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.

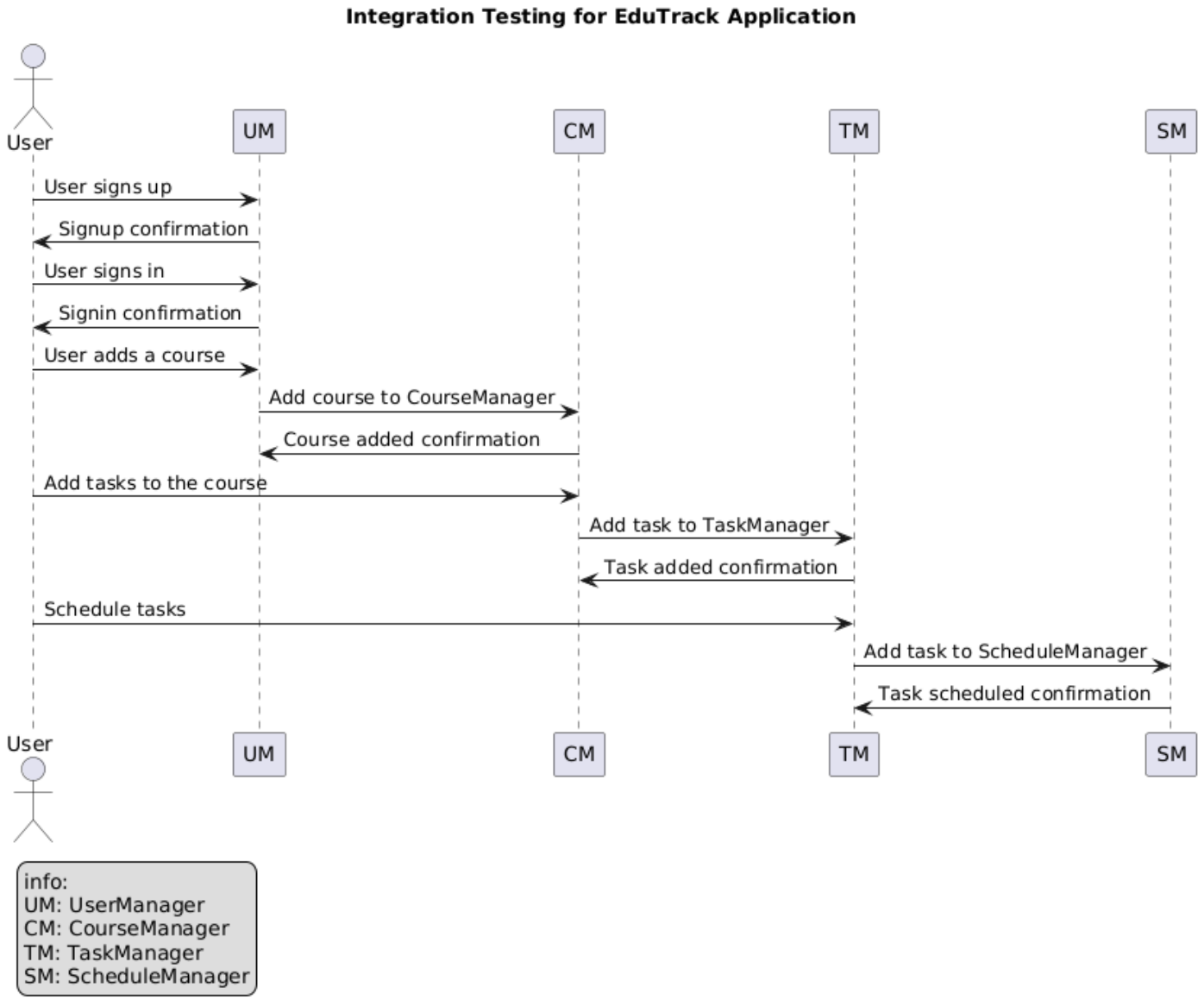
Integration Points

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.

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