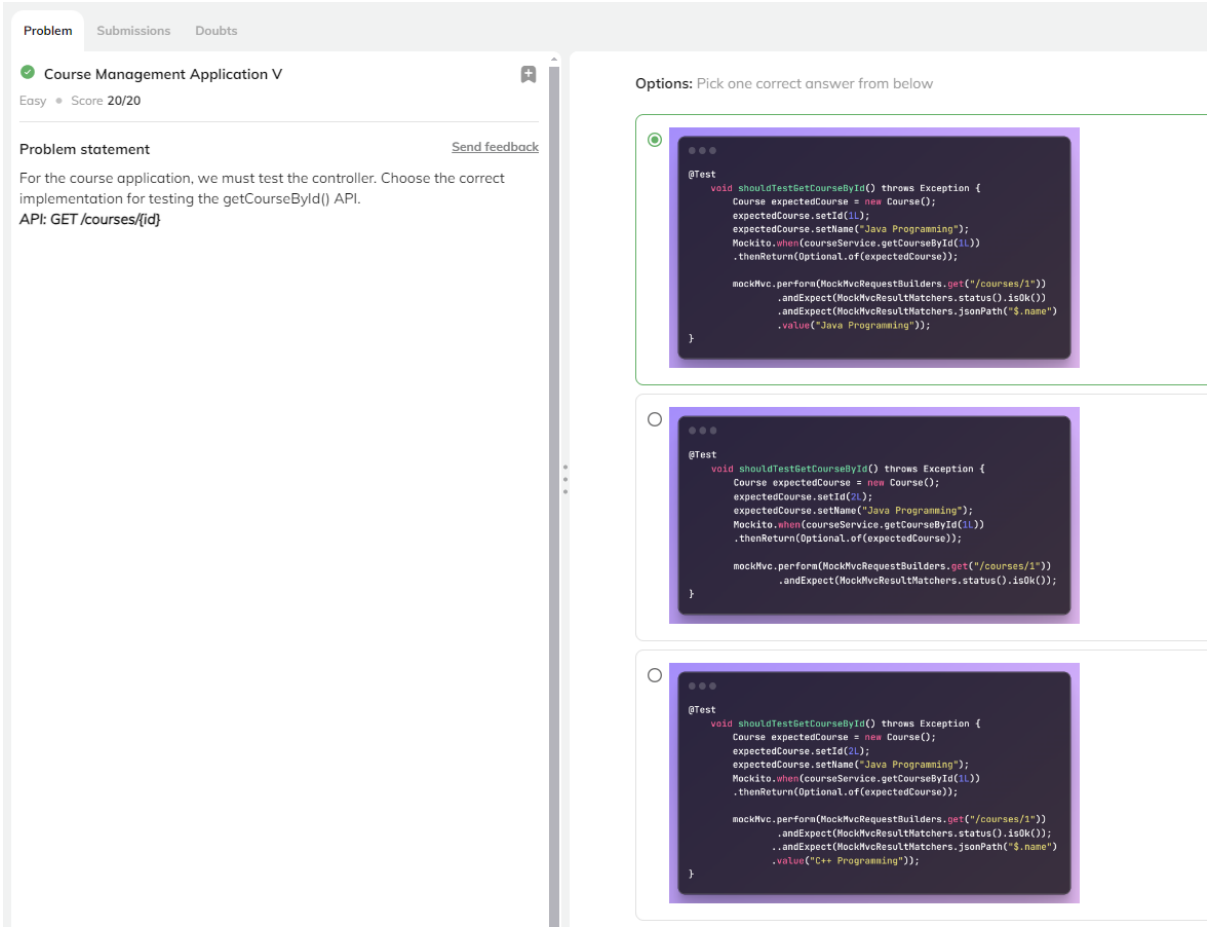
(8) Course Management System III



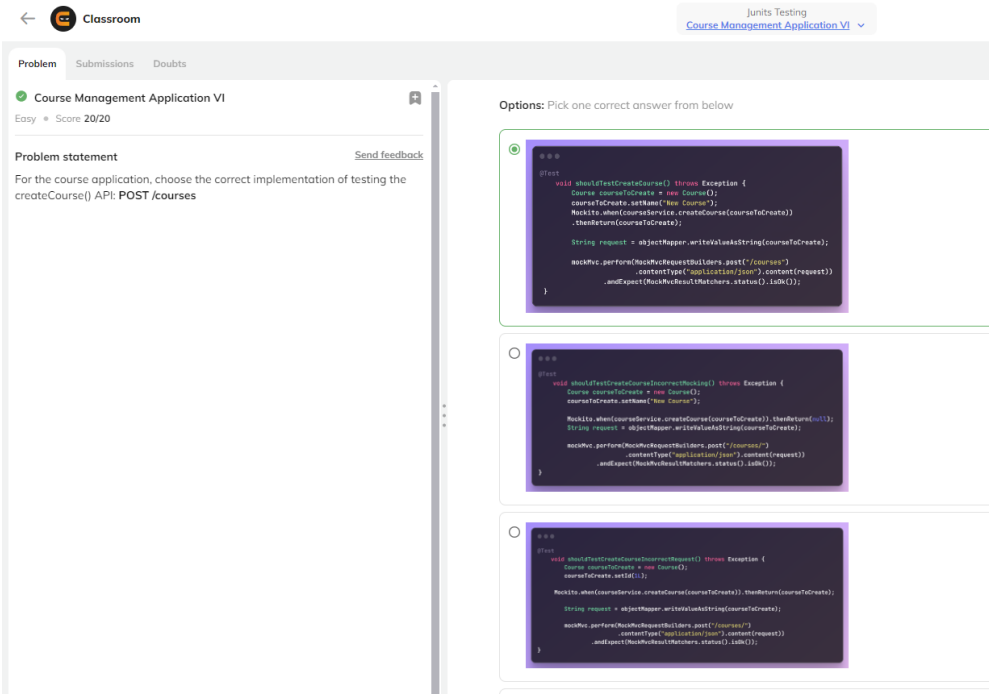
(9) Course Management Application IV



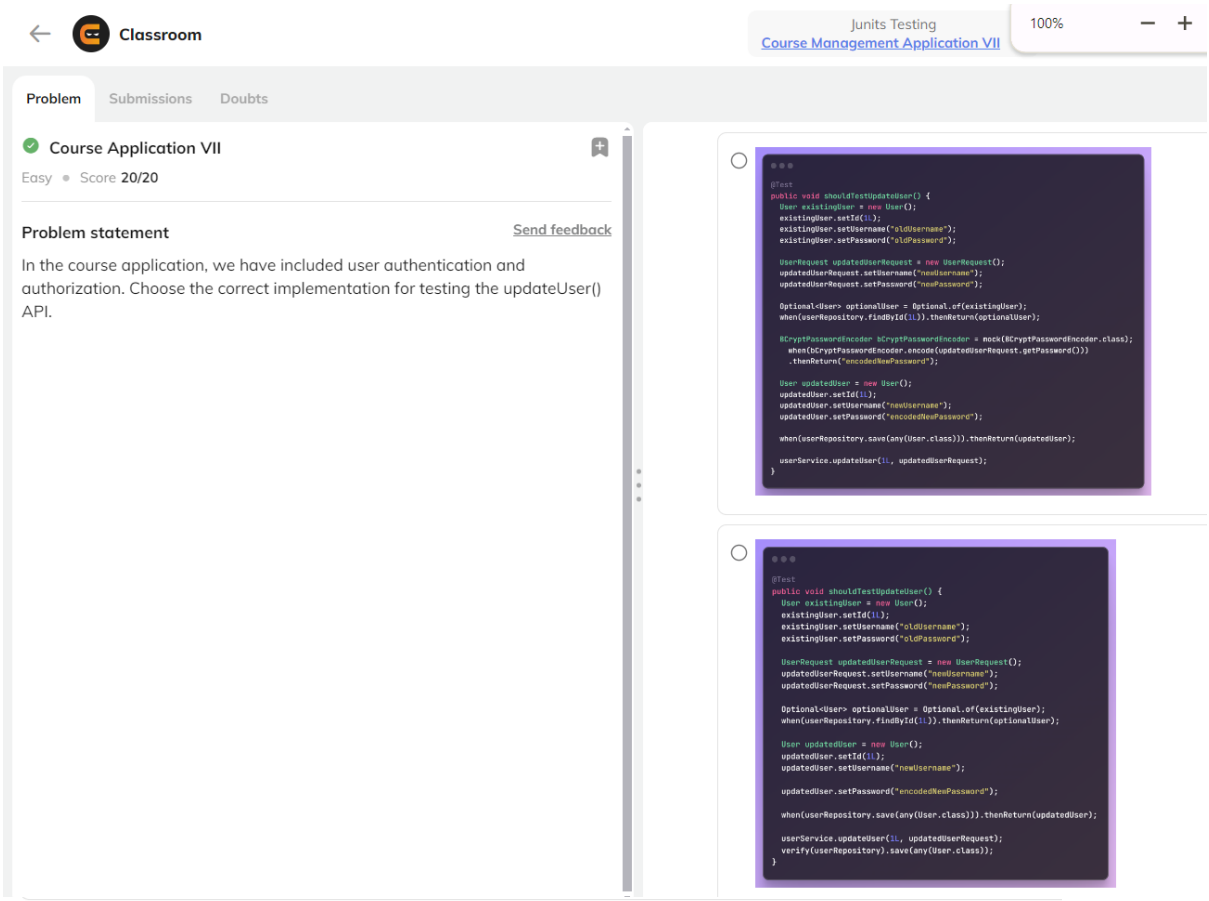
(10) Course Management Application V



(11) Course Management Application VI



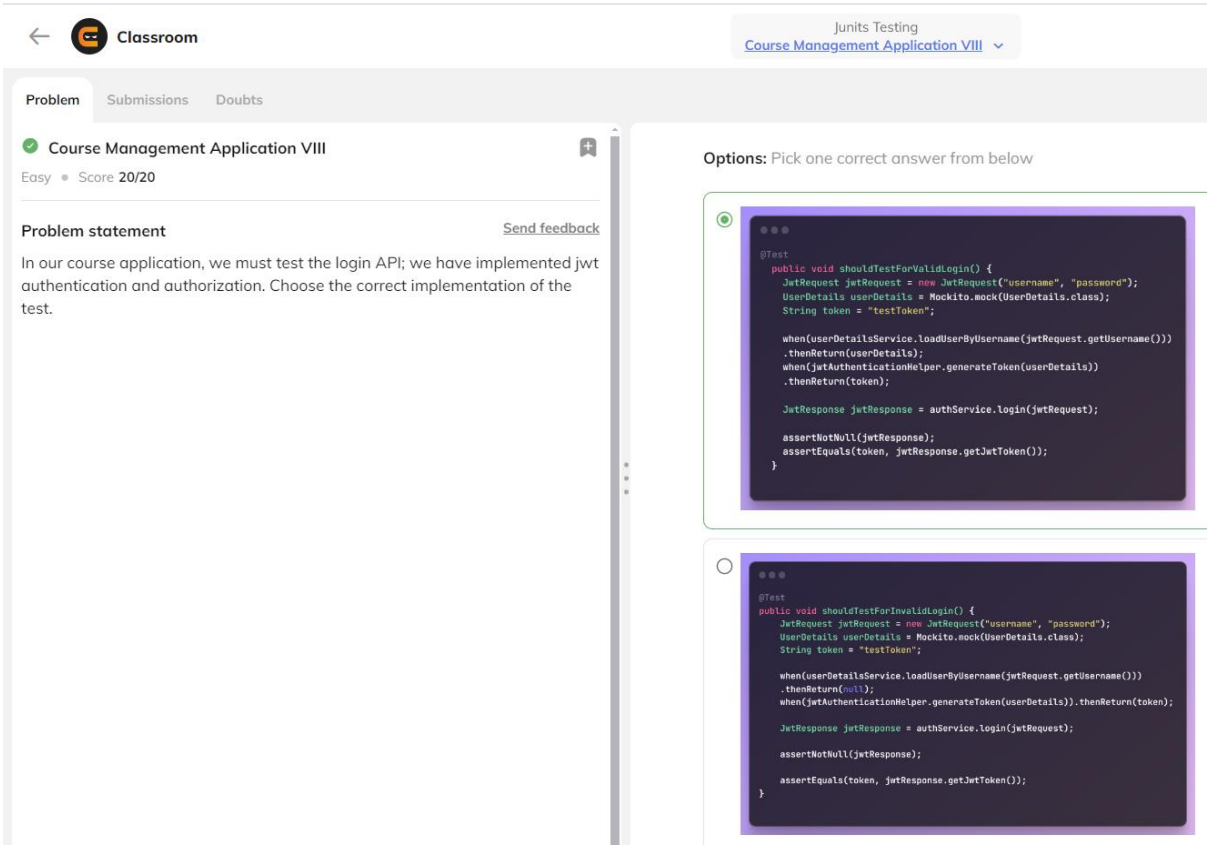
