**Integration Testing**

Integration testing focuses on testing the interfaces between modules. We'll perform integration testing on a subset of units in our application to ensure the modules interact correctly. We'll use graph coverage to visualize and ensure comprehensive integration testing.

**Selected Units for Integration Testing**

We'll focus on integrating the following units:

1. **UserManager**: Handles user-related operations.
2. **CourseManager**: Manages courses.
3. **TaskManager**: Manages tasks related to courses.
4. **ScheduleManager**: Manages the schedule of courses and tasks.

**Integration Points**

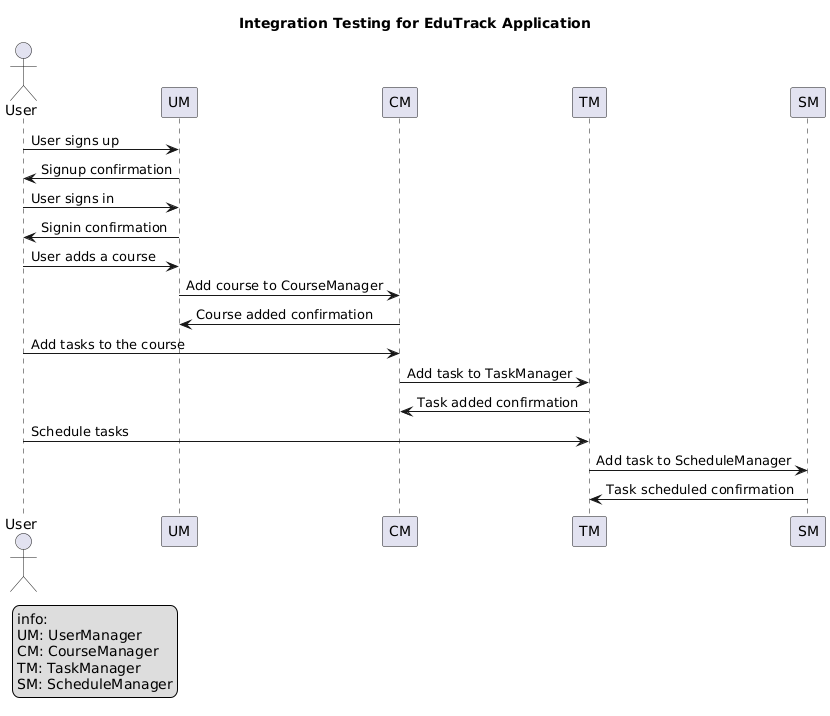
1. **UserManager** and **CourseManager**: Ensure a user can manage courses after signing up.
2. **CourseManager** and **TaskManager**: Ensure tasks can be added to courses.
3. **TaskManager** and **ScheduleManager**: Ensure tasks are correctly scheduled.

**Test Suits and Execution**

**Integration Test Suite**

1. **UserManager and CourseManager Integration**:
   * Verify a user can add a course after signing up.
2. **CourseManager and TaskManager Integration**:
   * Verify tasks can be added to a course.
3. **TaskManager and ScheduleManager Integration**:
   * Verify tasks are scheduled correctly.

**Diagram for Integration Testing**



**Java Implementation for Integration Tests**

**package** PathTesting;

**import** Proj\_375\_Classes.UserManager;

**import** Proj\_375\_Classes.CourseManager;

**import** Proj\_375\_Classes.TaskManager;

**import** Proj\_375\_Classes.ScheduleManager;

**import** Proj\_375\_Classes.User;

**import** Proj\_375\_Classes.Course;

**import** Proj\_375\_Classes.Task;

**import** org.junit.jupiter.api.BeforeEach;

**import** org.junit.jupiter.api.Test;

**import** java.util.Date;

**import** **static** org.junit.jupiter.api.Assertions.\*;

**class** IntegrationTest {

**private** UserManager userManager;

**private** CourseManager courseManager;

**private** TaskManager taskManager;

**private** ScheduleManager scheduleManager;

@BeforeEach

**void** setUp() {

userManager = **new** UserManager();

courseManager = **new** CourseManager();

taskManager = **new** TaskManager(courseManager); // Using the default constructor

scheduleManager = **new** ScheduleManager(taskManager); // Using the default constructor

}

@Test

**void** testUserManagerAndCourseManagerIntegration() {

userManager.signup("user1", "Password123!", "Password123!", "user1@example.com", "User One");

**boolean** courseAdded = courseManager.addCourse("CSE101", "Intro to Computer Science", 2024, "Fall");

*assertTrue*(courseAdded, "Course should be added successfully.");

}

@Test

**void** testCourseManagerAndTaskManagerIntegration() {

courseManager.addCourse("CSE101", "Intro to Computer Science", 2024, "Fall");

Date dueDate = **new** Date();

**boolean** taskAdded = taskManager.addTask("Assignment 1", dueDate, "CSE101", "Fall");

*assertTrue*(taskAdded, "Task should be added to the course successfully.");

}

@Test

**void** testTaskManagerAndScheduleManagerIntegration() {

Date dueDate = **new** Date();

**boolean** scheduleAdded = taskManager.addTask("Assignment 1", dueDate, "CSE101", "Fall");

*assertTrue*(scheduleAdded, "Task should be scheduled successfully.");

}

}

**Explanation**

1. **UserManager and CourseManager Integration**:
   * Tests if a user can add a course after signing up.
2. **CourseManager and TaskManager Integration**:
   * Tests if tasks can be added to a course.
3. **TaskManager and ScheduleManager Integration**:
   * Tests if tasks are scheduled correctly.

**Execution of Test Cases**

1. **Set up the Project**: Ensure your project is correctly set up with JUnit 5. Add the necessary dependencies if they are not already added.
2. **Run the Tests**: Execute the JUnit tests in your IDE or through a build tool like Maven or Gradle.
3. **Review Results**: Check the results to ensure all tests pass and verify the integration points.

These tests ensure that the selected units interact correctly, covering the essential integration points in the EduTrack application.