Tài liệu kiểm thử

MỤC LỤC

[Tài liệu Kiểm thử cho API và Dịch vụ Backend 3](#_Toc200798510)

[1. Tổng quan 3](#_Toc200798511)

[1.1. Mục đích 3](#_Toc200798512)

[1.2. Phạm vi 3](#_Toc200798513)

[2. Môi trường Kiểm thử 4](#_Toc200798514)

[2.1. Công cụ và Phụ thuộc 4](#_Toc200798515)

[2.2 Hướng dẫn Thiết lập 4](#_Toc200798516)

[3. Các File Kiểm thử và Trường hợp Kiểm thử 5](#_Toc200798517)

[3.1. assessmentController.test.js 5](#_Toc200798518)

[3.2. classController.test.js 7](#_Toc200798519)

[3.3. sprintService.test.js 9](#_Toc200798520)

[3.4. memberService.test.js 11](#_Toc200798521)

[3.5. taskService.test.js 13](#_Toc200798522)

[3.6. assessment-routes.test.js 13](#_Toc200798523)

[3.7. task-routes.test.js 16](#_Toc200798524)

[4. Thực thi Kiểm thử 17](#_Toc200798525)

[5. Giả định và Ràng buộc 18](#_Toc200798526)

[6. Bảo trì 18](#_Toc200798527)

[7. Phụ lục 18](#_Toc200798528)

[Tài liệu Kiểm thử cho Giao diện người dùng (Frontend) 20](#_Toc200798529)

[1. Tổng quan 20](#_Toc200798530)

[1.1. Mục đích 20](#_Toc200798531)

[1.2. Phạm vi 20](#_Toc200798532)

[2. Môi trường Kiểm thử 21](#_Toc200798533)

[2.1. Công cụ và Phụ thuộc 21](#_Toc200798534)

[2.2. Hướng dẫn Thiết lập 21](#_Toc200798535)

[3. Các File Kiểm thử và Trường hợp Kiểm thử 21](#_Toc200798536)

[3.1. Kiểm thử Logic (Hooks) 21](#_Toc200798537)

[3.2. Kiểm thử Thành phần Giao diện (Components) 26](#_Toc200798538)

[3.4. Kiểm thử Hàm Tiện ích (Utils) 30](#_Toc200798539)

[4. Thực thi Kiểm thử 32](#_Toc200798540)

[5. Giả định và Ràng buộc 32](#_Toc200798541)

[6. Bảo trì 33](#_Toc200798542)

[7. Phụ lục 33](#_Toc200798543)

# Tài liệu Kiểm thử cho API và Dịch vụ Backend

## 1. Tổng quan

Tài liệu này mô tả chi tiết các kiểm thử đơn vị (unit test) và kiểm thử tích hợp (integration test) cho các bộ điều khiển (controller), dịch vụ (service) và tuyến đường (route) của API backend trong ứng dụng quản lý dự án. Các kiểm thử đảm bảo rằng các điểm cuối API và logic dịch vụ hoạt động đúng, xử lý lỗi phù hợp và đáp ứng các yêu cầu đã đề ra. Bộ kiểm thử sử dụng **Jest** cho kiểm thử đơn vị và **Supertest** cho kiểm thử tích hợp các tuyến đường Express.

### 1.1. Mục đích

* Xác thực chức năng của các bộ điều khiển (assessmentController, classController), dịch vụ (sprintService, memberService, taskService) và tuyến đường (assessment-routes, task-routes).
* Đảm bảo xử lý lỗi đúng cách cho dữ liệu đầu vào không hợp lệ, truy cập không được phép và lỗi cơ sở dữ liệu.
* Xác nhận rằng hệ thống hoạt động theo đúng hành vi mong đợi cho các trường hợp thành công và trường hợp biên.

### 1.2. Phạm vi

**Các file kiểm thử:**

* assessmentController.test.js: Kiểm thử bộ điều khiển đánh giá cho các đánh giá ngang hàng và giảng viên.
* classController.test.js: Kiểm thử bộ điều khiển lớp học cho các thao tác quản lý lớp (ví dụ: nhập lớp, tham gia, tạo lớp).
* sprintService.test.js: Kiểm thử dịch vụ sprint cho các thao tác liên quan đến sprint.
* memberService.test.js: Kiểm thử dịch vụ thành viên cho dữ liệu nhóm và thành viên.
* taskService.test.js: Kiểm thử dịch vụ nhiệm vụ cho các thao tác liên quan đến nhiệm vụ.
* assessment-routes.test.js: Kiểm thử các tuyến đường API đánh giá.
* task-routes.test.js: Kiểm thử các tuyến đường API nhiệm vụ.

**Loại kiểm thử**: Kiểm thử đơn vị cho dịch vụ và kiểm thử tích hợp cho bộ điều khiển và tuyến đường.

**Loại trừ**: Kiểm thử đầu cuối (end-to-end), kiểm thử hiệu năng và kiểm thử giao diện người dùng không thuộc phạm vi.

## 2. Môi trường Kiểm thử

### 2.1. Công cụ và Phụ thuộc

* **Khung kiểm thử**: Jest (^29.x.x)
* Thư viện kiểm thử API: Supertest (^6.x.x)
* **Khung web**: Express (^4.x.x)
* **Giả lập (Mocking)**: Hàm giả lập của Jest cho các mô hình Sequelize và dịch vụ bên ngoài
* Phụ thuộc khác:
  + multer (^1.x.x) cho tải tệp trong classController.test.js
  + xlsx (^0.x.x) cho phân tích tệp Excel trong classController.test.js
  + sequelize (^6.x.x) cho giả lập mô hình cơ sở dữ liệu
* **Phiên bản Node.js**: ^18.x.x hoặc ^20.x.x
* **Hệ điều hành**: Hỗ trợ đa nền tảng (đã kiểm thử trên macOS/Linux/Windows)

### 2.2 Hướng dẫn Thiết lập

1. Cài đặt phụ thuộc: npm install --save-dev jest supertest express multer xlsx sequelize
2. Cấu hình Jest trong package.json hoặc jest.config.js:

{

"scripts": {

"test": "jest"

},

"jest": {

"testEnvironment": "node"

}

}

1. Đảm bảo các phụ thuộc giả lập (ví dụ: mô hình Sequelize, dịch vụ) được định nghĩa chính xác trong các file kiểm thử.
2. Chạy kiểm thử: **npm test**

## 3. Các File Kiểm thử và Trường hợp Kiểm thử

### 3.1. assessmentController.test.js

**Mục đích**: Xác thực các phương thức của assessmentController để xử lý đánh giá ngang hàng, thống kê nhiệm vụ và đánh giá giảng viên.