(12) Course Application VII



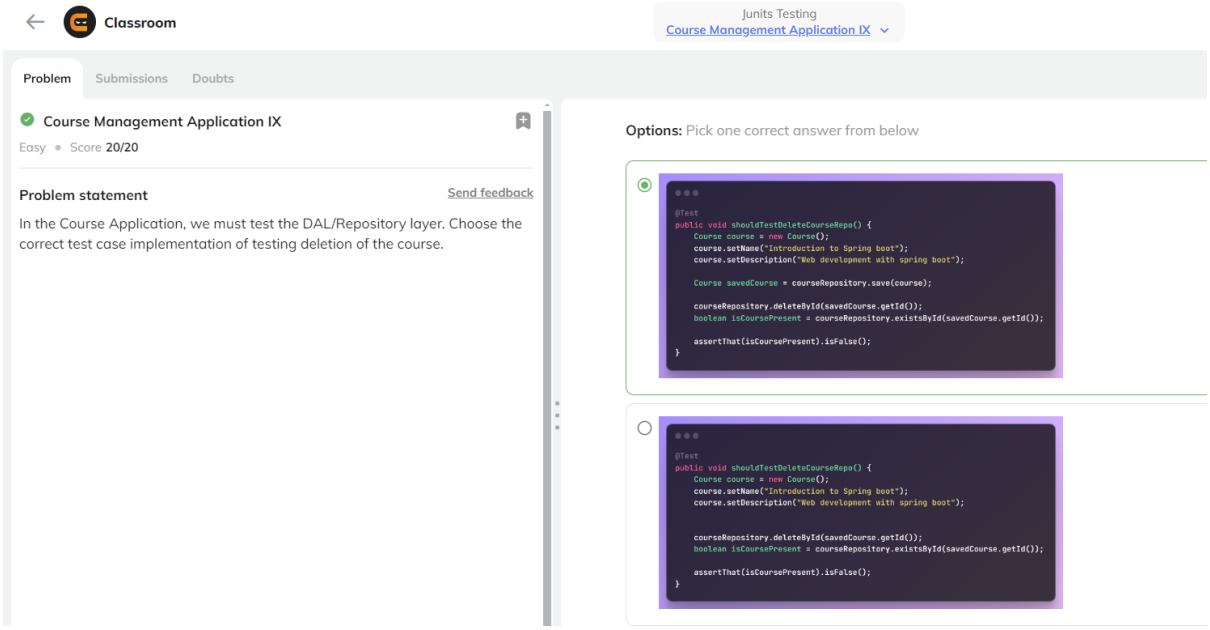
Solution description

1. Setup:
   - The test method initializes an existing User object (existingUser) representing a user in the system with some predefined details.
   - It creates a UserRequest object (updatedUserRequest) containing the updated information for the user.
2. Mock Behavior:
   - The userRepository.findById() method is mocked to return an Optional containing the existingUser when searching by the user's ID.
3. Updating User Information:
   - The updateUser() method of the UserService is invoked, passing the user ID and the updated user details.
4. Assertions and Verification:
   - The test validates whether the updateUser() method correctly handles the update process by ensuring the repository's save() method is called with the updated user information.
   - It verifies that the repository's save() method is called with the updated user object.

