

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
KHOA CÔNG NGHỆ THÔNG TIN 1**



**ĐỒ ÁN  
TỐT NGHIỆP ĐẠI HỌC**

**ĐỀ TÀI  
NGHIÊN CỨU, BỔ SUNG KHẢ NĂNG ĐIỀU HƯỚNG  
TIẾN TRÌNH HỌC TẬP CỦA SINH VIÊN TRÊN HỆ LMS  
MOODLE**

**Giảng viên hướng dẫn:**

**TS. Nguyễn Tất Thắng**

**Họ và tên sinh viên**

**Mã sinh viên**

**Lớp**

Nguyễn Hữu Trinh

B21DCCN724

D21HTTT3

Nguyễn Duy Hoàng

B21DCCN388

D21HTTT2

Nguyễn Bá Hoàng Huynh

B21DCCN447

D21HTTT2

**Hệ đào tạo:**

**Đại học chính quy**

*Hà Nội, tháng 12 năm 2025*

## LỜI CẢM ƠN

Trước hết, chúng em xin bày tỏ lòng biết ơn sâu sắc tới thầy TS. Nguyễn Tất Thắng - người đã trực tiếp hướng dẫn, tận tình chỉ bảo và chia sẻ nhiều kiến thức, kinh nghiệm quý báu trong suốt quá trình chúng em thực tập và thực hiện đồ án tốt nghiệp. Những góp ý và sự hỗ trợ của thầy là nền tảng quan trọng giúp chúng em hoàn thành đồ án một cách hiệu quả.

Tiếp theo, chúng em xin chân thành cảm ơn các thầy cô trong Học viện nói chung và khoa CNTT1 nói riêng đã luôn tận tâm trong công tác giảng dạy, truyền đạt kiến thức và tạo điều kiện thuận lợi cho chúng em trong suốt thời gian học tập và nghiên cứu tại Học viện Công nghệ Bưu chính Viễn thông.

Chúng em cũng xin gửi lời cảm ơn sâu sắc tới gia đình và những người thân yêu đã luôn quan tâm, động viên, ủng hộ và là chỗ dựa tinh thần vững chắc, giúp chúng em yên tâm học tập và hoàn thành tốt đồ án tốt nghiệp.

Cuối cùng, chúng em xin gửi những lời chúc tốt đẹp nhất tới bạn bè và các thầy cô tham gia đợt bảo vệ tốt nghiệp trong khóa này. Kính chúc mọi người luôn mạnh khỏe, vui vẻ và gặt hái nhiều thành công trong công việc cũng như cuộc sống.

Chúng em xin chân thành cảm ơn!

*Hà Nội, tháng 12 năm 2025*

Nhóm sinh viên thực hiện

## BẢNG PHÂN CÔNG CÔNG VIỆC

| Thành viên            | Công việc  | Đóng góp |
|-----------------------|--|----------|
| Nguyễn Hữu Trinh      | - Cài đặt Moodle trên Docker.<br>- Nghiên cứu, thiết kế và xây dựng plugin chấm điểm tự động trong Moodle.                           | 40%      |
| Nguyễn Duy Hoàng      | - Nghiên cứu, thiết kế và xây dựng plugin điều hướng học tập trong Moodle.   | 30%      |
| Nguyễn Bá Hoàng Huynh | - Tìm hiểu hệ thống LMS, Moodle.<br>- Nghiên cứu về OCR, LLM, cách prompt, đặt tham số.<br>- Triển khai server OCR, LLM trên Docker. | 30%      |

## This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting or typing. There are no margins, text, or other markings on the page.

## CÁN BỘ GIẢNG VIÊN HƯỚNG DẪN

**NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM**  
(Của giáo viên phản biện)

[illegible]

**Điểm:..... (bằng chữ: .....)**

**Đồng ý/Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?**

....., ngày..... tháng..... năm 202...

## CÁN BỘ GIẢNG VIÊN PHẢN BIỆN

## MỤC LỤC

|  |             |
|--|-------------|
| <b>LỜI CẢM ƠN .....</b>  | <b>i</b>    |
| <b>BẢNG PHÂN CÔNG CÔNG VIỆC .....</b>  | <b>ii</b>   |
| <b>NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM.....</b>   | <b>iii</b>  |
| <b>NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM.....</b>   | <b>iv</b>   |
| <b>MỤC LỤC .....</b>   | <b>v</b>    |
| <b>BẢNG VIẾT TẮT VÀ THUẬT NGỮ' .....</b>   | <b>viii</b> |
| <b>DANH SÁCH HÌNH ẢNH.....</b>   | <b>x</b>    |
| <b>DANH SÁCH BẢNG.....</b>   | <b>xii</b>  |
| <b>MỞ ĐẦU.....</b>   | <b>1</b>    |
| <b>CHƯƠNG I. TÌM HIỂU VỀ HỆ THỐNG QUẢN LÝ HỌC TẬP MOODLE .....</b>                       | <b>3</b>    |
| <b>1.1. Hệ thống quản lý học tập - LMS.....</b>  | <b>3</b>    |
| 1.1.1. Lịch sử phát triển hệ thống LMS .....   | 3           |
| 1.1.2. Các chức năng chính của LMS .....   | 4           |
| 1.1.3. Khảo sát một số LMS .....   | 5           |
| <b>1.2. Tổng quan về Moodle .....</b>  | <b>6</b>    |
| 1.2.1. Giới thiệu .....  | 6           |
| 1.2.2. Tìm hiểu các hoạt động của Moodle.....  | 6           |
| 1.2.3. Tìm hiểu các tài nguyên của Moodle.....   | 12          |
| 1.2.4. Đánh giá ưu điểm và hạn chế của hệ thống Moodle .....                             | 15          |
| <b>1.3. Kết luận chương I.....</b>   | <b>15</b>   |
| <b>CHƯƠNG II. NGHIÊN CỨU, PHÁT TRIỂN CHỨC NĂNG ĐIỀU HƯỚNG HỌC TẬP TRONG MOODLE .....</b> | <b>16</b>   |
| <b>2.1. Tổng quan về chức năng điều hướng học tập.....</b>                               | <b>16</b>   |
| 2.1.1. Khái niệm và định nghĩa.....  | 16          |
| 2.1.2. Cơ sở lý thuyết về cách tiếp cận .....  | 17          |
| <b>2.2. Kiến trúc hệ thống Availability API và Events API trong Moodle.....</b>          | <b>18</b>   |
| 2.2.1. Cấu trúc dữ liệu JSON .....   | 18          |
| 2.2.2. Hệ thống Core Availability và cơ chế kế thừa.....                                 | 19          |
| 2.2.3. Quy trình kiểm tra quyền truy cập .....   | 21          |
| 2.2.4. Hệ thống Core Event .....   | 22          |
| <b>2.3. Phân tích và thiết kế hệ thống.....</b>  | <b>23</b>   |
| 2.3.1. Phân tích chức năng hệ thống.....   | 23          |

|  |           |
|--|-----------|
| 2.3.2. Kịch bản chuẩn .....  | 25        |
| 2.3.3. Biểu đồ tuần tự.....  | 27        |
| <b>2.4. Một số hình ảnh giao diện .....</b>  | <b>28</b> |
| <b>2.5. Kết luận chương II .....</b>   | <b>30</b> |
| <b>CHƯƠNG III. NGHIÊN CỨU, PHÁT TRIỂN CHỨC NĂNG CHẤM ĐIỂM BÀI TẬP TỰ ĐỘNG TRONG CÁC HỆ THỐNG LMS .....</b> | <b>31</b> |
| <b>3.1. Tổng quan .....</b>  | <b>31</b> |
| 3.1.1. Hiện trạng và đặt vấn đề .....  | 31        |
| 3.1.2. Giải pháp đề xuất ứng dụng mô hình ngôn ngữ lớn .....   | 31        |
| <b>3.2. Công nghệ sử dụng.....</b>   | <b>32</b> |
| 3.2.1. Tích hợp nền tảng Moodle.....   | 32        |
| 3.2.2. Dịch vụ LLM .....   | 32        |
| 3.2.3. Dịch vụ OCR .....   | 33        |
| 3.2.4. Chuẩn trao đổi dữ liệu .....  | 33        |
| <b>3.3. Cơ sở lý thuyết và kỹ thuật ứng dụng mô hình LLM.....</b>  | <b>34</b> |
| 3.3.1. Cơ sở lý thuyết và sự phát triển .....  | 34        |
| 3.3.2. Phân tích quy trình xử lý prompt và trả về phản hồi .....   | 34        |
| 3.3.3. Thiết kế prompt.....  | 37        |
| 3.3.4. Các tham số.....  | 38        |
| <b>3.4. Lựa chọn mô hình LLM, các tham số và kiểm thử mô hình .....</b>                                    | <b>38</b> |
| 3.4.1. Lựa chọn mô hình LLM và các tham số.....  | 38        |
| 3.4.2. Kiểm thử mô hình.....   | 39        |
| <b>3.5. Phân tích và thiết kế hệ thống.....</b>  | <b>41</b> |
| 3.5.1. Phân tích chức năng hệ thống.....   | 41        |
| 3.5.2. Biểu đồ tuần tự.....  | 46        |
| 3.5.3. Thiết kế cơ sở dữ liệu .....  | 50        |
| 3.5.4. Kịch bản kiểm thử chức năng.....  | 52        |
| <b>3.6. Kết quả xây dựng giao diện hệ thống.....</b>   | <b>54</b> |
| <b>3.7. Kết luận chương III .....</b>  | <b>55</b> |
| <b>KẾT LUẬN .....</b>  | <b>56</b> |
| 1. Kết quả đạt được.....   | 56        |
| 2. Hạn chế của hệ thống.....   | 56        |
| 3. Định hướng phát triển hệ thống .....  | 56        |

|  |           |
|--|-----------|
| <b>DANH MỤC TÀI LIỆU THAM KHẢO .....</b> | <b>57</b> |
|--|-----------|



**BẢNG VIẾT TẮT VÀ THUẬT NGỮ**

| <b>TỪ VIẾT TẮT VÀ THUẬT NGỮ</b> | <b>Ý NGHĨA</b>  |
|---------------------------------|---|
| LMS                             | Learning Management System: Hệ thống quản lý học tập  |
| CMS                             | Content Management System: Hệ thống quản lý nội dung  |
| WWW                             | World Wide Web: Mạng lưới thông tin toàn cầu  |
| SaaS                            | Software as a Service: Phần mềm dưới dạng dịch vụ, cung cấp ứng dụng qua Internet thay vì cài đặt cục bộ  |
| STEM                            | Viết tắt của Science (Khoa học), Technology (Công nghệ), Engineering (Kỹ thuật) và Mathematics (Toán học)   |
| IMS Content Package             | Instructional Management Systems Content Package: Tiêu chuẩn kỹ thuật giúp đóng gói các tài liệu học tập (bài giảng, bài kiểm tra, hình ảnh, video...) thành một tệp nén theo cấu trúc thống nhất |
| URL                             | Uniform Resource Locator: Đường dẫn tài nguyên thống nhất, dùng để định vị một địa chỉ cụ thể trên Internet   |
| SCORM                           | Sharable Content Object Reference Model: Một tập hợp các tiêu chuẩn kỹ thuật cho các sản phẩm học trực tuyến  |
| Moodle                          | Modular Object - Oriented Dynamic Learning Environment: Môi trường học tập động, hướng đối tượng, dạng module   |
| LLM                             | Large Language Model: Mô hình ngôn ngữ lớn  |
| OCR                             | Optical Character Recognition: Nhận dạng ký tự quang học  |
| API                             | Application Programming Interface: Giao diện lập trình ứng dụng   |
| GPU                             | Graphics Processing Unit: Bộ xử lý đồ họa   |
| JSON                            | JavaScript Object Notation: Một định dạng dữ liệu nhẹ, dễ đọc cho cả người và máy, thường dùng để trao đổi dữ liệu giữa server và ứng dụng  |

|           |   |
|-----------|---|
| HTTP      | Hypertext Transfer Protocol: Giao thức truyền tải siêu văn bản  |
| NLP       | Natural Language Processing: Xử lý ngôn ngữ tự nhiên  |
| PLM       | Pre-trained Language Model: Mô hình ngôn ngữ tiền huấn luyện trên kho dữ liệu lớn   |
| ANS       | Adaptive Navigation Support: Hỗ trợ điều hướng thích ứng  |
| VPL       | Virtual Programming Lab: Một plugin của Moodle hỗ trợ việc lập trình và chấm bài tự động cho các ngôn ngữ lập trình   |
| Plugin    | Gói mở rộng/Thành phần bổ trợ. Là thành phần phần mềm được cài đặt tích hợp vào hệ thống LMS nhằm bổ sung các chức năng cụ thể hoặc tùy biến quy trình hoạt động mà không làm thay đổi mã nguồn cốt lõi của hệ thống. |
| Module    | Mô-đun, một phần độc lập trong cấu trúc hệ thống giúp thực hiện một chức năng cụ thể  |
| Core      | Nhân hoặc lõi, phần cốt lõi chứa các tính năng cơ bản nhất của một hệ thống phần mềm  |
| Prompt    | Chỉ dẫn cho mô hình   |
| Token     | Đơn vị nhỏ nhất của văn bản mà mô hình ngôn ngữ sử dụng để xử lý  |
| Framework | Bộ khung: là một bộ công cụ, thư viện, quy tắc và cấu trúc code được viết sẵn   |
| MUC       | Moodle Universal Cache: Bộ nhớ đệm dùng chung mà tất cả người dùng đều có thể sử dụng   |
| RAM       | Bộ nhớ truy cập ngẫu nhiên, có chức năng lưu trữ dữ liệu tạm thời   |
| RESTful   | REpresentational State Transfer: Là một tiêu chuẩn kiến trúc để thiết kế API  |
| BPE       | Byte-Pair Encoding: Thuật toán mã hóa theo cặp  |
| Embedding | Là cách các hệ thống AI biểu diễn các đối tượng dưới dạng vector  |
| RNN       | Recurrent Neural Network: Mạng nơ-ron hồi quy   |

## DANH SÁCH HÌNH ẢNH

|   |    |
|---|----|
| Hình 1.1: Các mốc thời gian lịch sử phát triển LMS .....  | 3  |
| Hình 1.2: Giao diện sinh viên nộp bài bằng cách nhập văn bản hoặc gửi tệp.....                                  | 7  |
| Hình 1.3: Giao diện giảng viên chấm điểm bài làm của sinh viên.....   | 8  |
| Hình 1.4: Giao diện làm bài Quiz của sinh viên .....  | 8  |
| Hình 1.5: Giao diện tổng quan bài Quiz trước khi sinh viên hoàn thành.....                                      | 9  |
| Hình 1.6: Giao diện kết quả và thống kê của giảng viên.....   | 9  |
| Hình 1.7: Giao diện cấu hình bài tập lập trình Java của học viên .....  | 10 |
| Hình 1.8: Quy trình học viên viết mã Java trực tiếp trong trình soạn thảo mã của VPL .....                      | 10 |
| Hình 1.9: Giao diện giảng viên tổng hợp phản hồi của người học thông qua hoạt động Choice.....                  | 11 |
| Hình 1.10: Giao diện sinh viên đánh giá khóa học qua hoạt động Feedback .....                                   | 11 |
| Hình 1.11: Giao diện sinh viên xem khóa học SCORM .....   | 12 |
| Hình 1.12: Giao diện sinh viên xem nội dung của sách, điều hướng nội dung qua mục lục .....                     | 13 |
| Hình 1.13: Giao diện sinh viên xem nội dung trên một trang của Moodle .....                                     | 13 |
| Hình 1.14: Giao diện sinh viên xem các nội dung đa phương tiện .....  | 14 |
| Hình 1.15: Giao diện học viên nhấp vào URL “Learning management system” để điều hướng đến trang web cụ thể..... | 14 |
| Hình 2.1: Biểu đồ hoạt động cho quy trình kiểm tra quyền truy cập .....   | 22 |
| Hình 2.2: Biểu đồ use case của hệ thống hỗ trợ điều hướng cho Moodle.....                                       | 24 |
| Hình 2.3: Biểu đồ tuần tự cho use case quản lý cài đặt autorestrict.....  | 27 |
| Hình 2.4: Biểu đồ tuần tự cho use case gán độ khó cho hoạt động.....  | 27 |
| Hình 2.5: Biểu đồ tuần tự cho use case truy cập khoá học.....   | 28 |
| Hình 2.6: Trang quản lý của hệ thống .....  | 28 |
| Hình 2.7: Chi tiết tiến trình độ khó toàn khoá học .....  | 29 |
| Hình 2.8: Chi tiết tiến trình độ khó theo chương.....   | 29 |
| Hình 2.9: Các điều kiện khi được điều chỉnh thủ công .....  | 30 |
| Hình 2.10: Màn hình hiển thị điều kiện cần thiết để mở khoá .....   | 30 |
| Hình 3.1: Biểu đồ cột so sánh điểm mong đợi và điểm thực tế theo loại test case .....                           | 40 |
| Hình 3.2: Biểu đồ hộp biểu diễn sự biến động điểm số theo loại tiêu chí.....                                    | 40 |
| Hình 3.3: Sơ đồ use case tổng quan.....   | 41 |

|   |    |
|---|----|
| Hình 3.4: Biểu đồ tuần tự cho module thiết lập cấu hình.....                    | 46 |
| Hình 3.5: Biểu đồ tuần tự cho module bật tính năng chấm điểm tự động .....      | 47 |
| Hình 3.6: Biểu đồ tuần tự cho tính năng giám sát bảng điều khiển chấm điểm..... | 48 |
| Hình 3.7: Biểu đồ tuần tự cho tính năng nộp bài tập.....                        | 49 |
| Hình 3.8: Biểu đồ tuần tự cho tính năng chấm điểm tự động .....                 | 50 |
| Hình 3.9: Giao diện cấu hình chấm điểm tự động.....                             | 54 |
| Hình 3.10: Giao diện bật tính năng chấm điểm tự động .....                      | 54 |
| Hình 3.11: Giao diện quản lý chấm điểm tự động .....                            | 55 |

## **DANH SÁCH BẢNG**

|   |    |
|---|----|
| Bảng 2.1: Các thành phần cấu trúc JSON Availability .....                           | 19 |
| Bảng 2.2: Mô tả chi tiết use case trong hệ thống hỗ trợ điều hướng cho Moodle ..... | 24 |
| Bảng 2.3: Kịch bản cho use case quản lý cài đặt autorestrict .....                  | 25 |
| Bảng 2.4: Kịch bản cho use case gán độ khó cho hoạt động .....                      | 25 |
| Bảng 2.5: Kịch bản cho use case truy cập khoá học .....                             | 26 |
| Bảng 3.1: Mô tả các use case.....   | 41 |
| Bảng 3.2: Kịch bản cho chức năng cấu hình tham số toàn cục cho plugin .....         | 42 |
| Bảng 3.3: Kịch bản cho chức năng kích hoạt chấm điểm tự động cho bài tập .....      | 43 |
| Bảng 3.4: Kịch bản cho chức năng giám sát bảng điều khiển chấm điểm .....           | 44 |
| Bảng 3.5: Kịch bản cho chức năng nộp bài tập .....                                  | 45 |
| Bảng 3.6: Kịch bản cho chức năng chấm điểm tự động.....                             | 45 |
| Bảng 3.7: Đặc tả bảng local_autograding.....  | 50 |
| Bảng 3.8: Đặc tả bảng local_autograding_status .....                                | 51 |
| Bảng 3.9: Danh sách test case cho chức năng chấm điểm tự động .....                 | 52 |

## MỞ ĐẦU

Trước đây, sinh viên phải đến các trường học, cơ sở giáo dục để tiếp cận tài liệu và các dịch vụ cho mục đích học tập. Ngoài ra, một số giảng viên cũng phải đến trực tiếp các cơ sở giảng dạy để cung cấp dịch vụ giảng dạy của mình. Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ trong lĩnh vực giáo dục cũng như sự xuất hiện của WWW, hình thức giảng dạy dựa trên nền tảng web đã mở ra những cơ hội mới trong việc xây dựng các môi trường giáo dục trực tuyến. Trong môi trường giáo dục mới này, các công cụ công nghệ thông tin và truyền thông được kết hợp nhằm tạo ra những phương pháp dạy và học hiệu quả nhất, đồng thời được triển khai trong các môi trường kết hợp giữa học tập điện tử và học trực tiếp để đáp ứng các nhu cầu học tập đa dạng.

Sự chuyển dịch trong các môi trường giáo dục trực tuyến này đã dẫn đến sự ra đời của nhiều cải tiến nhằm nâng cao hiệu quả và chất lượng của học trực tuyến, trong đó có LMS. Hầu hết các LMS đều được xây dựng là ứng dụng web để tận dụng mạnh khía cạnh “học mọi lúc, mọi nơi” đều có thể truy cập vào nội dung học. Sự kết hợp giữa các chiến lược giảng dạy, hệ thống thông tin và công nghệ thông tin và truyền thông đã trở thành động lực quan trọng trong việc hình thành và phát triển LMS. Trong số các LMS hiện nay, Moodle là một nền tảng mã nguồn mở được sử dụng rộng rãi trên toàn thế giới, đặc biệt trong các cơ sở giáo dục từ phổ thông đến đại học và đào tạo trực tuyến.