**Các trường hợp kiểm thử:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **AC-001** | Lấy danh sách đánh giá ngang hàng thành công | GET /peer-assessments/1/1/1 | Trạng thái: 200, Nội dung: Mảng đánh giá | Đạt |
| **AC-002** | Trả về 400 cho ID không hợp lệ | GET /peer-assessments/invalid/1/1 | Trạng thái: 400, Thông điệp: "ID không hợp lệ" | Đạt |
| **AC-003** | Trả về 404 nếu không tìm thấy dự án | GET /peer-assessments/1/1/1 (giả lập dự án null) | Trạng thái: 404, Thông điệp: "Không tìm thấy dự án" | Đạt |
| **AC-004** | Trả về 403 nếu người đánh giá không thuộc nhóm | GET /peer-assessments/1/1/1 (giả lập thành viên nhóm null) | Trạng thái: 403, Thông điệp: "Người dùng không thuộc nhóm" | Đạt |
| **AC-005** | Lưu đánh giá ngang hàng thành công | POST /assessments/1/1 với dữ liệu hợp lệ | Trạng thái: 200, Thông điệp: "Đã lưu đánh giá" | Đạt |
| **AC-006** | Trả về 400 cho điểm không hợp lệ | POST /assessments/1/1 với điểm > 5 | Trạng thái: 400, Thông điệp: "Điểm không hợp lệ" | Đạt |
| **AC-007** | Trả về 400 nếu tự đánh giá | POST /assessments/1/1 với cùng assessor và assessee | Trạng thái: 400, Thông điệp: "Không thể tự đánh giá" | Đạt |
| **AC-008** | Lấy thống kê nhiệm vụ thành công | GET /task-stats/1 | Trạng thái: 200, Nội dung: Đối tượng thống kê | Đạt |
| **AC-009** | Trả về 400 cho projectId không hợp lệ | GET /task-stats/invalid | Trạng thái: 400, Thông điệp: "projectId không hợp lệ" | Đạt |
| **AC-010** | Lấy thống kê nhiệm vụ thành viên thành công | GET /member-task-stats/1/1 | Trạng thái: 200, Nội dung: Mảng thống kê thành viên | Đạt |
| **AC-011** | Trả về 404 nếu không tìm thấy dự án | GET /member-task-stats/1/1 (giả lập dự án null) | Trạng thái: 404, Thông điệp: "Không tìm thấy dự án" | Đạt |
| **AC-012** | Cập nhật đánh giá ngang hàng thành công | PUT /assessments/1/1/1 với dữ liệu hợp lệ | Trạng thái: 200, Thông điệp: "Đã cập nhật đánh giá" | Đạt |
| **AC-013** | Trả về 404 nếu không tìm thấy đánh giá | PUT /assessments/1/1/1 (giả lập đánh giá null) | Trạng thái: 404, Thông điệp: "Không tìm thấy đánh giá" | Đạt |
| **AC-014** | Lấy danh sách đánh giá giảng viên thành công | GET /instructor-evaluations/1/1 | Trạng thái: 200, Nội dung: Mảng đánh giá | Đạt |
| **AC-015** | Trả về 400 cho ID không hợp lệ | GET /instructor-evaluations/invalid/1 | Trạng thái: 400, Thông điệp: "ID không hợp lệ" | Đạt |
| **AC-016** | Lưu đánh giá giảng viên thành công | POST /instructor-evaluations/1/1 với dữ liệu hợp lệ | Trạng thái: 201, Thông điệp: "Đã lưu đánh giá" | Đạt |
| **AC-017** | Trả về 403 nếu không phải giảng viên | POST /instructor-evaluations/1/1 (giả lập giảng viên khác) | Trạng thái: 403, Thông điệp: "Không được phép" | Đạt |
| **AC-018** | Cập nhật đánh giá giảng viên thành công | PUT /instructor-evaluations/1/1/1 với dữ liệu hợp lệ | Trạng thái: 200, Thông điệp: "Đã cập nhật đánh giá" | Đạt |
| **AC-019** | Trả về 404 nếu không tìm thấy đánh giá | PUT /instructor-evaluations/1/1/1 (giả lập đánh giá null) | Trạng thái: 404, Thông điệp: "Không tìm thấy đánh giá" | Đạt |

**Phụ thuộc giả lập**: GroupMember, User, PeerAssessment, Task, Sprint, Project, Group, Class, InstructorEvaluation.

### 3.2. classController.test.js

**Mục đích**: Xác thực các phương thức của classController để quản lý các thao tác lớp học, bao gồm nhập lớp, tham gia lớp và các thao tác CRUD.