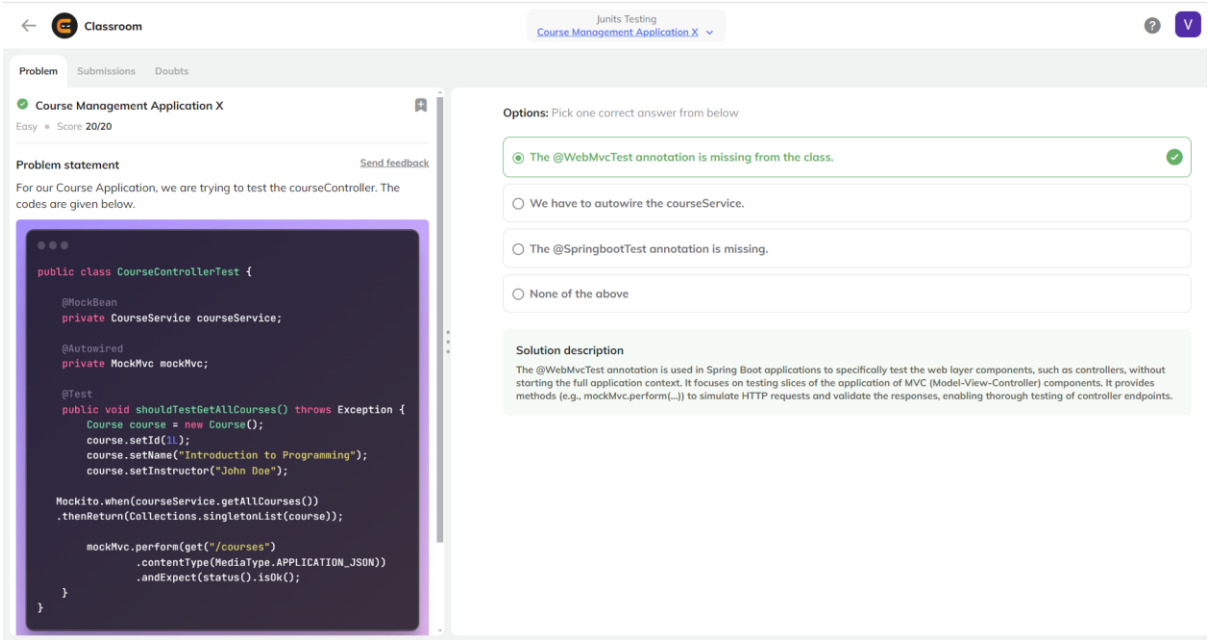
(13) Course Management Application VIII



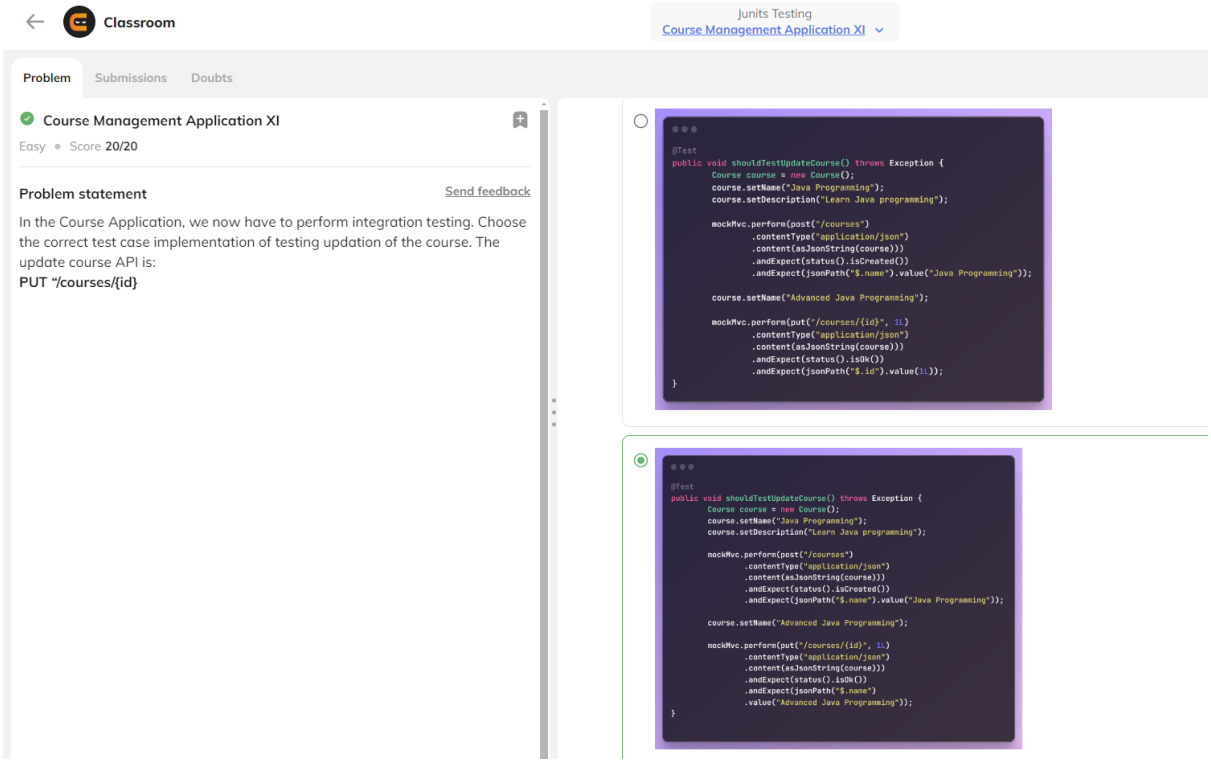
(14) Course Management Application IX