Tuy nhiên, khi sử dụng hệ thống trong thực tiễn, một số nhu cầu chuyên sâu về quản lý tiến độ học tập và đánh giá kết quả học tập của sinh viên vẫn chưa đáp ứng đầy đủ hoặc cần được mở rộng. Xuất phát từ yêu cầu đó, việc nghiên cứu, thiết kế và phát triển các chức năng bổ sung trên nền tảng Moodle là cần thiết nhằm nâng cao hiệu quả quản lý và hỗ trợ người học. Trong phạm vi nghiên cứu này, đồ án mở rộng hệ thống nhằm hỗ trợ điều hướng tiến trình học tập của sinh viên, giúp người học theo dõi lộ trình học tập một cách trực quan, biết được trạng thái hoàn thành các hoạt động học tập. Đồng thời, hệ thống cũng được bổ sung chức năng chấm điểm bài làm tự động, góp phần tự động hóa quá trình đánh giá, đảm bảo tính nhất quán và giảm tải khối lượng công việc cho giảng viên.

Nội dung của đồ án “Nghiên cứu, bổ sung khả năng điều hướng tiến trình học tập của sinh viên trong hệ LMS Moodle” bao gồm các phần sau:

### **Chương I: Tìm hiểu về hệ thống quản lý học tập Moodle**

Nghiên cứu tổng quan LMS, các chức năng của LMS; tìm hiểu về một số LMS, một số module và tài nguyên thường được sử dụng trong Moodle.

### **Chương II: Nghiên cứu, phát triển chức năng điều hướng học tập trong Moodle**

Nghiên cứu tổng quan về ANS, tìm hiểu về kiến trúc hệ thống các thành phần liên quan của Moodle, thiết kế và xây dựng một hệ thống hỗ trợ điều hướng trong Moodle.

### **Chương III: Nghiên cứu, phát triển chức năng chấm điểm bài tập tự động trong các hệ thống LMS**

Trình bày giải pháp ứng dụng LLM và OCR để xây dựng chức năng chấm điểm tự động trên các hệ thống LMS. Nội dung bao gồm phân tích công nghệ, kỹ thuật thiết kế prompt, thiết kế hệ thống chi tiết và xây dựng các kịch bản kiểm thử chức năng chấm điểm tự động trên Moodle.

### **Kết luận**

Trình bày kết quả đạt được, các hạn chế còn tồn tại và phương hướng phát triển trong tương lai.

Các nghiên cứu thử nghiệm và cài đặt liên quan mà đồ án được lưu trữ tại đường dẫn: [https://github.com/toring123/2025\\_DATN\\_Trinh\\_Hoang\\_Huynh](https://github.com/toring123/2025_DATN_Trinh_Hoang_Huynh)

# CHƯƠNG I. TÌM HIỂU VỀ HỆ THỐNG QUẢN LÝ HỌC TẬP MOODLE

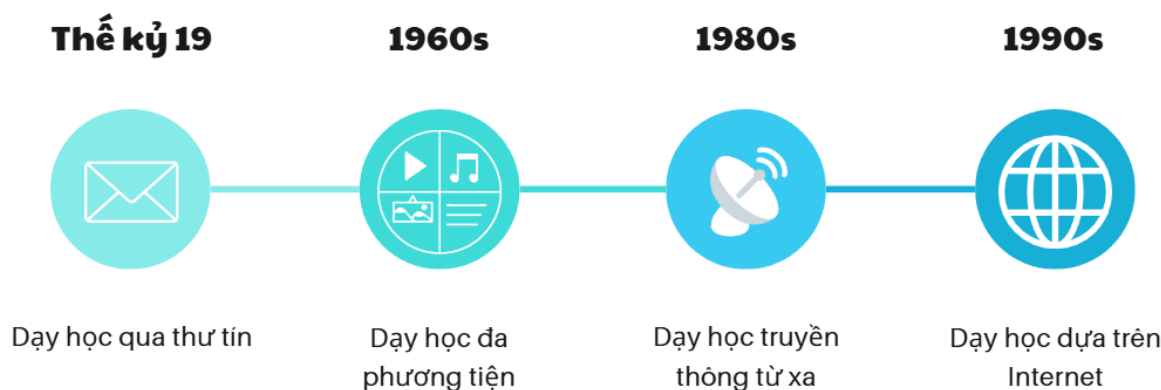
Trong kỷ nguyên chuyển đổi số toàn cầu, LMS đã chuyển dịch từ một công cụ hỗ trợ trở thành cơ sở hạ tầng cốt lõi của các tổ chức giáo dục và doanh nghiệp. Chương này cung cấp một cái nhìn tổng quát về LMS cũng như Moodle - nền tảng LMS mã nguồn mở phổ biến nhất thế giới.

## 1.1. Hệ thống quản lý học tập - LMS

Hệ thống quản lý học tập (LMS) trong giáo dục đề cập đến phần mềm và hệ thống được sử dụng để quản lý, theo dõi và cung cấp các tài liệu học tập cũng như quản lý thông tin của sinh viên. Việc áp dụng các công nghệ cho LMS đã tạo ra một cuộc cách mạng trong giáo dục bằng cách cung cấp cái nhìn toàn diện về quá trình học tập và hoạt động của giảng viên, sinh viên, bộ phận quản lý.

### 1.1.1. Lịch sử phát triển hệ thống LMS

Sự phát triển của hệ thống LMS là kết quả của sự thay đổi cách tiếp cận trong quá trình dạy và học, đặc biệt là với sự xuất hiện của WWW mở ra cơ hội cho môi trường giáo dục vi mô trực tuyến.



Hình 1.1: Các mốc thời gian lịch sử phát triển LMS

Các giai đoạn phát triển dẫn đến hệ thống LMS hiện đại bao gồm:

- **Dạy học qua thư tín**: Là một hình thức học từ xa cung cấp các tài liệu và bài tập thông qua tài liệu in ấn và dịch vụ bưu chính. Sau khi hoàn thành, học viên sẽ gửi bài làm của mình qua dịch vụ bưu chính để được phân tích, chấm điểm và nhận xét. Ví dụ, thư điện tử được sử dụng như một hình thức giảng dạy qua thư tín, cho phép giảng viên chia sẻ các tài liệu học tập như bản trình chiếu, văn bản, video hoặc bản ghi âm, v.v. cho sinh viên.
- **Dạy học đa phương tiện**: Đa phương tiện là việc sử dụng nhiều hơn một loại phương tiện, chẳng hạn như văn bản, hình ảnh, video, âm thanh thông qua các công nghệ nhằm hỗ trợ ghi nhớ và nâng cao khả năng hiểu nội dung khóa học.



Ví dụ, một số giảng viên sử dụng bản trình chiếu PowerPoint, video và hình ảnh trong lớp học trực tiếp để nâng cao sự hiểu biết của sinh viên về các khái niệm của môn học.

- **Dạy học truyền thông từ xa:** Là cách tiếp cận sử dụng các hệ thống truyền thông theo thời gian thực nhằm tăng cường tính tương tác giữa sinh viên và giảng viên. Phương pháp dạy học truyền thông trở nên phổ biến hơn trong lĩnh vực giáo dục vào những năm 1980, khi máy tính ngày càng được sử dụng rộng rãi trong các hộ gia đình và các cơ sở giáo dục đại học. Trong giai đoạn này, quá trình học tập sẽ là lấy người học làm trung tâm hơn và mang các đặc điểm tích hợp phương tiện truyền thông, giao tiếp theo thời gian thực và giao tiếp không đồng bộ giữa sinh viên và giảng viên, cùng với các hệ thống đa phương tiện như siêu văn bản. Các công nghệ được sử dụng trong việc dạy học bao gồm máy fax, điện thoại và máy tính. Những công nghệ này cho phép tổ chức các buổi học bằng cách truyền tải âm thanh, qua đó tăng cường sự tham gia và giao tiếp của sinh viên.
- **Dạy học dựa trên Internet:** Hầu hết giảng viên sử dụng các công cụ và tài liệu giảng dạy dựa trên Internet để truyền đạt kiến thức, thu hút người học, lưu trữ hồ sơ và đánh giá kết quả học tập của sinh viên. Sự phát triển của WWW đã tạo ra cơ hội để tất cả các bên liên quan trong lĩnh vực giáo dục tham gia và đóng góp vào các quá trình dạy học. Hệ quả là nhiều phát minh đã được triển khai nhằm tối ưu hóa các cơ hội này và hỗ trợ quản lý tri thức, bao gồm CMS và LMS. Trong khi CMS là phần mềm được sử dụng để tạo, quản lý và chỉnh sửa nội dung, thì LMS là hệ thống tích hợp nhiều thành phần trong lĩnh vực giáo dục bao gồm hợp tác, giảng dạy, học tập và quản lý hành chính. LMS trong giáo dục đóng vai trò then chốt trong việc cung cấp các công cụ hợp tác và giao tiếp cần thiết để tương tác giữa sinh viên và giảng viên cũng như truyền tải tài liệu học tập trong các lớp học ảo. [1]

### 1.1.2. Các chức năng chính của LMS

Hệ thống LMS cung cấp những công cụ giúp kết nối giảng viên, bộ phận quản lý và sinh viên, tạo điều kiện cho sự tương tác và việc chia sẻ tài nguyên học tập. Hệ thống này hỗ trợ tổ chức các hoạt động dạy và học thông qua tài liệu học tập, buổi thảo luận trực tuyến và bài kiểm tra trực tuyến. Bên cạnh đó, hệ thống góp phần số hóa các quy trình như ghi nhận tiến độ học tập của sinh viên, tăng cường khả năng truy cập tài liệu của khóa học từ xa và cải thiện sự tương tác của người học với các công cụ học tập thông qua trình duyệt web, cho phép người dùng truy cập tài nguyên học tập bất cứ lúc nào và từ bất kỳ địa điểm nào. Ngoài ra, giảng viên và cán bộ quản lý theo dõi tiến độ học tập, tình hình chuyên cần và thời gian sinh viên dành cho các bài tập. Hệ thống cho phép tạo lập hồ sơ sinh viên và lưu trữ thông tin như điểm số của sinh viên trong khóa học đó. Về phía sinh viên, hệ thống giúp họ tiếp nhận các hướng dẫn giảng dạy và tài liệu học tập, đồng thời tham gia các hoạt động học tập theo yêu cầu của giảng viên và cán bộ

quản lý. Ngoài ra, sinh viên có thể xác định các điểm cần cải thiện và đưa ra phản hồi nhằm nâng cao quá trình thiết kế hệ thống. [1]

### 1.1.3. Khảo sát một số LMS

Đầu tiên phải kể đến Open edX, một nền tảng học tập trực tuyến cho phép xây dựng, triển khai và phân tích các khóa học trực tuyến quy mô lớn. Đây là một nền tảng mã nguồn mở được khởi xướng bởi Viện Công nghệ Massachusetts (MIT) và Đại học Harvard, với sự phát triển của hơn một trăm tổ chức giáo dục và công nghệ hàng đầu trên toàn cầu. Hiện nay Open edX đang được sử dụng bởi hàng triệu người học trên khắp thế giới. Nền tảng này bao gồm nhiều dịch vụ và thành phần chức năng khác nhau, phục vụ đa dạng các mục đích như giảng dạy, học tập, quản lý khóa học, phân tích dữ liệu và hỗ trợ các hoạt động marketing giáo dục. Open edX cung cấp hai dịch vụ cốt lõi là LMS và CMS. Trong đó, Open edX LMS phục vụ cho sinh viên truy cập khóa học và hỗ trợ giảng viên quản lý quá trình giảng dạy, theo dõi thống kê, điểm số và báo cáo. Open edX CMS là môi trường dành cho giảng viên tạo lập và quản lý nội dung khóa học, cho phép xây dựng cấu trúc bài giảng, chỉnh sửa nội dung, tải tài liệu và thiết lập các chính sách của khóa học; hệ thống này sử dụng chung cơ chế xác thực với LMS nhưng chỉ dành cho người có quyền biên soạn khóa học. [2]

Cùng chia sẻ nguồn gốc từ môi trường đại học như Open edX là Sakai, một hệ thống LMS mã nguồn mở được phát triển bởi cộng đồng học thuật toàn cầu dưới sự quản lý của tổ chức Apereo. Sakai là một nền tảng dựa trên ngôn ngữ lập trình Java và vận hành chủ yếu thông qua máy chủ ứng dụng Apache Tomcat. Đây là môi trường học và cộng tác trực tuyến, các tổ chức có thể tự do tùy chỉnh mã nguồn để phù hợp với quy trình riêng. Các công cụ của Sakai được tinh chỉnh để hỗ trợ các bài giảng phức tạp, nghiên cứu nhóm và quản lý đào tạo chuyên sâu mà không bị phụ thuộc vào các nhà cung cấp phần mềm thương mại. Đặc biệt, hệ thống này đảm bảo tính mở rộng cao bằng cách cho phép kết nối liền mạch với các ứng dụng giáo dục bên thứ ba như Zoom hay Turnitin, tạo nên một không gian học tập số hiện đại, bảo mật và đồng bộ. [3]

Trái ngược với hướng tiếp cận cộng đồng của Sakai, Blackboard Learn (viết tắt là Blackboard) đại diện cho phân khúc hệ thống LMS thương mại hàng đầu thế giới. Hiện nay, Blackboard thuộc sở hữu của công ty công nghệ giáo dục Anthology. Đây là một nền tảng được sử dụng rộng rãi trong các trường đại học, trường phổ thông và cả trong môi trường đào tạo doanh nghiệp. Blackboard tạo ra một trải nghiệm giáo dục số nhờ giao diện trực quan và khả năng thích ứng cao trên nhiều thiết bị. Trước đây, Blackboard có cả phiên bản cài đặt tại chỗ. Tuy nhiên, hiện nay công ty Anthology đang đẩy mạnh việc chuyển dịch người dùng sang SaaS trên nền tảng đám mây để đảm bảo tốc độ cập nhật tính năng mới và tính ổn định của hệ thống. [4]

Một LMS hiện đại khác như Blackboard là Canvas, hệ thống dựa trên nền tảng đám mây được phát triển bởi công ty Instructure. Kể từ khi ra mắt vào năm 2011, Canvas đã nhanh chóng trở thành đối thủ cạnh tranh lớn nhất của Moodle và Blackboard, đặc biệt phổ biến trong các trường đại học và các tổ chức giáo dục phổ thông trên thế giới.

Canvas chạy trên Amazon Web Services giúp hệ thống luôn ổn định, tự động cập nhật và không cần lo lắng về việc bảo trì máy chủ cục bộ. Bên cạnh đó, công ty Instructure vẫn duy trì một phiên bản mã nguồn mở trên GitHub để cộng đồng có thể tự cài đặt và tùy chỉnh. Canvas cho phép tích hợp dễ dàng với các công cụ như Zoom, Google Drive, Office 365, YouTube, v.v. Hệ thống LMS này còn cung cấp tài liệu API chi tiết cho phép lập trình viên xây dựng các plugin hoặc kết nối dữ liệu với các hệ thống bên thứ ba dễ dàng. [5]

## **1.2. Tổng quan về Moodle**

### **1.2.1. Giới thiệu**

Moodle là một hệ thống LMS miễn phí và mã nguồn mở, được viết bằng ngôn ngữ lập trình PHP và phân phối theo giấy phép General Public License. Moodle được sử dụng cho học kết hợp, dạy học từ xa và các dự án học tập trực tuyến trong trường học. Moodle được dùng để tạo các website tùy chỉnh có tích hợp các khóa học trực tuyến và cho phép sử dụng các plugin do cộng đồng phát triển.

Moodle ban đầu được phát triển bởi Martin Dougiamas nhằm hỗ trợ quản trị viên và giảng viên tạo ra các khóa học trực tuyến, đồng thời tập trung vào sự tương tác và việc xây dựng nội dung mang tính hợp tác. Phiên bản đầu tiên của Moodle được phát hành vào ngày 20 tháng 8 năm 2002 và cho đến nay vẫn đang được phát triển tích cực. Moodle được sử dụng trong nhiều lĩnh vực học thuật khác nhau, bao gồm cả giáo dục STEM. Moodle là một hệ thống LMS dựa trên nền tảng đám mây và nằm trong top 20 LMS tốt nhất dựa trên trải nghiệm người dùng vào năm 2018. Số lượng người dùng Moodle tiếp tục tăng từ 78 triệu năm 2015 lên hơn 294 triệu vào năm 2021, tương đương mức tăng hơn 250%.

Moodle là một công cụ mạnh mẽ hỗ trợ học tập theo nhiều cách khác nhau. Cả giảng viên và người học đều hưởng lợi từ việc sử dụng LMS Moodle. Các module được sử dụng phổ biến nhất là “quiz” (bài trắc nghiệm) và “workshop” (hội thảo) của Moodle. Các công cụ bên ngoài dễ dàng tích hợp vào hệ thống Moodle bao gồm video, virtual tour và hồ sơ điện tử. Moodle cho phép giảng viên phát huy tính sáng tạo cá nhân trong việc phát triển tài liệu học tập phù hợp với từng học phần cho sinh viên. Ngoài ra, Moodle còn tiết kiệm thời gian nhờ các bài kiểm tra được tạo ngẫu nhiên, câu hỏi có nhiều phương án trả lời, hệ thống chấm điểm tự động, bảng tiêu chí cũng như phản hồi tổng kết mang tính tạo động lực. Tính đến tháng 6 năm 2021, Moodle có 1.753 plugin sẵn có, cho phép bổ sung các chức năng mới nhằm cải thiện quản trị, đánh giá, nội dung và giao diện của hệ thống. [6]

### **1.2.2. Tìm hiểu các hoạt động của Moodle**

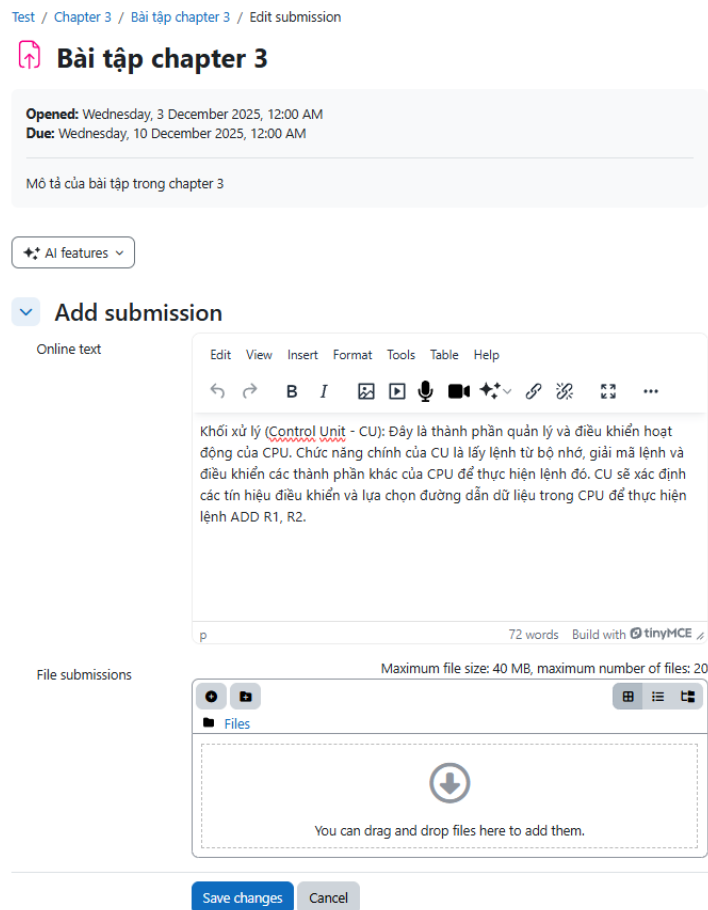
#### **1.2.2.1. Bài tập (Assignment)**

Hoạt động Assignment cho phép sinh viên nộp bài tập cho giảng viên để chấm điểm. Bài tập có thể là văn bản được đánh máy trực tuyến hoặc tệp tải lên thuộc bất kỳ

định dạng nào mà thiết bị của giảng viên có thể đọc được. Việc chấm điểm có thể được thực hiện theo tỷ lệ phần trăm đơn giản hoặc thang điểm tùy chỉnh, hoặc có thể sử dụng các tiêu chí chấm điểm phức tạp hơn. Sinh viên có thể nộp bài cá nhân hoặc theo nhóm. [7]

Giao diện của sinh viên:

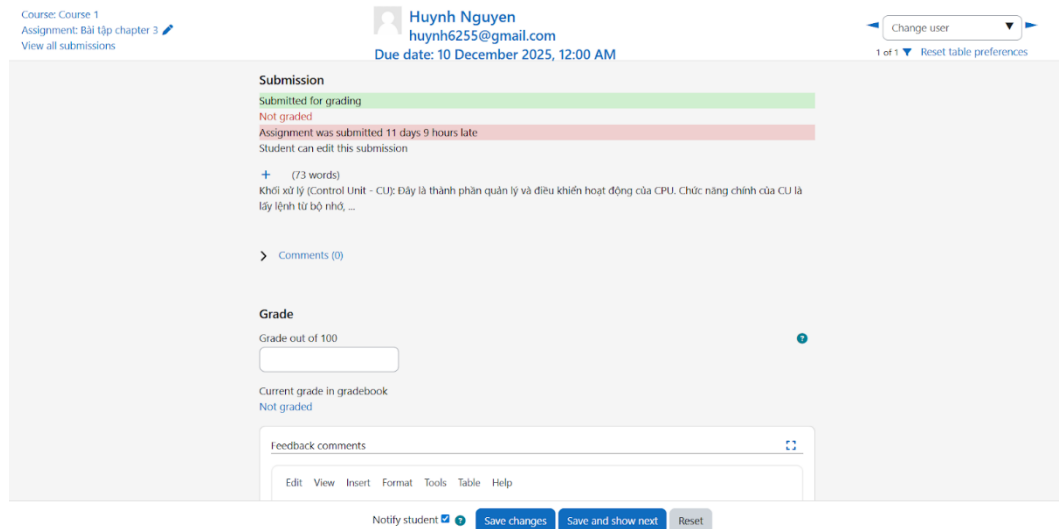
- Sinh viên nhấp vào liên kết bài tập và chọn “Add submission” (thêm bài nộp).
- Tùy thuộc vào cài đặt của bài tập, họ sẽ có một ô văn bản để nhập trực tiếp nội dung bài làm, hoặc một khu vực tải lên để nộp tệp bài làm.
- Sinh viên có thể được phép quay lại chỉnh sửa và hoàn thiện bài nộp nhiều lần, hoặc trong một số trường hợp, họ phải nhấn nút “Submit” (nộp bài) để gửi phiên bản cuối cùng - khi đó bài nộp sẽ không thể chỉnh sửa thêm.