**Các trường hợp kiểm thử:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **CC-001** | Nhập lớp thành công | POST /classes/1/import với tệp Excel hợp lệ | Trạng thái: 200, Thông điệp: "Nhập thành công" | Đạt |
| **CC-002** | Trả về 400 nếu không tải tệp | POST /classes/1/import không có tệp | Trạng thái: 400, Thông điệp: "Không có tệp" | Đạt |
| **CC-003** | Tham gia lớp thành công | POST /classes/join với mã hợp lệ | Trạng thái: 200, Thông điệp: "Đã tham gia lớp" | Đạt |
| **CC-004** | Trả về 404 nếu không tìm thấy lớp | POST /classes/join với mã không hợp lệ | Trạng thái: 404, Thông điệp: "Không tìm thấy lớp" | Đạt |
| **CC-005** | Trả về 400 nếu đã tham gia | POST /classes/join (giả lập thành viên hiện có) | Trạng thái: 400, Thông điệp: "Đã tham gia" | Đạt |
| **CC-006** | Tìm kiếm lớp thành công | GET /classes/search?searchText=Test | Trạng thái: 200, Nội dung: Mảng lớp học | Đạt |
| **CC-007** | Lấy danh sách lớp của người dùng | GET /classes | Trạng thái: 200, Nội dung: Mảng lớp học | Đạt |
| **CC-008** | Lấy tất cả lớp học | GET /classes/all | Trạng thái: 200, Nội dung: Mảng lớp học | Đạt |
| **CC-009** | Lấy lớp của giảng viên | GET /classes/instructor | Trạng thái: 200, Nội dung: Mảng lớp học | Đạt |
| **CC-010** | Tạo lớp thành công | POST /classes với dữ liệu hợp lệ | Trạng thái: 201, Nội dung: Đối tượng lớp | Đạt |
| **CC-011** | Trả về 400 nếu lớp đã tồn tại | POST /classes (giả lập lớp hiện có) | Trạng thái: 400, Thông điệp: "Lớp đã tồn tại" | Đạt |
| **CC-012** | Xóa lớp thành công | DELETE /classes/1 | Trạng thái: 200, Thông điệp: "Đã xóa lớp" | Đạt |
| **CC-013** | Trả về 404 nếu không tìm thấy lớp | DELETE /classes/1 (giả lập lớp null) | Trạng thái: 404, Thông điệp: "Không tìm thấy lớp" | Đạt |
| **CC-014** | Cập nhật lớp thành công | PUT /classes/1 với dữ liệu hợp lệ | Trạng thái: 200, Nội dung: Lớp đã cập nhật | Đạt |
| **CC-015** | Trả về 404 nếu không tìm thấy lớp | PUT /classes/1 (giả lập lớp null) | Trạng thái: 404, Thông điệp: "Không tìm thấy lớp" | Đạt |

**Phụ thuộc giả lập**: Group, Project, GroupMember, Class, User, ClassMember, fs, xlsx.

### 3.3. sprintService.test.js

**Mục đích**: Xác thực các phương thức của sprintService để quản lý các sprint trong dự án.

**Các trường hợp kiểm thử:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **SS-001** | Lấy danh sách sprint thành công | getSprintsByProjectId(1) | Mảng sprint | Đạt |
| **SS-002** | Ném lỗi nếu thiếu projectId | getSprintsByProjectId() | Lỗi: "Yêu cầu projectId" | Đạt |
| **SS-003** | Ném lỗi nếu lỗi cơ sở dữ liệu | getSprintsByProjectId(1) (giả lập lỗi DB) | Lỗi: "Không thể lấy sprint" | Đạt |
| **SS-004** | Lấy ID sprint hiện tại | getTargetSprintIdForFilter(1, 'current') | ID sprint | Đạt |
| **SS-005** | Lấy ID sprint cuối nếu không có sprint hiện tại | getTargetSprintIdForFilter(1, 'current') (giả lập không có sprint hiện tại) | ID sprint cuối | Đạt |
| **SS-006** | Lấy ID sprint cụ thể | getTargetSprintIdForFilter(1, '3') | 3 | Đạt |
| **SS-007** | Trả về null cho bộ lọc 'all' | getTargetSprintIdForFilter(1, 'all') | null | Đạt |
| **SS-008** | Ném lỗi nếu thiếu projectId | getTargetSprintIdForFilter() | Lỗi: "Yêu cầu projectId" | Đạt |
| **SS-009** | Tạo sprint thành công | createSprintForProject(1, { sprint\_name: 'Sprint 1' }) | Đối tượng sprint | Đạt |
| **SS-010** | Ném lỗi nếu thiếu projectId | createSprintForProject() | Lỗi: "Yêu cầu projectId" | Đạt |
| **SS-011** | Xử lý lỗi ràng buộc duy nhất | createSprintForProject(1, ...) (giả lập lỗi unique) | Lỗi: "Số sprint đã tồn tại" | Đạt |
| **SS-012** | Ném lỗi nếu lỗi cơ sở dữ liệu | createSprintForProject(1, ...) (giả lập lỗi DB) | Lỗi: "Không thể tạo sprint" | Đạt |

Phụ thuộc giả lập: Sprint.

### 3.4. memberService.test.js

**Mục đích**: Xác thực các phương thức của memberService để quản lý thành viên nhóm, đánh giá ngang hàng, sprint và thống kê dự án.

**Các trường hợp kiểm thử:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **MS-001** | Lấy nhóm cho giảng viên | getGroupsForUserByClass(1, 1, 'Instructor') | Mảng nhóm | Đạt |
| **MS-002** | Lấy nhóm cho sinh viên | getGroupsForUserByClass(1, 1, 'Student') | Mảng nhóm | Đạt |
| **MS-003** | Trả về mảng rỗng cho vai trò không xác định | getGroupsForUserByClass(1, 1, 'Unknown') | [] | Đạt |
| **MS-004** | Ném lỗi nếu thiếu tham số | getGroupsForUserByClass() | Lỗi: "Thiếu tham số" | Đạt |
| **MS-005** | Lấy thành viên nhóm với chi tiết nhiệm vụ | getGroupMembersWithTaskDetails(1) | Thành viên và dữ liệu sprint | Đạt |
| **MS-006** | Ném lỗi nếu không tìm thấy nhóm | getGroupMembersWithTaskDetails(1) (giả lập nhóm null) | Lỗi: "Không tìm thấy nhóm" | Đạt |
| **MS-007** | Trả về thành viên rỗng nếu không có thành viên | getGroupMembersWithTaskDetails(1) (giả lập thành viên rỗng) | { members: [], sprintData: {} } | Đạt |
| **MS-008** | Lấy đánh giá ngang hàng | getPeerAssessmentsForGroup(1) | Mảng đánh giá ngang hàng | Đạt |
| **MS-009** | Ném lỗi nếu không tìm thấy nhóm | getPeerAssessmentsForGroup(1) (giả lập không có nhóm) | Lỗi: "Không tìm thấy nhóm" | Đạt |
| **MS-010** | Ném lỗi cho groupId không hợp lệ | getPeerAssessmentsForGroup('invalid') | Lỗi: "groupId không hợp lệ" | Đạt |
| **MS-011** | Lấy sprint cho nhóm | getSprintsOfProjectInGroup(1) | Mảng sprint | Đạt |
| **MS-012** | Trả về mảng rỗng nếu không có dự án | getSprintsOfProjectInGroup(1) (giả lập không có dự án) | [] | Đạt |
| **MS-013** | Lấy thống kê dự án | getProjectOverallStatsForGroup(1, 'current') | Đối tượng thống kê | Đạt |
| **MS-014** | Ném lỗi nếu không có dự án | getProjectOverallStatsForGroup(1) (giả lập không có dự án) | Lỗi: "Không tìm thấy dự án" | Đạt |