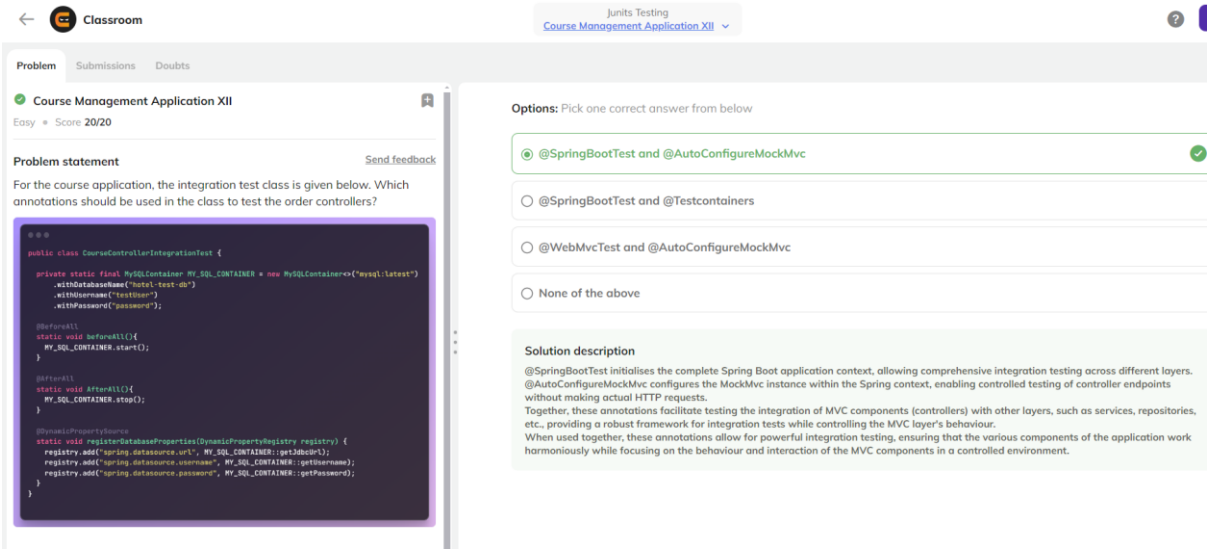
(15) Course Management Application X



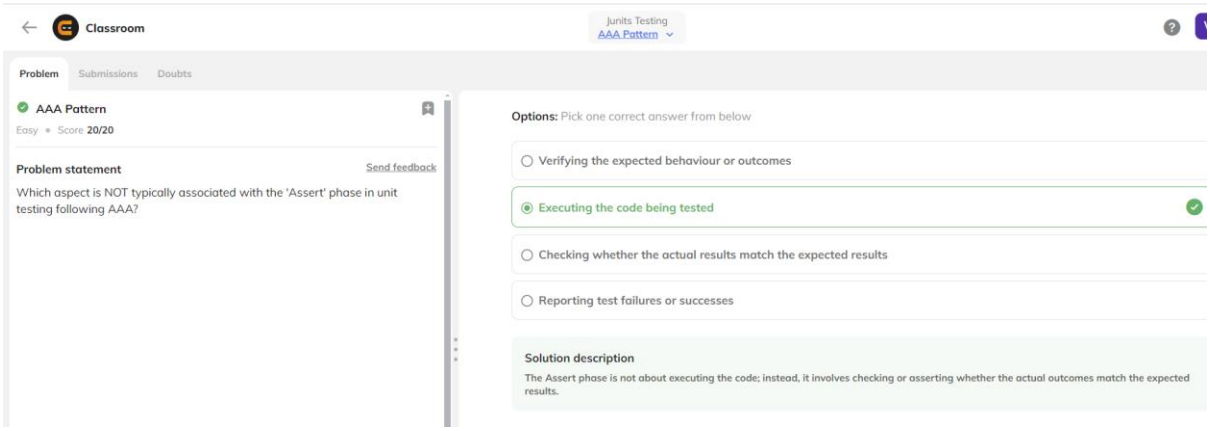
(16) Course Management Application XI