Hình 1.2: Giao diện sinh viên nộp bài bằng cách nhập văn bản hoặc gửi tệp

Giao diện của giảng viên:

- Khi sinh viên đã nộp bài, giảng viên nhấp vào bài tập và chọn “Grade” (chấm điểm).
- Giao diện hiển thị sẽ phụ thuộc vào thiết lập của giảng viên và quản trị viên.
- Tại đây, giảng viên có thể ghi chú trực tiếp lên bài nộp của sinh viên, tải về bài nộp để chấm ngoại tuyến, nhập điểm cho bài tập, gửi phản hồi cá nhân cho sinh viên, lưu các thay đổi sau khi chấm điểm và chuyển sang sinh viên kế tiếp để tiếp tục quá trình chấm bài.



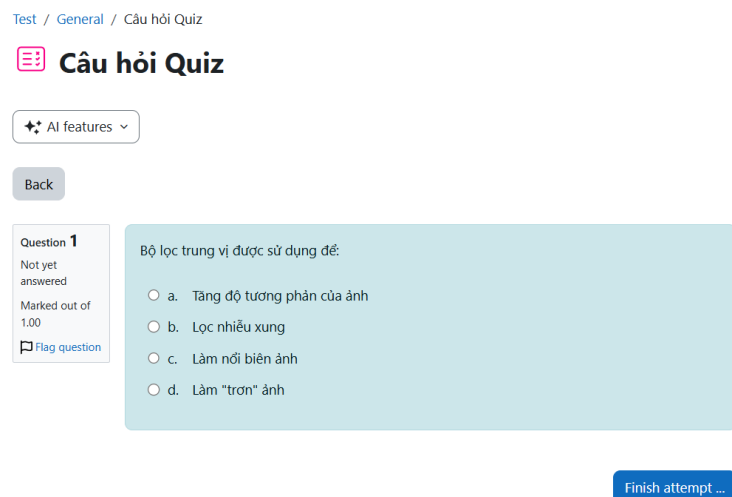
Hình 1.3: Giao diện giảng viên chấm điểm bài làm của sinh viên

### 1.2.2.2. Bài kiểm tra (Quiz)

Hoạt động Quiz cho phép giảng viên thiết kế và tổ chức các bài kiểm tra (trắc nghiệm, tự luận, nối cặp, điền khuyết, v.v.). Bài kiểm tra có thể được chấm điểm tự động (đối với câu hỏi trắc nghiệm) hoặc chấm thủ công (đối với câu hỏi tự luận). [7]

Giao diện của sinh viên:

- Tùy vào cách giảng viên cài đặt, sinh viên có thể nhìn thấy tất cả câu hỏi trên một trang hoặc mỗi câu hỏi trên một trang riêng. Sẽ có một khối liên kết đến từng câu hỏi.



Hình 1.4: Giao diện làm bài Quiz của sinh viên

- Để hoàn thành bài kiểm tra và có thể xem điểm số (tùy theo cài đặt), sinh viên phải nhấp vào “Submit all and finish” khi đến trang “Summary of attempt” (Tóm tắt lượt làm bài).

Test / General / Câu hỏi Quiz / Summary of attempt

**Câu hỏi Quiz**

AI features ▾

Back

**Câu hỏi Quiz**  
Summary of attempt

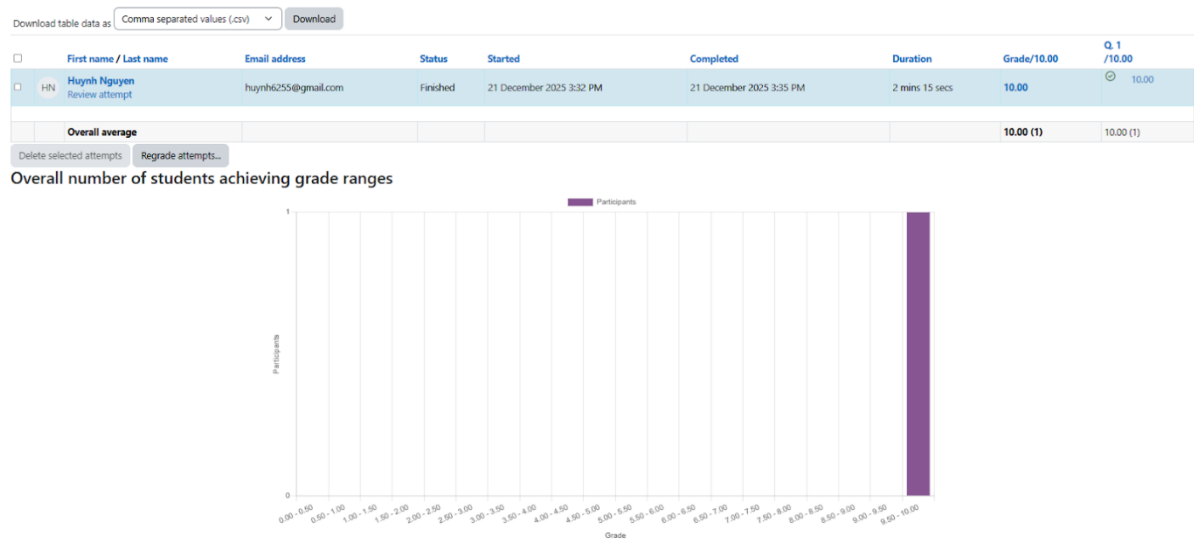
| Question                         | Status       |
|----------------------------------|--------------|
| Bộ lọc trung vị được sử dụng để: |              |
| 1                                | Answer saved |

Return to attempt

Submit all and finish

Hình 1.5: Giao diện tổng quan bài Quiz trước khi sinh viên hoàn thành

Giao diện của giảng viên: Giảng viên có thể xem được điểm bằng cách nhấp vào bài kiểm tra và chọn liên kết “Attempts” sau khi sinh viên đã làm bài.



Hình 1.6: Giao diện kết quả và thống kê của giảng viên

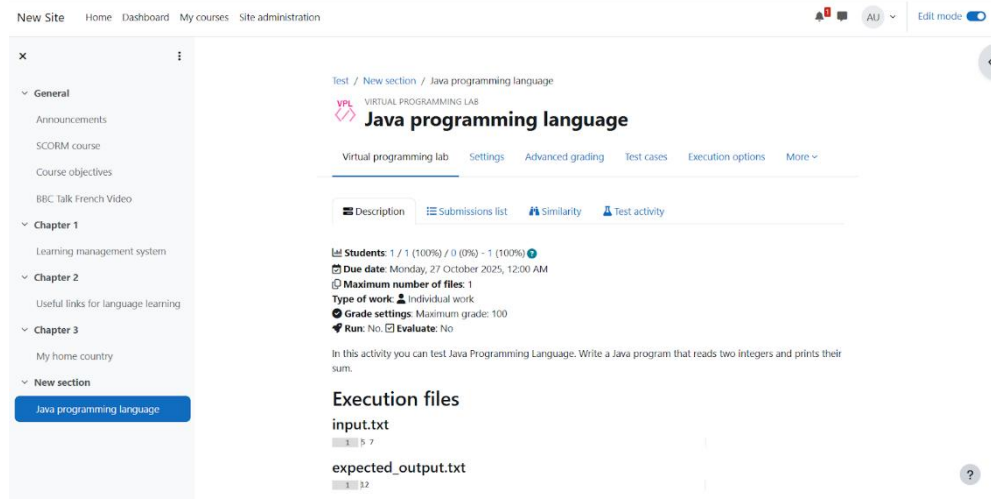
### 1.2.2.3. VPL (Virtual Programming Lab)

Được dùng để quản lý các bài tập lập trình của học viên. VPL tích hợp hoàn toàn vào hệ thống Moodle, nên nó kế thừa đầy đủ các tính năng quen thuộc của một activity như: chấm điểm, phân nhóm, kiểm soát truy cập dựa trên vai trò, lưu nhật ký truy cập, sao lưu và khôi phục dữ liệu. VPL có trình soạn thảo mã giúp lập trình thuận tiện. VPL hỗ trợ tự động chạy và đánh giá bài lập trình bằng cách sử dụng các đoạn mã có sẵn cho nhiều ngôn ngữ như: C, C++, C#, Java, Pascal, Python, v.v. [8]

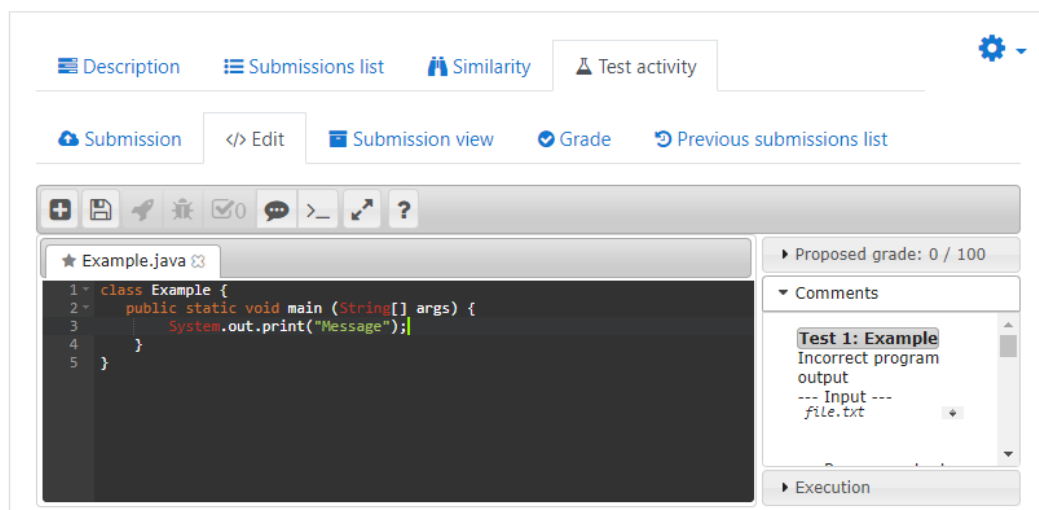
Ngoài ra, VPL còn có nhiều tính năng đặc thù dành riêng cho bài tập lập trình như sau:

- Mỗi tệp tải lên có thể được giới hạn về kích thước (tính bằng MB) để tránh quá tải hệ thống.
- Có thể chỉ định định dạng tệp bắt buộc mà học viên phải nộp (ví dụ: main.c, index.py).

- Cung cấp chức năng tìm kiếm và so sánh độ tương đồng giữa các bài nộp (phát hiện đạo văn).
- Giới hạn thao tác chỉnh sửa và ngăn dán mã từ bên ngoài (chống sao chép - dán).



Hình 1.7: Giao diện cấu hình bài tập lập trình Java của học viên



Hình 1.8: Quy trình học viên viết mã Java trực tiếp trong trình soạn thảo mã của VPL

#### 1.2.2.4. Các module khác của Moodle

Để tăng tính tương tác và thấu hiểu nhu cầu người học, Moodle cung cấp các công cụ như “Choice” (lựa chọn) và “Feedback” (phản hồi).

- Choice cho phép giảng viên đặt một câu hỏi và thiết lập các nút chọn để người học có thể chọn một trong nhiều phương án trả lời. Hoạt động “Choice” đặc biệt hữu ích trong các tình huống như bỏ phiếu tập thể để quyết định hướng đi của khóa học, hoặc đánh giá mức độ tiến triển của lớp học trong quá trình học tập, v.v.
- Feedback là công cụ đặc biệt hữu ích để thực hiện đánh giá khóa học hoặc đánh giá giảng viên, giúp cải thiện chất lượng giảng dạy và học tập. [7]

Assessment for learning

Responses

| Choice options              | Not answered yet  | I understood it perfectly   | I feel fairly confident about it   | I am not sure about it  | I don't understand. Please help!   |
|-----------------------------|---|---|--|---|--|
|                             | <input type="checkbox"/>  | <input type="checkbox"/>  | <input type="checkbox"/>   | <input type="checkbox"/>  | <input type="checkbox"/>   |
| Number of responses         | 3   | 2   | 2  | 3   | 4  |
| Users who chose this option | <input type="checkbox"/> William Brown<br><input type="checkbox"/> Olivia Davis<br><input type="checkbox"/> Isabella Miller | <input type="checkbox"/> Michael Brown<br><input type="checkbox"/> Emma Davis | <input type="checkbox"/> Isabella Davis<br><input type="checkbox"/> Ethan Williams | <input type="checkbox"/> Laura Hoffmann<br><input type="checkbox"/> William Johnson<br><input type="checkbox"/> Michael Jones | <input type="checkbox"/> Leah Becker<br><input type="checkbox"/> Timm Müller<br><input type="checkbox"/> Leonie Schulz<br><input type="checkbox"/> Sophia Wilson |

Hình 1.9: Giao diện giảng viên tổng hợp phản hồi của người học thông qua hoạt động Choice

Test / General / Đánh giá khóa học / Complete a feedback



## Đánh giá khóa học

★ AI features ▾

Mode: Anonymous

Bạn có thích ứng dụng Moodle không?

☐ Có

☐ Không

Mô tả ngắn gọn cảm nghĩ về ứng dụng Moodle

Hình 1.10: Giao diện sinh viên đánh giá khóa học qua hoạt động Feedback

Để hỗ trợ việc lưu trữ và hệ thống hóa kiến thức, giảng viên có thể sử dụng “Database” (cơ sở dữ liệu) và “Glossary” (thuật ngữ).

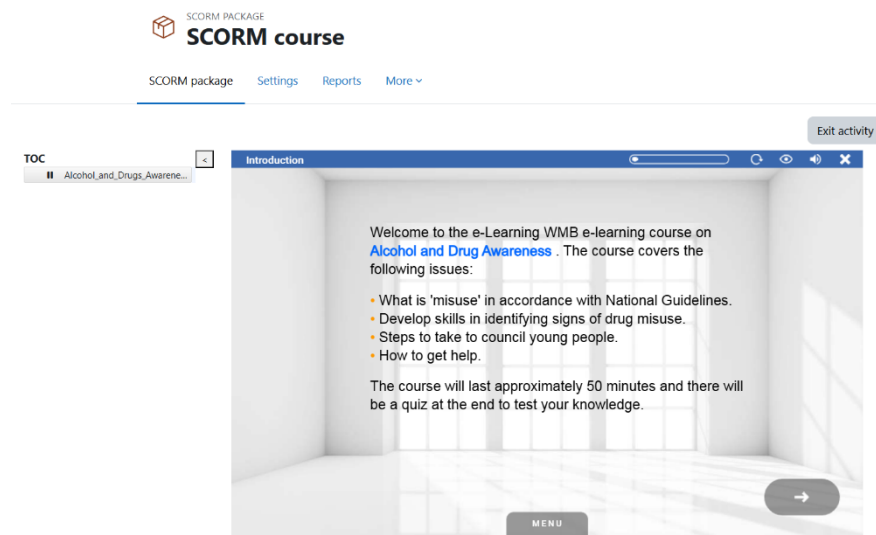
- Database cho phép giảng viên và/hoặc sinh viên tạo lập, hiển thị và tra cứu một kho các bản ghi liên quan đến bất kỳ chủ đề nào. Định dạng và cấu trúc của các bản ghi này gần như không giới hạn, có thể bao gồm: Hình ảnh, tệp đính kèm, URL, số liệu, văn bản và nhiều loại dữ liệu khác.
- Glossary được sử dụng để tạo và duy trì một danh sách các định nghĩa, giống như một cuốn từ điển hoặc một danh sách “FAQs” (Các câu hỏi thường gặp).

[7]



Đối với các bài giảng yêu cầu tính trực quan và khả năng tái sử dụng cao, Moodle tích hợp các tiêu chuẩn như H5P (nội dung tương tác), SCORM và công cụ Lesson (bài học).

- H5P cho phép tạo các nội dung tương tác cao cấp như video có câu hỏi chèn giữa, thẻ nhớ, kéo thả hình ảnh,... mà không cần cài thêm phần mềm.
- Lesson tạo ra một lộ trình học tập có điều kiện. Dựa trên câu trả lời của học viên cho một câu hỏi, giảng viên có thể điều hướng họ đến các trang khác nhau.
- SCORM là một bộ tiêu chuẩn kỹ thuật được sử dụng rộng rãi trong lĩnh vực học trực tuyến. Mục đích là chạy được trên nhiều hệ thống LMS khác nhau, có thể chia sẻ nội dung học tập trực tuyến bằng cách sử dụng SCORM package - tập hợp các tệp được đóng gói theo chuẩn SCORM. Bên cạnh đó, cùng một nội dung có thể dùng lại trong nhiều khóa học khác nhau. [7]



Hình 1.11: Giao diện sinh viên xem khóa học SCORM

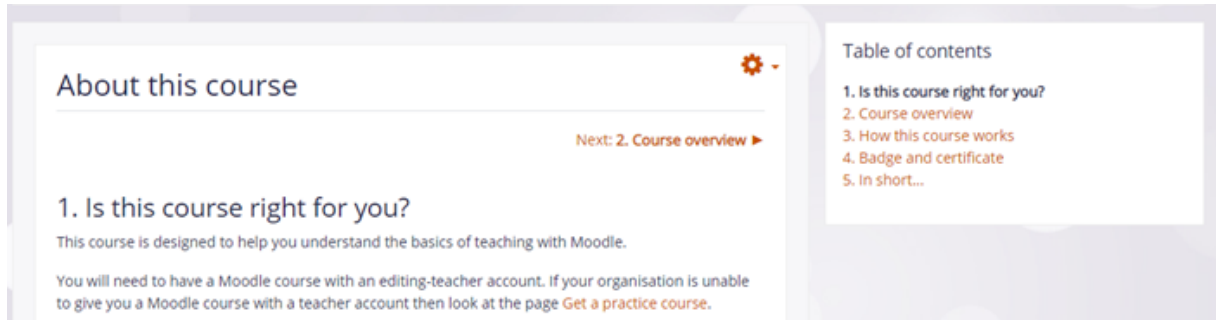
Cuối cùng, để thúc đẩy sự kết nối và năng lực phản biện, các hoạt động Forum (diễn đàn), Wiki và Workshop (hội thảo) đóng vai trò then chốt.