**Phụ thuộc giả lập**: Group, GroupMember, User, PeerAssessment, sequelize, projectService, taskService, contributionService, sprintService.

### 3.5. taskService.test.js

**Mục đích**: Xác thực các phương thức của taskService để quản lý nhiệm vụ và thống kê nhiệm vụ.

**Các trường hợp kiểm thử:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **TS-001** | Lấy danh sách nhiệm vụ thành công | getTasksByProjectId(1) | Mảng nhiệm vụ | Đạt |
| **TS-002** | Ném lỗi nếu lỗi cơ sở dữ liệu | getTasksByProjectId(1) (giả lập lỗi DB) | Lỗi: "Không thể lấy nhiệm vụ" | Đạt |
| **TS-003** | Lấy thống kê nhiệm vụ với bộ lọc sprint | getTaskStatsForProject(1, 1) | Đối tượng thống kê | Đạt |
| **TS-004** | Lấy thống kê nhiệm vụ không có bộ lọc | getTaskStatsForProject(1) | Đối tượng thống kê | Đạt |
| **TS-005** | Ném lỗi nếu lỗi cơ sở dữ liệu | getTaskStatsForProject(1) (giả lập lỗi DB) | Lỗi: "Không thể lấy thống kê" | Đạt |

Phụ thuộc giả lập: Task, Sprint.

### 3.6. assessment-routes.test.js

**Mục đích**: Xác thực các tuyến đường API được định nghĩa trong assessment-routes.js cho các thao tác liên quan đến đánh giá.

**Các trường hợp kiểm thử:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **AR-001** | Lấy đánh giá ngang hàng | GET /api/assessments/groups/1/projects/1/peerassessments/1 | Trạng thái: 200, Gọi controller | Đạt |
| **AR-002** | Lưu đánh giá ngang hàng | POST /api/assessments/groups/1/projects/1/peerassessments | Trạng thái: 200, Gọi controller | Đạt |
| **AR-003** | Lấy thống kê nhiệm vụ | GET /api/assessments/groups/1/projects/1/task-stats | Trạng thái: 200, Gọi controller | Đạt |
| **AR-004** | Lấy thống kê nhiệm vụ thành viên | GET /api/assessments/groups/1/projects/1/member-task-stats | Trạng thái: 200, Gọi controller | Đạt |
| **AR-005** | Cập nhật đánh giá ngang hàng | PUT /api/assessments/groups/1/projects/1/peerassessments/1 | Trạng thái: 200, Gọi controller | Đạt |
| **AR-006** | Lấy đánh giá giảng viên (giảng viên) | GET /api/assessments/.../instructor-evaluations | Trạng thái: 200, Mảng đánh giá | Đạt |
| **AR-007** | Lấy đánh giá giảng viên (sinh viên) | GET /api/assessments/... (vai trò sinh viên) | Trạng thái: 200, Mảng đánh giá | Đạt |
| **AR-008** | Trả về 400 cho ID không hợp lệ | GET /api/assessments/groups/invalid/... | Trạng thái: 400, Thông điệp: "ID không hợp lệ" | Đạt |
| **AR-009** | Trả về 404 nếu không tìm thấy nhóm | GET /api/assessments/... (giả lập nhóm null) | Trạng thái: 404, Thông điệp: "Không tìm thấy nhóm" | Đạt |
| **AR-010** | Trả về 403 cho giảng viên sai | GET /api/assessments/... (giả lập giảng viên khác) | Trạng thái: 403, Thông điệp: "Không được phép" | Đạt |
| **AR-011** | Trả về 403 cho sinh viên không trong nhóm | GET /api/assessments/... (giả lập không có thành viên) | Trạng thái: 403, Thông điệp: "Không trong nhóm" | Đạt |
| **AR-012** | Trả về 403 cho vai trò không hợp lệ | GET /api/assessments/... (vai trò admin) | Trạng thái: 403, Thông điệp: "Vai trò không hợp lệ" | Đạt |
| **AR-013** | Lưu đánh giá giảng viên | POST /api/assessments/.../instructor-evaluations | Trạng thái: 201, Gọi controller | Đạt |
| **AR-014** | Cập nhật đánh giá giảng viên | PUT /api/assessments/.../instructor-evaluations/1 | Trạng thái: 200, Gọi controller | Đạt |