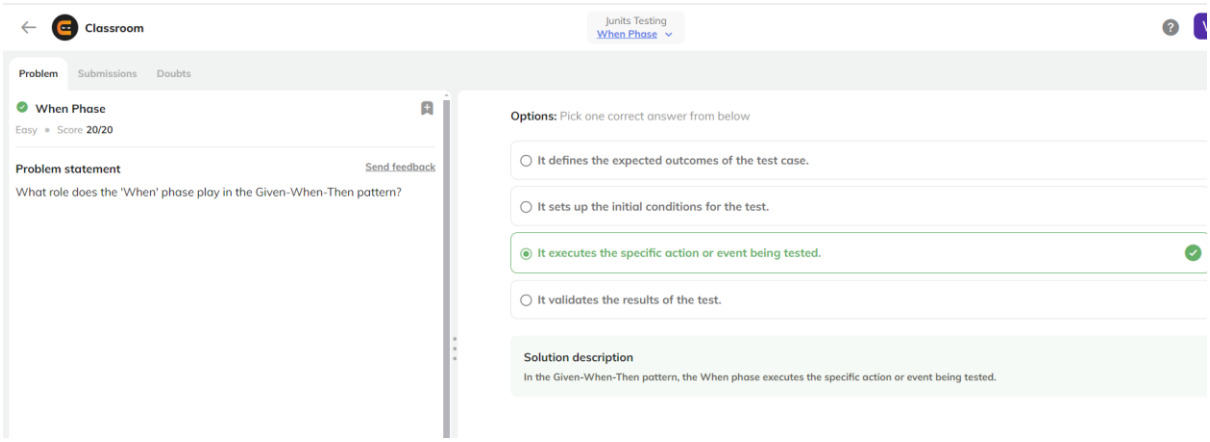
(17) Course Management Application XII



(18) AAA Pattern



(19) When Phase



(20) Timeout in test cases