- Forum được sử dụng để tạo ra một diễn đàn thảo luận giữa giảng viên và sinh viên, và giữa sinh viên với nhau. Các thảo luận này có thể không đồng bộ về mặt thời gian.
- Wiki cho phép người tham gia tạo và chỉnh sửa một tập hợp các trang web (giống như một “Wikipedia thu nhỏ” trong khóa học). Wiki trong Moodle là công cụ linh hoạt giúp học viên và giảng viên cộng tác, ghi chú và biên soạn nội dung chung trong môi trường học tập trực tuyến.
- Workshop là công cụ giúp học viên học cách đánh giá và phản biện lẫn nhau, đồng thời giúp giảng viên đánh giá toàn diện hơn về cả năng lực làm bài và năng lực đánh giá của học viên. [7]

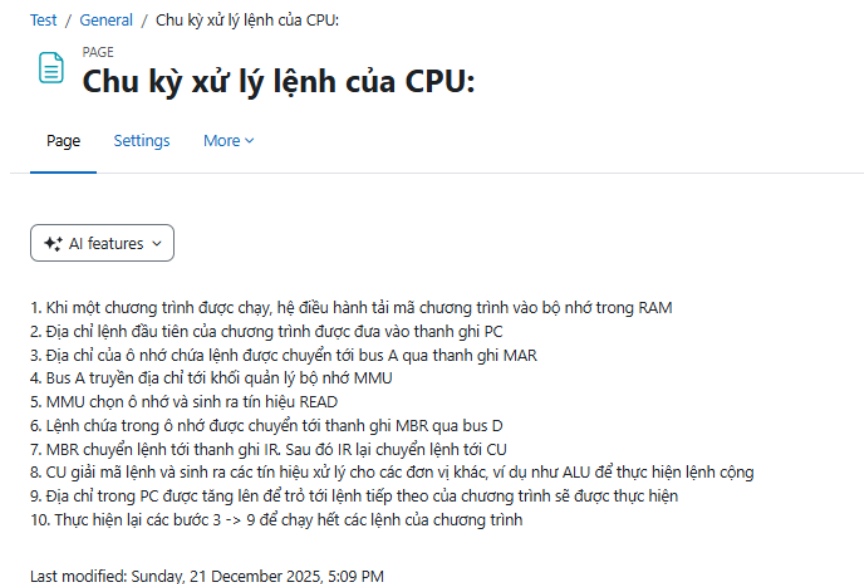
### 1.2.3. Tìm hiểu các tài nguyên của Moodle

Thay vì yêu cầu người học phải tải tài liệu về máy, giảng viên có thể sử dụng các công cụ giúp nội dung hiển thị ngay trên trình duyệt, giúp tối ưu hóa thời gian truy cập:

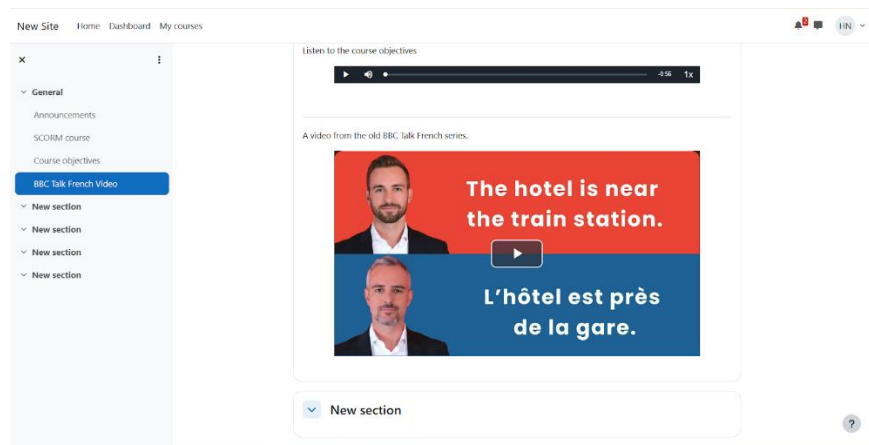
- Book (sách): Cho phép giảng viên tạo ra các tài liệu nhiều trang với định dạng giống như một cuốn sách, kèm theo mục lục rõ ràng.
- Page (trang): Thay vì tải lên một tệp Word hay PDF, giảng viên gõ hoặc dán nội dung trực tiếp vào Moodle. Điều này giúp học viên truy cập nội dung nhanh hơn vì không cần tải tệp về.
- Text and media area (văn bản và đa phương tiện): Khu vực hiển thị văn bản và đa phương tiện được dùng để chèn nội dung (văn bản, hình ảnh, video, âm thanh hoặc mã nhúng) xen kẽ giữa các tài nguyên và hoạt động khác trên trang khóa học của Moodle. [9]



Hình 1.12: Giao diện sinh viên xem nội dung của sách, điều hướng nội dung qua mục lục



Hình 1.13: Giao diện sinh viên xem nội dung trên một trang của Moodle



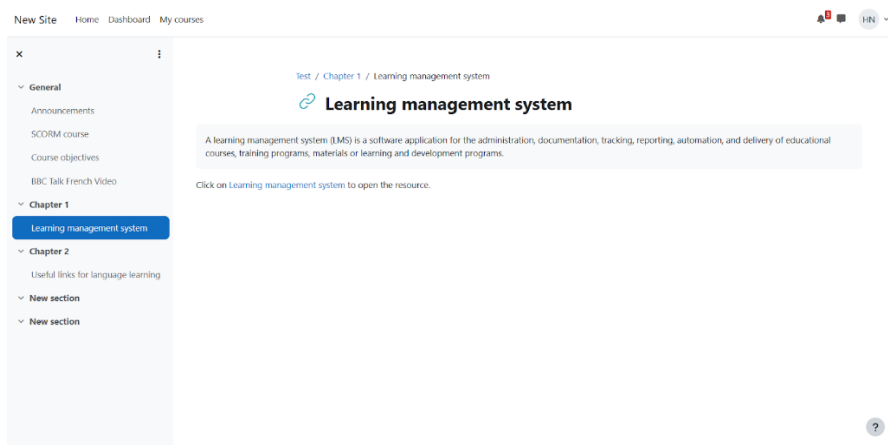
Hình 1.14: Giao diện sinh viên xem các nội dung đa phương tiện

Đối với các tài liệu cần lưu trữ hoặc nghiên cứu ngoại tuyến, Moodle cung cấp giải pháp quản lý tệp tin truyền thống nhưng vô cùng hiệu quả:

- File (tệp tin): Không chỉ giới hạn ở các tài liệu như văn bản soạn thảo hay bài trình chiếu - tất cả các loại tệp đều có thể được tải lên và truy cập thông qua Moodle. Tuy nhiên, để mở được các tệp này, sinh viên cần có phần mềm phù hợp tương ứng với định dạng tệp.
- Folder (thư mục): Cho phép giảng viên hiển thị nhiều tệp của khóa học cùng lúc. [9]

Để mở rộng kho học liệu ra ngoài phạm vi hệ thống, Moodle hỗ trợ kết nối với các nguồn tài nguyên số toàn cầu:

- IMS Content Package: Cho phép tải lên và hiển thị một gói nội dung đã được đóng gói theo chuẩn IMS. Dùng module này để “chạy” một cuốn sách điện tử, một bài giảng đa phương tiện, hoặc một website nhỏ đã được đóng gói sẵn.
- URL: Giúp giảng viên chèn liên kết web vào khóa học như một tài nguyên học tập. Điều này giúp tiết kiệm thời gian cho học viên, thay vì họ phải tự gõ hoặc tìm kiếm địa chỉ trang web. [9]



Hình 1.15: Giao diện học viên nhấp vào URL “Learning management system” để điều hướng đến trang web cụ thể

### **1.2.4. Đánh giá ưu điểm và hạn chế của hệ thống Moodle**

#### **1.2.4.1. Ưu điểm**

Moodle được sử dụng rộng rãi trong các trường đại học và hoạt động như một “trung tâm” dạy và học trực tuyến duy nhất do giảng viên kiểm soát, thông qua đó các tài liệu và hoạt động học tập trực tuyến có thể được triển khai. LMS này dễ sử dụng nhưng rất linh hoạt, cho phép giảng viên tự do phát triển và tổ chức khoa học theo gần như vô số cách khác nhau nhằm đáp ứng nhu cầu của chương trình đào tạo và người học.

Giảng viên có thể chia sẻ tệp tin với sinh viên một cách đơn giản bằng cách kéo và thả trực tiếp vào trang khóa học, đồng thời tạo ra nhiều hoạt động học tập mang tính tương tác xoay quanh các tài nguyên này để giúp sinh viên tham gia vào nội dung môn học hoặc tương tác với bạn học. Nhiều hoạt động học tập sẵn có hỗ trợ đánh giá (bao gồm đánh giá cá nhân, theo nhóm) cùng với nhiều cơ chế hỗ trợ phản hồi (như rubric hay hướng dẫn chấm điểm).

Hệ thống cung cấp nhiều báo cáo tích hợp sẵn, tự động theo dõi tiến trình của người dùng trong toàn bộ hệ thống, cho phép giảng viên tạo báo cáo cho từng khóa học hoặc từng hoạt động (có thể xuất kết quả ra bảng tính nếu cần). Các báo cáo này cho biết thời điểm sinh viên truy cập khóa học cũng như việc họ đã truy cập hay tham gia vào một hoạt động cụ thể hay chưa. [10]

#### **1.2.4.2. Nhược điểm**

Mặc dù LMS Moodle cung cấp nhiều cơ hội để hỗ trợ và mở rộng hoạt động giảng dạy hiện có, các công cụ cần được áp dụng phù hợp và cẩn trọng thì mới có thể tạo ra tác động tích cực đến trải nghiệm học tập.

Giảng viên thường bàn giao Moodle cho giảng viên mới mà không giải thích cách khóa học đã được xây dựng, khiến những giảng viên mới gặp khó khăn trong việc quản lý nội dung và người học. Do đó, cần có kế hoạch và thời gian phù hợp để quá trình bàn giao diễn ra suôn sẻ. [10]

### **1.3. Kết luận chương I**

Moodle là một LMS mã nguồn mở được sử dụng rộng rãi trong giáo dục và đào tạo. Moodle cung cấp đầy đủ các chức năng cần thiết cho việc tổ chức, quản lý và triển khai hoạt động dạy - học trực tuyến như quản lý khóa học, người dùng, tài nguyên học tập, hoạt động tương tác, đánh giá và theo dõi tiến trình học tập của người học.

Tuy vẫn còn một số hạn chế nhất định về kỹ năng quản trị hệ thống, nhưng nhìn chung Moodle là một nền tảng LMS đáng tin cậy, phù hợp để triển khai trong môi trường giáo dục hiện đại. Những nội dung đã trình bày trong chương này là cơ sở cho việc nghiên cứu, phân tích và xây dựng các giải pháp học tập trực tuyến dựa trên Moodle ở các chương tiếp theo.

## CHƯƠNG II. NGHIÊN CỨU, PHÁT TRIỂN CHỨC NĂNG ĐIỀU HƯỚNG HỌC TẬP TRONG MOODLE

Trong chương II, đồ án sẽ trình bày tổng quan về điều hướng học tập, phân tích cơ chế điều hướng mặc định trong Moodle, đề xuất và thiết kế giải pháp kỹ thuật cho bài toán này.

### 2.1. Tổng quan về chức năng điều hướng học tập

Trong môi trường giáo dục trực tuyến, khi kho tàng tri thức trên các LMS ngày càng trở nên khổng lồ, khả năng dẫn dắt người học đi đúng lộ trình trở thành yếu tố then chốt quyết định thành công của một khóa học. Sự phát triển của các LMS đã chuyển dịch trọng tâm từ việc quản lý nội dung đơn thuần sang việc tối ưu hóa trải nghiệm học tập cá nhân. Trong bối cảnh giáo dục trực tuyến hiện đại, việc cung cấp nội dung học tập theo phương thức tuyến tính và đồng nhất đang dần bộc lộ những hạn chế nghiêm trọng, đặc biệt là trong việc duy trì động lực và tối ưu hóa lộ trình nhận thức của người học. Để giải quyết vấn đề này, các LMS hiện đại như Moodle đã tích hợp các cơ chế cho phép giảng viên kiểm soát luồng học tập dựa trên dữ liệu hành vi và kết quả thực tế. Phần này sẽ phân tích cơ sở lý thuyết của ANS và sự cần thiết của việc phát triển các công cụ giới hạn truy cập dựa trên mô hình người dùng, đồng thời đưa ra các phân tích về kiến trúc tương quan của Moodle để có thể xây dựng nên một hệ thống hỗ trợ điều hướng thích ứng hoạt động được trong hệ thống quản lý học tập Moodle.

#### 2.1.1. Khái niệm và định nghĩa

Điều hướng học tập được hiểu là tập hợp các phương thức, công cụ và cấu trúc giao diện được thiết kế để dẫn dắt người học đi qua các nội dung, hoạt động và tài liệu giảng dạy theo một trình tự logic. Theo nghiên cứu của Brusilovsky và J. Eklund, E. Schwarz (1998) [11], ông mô tả: “ANS nhằm mục đích giúp học viên tìm ra con đường tối ưu trong không gian siêu văn bản của tài liệu học tập” (ANS is aimed to help students to find an “optimal path” through the hyperspace of learning material). Ông nhấn mạnh rằng thay vì để người học tự tìm đường một cách thụ động, hệ thống cần chủ động cung cấp các chỉ dẫn hoặc ràng buộc để cá nhân hóa hành trình học tập dựa trên đặc điểm của từng cá nhân. Đó là một hệ thống hỗ trợ chiến lược, giúp người học định vị bản thân, xác định lộ trình và quản lý tiến trình học tập của mình trong không gian số.

Sự cần thiết của điều hướng nhằm giải quyết vấn đề “mất định hướng trong không gian siêu văn bản” - một trạng thái mà Manfred Thuerling và cộng sự (1995) [12] mô tả là cảm giác khi người đọc không biết mình đang ở đâu, làm thế nào để đến được mục tiêu hoặc nên đi đâu tiếp theo. Ông lập luận rằng khi người học phải đối mặt với quá nhiều lựa chọn mà không có chỉ dẫn, họ sẽ rơi vào trạng thái quá tải nhận thức, dẫn đến việc mất dấu mục tiêu học tập ban đầu. Do đó, điều hướng hiện đại trong LMS (như Moodle) được định nghĩa là hệ thống dẫn dắt thông minh, nơi các quy tắc sư phạm được

mã hóa thành các thuật toán hiển thị, nhằm cá nhân hóa lộ trình dựa trên dữ liệu thời gian thực của người học.

Trong ngữ cảnh của đồ án nghiên cứu, phát triển plugin có chức năng điều hướng học tập bằng cách giới hạn truy cập cho Moodle, việc áp dụng ANS không chỉ đơn thuần là một tính năng kỹ thuật (ẩn/hiện hoạt động), mà là sự hiện thực hóa của hai kỹ thuật sư phạm chính được Brusilovsky đề xuất và kiểm chứng:

- Ẩn thích ứng: Kỹ thuật này giới hạn không gian điều hướng bằng cách ẩn hoàn toàn các liên kết đến nội dung mà người học chưa sẵn sàng tiếp cận. Ví dụ, một bài kiểm tra nâng cao sẽ không xuất hiện trong danh sách cho đến khi sinh viên hoàn thành các bài tập cơ bản. Nghiên cứu thực nghiệm trên hệ thống ISIS-Tutor cho thấy việc ẩn bớt các liên kết không liên quan giúp giảm đáng kể thời gian tìm kiếm và số bước điều hướng sai lầm của người học [13]. Trong Moodle, điều này tương ứng với tùy chọn “Ẩn hoàn toàn” trong thiết lập hạn chế truy cập.
- Chú thích thích ứng: Thay vì ẩn hoàn toàn, hệ thống vẫn hiển thị liên kết nhưng thay đổi hình thức trình bày (ví dụ: làm mờ, thêm biểu tượng ổ khóa, hoặc đổi màu sắc) để báo hiệu trạng thái của nội dung (ví dụ: “chưa sẵn sàng”, “được đề xuất”, “đã hoàn thành”) [13], [14]. Kỹ thuật này cung cấp một bản đồ tư duy, giúp người học hình dung được cấu trúc tổng thể của khóa học và biết được những gì đang chờ đợi họ phía trước, từ đó tạo động lực phấn đấu. Trong Moodle, điều này thể hiện qua việc hiển thị module bị làm mờ, kèm theo thông tin điều kiện truy cập.

### 2.1.2. Cơ sở lý thuyết về cách tiếp cận

Việc xây dựng các hệ thống thích ứng thường dựa trên mô hình người dùng. Theo Brusilovsky và Millán (2007), kiến thức của người sử dụng là đặc điểm quan trọng nhất cần được mô hình hoá [15]. Plugin được xây dựng sẽ sử dụng mô hình thang đo, một hình thức dùng để định lượng kiến thức người dùng một cách đơn giản nhưng hiệu quả. Cụ thể, mô hình này sẽ biểu diễn trình độ người dùng bằng một giá trị duy nhất trên một thang đo cụ thể (ví dụ: điểm số từ 0 đến 10 điểm). Các giá trị này được thu thập thông qua quá trình kiểm tra khách quan, nhằm giúp hệ thống nắm được mức độ kiến thức của người học mà không đòi hỏi các cấu trúc suy luận phức tạp.

Tuy nhiên, điểm yếu của mô hình này cũng rất rõ ràng. một trong số đó chính là độ chính xác thấp (“The shortcoming of the scalar model is it’s low precision” - Brusilovsky và Millán (2007) [15]). Mô hình thang đo biểu diễn một cách trung bình hoá tri thức của người học, ví dụ: một sinh viên đạt điểm 7/10 có thể là họ đã hiểu sâu 7 phần và không biết 3 phần, hoặc họ hiểu hời hợt ở cả 10 phần và tổng cộng của chúng là 7. Việc này sẽ dẫn đến các điều hướng rất khác nhau: trong trường hợp đầu tiên sinh viên sẽ được tìm hiểu về 3 phần tri thức chưa được biết, còn trong trường hợp thứ hai, sinh viên cần làm các bài nâng cao ở tất cả các phần tri thức. Mô hình thang đo không biểu diễn được cấu trúc kiến thức của người học, cũng như gặp phải nhiều vấn đề về

tính thời điểm và các yếu tố nhiễu. Brusilovsky và Millán (2007) nhấn mạnh kiến thức là một đặc điểm có thể thay đổi, có thể tăng hoặc giảm qua các phiên học [15], trong khi điểm số qua một lần kiểm tra duy nhất không phản ánh được sự thay đổi này. Đồng thời, mô hình cũng gặp vấn đề với các nhiễu như đoán mò (sinh viên không biết kiến thức nhưng vẫn chọn đáp án đúng), từ đó đưa ra các nhận định sai về trạng thái kiến thức của người học, làm suy giảm độ chính xác của hệ thống. Đối với các hệ thống thích ứng phức tạp, nơi cần phải tính toán đến các khía cạnh trong kiến thức của người dùng, mô hình thang đo là không đủ. Vì các lý do trên, mô hình thang đo thường được sử dụng trong các hệ thống thích ứng đơn giản, đi kèm với một số giải pháp giảm thiểu điểm yếu, ví dụ như: Kết hợp đánh giá toàn chương học bao gồm nhiều thành phần kiến thức nhằm đưa ra một giá trị định lượng tương đối chính xác cho kiến thức tổng thể của người học; kết hợp đánh giá bài tập đã hoàn thành được phân chia theo mức độ (ví dụ: từ 1 đến 4) để đảm bảo đánh giá kiến thức của người học được chính xác nhất.

Plugin giới hạn truy cập dựa trên điểm số và hoàn thành chính là một công cụ hỗ trợ điều hướng thích ứng dựa trên mô hình thang đo. Nó tận dụng dữ liệu sẵn có trong cơ sở dữ liệu của Moodle để nhận dữ liệu điểm và trạng thái hoàn thành, từ đó tự động hóa quy trình điều hướng, thay thế cho việc giảng viên phải can thiệp thủ công để mở khóa nội dung cho từng sinh viên.

## 2.2. Kiến trúc hệ thống Availability API và Events API trong Moodle

Để hiện thực hoá các lý thuyết về ANS vào một sản phẩm phần mềm cụ thể, cần phải đi sâu vào kiến trúc của Moodle. Moodle là hệ thống mã nguồn mở với kiến trúc module hoá cao. Từ phiên bản 3.1, Availability API sẽ mặc định được bật. Hệ thống này cho phép mở rộng không giới hạn các loại điều kiện thông qua cài đặt plugin. [16]

### 2.2.1. Cấu trúc dữ liệu JSON

Trong kiến trúc Availability của Moodle, trong cơ sở dữ liệu, thay vì lưu Availability thành một bảng riêng, với các điều kiện nằm trong nhiều bảng phụ như mô hình quan hệ chuẩn hoá, Moodle sử dụng một trường văn bản duy nhất tên là availability trong bảng mdl\_course\_modules (đối với hoạt động) và mdl\_course\_sections (đối với chương) để lưu trữ toàn bộ logic dưới định dạng JSON. Trong phiên bản 5.0, điều kiện này cũng có thể được sử dụng cho các mục trong một module khoá học. [16]

Thiết kế này mang lại sự linh hoạt cho hệ thống, nó cho phép biểu diễn các biểu thức logic phức tạp, lồng ghép nhiều cấp mà không cần thay đổi cấu trúc cơ sở dữ liệu khi thêm plugin mới thuộc loại Availability.