**Phụ thuộc giả lập**: Group, Project, Class, InstructorEvaluation, User, GroupMember, assessmentController.

### 3.7. task-routes.test.js

**Mục đích**: Xác thực các tuyến đường API được định nghĩa trong task-routes.js cho các thao tác liên quan đến nhiệm vụ.

Các trường hợp kiểm thử:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **TR-001** | Thêm bình luận | POST /api/tasks/comments | Trạng thái: 201, Gọi controller | Đạt |
| **TR-002** | Cập nhật mục danh sách kiểm tra | PATCH /api/tasks/checklists/1 | Trạng thái: 200, Gọi controller | Đạt |
| **TR-003** | Xóa mục danh sách kiểm tra | DELETE /api/tasks/checklists/1 | Trạng thái: 200, Gọi controller | Đạt |
| **TR-004** | Lấy danh sách sprint | GET /api/tasks/sprints | Trạng thái: 200, Gọi controller | Đạt |
| **TR-005** | Tạo sprint | POST /api/tasks/sprints | Trạng thái: 201, Gọi controller | Đạt |
| **TR-006** | Lấy thành viên nhóm theo dự án | GET /api/tasks/group-members-by-project | Trạng thái: 200, Gọi controller | Đạt |
| **TR-007** | Lấy danh sách nhiệm vụ | GET /api/tasks | Trạng thái: 200, Gọi controller | Đạt |
| **TR-008** | Tạo nhiệm vụ | POST /api/tasks | Trạng thái: 201, Gọi controller | Đạt |
| **TR-009** | Lấy chi tiết nhiệm vụ | GET /api/tasks/1 | Trạng thái: 200, Gọi controller | Đạt |
| **TR-010** | Cập nhật nhiệm vụ | PATCH /api/tasks/1 | Trạng thái: 200, Gọi controller | Đạt |
| **TR-011** | Xóa nhiệm vụ | DELETE /api/tasks/1 | Trạng thái: 200, Gọi controller | Đạt |

**Phụ thuộc giả lập**: taskController.

## 4. Thực thi Kiểm thử

* **Lệnh**: npm test hoặc jest <filename>.test.js
* **Độ phủ (Coverage)**: Chạy jest --coverage để tạo báo cáo độ phủ.
* **Kết quả mong đợi**: Tất cả các trường hợp kiểm thử được đánh dấu "Đạt" nên thực thi mà không có lỗi. Bất kỳ lỗi nào cho thấy vấn đề trong mã hoặc thiết lập kiểm thử.
* **Nhật ký (Logs)**: Các nhật ký console trong file kiểm thử (ví dụ: console.error) hỗ trợ gỡ lỗi khi có lỗi.

## 5. Giả định và Ràng buộc

* **Giả định:**
  + Middleware verifyToken thiết lập req.user (id, role) cho tuyến đường đánh giá và req.userId, req.isTeamLead cho tuyến đường nhiệm vụ.
  + Mô hình Sequelize và dịch vụ (projectService, contributionService) tuân theo cấu trúc giả lập.
  + Trigger cơ sở dữ liệu xử lý sprint\_number trong sprintService.
* **Ràng buộc:**
  + Kiểm thử được cô lập (không truy cập cơ sở dữ liệu thật).
  + Kiểm thử tải tệp (classController) giả định hệ thống tệp giả lập.
  + Các truy vấn SQL thô trong memberService được giả lập với dữ liệu đơn giản hóa.

## 6. Bảo trì

* **Cập nhật**: Thêm trường hợp kiểm thử mới cho các điểm cuối hoặc phương thức dịch vụ bổ sung.
* **Giả lập**: Cập nhật giả lập nếu lược đồ mô hình hoặc giao diện dịch vụ thay đổi.
* **Độ phủ**: Đặt mục tiêu độ phủ >80%; xử lý các nhánh chưa được phủ trong các lần lặp sau.
* **Liên hệ**: Phân công một lập trình viên để bảo trì kiểm thử (ví dụ: trưởng nhóm backend).

## 7. Phụ lục

* **Chạy kiểm thử**:

ĐẠT tests/assessmentController.test.js

ĐẠT tests/[classController.test.js](http://classcontroller.test.js)

ĐẠT tests/[sprintService.test.js](http://sprintservice.test.js)

ĐẠT tests/[memberService.test.js](http://memberservice.test.js)

ĐẠT tests/[taskService.test.js](http://taskservice.test.js)

ĐẠT tests/[assessment-routes.test.js](http://assessment-routes.test.js)

ĐẠT tests/task-routes.test.js

* **Độ phủ**:

Độ phủ: 85% câu lệnh, 80% nhánh, 90% hàm, 88% dòng

# Tài liệu Kiểm thử cho Giao diện người dùng (Frontend)

## 1. Tổng quan

Tài liệu này mô tả chi tiết các kiểm thử đơn vị (unit test) và kiểm thử tích hợp (integration test) cho các thành phần (components), hooks, dịch vụ (services) và hàm tiện ích (utils) của ứng dụng Frontend. Các kiểm thử đảm bảo rằng giao diện người dùng hiển thị đúng, logic nghiệp vụ hoạt động chính xác, và việc tương tác với API backend được xử lý phù hợp.

### 1.1. Mục đích

* **Xác thực chức năng**: Đảm bảo các thành phần React render đúng với các props và trạng thái khác nhau. Xác thực logic nghiệp vụ trong các custom hooks.
* **Kiểm tra tương tác**: Giả lập hành vi người dùng (click, nhập liệu) để đảm bảo các sự kiện được xử lý đúng và giao diện được cập nhật tương ứng.
* **Đảm bảo xử lý lỗi**: Kiểm tra ứng dụng hiển thị thông báo lỗi thân thiện khi có lỗi từ API hoặc lỗi logic.
* **Cô lập và Tái sử dụng**: Kiểm thử các đơn vị code một cách độc lập để dễ dàng xác định nguồn gốc lỗi và tăng cường khả năng bảo trì.

### 1.2. Phạm vi

* **Trong phạm vi**:
  + **Components**: Kiểm thử render và tương tác của các thành phần giao diện, đặc biệt là các component có logic phức tạp.
  + **Hooks**: Kiểm thử logic xử lý trạng thái, vòng đời và các side effect.
  + **Services**: Kiểm thử việc gọi API và xử lý dữ liệu trả về (mocking API calls).
  + **Utils**: Kiểm thử các hàm tiện ích thuần túy (pure functions).
* **Ngoài phạm vi**:
  + Kiểm thử End-to-End (E2E) toàn bộ luồng người dùng.
  + Kiểm thử hiệu năng (Performance Testing).
  + Kiểm thử tính đúng đắn của API Backend (việc này thuộc về kiểm thử Backend).

## 2. Môi trường Kiểm thử

### 2.1. Công cụ và Phụ thuộc

* **Khung kiểm thử**: Jest (^27.x.x hoặc mới hơn)
* **Thư viện kiểm thử React**: React Testing Library (RTL) (^13.x.x)
* **Thư viện giả lập tương tác người dùng**: User Event (^14.x.x)
* **Môi trường thực thi**: JSDOM (thông qua jest-environment-jsdom)
* **Giả lập (Mocking)**: Jest Mock Functions (jest.fn(), jest.mock()) được sử dụng để giả lập các module (axios, services) và các hàm của trình duyệt (sessionStorage, localStorage, fetch).

### 2.2. Hướng dẫn Thiết lập

1. **Cài đặt phụ thuộc**: Chạy npm install hoặc yarn install tại thư mục gốc frontend để cài đặt tất cả các gói cần thiết được định nghĩa trong package.json.
2. **Cấu hình Jest**: Cấu hình Jest đã được thiết lập trong file jest.config.js và src/setupTests.js để hỗ trợ import các tiện ích của jest-dom và mock các file tĩnh (CSS, ảnh).
3. **Chạy kiểm thử**: Sử dụng lệnh npm test để thực thi tất cả các file kiểm thử.

## 3. Các File Kiểm thử và Trường hợp Kiểm thử

### 3.1. Kiểm thử Logic (Hooks)

#### useCommitActivity.test.js

**Mục đích**: Xác thực hook useCommitActivity xử lý việc lấy, xử lý, và cung cấp dữ liệu hoạt động commit hàng tuần.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-HK-001** | Xử lý đúng các trạng thái loading và thành công | Render hook với projectId hợp lệ | Ban đầu loading là true. Sau khi hoàn tất, loading là false,  weeklyActivity và  projectUsers chứa dữ liệu đã được xử lý. | Đạt |
| **FE-HK-002** | Xử lý trạng thái lỗi khi API thất bại | Render hook, mock API trả về lỗi | loading trở thành false, error chứa thông báo lỗi, weeklyActivity và  projectUsers là mảng rỗng. | Đạt |
| **FE-HK-003** | Tự động fetch lại dữ liệu khi selectedUser thay đổi | Render hook, sau đó gọi hàm setSelectedUser | Hàm gọi API (axios.get) được gọi lại. | Đạt |
| **FE-HK-004** | Tự động fetch lại dữ liệu khi refreshNonce thay đổi | Render hook với nonce mới | Hàm gọi API (axios.get) được gọi lại. | Đạt |

#### useDashboardCoreData.test.js

**Mục đích**: Quản lý và cung cấp dữ liệu lõi cho trang Dashboard, bao gồm danh sách nhóm, sprints, và các chỉ số thống kê.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-HK-005** | Lấy danh sách nhóm và vai trò người dùng thành công | Render hook với userId và classId hợp lệ | loadingGroups là false, groups chứa danh sách nhóm, currentUserRole được thiết lập. | Đạt |
| **FE-HK-006** | Tự động chọn nhóm từ context nếu có | Render hook với initialGroupIdFromContext hợp lệ | selectedGroup được tự động gán bằng nhóm có ID tương ứng. | Đạt |
| **FE-HK-007** | Tự động lấy sprints và stats sau khi chọn nhóm | selectedGroup được thiết lập | Các API lấy sprints và stats được gọi, sprints và statData được cập nhật. | Đạt |
| **FE-HK-008** | Xử lý khi không có userId hoặc classId | Render hook với userId=null | Không gọi API, các trạng thái dữ liệu rỗng. | Đạt |

#### useLOCGrowthData.test.js

**Mục đích**: Lấy và xử lý dữ liệu về thay đổi số dòng code (Lines of Code) hàng tuần.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-HK-009** | Lấy và xử lý dữ liệu LOC thành công | Render hook với projectId hợp lệ | loading là false, weeklyLOCActivity chứa dữ liệu LOC đã được xử lý. | Đạt |
| **FE-HK-010** | Xử lý lỗi khi không lấy được dữ liệu LOC | Render hook, mock API trả về lỗi | loading là false, error chứa thông báo lỗi. | Đạt |
| **FE-HK-011** | Tự động fetch lại dữ liệu khi selectedUser thay đổi | Render hook, sau đó gọi hàm setSelectedUser | Hàm gọi API (axios.get) được gọi lại. | Đạt |

#### useLateTasksData.test.js

**Mục đích**: Cung cấp dữ liệu về các task bị trễ của thành viên trong nhóm.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-HK-012** | Lấy dữ liệu task trễ thành công | Render hook với groupId hợp lệ | loading là false, members và sprintData chứa dữ liệu. | Đạt |
| **FE-HK-013** | Xử lý lỗi khi không có groupId | Render hook với groupId=null | loading là false, error chứa thông báo "No Group ID provided". | Đạt |

#### useMemberCompletionData.test.js

**Mục đích**: Lấy dữ liệu thống kê hoàn thành task của từng thành viên.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-HK-014** | Lấy dữ liệu và định dạng joinDate | Render hook với groupId hợp lệ | loading là false,  members chứa dữ liệu, joinDate của member được định dạng là "YYYY-MM-DD". | Đạt |

#### usePeerReviewData.test.js

**Mục đích**: Lấy dữ liệu đánh giá ngang hàng (peer review) cho một nhóm.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Hành động / Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-HK-015** | Lấy dữ liệu đánh giá thành công | Render hook với groupId  hợp lệ | loading là false,  peerReviewData chứa mảng dữ liệu đánh giá. | Đạt |
| **FE-HK-016** | Xử lý lỗi khi groupId không hợp lệ | Render hook với  groupId=null | error chứa thông báo "Group ID is required...". | Đạt |

#### useSprintPerformanceData.test.js

**Mục đích**: Xử lý và tổng hợp dữ liệu hiệu suất của các sprint.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-HK-017** | Lấy và xử lý dữ liệu hiệu suất sprint | Render hook với groupId hợp lệ | loading là false, processedSprints chứa dữ liệu đã được tổng hợp từ các thành viên. | Đạt |