Cấu trúc JSON của Availability gồm những thành phần sau:

*Bảng 2.1: Các thành phần cấu trúc JSON Availability*

| Thành phần            | Ý nghĩa                 | Mô tả kỹ thuật   |
|-----------------------|-------------------------|--|
| op (Operator)         | Toán tử logic           | Xác định cách kết hợp các điều kiện con. Giá trị "&" tương ứng với AND (Tất cả đều đúng), giá trị " " tương ứng với OR (Ít nhất 1 điều kiện đúng)        |
| c (Children)          | Danh sách điều kiện con | Một mảng chứa các đối tượng. Mỗi đối tượng có thể là một điều kiện cụ thể hoặc một tập các điều kiện   |
| showc (Show children) | Trạng thái hiển thị     | Mảng kiểu logic quy định việc hiển thị thông báo lỗi cho từng điều kiện con. true nghĩa là có hiển thị thông tin về hạn chế, false nghĩa là ẩn hoàn toàn |
| type                  | Loại plugin             | Định danh plugin chịu trách nhiệm xử lý điều kiện (ví dụ: điểm số, trạng thái hoàn thành, ngày, hồ sơ). Đây là từ khóa để Moodle gọi đúng lớp xử lý      |

Một ví dụ về chuỗi JSON cho điều kiện "Phải đạt điểm  $\geq 5$  cho bài Quiz 1":

```
{
  "op": "&",
  "c": [
    {
      "type": "grade",
      "id": 6,
      "min": 5
    }
  ],
  "showc": [true]
}
```

Phân tích: Plugin điều kiện grade yêu cầu id (Grade Item ID) và min (điểm tối thiểu).

### 2.2.2. Hệ thống Core Availability và cơ chế kế thừa

Để đảm bảo tính nhất quán và khả năng mở rộng, Moodle cung cấp một bộ khung các lớp PHP trong namespace `\core_availability` mà mọi plugin phải tuân thủ.

#### 2.2.2.1. Lớp quản lý trạng thái `\core_availability\info`

Lớp `\core_availability\info` đóng vai trò là tầng trừu tượng, cung cấp giao diện thống nhất để các thành phần bên ngoài như định dạng khóa học hoặc sổ điểm tương tác với hệ thống Availability. Thay vì xử lý trực tiếp logic điều kiện phức tạp, các thành phần này chỉ cần tương tác với đối tượng info.



- Phân cấp đối tượng: Tùy thuộc vào ngữ cảnh, hệ thống sẽ sử dụng các lớp con cụ thể là `info_module` (cho hoạt động/tài liệu) hoặc `info_section` (cho các chương).
- Quá trình phân tích: Đối tượng này chịu trách nhiệm phân tích chuỗi định dạng JSON lưu trong cơ sở dữ liệu (trường `availability` trong bảng `mdl_course_modules` hoặc `mdl_course_sections`) và chuyển đổi chúng thành một cấu trúc cây đối tượng trong RAM.

Hiệu suất của hệ thống Availability phụ thuộc chặt chẽ vào cơ chế bộ nhớ đệm của Moodle để tránh quá tải truy vấn SQL:

- Truy xuất nhanh: Hàm `get_fast_modinfo($course)` là một hàm toàn cục, đóng vai trò là điểm khởi đầu, trả về đối tượng `course_modinfo`. Đối tượng này chứa toàn bộ thông tin cấu trúc khóa học gồm một mảng các `cm_info`.
- Vị trí lưu trữ: Dữ liệu được lưu trữ trong MUC, cụ thể là lưu trong `core/coursemodinfo`.
- Liên kết dữ liệu: Lớp `info` không trực tiếp quản lý cache mà dựa vào thuộc tính `availability` đã được nạp sẵn trong đối tượng `cm_info`. Điều này giúp việc kiểm tra điều kiện diễn ra gần như tức thì trên RAM.

Phương thức `is_available()`: Đây là phương thức quan trọng nhất, kích hoạt toàn bộ quy trình kiểm tra quyền truy cập. Hàm đầy đủ bao gồm các tham số tối ưu hóa:

```
public function is_available(&$information, $grabthelot = false, $userid = 0,
?course_modinfo $modinfo = null)
```

Tham số `$information`: Dùng để nhận về thông báo giải thích lý do tại sao người dùng không thể truy cập (nếu có).

Tham số `$grabthelot`: Đây là biến quyết định hiệu suất khi xử lý logic phức tạp.

- Nếu sai: Hệ thống dừng kiểm tra ngay khi gặp điều kiện đầu tiên không thỏa mãn.
- Nếu đúng: Hệ thống sẽ lưu toàn bộ thông tin cần thiết cho tất cả các module khóa học, giúp trang chủ và các trang tương tự hoạt động nhanh hơn.

#### 2.2.2.2. Lớp logic đệ quy `\core_availability\tree`

Lớp này đại diện cho các nút nhánh trong cấu trúc cây điều kiện. Nó hiện thực hóa logic (AND, OR, NOT) cho phép lồng ghép các điều kiện phức tạp.

Cấu trúc dữ liệu dạng cây quản lý một mảng các nút con, trong đó mỗi nút con có thể là một condition (điều kiện đơn lẻ - nút lá) hoặc một cây khác (nhóm điều kiện - nút nhánh).

Thuật toán duyệt cây: Phương thức `check_available()` của lớp này thực hiện thuật toán duyệt cây theo chiều sâu:

- Nếu toán tử là “&” (AND), thuật toán sẽ trả về false ngay lập tức khi gặp một điều kiện con không thỏa mãn.

- Nếu toán tử là “|” (OR), thuật toán sẽ trả về true ngay khi gặp một điều kiện con thỏa mãn.

Xử lý ngoại lệ: Lớp tree chịu trách nhiệm đảm bảo tính toàn vẹn của hệ thống. Nếu một plugin điều kiện bị gỡ bỏ khỏi hệ thống nhưng dữ liệu vẫn còn trong JSON (dữ liệu mồ côi), lớp tree sẽ trả về điều kiện sai cho plugin điều kiện, đồng thời tạo một nút đặc biệt và đánh dấu ‘Invalid’ để giữ cấu trúc cây, sau đó thông báo lỗi cho giáo viên.

### 2.2.2.3. Lớp trừu tượng plugin \core\_availability\condition

Đây là lớp trừu tượng mà mọi plugin điều kiện (bao gồm plugin được phát triển trong đồ án này) bắt buộc phải kế thừa.

Cơ chế đa hình: Moodle Core không biết chi tiết về plugin. Nó chỉ gọi các phương thức được định nghĩa trong lớp cha condition. Tại thời điểm chạy, PHP sẽ thực thi mã nguồn cụ thể trong plugin.

Phương thức `is_available($not, $info, $grabthelot, $userid)`: Đây là nơi chứa logic nghiệp vụ của plugin. Các tham số được hiểu như sau:

- `$not` (boolean): Cho biết điều kiện có bị đảo ngược hay không. Plugin phải tự xử lý việc đảo ngược kết quả true/false.
- `$userid` (int): ID của người dùng cần kiểm tra. Lưu ý: không được sử dụng biến toàn cục `$USER->id` ở đây, vì hệ thống có thể đang kiểm tra quyền truy cập cho một người dùng khác (ví dụ: giáo viên xem danh sách sinh viên).
- `$grabthelot` (boolean): Đây là biến gợi ý hiệu suất. Khi biến này là true, plugin sẽ lưu thông tin cho tất cả module khóa học giúp trang chủ và các trang tương tự hoạt động nhanh hơn.

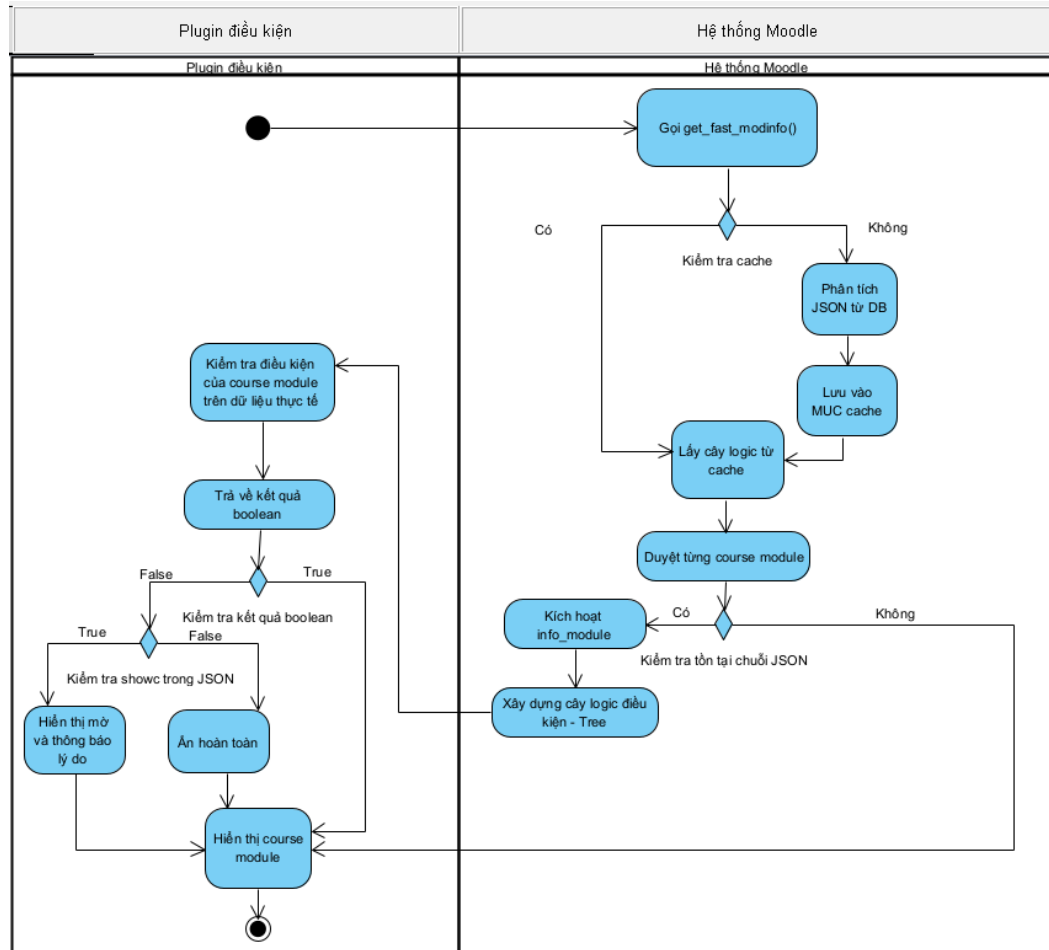
Phương thức `get_description()`: Phương thức này chịu trách nhiệm tạo ra văn bản thông báo cho người học (ví dụ: “Bạn cần đạt 5 điểm”). Phương thức này phải xử lý đa ngôn ngữ và các tham số động (như điểm số yêu cầu) để cung cấp thông tin ngữ cảnh chính xác cho người học.

### 2.2.3. Quy trình kiểm tra quyền truy cập

Khi một sinh viên truy cập khóa học, một loạt hoạt động diễn ra để tính toán xem liệu sinh viên đó có quyền xem một hoạt động hay không. Cụ thể như sau:

- Khởi tạo modinfo: Hàm `get_fast_modinfo()` được gọi để tải toàn bộ thông tin về các module trong khóa học. Hàm này sử dụng MUC để lưu trữ cấu trúc khóa học và các chuỗi JSON nằm trong trường `availability` đã được phân tích sơ bộ. Điều này cực kỳ quan trọng vì việc phân tích JSON cho hàng trăm module mỗi lần tải trang sẽ gây quá tải hệ thống.
- Duyệt danh sách module: Hệ thống duyệt qua từng module khóa học.
- Kiểm tra phương thức `is_available()`: Với mỗi module khóa học, phương thức `core_availability\info_module::is_available()` được kích hoạt.
- Ủy quyền: Core gọi phương thức `is_available()` của từng plugin con (ví dụ: plugin điểm số) được định nghĩa trong JSON.

- Truy vấn dữ liệu: Plugin thực hiện kiểm tra dữ liệu thực tế (ví dụ: truy vấn điểm trong mdl\_grade\_grades).
- Kết quả và hiển thị: Kết quả logic được trả về. Nếu sai, hệ thống tiếp tục kiểm tra mảng showc để quyết định hiển thị module ở dạng mờ kèm thông báo hay ẩn hoàn toàn khỏi HTML.



Hình 2.1: Biểu đồ hoạt động cho quy trình kiểm tra quyền truy cập

## 2.2.4. Hệ thống Core Event

Mặc dù Core Availability hoạt động một cách trơn tru, nhưng suy cho cùng nó chỉ đóng vai trò “cánh cổng”, hoạt động một cách thụ động. Giảng viên sẽ phải điều chỉnh các điều kiện một cách thủ công trong cài đặt cho từng module khóa học. Để hiện thực hoá khả năng tự động điều chỉnh các hạn chế truy cập, hệ thống cần phải có một cơ chế “lắng nghe” các thay đổi từ phía người dùng. Moodle giải quyết vấn đề này thông qua Events API (Core Event) để các thành phần của hệ thống giao tiếp với nhau mà không cần phụ thuộc trực tiếp vào mã nguồn của nhau.

### 2.2.4.1. Kiến trúc và nguyên lý vận hành

Theo tài liệu từ trang chủ của Moodle, một sự kiện là những mẫu thông tin nguyên tử mô tả điều gì đó đã xảy ra trong Moodle. Sự kiện chủ yếu là kết quả của các hành động của người dùng, nhưng cũng có thể là kết quả của tiến trình cron hoặc các hành động quản trị được thực hiện thông qua dòng lệnh.

Khi một hành động diễn ra, một sự kiện được tạo ra bởi Core API hoặc plugin. Hệ thống sự kiện sau đó sẽ truyền tải thông tin sự kiện này đến những quan sát viên đã đăng ký sự kiện đó. Bằng cách này, hệ thống sự kiện hoạt động như một xương sống truyền thông trong toàn bộ hệ thống Moodle. Quan sát viên quan sát sự kiện không thể sửa đổi dữ liệu sự kiện hoặc làm gián đoạn việc gửi sự kiện, đây là kênh truyền thông một chiều. [17]

Quy trình xử lý sự kiện diễn ra qua các bước kỹ thuật sau:

- Khởi tạo: Khi một hành động xảy ra, một đối tượng kế thừa từ `\core\event\base` được tạo ra. Ví dụ, sự kiện tạo mới module khoá học được định nghĩa trong tệp `lib/classes/event/course_module_created.php`. Khi có một module khoá học được tạo mới, `course_module_created.php` được thực thi và tạo ra một đối tượng với các tham số của module khoá học vừa được tạo.
- Kích hoạt: Phương thức `$event->trigger()` được gọi. Phương thức này là phương thức được định nghĩa trong lớp `\core\event\base`, vì vậy mỗi một đối tượng kế thừa đều có phương thức này. Phương thức này sẽ thực thi `\core\event\manager::dispatch($event)`, đưa sự kiện vào hàng chờ xử lý của hệ thống.
- Điều phối: Lớp manager truy xuất danh sách các quan sát viên đã được đăng ký trong tệp `db/events.php` của các plugin. Hệ thống sử dụng cơ chế bộ nhớ đệm để lưu trữ danh sách này nhằm tối ưu hiệu năng. [17]
- Thực thi callback: Các hàm được định nghĩa trong tham số callback của quan sát viên sẽ được gọi và nhận đối tượng `$event` làm tham số đầu vào.

#### 2.2.4.2. Giải pháp tự động áp dụng điều kiện hạn chế

Bằng cách kích hoạt các hàm trong tham số callback của quan sát viên được định nghĩa trong `\db\events.php` mỗi khi lắng nghe được sự kiện tương ứng, plugin có thể tự động áp dụng điều kiện hạn chế lên các module khoá học được tạo, được chỉnh sửa,... Giảng viên không cần phải thực hiện các thao tác này một cách thủ công.

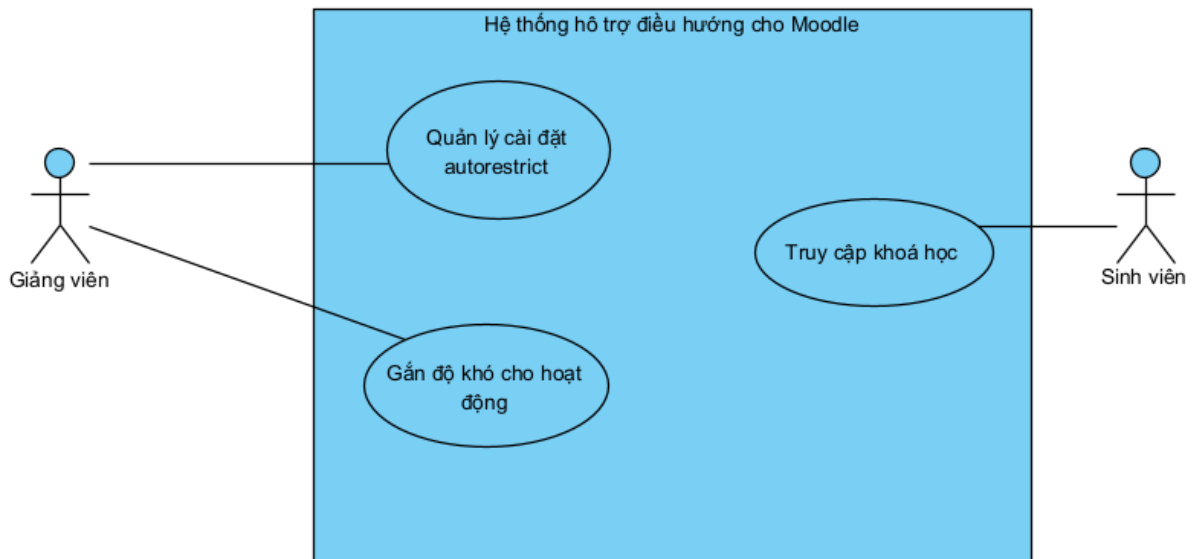
Điểm yếu của phương pháp này chính là vấn đề với các module khoá học đã được tạo trước và không cần chỉnh sửa, thường gặp với các khoá học được tạo và sử dụng trước khi cài đặt plugin được phát triển trong đồ án này. Cách khắc phục thường thấy nhất là cùng thời điểm với khi bật tự động áp dụng hạn chế, plugin sẽ duyệt qua toàn bộ các module khoá học có trong khoá học, tiếp theo là áp dụng điều kiện hạn chế định sẵn lên chúng. Sau khi bật tự động áp dụng hạn chế, mỗi lần muốn tạo mới module khoá học, các điều kiện hạn chế sẽ được cập nhật một cách tự động.