#### useTaskSummaryData.test.js

**Mục đích**: Cung cấp dữ liệu tóm tắt trạng thái các task và danh sách sprint.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-HK-018** | Lấy dữ liệu tóm tắt cho "tất cả" sprint | Render hook với initialSelectedSprint='all' | taskSummary và sprintOptions được điền dữ liệu. | Đạt |
| **FE-HK-019** | Fetch lại dữ liệu khi  selectedSprint  thay đổi | Gọi hàm setSelectedSprint với ID sprint mới | API được gọi lại với sprintId mới, taskSummary được cập nhật. | Đạt |

### 3.2. Kiểm thử Thành phần Giao diện (Components)

#### LoginForm.test.js

**Mục đích**: Kiểm tra component LoginForm hoạt động đúng, bao gồm hiển thị, nhập liệu, xử lý lỗi và đăng nhập thành công.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-CP-001** | Hiển thị form đăng nhập chính xác | Render component  LoginForm | Các trường Email,  Password, và nút Login đều hiển thị. | Đạt |
| **FE-CP-002** | Hiển thị thông báo lỗi khi đăng nhập thất bại | Nhập sai thông tin và nhấn Login. Mock service loginUser trả về lỗi. | Component hiển thị thông báo lỗi "Invalid credentials". | Đạt |
| **FE-CP-003** | Đăng nhập thành công và chuyển hướng | Nhập đúng thông tin và nhấn Login. Mock service loginUser trả về thành công. | localStorage và  sessionStorage lưu đúng thông tin. Người dùng được chuyển hướng đến trang tương ứng với vai trò. | Đạt |

#### RegisterForm.test.js

**Mục đích**: Đảm bảo form đăng ký xác thực dữ liệu và xử lý việc tạo tài khoản mới.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-CP-004** | Hiển thị lỗi khi mật khẩu không khớp | Nhập mật khẩu và xác nhận mật khẩu khác nhau, sau đó nhấn "Register" | Thông báo lỗi "Mật khẩu không khớp" được hiển thị. registerUser không được gọi. | Đạt |
| **FE-CP-005** | Gọi API đăng ký và chuyển hướng khi thành công | Điền đầy đủ và hợp lệ thông tin, sau đó nhấn "Register". Mock registerUser trả về thành công. | Hàm registerUser được gọi với đúng dữ liệu. Người dùng được chuyển hướng đến trang /login. | Đạt |

#### RequireAuth.test.js & RequireRole.test.js

**Mục đích**: Kiểm tra các component bậc cao (HOC) bảo vệ route hoạt động chính xác.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-CP-004** | Cho phép truy cập khi có token/đúng vai trò | Truy cập route được bọc bởi HOC với token hợp lệ / user.role hợp lệ trong localStorage. | Component con (children) được render. | Đạt |
| **FE-CP-005** | Chặn truy cập và chuyển hướng khi không có token/sai vai trò | Truy cập route được bọc bởi HOC nhưng không có token / user.role không hợp lệ. | Hiển thị thông báo lỗi truy cập. Sau một khoảng thời gian, tự động chuyển hướng về trang /login. | Đạt |

#### KanbanView.test.js

**Mục đích**: Xác thực bảng Kanban hiển thị và hoạt động đúng với các quyền hạn khác nhau.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-CP-006** | Hiển thị các task đúng cột trạng thái | Render component với dữ liệu task mẫu | Task "To Do" nằm trong cột "To-Do". Task "In Progress" nằm trong cột "In-Progress". | Đạt |
| **FE-CP-007** | Team Lead có thể xóa một task | Render với isTeamLead = true. Click nút xóa task. | window.confirm được gọi. Hàm deleteTask trong service được gọi với đúng taskId. | Đạt |
| **FE-CP-008** | Thành viên thường không thấy nút xóa task | Render với isTeamLead = false. | Nút xóa task không hiển thị trên giao diện. | Đạt |

#### Pagination.test.js

**Mục đích**: Đảm bảo component phân trang hiển thị và hoạt động chính xác.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-CP-011** | Hiển thị đúng thông tin và trạng thái active | Render với currentPage=3, totalPages=10 | Nút trang '3' có class active. Hiển thị đúng tổng số dòng. | Đạt |
| **FE-CP-012** | Gọi callback khi chuyển trang | Click vào nút trang '5' | Hàm onPageChange được gọi với tham số là 5. | Đạt |
| **FE-CP-013** | Gọi callback khi thay đổi số dòng/trang | Chọn giá trị '20' trong ô select | Hàm onItemsPerPageChange được gọi. | Đạt |
| **FE-CP-014** | Vô hiệu hóa nút khi ở trang đầu/cuối | Render với currentPage=1 | Các nút "First" và "Previous" bị vô hiệu hóa (disabled). | Đạt |

#### ProjectRate.test.js & InstructorProjectRate.test.js

**Mục đích**: Đảm bảo trang đánh giá cho phép người dùng (sinh viên và giảng viên) xem, chỉnh sửa và gửi đánh giá.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-CP-009** | Hiển thị dữ liệu đánh giá có sẵn | Render component. Mock service trả về dữ liệu đánh giá và thống kê task. | Tên thành viên, thống kê task, và điểm/ghi chú đã có được hiển thị chính xác. | Đạt |
| **FE-CP-010** | Cho phép chỉnh sửa, thay đổi và lưu đánh giá | Click nút "Edit", thay đổi điểm (click sao) hoặc nhập ghi chú mới, sau đó click "Confirm". | Hàm service updatePeerAssessment (hoặc updateInstructorEvaluation) được gọi với dữ liệu đã được cập nhật. | Đạt |

### 3.4. Kiểm thử Hàm Tiện ích (Utils)

#### generateCommitChartConfig.test.js

**Mục đích**: Đảm bảo hàm tạo cấu hình cho biểu đồ commit hoạt động chính xác.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-UT-001** | Tạo cấu hình cho "Tất cả user" | Gọi hàm với selectedUser='all' | chartData.labels và chartData.datasets[0].data khớp với dữ liệu đầu vào. Label của dataset là "Total Commits per Week". | Đạt |
| **FE-UT-002** | Tạo cấu hình cho một user cụ thể | Gọi hàm với selectedUser='1' | Label của dataset chứa tên của user được chọn. | Đạt |