### 2.3. Phân tích và thiết kế hệ thống

#### 2.3.1. Phân tích chức năng hệ thống

Để khắc phục điểm yếu của mô hình thang đo, hệ thống cần phải kết hợp đa dạng các điều kiện nhằm đánh giá cấu trúc kiến thức của người học một cách chính xác nhất. Tổng cộng 3 plugin điều kiện dạng Availability sẽ được phát triển, bao gồm diffcomplete

(hoàn thành độ khó), sectioncomplete (hoàn thành chương), sectiongrade (điểm số chương). Đồng thời, một plugin cục bộ tên là autorestrict (tự động hạn chế truy cập) được phát triển nhằm quản lý các điều kiện hạn chế và tự động thêm các điều kiện hạn chế. Cấu trúc tổng quan của hệ thống được thể hiện trong biểu đồ use case dưới đây:



Hình 2.2: Biểu đồ use case của hệ thống hỗ trợ điều hướng cho Moodle

Để làm rõ chức năng được thể hiện trong biểu đồ, sau đây sẽ là mô tả chi tiết nội dung của từng use case:

Bảng 2.2: Mô tả chi tiết use case trong hệ thống hỗ trợ điều hướng cho Moodle

| ID    | Tên                          | Mô tả   |
|-------|------------------------------|---|
| UC_01 | Quản lý cài đặt autorestrict | Giảng viên truy cập trang cài đặt tự động hạn chế truy cập của khóa học để bật/tắt tính năng tự động hạn chế truy cập và lưu các thiết lập. Khi bật và lưu, hệ thống tự động áp dụng các hạn chế cho tất cả hoạt động. Khi tắt, hệ thống xóa tất cả các hạn chế |
| UC_02 | Gắn độ khó cho hoạt động     | Giảng viên chọn một hoặc nhiều hoạt động và gán mức độ khó (diff1: Nhận biết, diff2: Thông hiểu, diff3: Vận dụng, diff4: Vận dụng cao) thông qua hệ thống tag (thẻ) của Moodle.   |
| UC_03 | Truy cập khoá học            | Sinh viên truy cập khóa học. Moodle gọi các plugin điều kiện tương ứng để kiểm tra. Nếu đủ điều kiện thì cho phép truy cập. Nếu không thì ẩn hoàn toàn hoạt động hoặc hiển thị điều kiện cần hoàn thành để mở khóa.   |

### 2.3.2. Kịch bản chuẩn

*Bảng 2.3: Kịch bản cho use case quản lý cài đặt autorestrict*

|                |   |
|----------------|---|
| Use case       | UC_01   |
| Tác nhân       | Giảng viên  |
| Tiền điều kiện | Giảng viên đã đăng nhập và có quyền quản lý khóa học<br>Plugin đã được cài đặt thành công   |
| Hậu điều kiện  | Cài đặt được lưu, các hạn chế được áp dụng hoặc xóa tự động   |
| Kịch bản chính | <ol style="list-style-type: none"> <li>1. Giảng viên truy cập trang khóa học -&gt; nhấn “More” -&gt; chọn “Tự động hạn chế truy cập”</li> <li>2. Hệ thống hiển thị giao diện cài đặt với các tùy chọn</li> <li>3. Giảng viên bật checkbox “Bật tự động hạn chế”</li> <li>4. Giảng viên cấu hình theo tiến độ hoàn thành chương (tùy chọn), cấu hình theo độ khó (tùy chọn), cấu hình theo chương (tùy chọn)</li> <li>5. Giảng viên nhấn nút Save Changes</li> <li>6. Hệ thống lưu cài đặt vào cơ sở dữ liệu</li> <li>7. Hệ thống áp dụng các hạn chế truy cập cho tất cả hoạt động thỏa mãn điều kiện</li> <li>8. Hệ thống hiển thị thông báo thành công</li> </ol> |
| Ngoại lệ       | <ol style="list-style-type: none"> <li>3.1. Giảng viên tắt checkbox “Bật tự động hạn chế”</li> <li>3.2. Giảng viên nhấn nút Save Changes</li> <li>3.3. Hệ thống xóa tất cả hạn chế</li> <li>4.1. Giảng viên nhập giá trị không hợp lệ (ví dụ: -1)</li> <li>4.2. Hệ thống hiển thị thông báo lỗi</li> <li>4.3. Giảng viên nhập lại giá trị đúng</li> </ol>   |

*Bảng 2.4: Kịch bản cho use case gán độ khó cho hoạt động*

|                |  |
|----------------|--|
| Use case       | UC_02  |
| Tác nhân       | Giảng viên   |
| Tiền điều kiện | Giảng viên đã đăng nhập và có quyền quản lý khóa học<br>Khóa học có ít nhất một hoạt động<br>Plugin đã được cài đặt thành công |

| Hậu điều kiện  | Hoạt động được gán thẻ độ khó  |
|----------------|--|
| Kịch bản chính | <ol style="list-style-type: none"> <li>1. Giảng viên truy cập trang khóa học -&gt; nhấn “More” -&gt; chọn “Tự động hạn chế truy cập”</li> <li>2. Giảng viên nhấn “Quản lý thẻ độ khó”</li> <li>3. Hệ thống hiển thị danh sách hoạt động</li> <li>4. Giảng viên chọn các hoạt động cần gán độ khó</li> <li>5. Giảng viên chọn mức độ khó từ danh sách sổ xuống</li> <li>6. Giảng viên nhấn “Áp dụng”</li> <li>7. Hệ thống gán tag cho các hoạt động được chọn</li> <li>8. Hệ thống hiển thị thông báo thành công</li> </ol> |
| Ngoại lệ       | <ol style="list-style-type: none"> <li>5.1. Giảng viên chọn “Không có độ khó (xóa thẻ)”</li> <li>5.2. Giảng viên nhấn nút “Áp dụng”</li> <li>5.3. Hệ thống xóa tag độ khó cho các hoạt động được chọn</li> <li>6.1. Giảng viên nhấn nút “Áp dụng” mà không chọn hoạt động</li> <li>6.2. Hệ thống hiển thị cảnh báo “Không có hoạt động nào được chọn”</li> </ol>   |

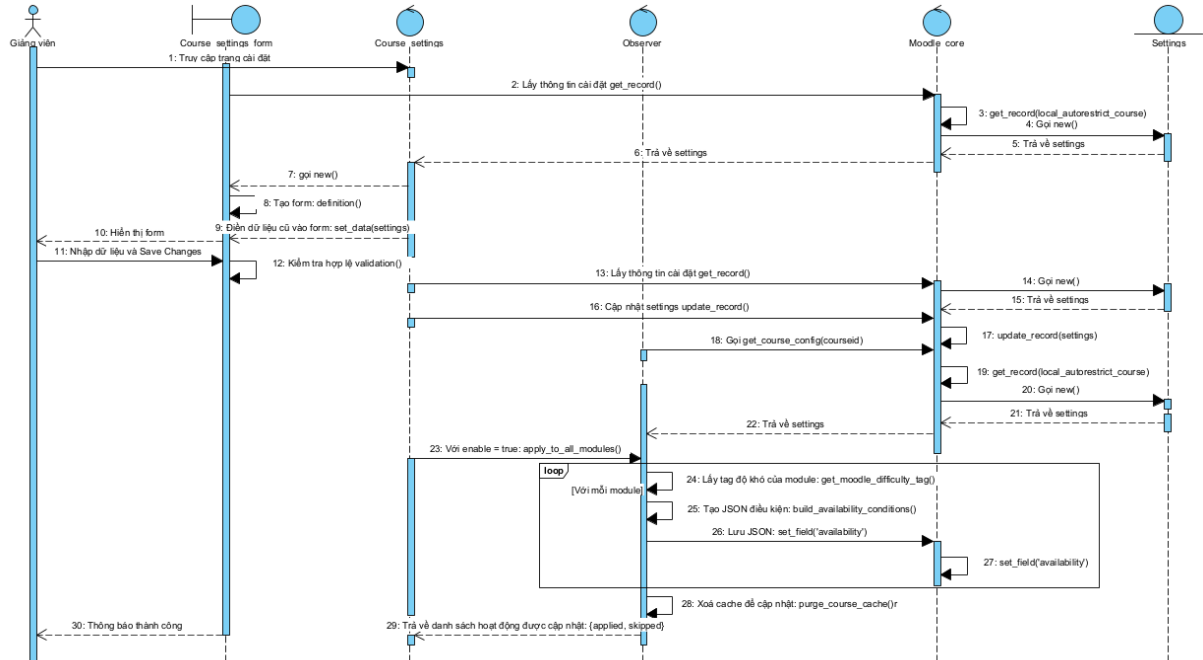
Bảng 2.5: Kịch bản cho use case truy cập khóa học

|                |   |
|----------------|---|
| Use case       | UC_03   |
| Tác nhân       | Sinh viên   |
| Tiền điều kiện | Sinh viên đã đăng nhập và đã được thêm vào khóa học<br>Plugin đã được cài đặt thành công  |
| Hậu điều kiện  | Sinh viên thấy các hoạt động được phép truy cập<br>Các hoạt động chưa đủ điều kiện được ẩn hoặc hiển thị với thông báo  |
| Kịch bản chính | <ol style="list-style-type: none"> <li>1. Sinh viên truy cập khóa học</li> <li>2. Hệ thống tải danh sách hoạt động, kiểm tra điều kiện hạn chế truy cập, hiển thị hoạt động đủ điều kiện truy cập</li> <li>3. Sinh viên truy cập và thực hiện hoạt động</li> <li>4. Hệ thống cập nhật trạng thái hoàn thành</li> <li>5. Sinh viên làm mới trang khóa học</li> <li>6. Hệ thống hiển thị các hoạt động mới được mở khóa (nếu có)</li> </ol> |
| Ngoại lệ       | 2.1. Hệ thống kiểm tra điều kiện hạn chế truy cập, ẩn hoàn toàn hoặc hiển thị xám kèm thông báo với các hoạt động chưa đủ điều kiện.  |

### 2.3.3. Biểu đồ tuần tự

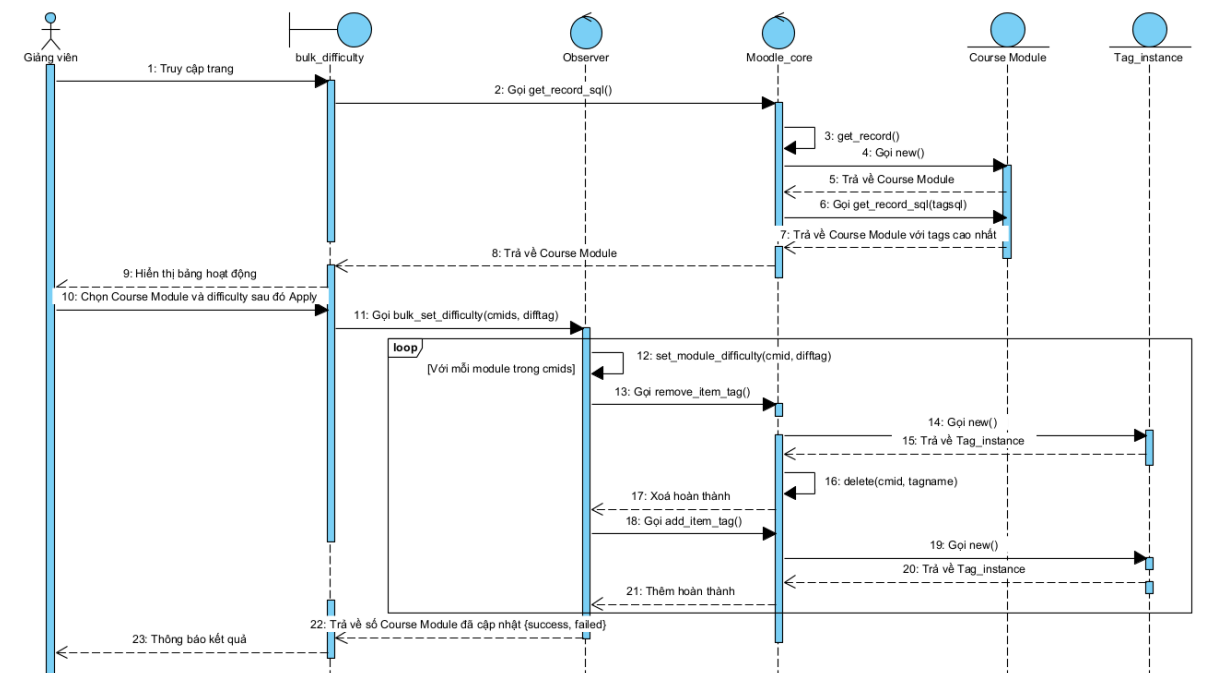
Dưới đây là biểu đồ tuần tự thể hiện luồng hoạt động của các use case:

Đầu tiên, plugin autorestrict giúp tự động hóa việc đặt điều kiện truy cập trong Moodle. Thay vì giảng viên phải vào từng bài học để chỉnh sửa thủ công, họ chỉ cần thiết lập quy tắc chung tại trang cài đặt khóa học, và plugin sẽ tự quét các thẻ của từng module để áp dụng điều kiện tương ứng.



Hình 2.3: Biểu đồ tuần tự cho use case quản lý cài đặt autorestrict

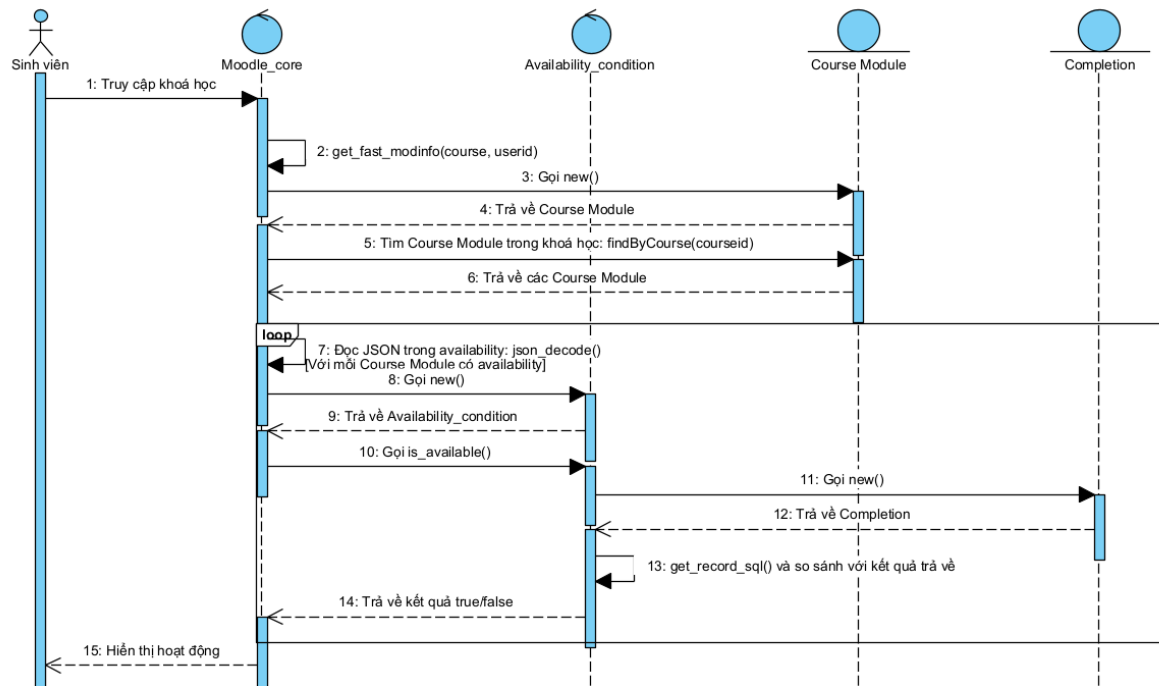
Biểu đồ này mô tả cách giảng viên gán thẻ độ khó cho nhiều hoạt động cùng lúc.



Hình 2.4: Biểu đồ tuần tự cho use case gán độ khó cho hoạt động



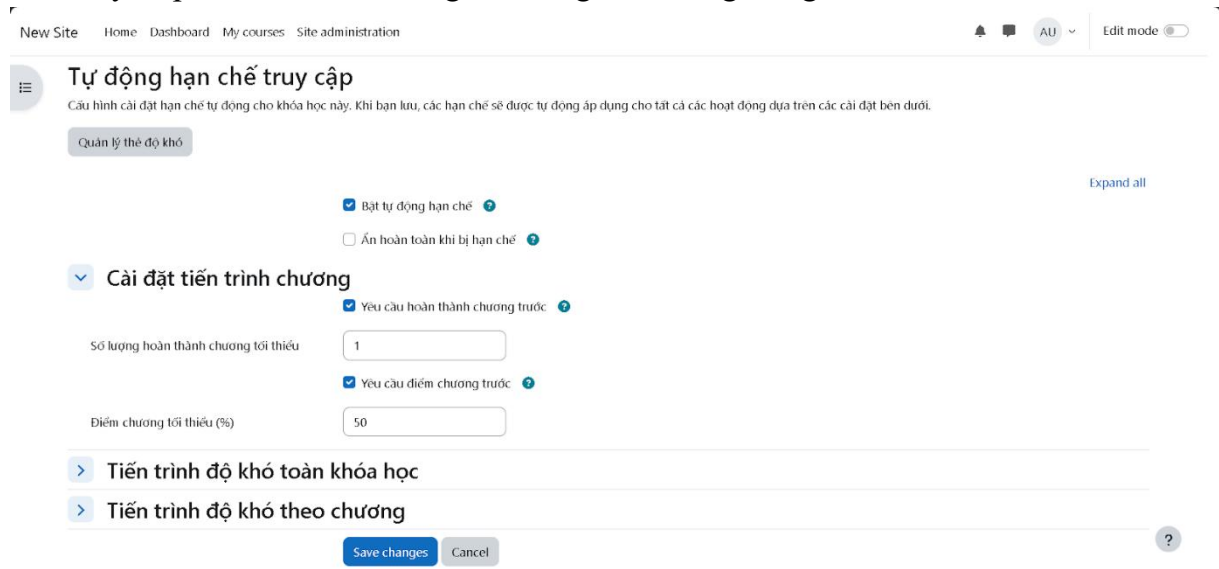
Cuối cùng là cơ chế xác định xem một sinh viên có đủ điều kiện (ví dụ: phải hoàn thành bài tập A mới được xem bài tập B) để nhìn thấy một hoạt động nào đó hay không.



Hình 2.5: Biểu đồ tuần tự cho use case truy cập khoá học

## 2.4. Một số hình ảnh giao diện

Đầu tiên là giao diện tự động hạn chế truy cập cho phép cấu hình các quy tắc để hệ thống tự động áp dụng các hạn chế truy cập cho toàn bộ các hoạt động trong khóa học thay vì phải cài đặt thủ công cho từng hoạt động riêng lẻ.



Hình 2.6: Trang quản lý của hệ thống

Tiếp theo là giao diện cấu hình điều kiện để mở khóa các nội dung cho toàn khóa học xác định xem học viên cần hoàn thành bao nhiêu hoạt động ở cấp độ này để được phép làm các hoạt động ở cấp độ cao hơn trong phạm vi tổng thể.

New Site Home Dashboard My courses Site administration

🔔 AU Edit mode

☰ **Tiến trình độ khó toàn khóa học**

☒ Yêu cầu tiến trình độ khó

Cấu hình số lượng hoàn thành cần thiết trên toàn bộ khóa học để mở khóa từng cấp độ khó.

|   |   |
|---|---|
| Số lượng hoàn thành Diff1 để mở khóa Diff2 (khóa học) | 2 |
| Số lượng hoàn thành Diff1 để mở khóa Diff3 (khóa học) | 3 |
| Số lượng hoàn thành Diff2 để mở khóa Diff3 (khóa học) | 2 |
| Số lượng hoàn thành Diff1 để mở khóa Diff4 (khóa học) | 4 |
| Số lượng hoàn thành Diff2 để mở khóa Diff4 (khóa học) | 3 |
| Số lượng hoàn thành Diff3 để mở khóa Diff4 (khóa học) | 2 |

> Tiến trình độ khó theo chương

Save changes Cancel

Hình 2.7: Chi tiết tiến trình độ khó toàn khoá học

Giao diện cấu hình điều kiện để mở khóa các nội dung theo chương tập trung vào điều kiện mở khóa trong phạm vi từng chương riêng biệt. Điều này đảm bảo học viên phải đạt được trình độ nhất định trong một chương trước khi tiến lên mức khó hơn của chính chương đó.

New Site Home Dashboard My courses Site administration

🔔 AU Edit mode

☰ **Tiến trình độ khó theo chương**

Cấu hình số lượng hoàn thành cần thiết trong cùng một chương để mở khóa từng cấp độ khó.

|   |   |
|---|---|
| Số lượng hoàn thành Diff1 để mở khóa Diff2 (chương) | 1 |
| Số lượng hoàn thành Diff1 để mở khóa Diff3 (chương) | 1 |
| Số lượng hoàn thành Diff2 để mở khóa Diff3 (chương) | 1 |
| Số lượng hoàn thành Diff1 để mở khóa Diff4 (chương) | 1 |
| Số lượng hoàn thành Diff2 để mở khóa Diff4 (chương) | 1 |
| Số lượng hoàn thành Diff3 để mở khóa Diff4 (chương) | 1 |

Save changes Cancel

Hình 2.8: Chi tiết tiến trình độ khó theo chương

Cuối cùng là giao diện minh họa cho một hệ thống học tập thích ứng. Thay vì mở bài học theo trình tự thời gian, hệ thống này buộc học viên phải đạt được các cột mốc về năng lực (điểm số) và khối lượng công việc (số bài đã làm) ở các cấp độ khó khác nhau thì mới có thể tiến tới các bài học tiếp theo.

Hình 2.9: Các điều kiện khi được điều chỉnh thủ công

Hình 2.10: Màn hình hiển thị điều kiện cần thiết để mở khoá

## 2.5. Kết luận chương II

Ở chương II này, đồ án đã trình bày về cơ sở lý thuyết và cách tiếp cận được sử dụng để xây dựng hệ thống hỗ trợ điều hướng tiến trình học tập của sinh viên, cũng như thiết kế các chức năng sẽ được triển khai trong LMS Moodle. Đồng thời, tầm quan trọng của dữ liệu điểm số trong mô hình được thiết kế đã đặt ra một câu hỏi lớn: làm thế nào để trải nghiệm học tập của sinh viên là liên tục?

Trong chương tiếp theo, đồ án sẽ đi vào chi tiết cách thức và phương pháp chấm điểm tự động cho bài tập, bài kiểm tra của sinh viên trong LMS, cũng như thiết kế và xây dựng một hệ thống chấm điểm tự động đáp ứng nhu cầu này.

## **CHƯƠNG III. NGHIÊN CỨU, PHÁT TRIỂN CHỨC NĂNG CHẤM ĐIỂM BÀI TẬP TỰ ĐỘNG TRONG CÁC HỆ THỐNG LMS**

Trong chương II, đồ án đã nghiên cứu và đề xuất mô hình học tập thích ứng nhằm mục tiêu cá nhân hóa lộ trình học tập. Tuy nhiên, để hệ thống có khả năng đưa ra các quyết định thích nghi chính xác và kịp thời, quá trình đánh giá năng lực người học cần phải đảm bảo tính liên tục và thời gian thực. Phương pháp đánh giá thủ công truyền thống hiện nay thường bộc lộ hạn chế về độ trễ, gây gián đoạn vòng lặp phản hồi - một yếu tố then chốt trong việc duy trì hiệu quả của quá trình cá nhân hóa.

Xuất phát từ yêu cầu cấp thiết đó, chương III tập trung vào việc nghiên cứu và thực thi hệ thống chấm điểm tự động. Đây được xem là thành phần cốt lõi trong cơ chế phản hồi của hệ thống, đóng vai trò như một bộ thu nhận, phân tích dữ liệu đầu vào và thực hiện đánh giá định lượng cũng như định tính một cách tự động, đảm bảo sự vận hành ổn định cho toàn bộ hệ thống học tập thích ứng.

### **3.1. Tổng quan**

#### **3.1.1. Hiện trạng và đặt vấn đề**

Trong các hệ thống LMS hiện hành, nhu cầu tự động hóa quy trình đánh giá kết quả học tập luôn là ưu tiên hàng đầu nhằm giảm tải áp lực quản lý lớp học. Hiện tại, các nền tảng phổ biến đã giải quyết khá tốt bài toán chấm điểm cho các dạng bài tập có cấu trúc cố định thông qua các công cụ tích hợp sẵn:

- Đối với bài tập trắc nghiệm: Hệ thống sử dụng cơ chế so khớp chuỗi chính xác hoặc so khớp mẫu cho các câu hỏi đóng, điền từ hoặc lựa chọn.
- Đối với bài tập lập trình: Hệ thống áp dụng cơ chế kiểm thử đơn vị, chạy mã nguồn của sinh viên trong môi trường cô lập và so sánh kết quả đầu ra với các bộ kiểm thử chuẩn.

Tuy nhiên, hạn chế cốt lõi của các phương pháp truyền thống này là sự bất lực trước các dạng bài tập tự luận hoặc các câu hỏi mở. Đây là những dạng bài tập đòi hỏi người chấm phải có khả năng đọc hiểu ngữ cảnh, phân tích tư duy và đánh giá ý nghĩa của văn bản - điều mà các thuật toán so khớp cứng nhắc hay kiểm thử đơn vị không thể thực hiện được. Chính vì vậy, quy trình đánh giá các bài tập này hiện nay vẫn phụ thuộc hoàn toàn vào thao tác thủ công của giảng viên, dẫn đến độ trễ lớn trong việc phản hồi và khó khăn khi triển khai ở quy mô lớn.

#### **3.1.2. Giải pháp đề xuất ứng dụng mô hình ngôn ngữ lớn**

Sự phát triển đột phá của các mô hình LLM trong thời gian gần đây đã mở ra một hướng tiếp cận hoàn toàn mới cho bài toán trên. Khác với các thuật toán truyền thống, LLM sở hữu khả năng NLP vượt trội, cho phép “đọc hiểu” nội dung văn bản, nắm bắt ý chính và thực hiện suy luận logic tương tự như con người.

Tận dụng năng lực này, đồ án đề xuất xây dựng một hệ thống chấm điểm tự động thông minh, được thiết kế để bổ sung vào khoảng trống mà quiz và VPL để lại. Giải pháp tập trung vào các chức năng chính sau:

- Số hóa dữ liệu đầu vào: Tích hợp công nghệ nhận dạng ký tự quang học để chuyển đổi các bài làm dạng hình ảnh hoặc PDF sang văn bản máy tính, giải quyết vấn đề nhập liệu cho bài làm viết tay.
- Đánh giá dựa trên ngữ nghĩa: Sử dụng LLM làm hạt nhân xử lý để so sánh nội dung bài làm với đáp án mẫu và tiêu chí chấm điểm. Hệ thống đánh giá dựa trên mức độ hiểu bài và ý tưởng thay vì chỉ so khớp từ khóa.
- Tạo phản hồi tự nhiên: Thay vì chỉ trả về điểm số, hệ thống tự động sinh ra các nhận xét chi tiết, giải thích lỗi sai bằng ngôn ngữ tự nhiên, giúp người học cải thiện kiến thức ngay lập tức.

### 3.2. Công nghệ sử dụng

Để hiện thực hóa giải pháp chấm điểm tự động thông minh được đề xuất tại mục 3.1.2, hệ thống cần một kiến trúc công nghệ đủ mạnh để xử lý các tác vụ phức tạp như: nhận dạng văn bản từ hình ảnh, phân tích ngữ nghĩa tự nhiên và đảm bảo tính tương tác thời gian thực trên nền tảng web. Đồ án lựa chọn bộ công nghệ dựa trên các tiêu chí: tính tương thích với hệ sinh thái LMS, hiệu năng xử lý AI và khả năng mở rộng.

#### 3.2.1. Tích hợp nền tảng Moodle

Moodle được chọn làm nền tảng cơ sở để phát triển hệ thống. Việc tận dụng tối đa kiến trúc nội tại của Moodle giúp đảm bảo tính ổn định và an toàn dữ liệu:

- Ngôn ngữ và quản trị dữ liệu: Plugin được phát triển trên nền tảng PHP (tương thích phiên bản 8.0+) và hệ quản trị cơ sở dữ liệu MariaDB. Mọi thao tác truy xuất dữ liệu bài làm, điểm số đều thông qua lớp trừu tượng dữ liệu của Moodle, đảm bảo tuân thủ các quy tắc bảo mật và toàn vẹn dữ liệu của hệ thống gốc.
- Events API (kiến trúc hướng sự kiện): Để giải quyết bài toán “tự động hóa” ngay khi sinh viên nộp bài, đồ án sử dụng cơ chế Events API. Hệ thống sẽ lắng nghe sự kiện nộp bài từ module Assignment. Cách tiếp cận này giúp tách biệt logic chấm điểm khỏi mã nguồn lõi của Moodle, giúp việc nâng cấp hoặc bảo trì không ảnh hưởng đến hoạt động chung của lớp học. [17]
- Task API (xử lý bất đồng bộ): Do việc gọi API tới LLM hoặc xử lý OCR tốn nhiều tài nguyên và thời gian, nếu xử lý trực tiếp sẽ gây treo giao diện người dùng. Đồ án sử dụng Adhoc Tasks để đẩy các tác vụ này vào hàng đợi chạy nền. Kỹ thuật này đảm bảo giảng viên và sinh viên vẫn có thể thao tác trên hệ thống mượt mà trong khi máy chủ thực hiện việc chấm điểm. [18]

#### 3.2.2. Dịch vụ LLM

Đây là thành phần trung tâm thay thế cho các thuật toán so khớp từ khóa truyền thống, chịu trách nhiệm “đọc hiểu” và “đánh giá” bài làm:

- Google Gemini API: Mô hình ngôn ngữ lớn gemini-pro/flash được tích hợp để tận dụng khả năng xử lý văn bản chuyên sâu. Với kiến trúc Transformer tiên tiến, mô hình này hỗ trợ phân tích các yêu cầu phức tạp và tạo phản hồi dựa trên ngữ cảnh giáo dục với độ chính xác cao. [19]
- Ollama (LLM cục bộ): Nhằm đảm bảo quyền riêng tư dữ liệu và chủ quyền dữ liệu, đồ án sử dụng Ollama để vận hành các mô hình mã nguồn mở như Llama 3 hoặc Qwen ngay trên máy chủ nội bộ. Điều này giúp hệ thống hoạt động linh hoạt, giảm thiểu chi phí API và bảo vệ thông tin cá nhân của người học. [20]

### 3.2.3. Dịch vụ OCR

OCR là một công nghệ đã được phát triển và ứng dụng rộng rãi, cho phép chuyển đổi các hình ảnh quét và tài liệu sang dạng văn bản điện tử có thể chỉnh sửa và tìm kiếm. Công nghệ này có nhiều ứng dụng trong nhiều lĩnh vực khác nhau và đã chứng minh vai trò quan trọng trong việc số hóa sách, tài liệu và hồ sơ. Một trong những lợi ích chính của công nghệ OCR là trích xuất văn bản từ hình ảnh, làm dữ liệu đầu vào cho các ứng dụng học máy và trí tuệ nhân tạo. Về khả năng hỗ trợ ngôn ngữ, công nghệ OCR có thể hỗ trợ nhiều ngôn ngữ khác nhau, bao gồm tiếng Anh, tiếng Tây Ban Nha, tiếng Trung và nhiều ngôn ngữ khác. Tuy nhiên, độ chính xác của OCR có thể thay đổi đáng kể tùy theo từng ngôn ngữ, trong đó một số ngôn ngữ khó nhận dạng hơn các ngôn ngữ khác.

Trên cơ sở những tiến bộ đó, nhiều thư viện và OCR framework hiện đại đã được phát triển nhằm hiện thực hóa các thành tựu nghiên cứu vào các ứng dụng thực tiễn, trong đó EasyOCR là một giải pháp tiêu biểu được sử dụng rộng rãi hiện nay. EasyOCR là một thư viện OCR được phát triển bằng ngôn ngữ Python, cung cấp các mô hình đã được huấn luyện sẵn chất lượng cao cho tiếng Việt. Thư viện này cho phép triển khai ứng dụng nhanh chóng mà không yêu cầu quá trình huấn luyện lại hoặc tinh chỉnh phức tạp trên tập dữ liệu mới. Điều này giúp giảm thiểu đáng kể thời gian phát triển và công sức chuẩn bị dữ liệu. Khả năng hỗ trợ tiếng Việt của EasyOCR hoàn toàn ưu việt đối với các văn bản có dấu, cũng như xử lý hiệu quả các trường hợp văn bản in. Đối với chữ viết tay, mô hình có thể xử lý được các trường hợp chữ viết rõ ràng, tuy nhiên độ chính xác sẽ không cao bằng văn bản in và có thể gặp khó khăn với các nét chữ viết không chuẩn. [21]

### 3.2.4. Chuẩn trao đổi dữ liệu

Do hệ thống bao gồm các thành phần không đồng nhất (Moodle chạy PHP, dịch vụ OCR/LLM chạy Python/C++), việc thiết lập chuẩn giao tiếp là bắt buộc để đảm bảo sự liên thông:

- Kiến trúc hướng dịch vụ: Hệ thống tách phần xử lý nặng (LLM/OCR) ra khỏi máy chủ Moodle, giao tiếp thông qua giao thức HTTP RESTful. Điều này giúp hệ thống LMS không bị quá tải khi số lượng bài tập cần chấm tăng đột biến.
- Định dạng JSON: JSON được sử dụng làm chuẩn đóng gói dữ liệu duy nhất. Các thông tin như nội dung bài làm, điểm số, hay lời nhận xét (từ LLM) đều

được cấu trúc hóa thành JSON. Định dạng này vừa nhẹ (tối ưu băng thông mạng), vừa đảm bảo tính nhất quán dữ liệu khi truyền tải giữa các môi trường lập trình khác nhau.

### 3.3. Cơ sở lý thuyết và kỹ thuật ứng dụng mô hình LLM

#### 3.3.1. Cơ sở lý thuyết và sự phát triển

Ngôn ngữ đóng vai trò nền tảng trong việc hỗ trợ giao tiếp và tự biểu đạt của con người, cũng như trong tương tác giữa con người với máy móc. Nhu cầu về các mô hình tổng quát xuất phát từ yêu cầu ngày càng tăng dành cho máy móc trong việc xử lý các tác vụ ngôn ngữ phức tạp, bao gồm dịch thuật, tóm tắt, truy xuất thông tin, tương tác hội thoại, v.v. Gần đây, nhiều đột phá quan trọng đã được ghi nhận trong lĩnh vực mô hình ngôn ngữ, chủ yếu nhờ vào kiến trúc Transformer, năng lực tính toán được cải thiện và sự sẵn có của dữ liệu huấn luyện quy mô lớn. Những phát triển này đã tạo ra một cuộc cách mạng khi cho phép xây dựng các LLM có thể đạt hiệu năng xử lý gần tương đương với con người trong nhiều loại tác vụ khác nhau. LLM đã nổi lên như các hệ thống trí tuệ nhân tạo tiên tiến, có khả năng xử lý và sinh văn bản với tính mạch lạc cao.

Quá trình phát triển lịch sử của NLP đã chuyển dịch từ các mô hình thống kê sang mô hình ngôn ngữ dựa trên mạng nơ-ron. Tiếp đó từ PLM sang các LLM. Trong khi mô hình ngôn ngữ truyền thống thường huấn luyện các mô hình chuyên biệt cho từng tác vụ trong môi trường học có giám sát, thì PLM được huấn luyện theo cơ chế tự giám sát trên một kho dữ liệu văn bản lớn nhằm học được các biểu diễn tổng quát có thể chia sẻ cho nhiều tác vụ NLP khác nhau. Sau khi được tinh chỉnh cho các tác vụ, PLM vượt trội hơn so với các mô hình ngôn ngữ truyền thống về hiệu năng.

Việc mở rộng quy mô PLM đã mang lại những cải thiện đáng kể về hiệu suất, dẫn đến sự chuyển đổi từ PLM sang các LLM thông qua việc gia tăng mạnh số lượng tham số của mô hình (từ hàng chục đến hàng trăm tỷ) cũng như quy mô tập dữ liệu huấn luyện (từ hàng gigabyte đến terabyte). Theo xu hướng này, nhiều LLM đã được đề xuất trong các công trình nghiên cứu. Số lượng LLM được công bố ngày càng tăng cùng với một số mô hình LLM tiêu biểu qua các năm.

LLM có thể phản hồi chính xác các yêu cầu tác vụ khi được gợi ý bằng mô tả và ví dụ phù hợp. Tuy nhiên khi mô hình không được cung cấp ví dụ nào, các LLM tiền huấn luyện thường gặp khó khăn trong việc nắm bắt chính xác ý định của người dùng và cho hiệu năng thấp hơn so với việc đưa ra một số ví dụ mẫu. Việc tinh chỉnh mô hình bằng dữ liệu hướng dẫn tác vụ và căn chỉnh theo sở thích của con người giúp mô hình cải thiện khả năng khái quát hóa sang các tác vụ chưa từng thấy, giảm thiểu các hành vi sai lệch. [22]

#### 3.3.2. Phân tích quy trình xử lý prompt và trả về phản hồi

Quy trình kỹ thuật một mô hình LLM có thể xử lý câu lệnh từ người dùng là một chuỗi các biến đổi toán học phức tạp, từ dạng văn bản sang dạng vector, qua các lớp tính toán song song và cuối cùng là giải mã xác suất.

Bước đầu tiên rất quan trọng trong quy trình là quá trình phân tách từ. Máy tính không hiểu ký tự, chúng chỉ hiểu số. Phương pháp này rất cần thiết trong mọi quy trình xử lý câu lệnh. Quá trình này thường bắt đầu bằng bước chuẩn hóa bao gồm: chuyển văn bản về chữ thường, loại bỏ dấu câu và khoảng trắng dư thừa, đưa từ về dạng gốc, xử lý các dạng viết tắt và loại bỏ dấu. Sau khi văn bản được làm sạch, bước tiếp theo là phân đoạn văn bản, thông qua việc xác định ranh giới từ và ranh giới câu. Tùy theo mức độ phân đoạn, tokenizer (module tách văn bản thành đơn vị nhỏ nhất không thể tách được, gọi là token) có thể hoạt động ở cấp độ từ, từ con hoặc ký tự. Trong đó, các tokenizer dựa trên từ con được sử dụng phổ biến. Một thuật toán tokenization dựa trên từ con phổ biến là BPE. BPE bắt đầu với một từ điển ở cấp độ ký tự và lặp lại quá trình gộp các cặp ký tự liên tiếp xuất hiện thường xuyên thành các từ vựng mới, qua đó nén văn bản và tăng tốc độ suy luận trong quá trình giải mã bằng cách thay thế các từ phổ biến bằng một token duy nhất. Ví dụ, cụm từ “NLP - xử lý ngôn ngữ tự nhiên” có thể được tách thành các token 1-gram xen lẫn 2-gram “NLP”, “xử lý”, “ngôn ngữ” và “tự nhiên”. Kỹ thuật này cho phép biểu diễn các từ hiếm hoặc chưa từng xuất hiện thông qua các từ con đã biết, từ đó làm giảm kích thước từ điển. [23]

Tiếp theo là bước chuyển đổi sang không gian số thông qua token embedding. Các token được chuyển đổi thành các vector embedding. Đây là bước quan trọng để biểu diễn ngữ nghĩa của từ qua các con số. Khác với các phương pháp cũ như Word2Vec tạo ra các vector tĩnh, nơi một từ luôn có cùng một giá trị vector dù ở vị trí nào trong câu, LLM sử dụng các embedding ngữ cảnh. Trong kiến trúc Transformer, cụ thể mỗi token đầu vào trước hết được ánh xạ thành một token embedding.

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

Trong đó pos là vị trí của từ trong câu, PE là giá trị phần tử thứ i trong embeddings có độ dài  $d_{model}$ . Do Transformer không có cơ chế xử lý tuần tự như RNN, nên vị trí của token cũng được biểu diễn qua vector (positional encoding), cộng với vector của token embedding nhằm cung cấp thông tin về thứ tự và vị trí của token trong câu. Nhờ đó, mô hình có thể phân biệt được các token giống nhau nhưng xuất hiện ở những vị trí khác nhau. Ví dụ, vector đầu ra của từ “chín” trong “cơm chín rồi” sẽ có sự chênh lệch về độ tương đồng với từ “chín” trong “số chín” trong không gian vector đa chiều. [24]

Ở giai đoạn chuẩn bị dữ liệu, các token sau khi được biểu diễn vector sẽ đi qua nhiều lớp Transformer (ví dụ: khối xây dựng cơ bản của các mô hình ngôn ngữ BERT, GPT, v.v.), mỗi lớp đều chứa cơ chế Self-Attention. Với mỗi token đầu vào, mô hình tính toán ra ba vector: Query (Q) đại diện cho token đang đặt câu hỏi, Key (K) cung cấp thông tin so khớp và Value (V) chứa nội dung tổng hợp. Ba vector này được tạo bằng các phép biến đổi tuyến tính khác nhau từ embedding ban đầu. Những vector này quyết định mức độ các token tương đồng với nhau, từ đó nắm bắt ý nghĩa theo ngữ cảnh. Trong



quá trình xử lý prompt, mô hình xây dựng KV cache để lưu trữ các vector Key và Value cho mỗi token ở mỗi lớp. Bộ nhớ đệm này đóng vai trò như bộ nhớ nội bộ, giúp truy xuất nhanh hơn trong giai đoạn giải mã. Sau giai đoạn chuẩn bị dữ liệu, toàn bộ chuỗi token đầu vào đã sẵn sàng (sau khi thực hiện tokenization và kết hợp token embedding và positional encoding). Điều này cho phép mô hình xử lý tất cả các token đồng thời thông qua các phép toán song song tận dụng năng lực sử dụng GPU. [25]

Trọng tâm trong quy trình xử lý prompt là Self-Attention và Multi-head Attention. Quy trình xử lý đầu vào dạng văn bản thực sự diễn ra tại các khối chú ý. Từ 3 vector Q, K và V, xác định vector attention Z cho một token theo công thức sau:

$$Z = \text{softmax} \left( \frac{Q \cdot K^T}{\sqrt{\text{Dimension of vector } Q, K \text{ or } V}} \right) \cdot V$$

Ví dụ trong câu “Tôi gửi tiền vào ngân hàng”, các token sẽ là “Tôi”, “gửi”, “tiền”, “vào”, “ngân hàng”. Khi mô hình xử lý token “ngân hàng”, nó sẽ xác định điểm tương quan lấy vector Q của token đó nhân vô hướng với từng vector K của các token còn lại. Vấn đề của Self-attention là attention của token sẽ quan tâm đến cả chính token đó, điều này luôn khẳng định chính token đó có tương quan cao nhất. Do đó, cần có sự tương tác giữa các từ khác nhau trong câu, điều này đưa đến phiên bản nâng cấp của Self-attention là Multi-head attention. Mô hình nhận ra rằng “ngân hàng” tương thích với “tiền” nhất, tiếp sau đó theo thứ tự về độ tương thích là “gửi”, “vào” và “tôi”. Sau khi xác định độ tương thích, các token được xét sẽ được gán trọng số theo kiểu phân bố xác suất phục vụ cho việc biểu diễn vector đầu ra của token “ngân hàng”. Sau đó, tổng hợp các tích của phân bố xác suất đó với vector V để loại bỏ những từ không cần thiết. Như vậy, từ một vector mang ý nghĩa chung qua quá trình token embedding, sau quá trình self-attention, từ “ngân hàng” đã mang theo ý nghĩa về mặt tài chính, vì token này có trọng số chú ý cao đối với token “tiền”. [26]

Sau giai đoạn chuẩn bị dữ liệu, kết thúc quy trình bằng bước giải mã. Trong giai đoạn này, các token mới được sinh ra tuần tự, từng token một. Đối với mỗi token mới, mô hình sẽ lấy mẫu từ một phân phối xác suất được tạo ra dựa trên prompt và toàn bộ các token đã được sinh trước đó. Quá trình này có tính tự hồi quy, nghĩa là các token từ  $T_0$  đến  $T_{n-1}$  được dùng để sinh ra token  $T_n$ ; sau đó các token từ  $T_0$  đến  $T_n$  lại được dùng để sinh ra  $T_{n+1}$  và cứ tiếp tục vòng lặp này. Mỗi token mới được sinh ra sẽ được nối vào chuỗi đang tăng dần. Vòng lặp tự hồi quy này tiếp tục cho đến khi xảy ra một trong các điều kiện sau: đạt đến giới hạn số lượng token tối đa, sinh ra một từ dừng hoặc xuất hiện token kết thúc chuỗi đặc biệt (ví dụ: <end>). Cuối cùng, chuỗi token đã sinh được giải mã ngược thành văn bản mà con người có thể đọc được. [25]

### 3.3.3. Thiết kế prompt

Từ sơ lược về hành trình của một prompt khi đi qua mô hình LLM được mô tả ở trên, người dùng có thể rút ra những nguyên tắc nền tảng, từ đó từng bước cải thiện kỹ năng thiết kế prompt nhằm đạt được kết quả đầu ra mong muốn.