#### generateLateTasksChartConfig.test.js

**Mục đích**: Đảm bảo hàm tạo cấu hình cho biểu đồ task trễ hoạt động chính xác.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-UT-003** | Tạo cấu hình với dữ liệu hợp lệ | Gọi hàm với members và sprintData | data.labels là tên các thành viên. data.datasets[0].  data là tổng số task trễ của mỗi thành viên. | Đạt |

#### generateLOCGrowthChartConfig.test.js

**Mục đích**: Đảm bảo hàm tạo cấu hình cho biểu đồ tăng trưởng LOC hoạt động chính xác.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-UT-004** | Tạo cấu hình với dữ liệu LOC hợp lệ | Gọi hàm với weeklyLOCActivity | chartData chứa 2 dataset: "Lines Added" và "Lines Removed" với dữ liệu chính xác. | Đạt |

#### generateMemberCompletionChartConfig.test.js

**Mục đích**: Đảm bảo hàm tạo cấu hình cho biểu đồ phân phối task của thành viên.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-UT-005** | Tạo cấu hình biểu đồ cột chồng | Gọi hàm với dữ liệu members | data.datasets chứa 5 loại trạng thái task. Options scales.x.stacked và scales.y.stacked là true | Đạt |

#### generateSprintPerformanceChartConfig.test.js

**Mục đích**: Đảm bảo hàm tạo cấu hình cho biểu đồ hiệu suất sprint.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-UT-006** | Tạo cấu hình từ dữ liệu sprint đã xử lý | Gọi hàm với mảng sprints | data.labels là tên các sprint. data.datasets chứa dữ liệu "Completed" và "Late". | Đạt |

#### generateTaskChartConfig.test.js

**Mục đích**: Đảm bảo hàm tạo cấu hình cho biểu đồ tròn trạng thái task.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã kiểm thử** | **Mô tả** | **Đầu vào** | **Kết quả mong đợi** | **Trạng thái** |
| **FE-UT-007** | Tạo cấu hình và loại bỏ dữ liệu bằng 0 | Gọi hàm với  taskSummary | dataForChart.datasets[0].data không chứa các giá trị 0. totalTasks được tính chính xác. | Đạt |

## 4. Thực thi Kiểm thử

* **Chạy tất cả các bài test**: npm test
* **Chạy một file test cụ thể**: npm test -- src/path/to/your/file.test.js
* **Tạo báo cáo độ phủ (coverage report)**: npm test -- --coverage. Báo cáo sẽ được tạo trong thư mục /coverage.

## 5. Giả định và Ràng buộc

* **Giả định**:
  + API Backend được giả định là hoạt động và trả về dữ liệu theo đúng định dạng đã được mock.
  + Các HOC như useOutletContext được giả định là cung cấp đúng dữ liệu context cần thiết cho các component con.
* **Ràng buộc**:
  + Các bài test hoàn toàn được cô lập ở phía frontend, không thực hiện bất kỳ cuộc gọi mạng thực tế nào.
  + Việc kiểm thử không bao gồm các yếu tố về style hoặc vị trí chính xác của các phần tử trên trang, mà tập trung vào sự tồn tại và khả năng tương tác của chúng.

## 6. Bảo trì

* **Cập nhật**: Khi có sự thay đổi về logic trong một component/hook hoặc thay đổi về cấu trúc dữ liệu từ API, file test tương ứng cần được cập nhật để phản ánh những thay đổi đó.
* **Độ phủ**: Duy trì mục tiêu độ phủ code trên 80%. Các pull request mới cần bao gồm các bài test cho các chức năng mới hoặc các thay đổi logic.
* **Phân công**: Lập trình viên phát triển tính năng nào sẽ chịu trách nhiệm viết và bảo trì unit test cho tính năng đó.

## 7. Phụ lục

* **Chạy kiểm thử**:
  + ĐẠT tests/components/auth/LoginForm.test.js
  + ĐẠT tests/components/auth/RegisterForm.test.js
  + ĐẠT tests/components/auth/RequireAuth.test.js
  + ĐẠT tests/components/auth/RequireRole.test.js
  + ĐẠT tests/components/shared/Pagination.test.js
  + ĐẠT tests/components/shared/Rate/ProjectRate.test.js
  + ĐẠT tests/components/shared/Rate/InstructorProjectRate.test.js
  + ĐẠT tests/components/shared/task/KanbanView.test.js
  + ĐẠT tests/hooks/useCommitActivity.test.js
  + ĐẠT tests/hooks/useDashboardCoreData.test.js
  + ĐẠT tests/hooks/useLOCGrowthData.test.js
  + ĐẠT tests/hooks/useLateTasksData.test.js
  + ĐẠT tests/hooks/useMemberCompletionData.test.js
  + ĐẠT tests/hooks/usePeerReviewData.test.js
  + ĐẠT tests/hooks/useSprintPerformanceData.test.js
  + ĐẠT tests/hooks/useTaskSummaryData.test.js
  + ĐẠT tests/services/api-client.test.js
  + ĐẠT tests/services/auth-service.test.js
  + ĐẠT tests/services/class-service.test.js
  + ĐẠT tests/services/group-service.test.js
  + ĐẠT tests/services/peer-assessment-service.test.js
  + ĐẠT tests/services/profile-service.test.js
  + ĐẠT tests/services/user-service.test.js
  + ĐẠT tests/utils/generateCommitChartConfig.test.js
  + ĐẠT tests/utils/generateLateTasksChartConfig.test.js
  + ĐẠT tests/utils/generateLOCGrowthChartConfig.test.js
  + ĐẠT tests/utils/generateMemberCompletionChartConfig.test.js
  + ĐẠT tests/utils/generateSprintPerformanceChartConfig.test.js
  + ĐẠT tests/utils/generateTaskChartConfig.test.js
* **Độ phủ (Coverage)**:
  + **Mục tiêu**: >80% cho Statements, Branches, Functions, Lines.
  + **Kết quả ví dụ**: 89% Statements, 85% Branches, 92% Functions, 89% Lines.