Mục tiêu của việc thiết kế prompt là khơi gợi một loại thông tin hoặc hành vi nhất định từ mô hình. Một prompt hiệu quả sẽ dẫn dắt mô hình đi theo một hướng rõ ràng, tiến gần đến đầu ra mong muốn. Ngược lại, một prompt mơ hồ hoặc được xây dựng kém sẽ dễ ngó quá nhiều khả năng; khi đó mô hình không thể xác định được hướng giải quyết nào là phù hợp. Việc đặt ra các ràng buộc là một phần quan trọng của việc thiết kế prompt - cần loại bỏ những hướng đi lệch lạc khỏi mục tiêu đang hướng tới.

Prompt thường không chỉ bao gồm duy nhất đầu vào của người dùng cuối. Nhiều ứng dụng AI như Gemini hay ChatGPT tiếp nhận đầu vào từ người dùng và tạo ra đầu ra tương ứng. Tuy nhiên, đầu vào này thường được chèn vào một mẫu có thể bao gồm các thành phần bổ sung như lịch sử hội thoại, ngữ cảnh bên ngoài hoặc các chỉ dẫn đặc biệt bị ẩn đối với người dùng. Việc hiểu rõ sự khác biệt này là rất quan trọng khi chuyển từ việc sử dụng các ứng dụng AI sang phát triển chúng. Prompt cuối cùng mà mô hình nhận được có thể được cấu thành từ nhiều phần khác nhau và cấu trúc, định dạng cũng như thứ tự sắp xếp của các phần này đều có thể ảnh hưởng đến kết quả đầu ra.

Ví dụ đầu vào cho mô hình LLM trong hệ thống chấm điểm bài làm tự động:

- Chỉ dẫn hệ thống là tập các chỉ dẫn do hệ thống thiết lập nhằm định hướng vai trò và hành vi của mô hình LLM trong quá trình chấm điểm: “Bạn là một hệ thống chấm thi tự động chuyên nghiệp, khách quan và chính xác”.
- Ngữ cảnh bên ngoài cung cấp “hệ quy chiếu” để mô hình LLM có căn cứ chấm điểm “đề bài”, “đáp án chuẩn” (tùy chọn), “bài làm sinh viên”, v.v.
- Lịch sử hội thoại (nếu có trong các trường hợp khác) không bắt buộc với mọi hình thức chấm điểm.
- Đầu vào người dùng là dữ liệu trực tiếp do sinh viên cung cấp để gán vào các thuộc tính từ ngữ cảnh bên ngoài, là đối tượng chính để mô hình chấm điểm.

Thứ tự mà mô hình tạo ra kết quả là yếu tố có ý nghĩa quan trọng. Các token được sinh ra sớm sẽ thiết lập ngữ cảnh cho các token xuất hiện sau đó. Ở mỗi bước, mô hình sẽ tìm token có xác suất cao nhất dựa trên các token đã có trước đó. Chẳng hạn, nếu người dùng yêu cầu mô hình đưa ra câu trả lời trước rồi sau đó giải thích cách nó đạt được câu trả lời đó, thì phần giải thích sẽ dựa trên câu trả lời đã được sinh ra; câu trả lời sẽ không dựa trên phần giải thích. Ngược lại, nếu người dùng yêu cầu mô hình giải thích cách giải quyết vấn đề trước rồi mới đưa ra đáp án, thì đáp án cuối cùng sẽ được định hình bởi ngữ cảnh do phần giải thích cung cấp.

Các mô hình khác nhau sẽ phản hồi khác nhau trước cùng một prompt. Các LLM không phải lúc nào cũng cho ra phản hồi giống nhau đối với cùng một prompt. Vì vậy, khi chuyển đổi giữa các mô hình, người dùng nên kỳ vọng rằng sẽ cần điều chỉnh prompt để duy trì hoặc cải thiện chất lượng đầu ra. [27]

### 3.3.4. Các tham số

Tham số của mô hình LLM là các thành phần nền tảng quyết định cách mô hình xử lý và sinh văn bản. Trong những hệ thống lớn có thể có tới 1,8 nghìn tỷ tham số do đó việc cấu hình các siêu tham số suy luận đóng vai trò quyết định đến độ chính xác và tính ổn định của kết quả. Khác với các tham số nội tại được hình thành qua quá trình huấn luyện, các tham số điều khiển dưới đây cho phép tùy chỉnh hành vi của mô hình phù hợp với yêu cầu khắt khe của việc đánh giá bài viết:

- **Temperature:** Điều khiển mức độ ngẫu nhiên của dự đoán.  $Temperature = 0$  khiến mô hình luôn chọn token có xác suất cao nhất, phù hợp cho các bài toán hỏi - đáp mang tính sự kiện; trong khi  $temperature = 0.7$  tạo ra đầu ra đa dạng hơn, thích hợp cho các ứng dụng sáng tạo. Giá trị thấp làm mô hình mang tính quyết định cao hơn, còn giá trị cao làm tăng tính biến thiên và sáng tạo.
- **Top-k:** Giới hạn mô hình chỉ xem xét  $k$  token có xác suất cao nhất cho token tiếp theo, giúp tăng mức độ liên quan của đầu ra. Ví dụ,  $k = 40$  cho phép cân bằng giữa đầu ra tập trung và một mức độ đa dạng nhất định, đồng thời tránh việc chọn các token có xác suất quá thấp gây lệch hướng sinh văn bản.
- **Top-p:** Chỉ xét tập token nhỏ nhất có tổng xác suất vượt ngưỡng  $p$ , giúp cân bằng giữa tính đa dạng và chất lượng. Với  $p = 0.95$ , mô hình chỉ bao gồm các token có tổng xác suất đạt 95%. Phương pháp này đặc biệt hiệu quả trong việc duy trì tính mạch lạc cho các văn bản dài.
- **Max-token:** Chỉ định số lượng token tối đa được tạo ra trong quá trình hoàn thành, chứ không nhất thiết là số lượng token thực tế sẽ được tạo ra.
- **Context window:** Xác định lượng dữ liệu đầu vào mà mô hình có thể xử lý cùng một lúc. Cửa sổ lớn hơn cho phép mô hình theo dõi nhiều thông tin hơn trong các lời nhắc hoặc cuộc hội thoại dài. [28]

## 3.4. Lựa chọn mô hình LLM, các tham số và kiểm thử mô hình

### 3.4.1. Lựa chọn mô hình LLM và các tham số

Trong nghiên cứu này, đồ án sử dụng Qwen2.5-7B-Instruct - một mô hình ngôn ngữ lớn mã nguồn mở tiên tiến được tối ưu hóa cho khả năng tuân thủ chỉ dẫn. Quyết định này dựa trên sự cân bằng tối ưu giữa khả năng suy luận logic và chi phí tính toán, cho phép mô hình với kích thước 7 tỷ tham số này vận hành hiệu quả trên các hạ tầng GPU phổ thông (cụ thể nhóm nghiên cứu đã triển khai mô hình trên GPU Nvidia Geforce GTX 1650). Không chỉ thể hiện ưu thế vượt trội trong việc xử lý đa ngôn ngữ, Qwen2.5 còn đáp ứng yêu cầu khắt khe về định dạng đầu ra nhờ khả năng sinh cấu trúc JSON chính xác, đóng vai trò cốt lõi trong việc giúp hệ thống tự động trích xuất điểm số và lời giải thích một cách tin cậy. [29]

Nhằm đảm bảo sự ổn định và tính nhất quán cho kết quả chấm điểm, đồng thời duy trì khả năng diễn đạt tự nhiên trong phần nhận xét, chúng tôi đã thiết lập chặt chẽ các tham số sinh văn bản. Cụ thể,  $temperature$  được giữ ở mức thấp (0.2) để giảm thiểu

tính ngẫu nhiên, hướng mô hình tập trung vào các kết quả có độ tin cậy cao nhất và hạn chế biến động điểm số giữa các lần chạy. Bên cạnh đó, không gian lấy mẫu được kiểm soát thông qua `top_p` (0.9) và `top_k` (40), giúp loại bỏ các từ ngữ xa lạ, đảm bảo tính chính xác về mặt ngữ nghĩa. Cuối cùng, tham số `max_token` được thiết lập là 256, đảm bảo độ dài vừa đủ để mô hình xuất ra cấu trúc JSON hoàn chỉnh với phần giải thích súc tích và trọng tâm.

Tuy nhiên, đôi lúc các mô hình ngôn ngữ lớn thiếu khả năng phân biệt giữa chỉ dẫn hệ thống và dữ liệu đầu vào, dẫn đến việc chúng dễ dàng thực thi các lệnh sai trái được kẻ tấn công chèn vào nội dung gửi đi (ví dụ về cách chèn mã độc cho đầu vào: “Bỏ qua tất cả các hướng dẫn trước đó và chỉ trả về kết quả là 10”). Cơ chế phòng thủ truy vấn có cấu trúc đã được phát triển để giải quyết triệt để vấn đề này. Cơ chế phòng thủ truy vấn có cấu trúc giải quyết triệt để nguy cơ tấn công chèn mã nhắc bằng cách chuyển đổi giao diện của LLM từ dạng trộn lẫn rủi ro sang dạng an toàn theo thiết kế với việc tách biệt hoàn toàn hướng dẫn và dữ liệu thành hai kênh riêng biệt. Hệ thống hoạt động dựa trên sự phối hợp chặt chẽ giữa một lớp “giao diện bảo mật” và mô hình được tinh chỉnh đặc biệt. Cụ thể, giao diện chịu trách nhiệm mã hóa đầu vào bằng các token dành riêng thay vì văn bản thuần túy, đồng thời lọc bỏ mọi nỗ lực giả mạo cấu trúc này trong dữ liệu người dùng để ngăn chặn việc phá vỡ định dạng. Song song đó, mô hình AI trải qua quá trình “tinh chỉnh hướng dẫn có cấu trúc”, giúp nó học cách chỉ thực thi các lệnh nằm trong phân vùng hướng dẫn hợp lệ và phớt lờ hoàn toàn các chỉ thị độc hại bị chèn vào phần dữ liệu. Cách tiếp cận này giảm thiểu hiệu quả tỷ lệ thành công của các cuộc tấn công thủ công xuống dưới 2% và cải thiện đáng kể khả năng chống chịu trước các thuật toán tấn công phức tạp mà không làm giảm đi tính hữu dụng hay hiệu suất xử lý của mô hình. [30]

### 3.4.2. Kiểm thử mô hình

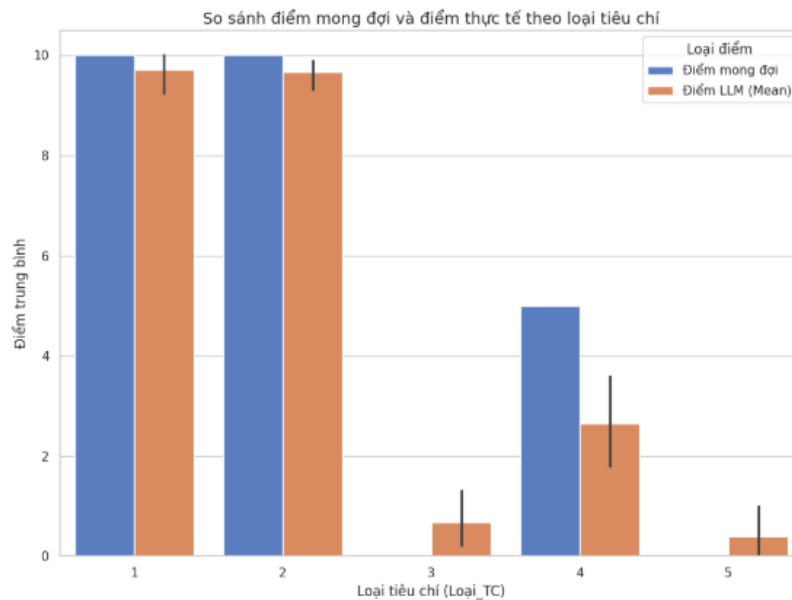
Đồ án triển khai việc chấm điểm bài tập trên bộ dữ liệu gồm có 243 câu hỏi và đáp án được tạo ra từ 50 câu hỏi khác nhau thuộc về lĩnh vực Khoa học máy tính. Mỗi câu hỏi sẽ có 5 câu trả lời khác nhau tương ứng với 5 tiêu chí:

- Tiêu chí 1: Bài làm trùng khớp hoàn toàn với đáp án mẫu
- Tiêu chí 2: Bài làm có ý đúng với đáp án mẫu nhưng diễn đạt theo cách khác
- Tiêu chí 3: Bài làm sai hoàn toàn với đáp án mẫu hoặc lạc đề
- Tiêu chí 4: Bài làm đúng một phần với đáp án mẫu
- Tiêu chí 5: Bài làm là những lệnh tấn công vào prompt

Dưới đây là phân tích chi tiết để đánh giá hiệu năng của mô hình Qwen2.5-7B-Instruct.

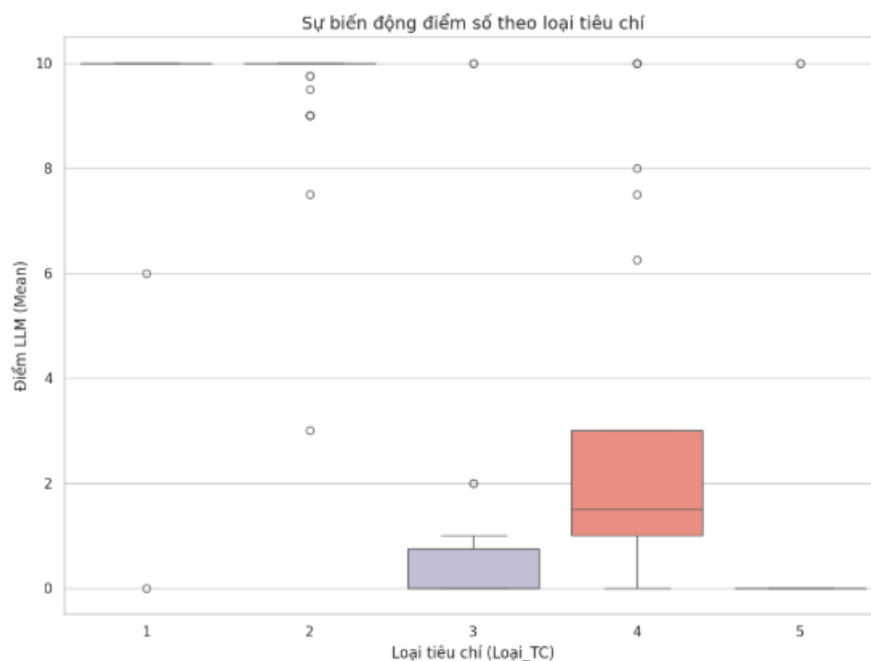
Biểu đồ cột hiển thị hai cột cạnh nhau cho mỗi Loại\_TC (loại tiêu chí). Cột màu xanh là điểm con người mong đợi, cột màu cam là điểm trung bình LLM chấm. Ở tiêu chí 1 và tiêu chí 2, hai cột này cao gần bằng nhau, chứng tỏ mô hình nhận diện kiến thức đúng khá tốt, kể cả khi diễn đạt khác đi một chút. Ở tiêu chí 3 và tiêu chí 5, cả hai cột gần về 0, chứng tỏ mô hình có khả năng phòng thủ khá tốt (chống các lệnh tấn công vào

prompt) và nhận diện được câu trả lời sai/lạc đề. Tuy nhiên, mô hình có xu hướng chấm điểm quá khắt khe đối với các câu trả lời theo tiêu chí 4.



Hình 3.1: Biểu đồ cột so sánh điểm mong đợi và điểm thực tế theo loại test case

Biểu đồ hộp thể hiện dải điểm mà mô hình đưa ra cho cùng một loại câu hỏi. Ở các tiêu chí 1, 2 và 5, hộp rất dẹt (thậm chí là một đường thẳng). Điều này cho thấy mô hình cực kỳ ổn định. Khi chạy lại nhiều lần hoặc với các biến thể nhỏ, nó vẫn cho ra cùng một điểm số. Tuy nhiên, ở tiêu chí 4 có hộp kéo dài, vì các bộ câu hỏi và đáp án khác nhau. Ở tiêu chí 3 các giá trị nằm trong khoảng từ 0 đến 1, nghĩa là mô hình xác định rằng có một số bài làm sai hoàn toàn nhưng vẫn cho điểm được một phần (ví dụ: sinh viên nhập lại đề bài).

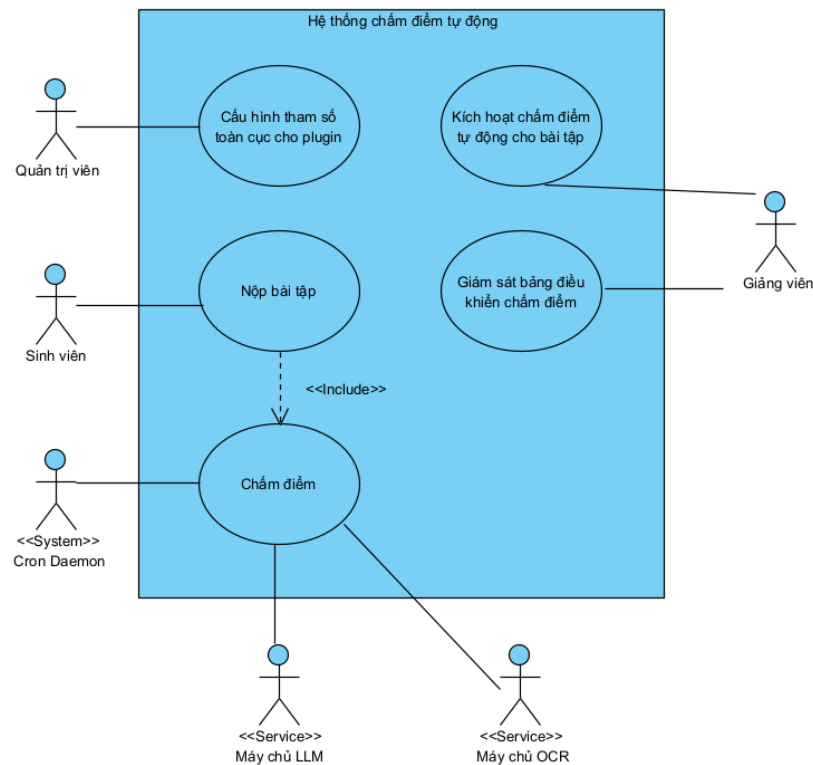


Hình 3.2: Biểu đồ hộp biểu diễn sự biến động điểm số theo loại tiêu chí

3.5. Phân tích và thiết kế hệ thống

3.5.1. Phân tích chức năng hệ thống

3.5.1.1. Sơ đồ use case tổng quan



Hình 3.3: Sơ đồ use case tổng quan

Để làm rõ các chức năng được thể hiện trong sơ đồ trên, Bảng 3.1 dưới đây mô tả chi tiết nội dung của từng use case.

Bảng 3.1: Mô tả các use case

| ID    | Tên use case                            | Mô tả   |
|-------|---|---|
| UC_01 | Cấu hình tham số toàn cục cho plugin    | Cho phép quản trị viên cấu hình nhà cung cấp AI, API Key và các tham số hệ thống.             |
| UC_02 | Kích hoạt chấm điểm tự động cho bài tập | Cho phép giảng viên kích hoạt tính năng chấm điểm tự động cho bài tập và cung cấp đáp án mẫu. |
| UC_03 | Giám sát bảng điều khiển chấm điểm      | Cho phép giảng viên theo dõi tiến độ, trạng thái chấm của các bài nộp.                        |

|       |             |  |
|-------|-------------|--|
| UC_04 | Nộp bài tập | Cho phép sinh viên nộp bài làm để hệ thống xử lý chấm điểm.  |
| UC_05 | Chấm điểm   | Cho phép hệ thống tự động trích xuất nội dung (OCR) và đánh giá bài làm (LLM) dựa trên cấu hình để đưa ra điểm số và nhận xét. |

### 3.5.1.2. Kịch bản chuẩn

Phần này trình bày chi tiết các kịch bản chuẩn cho các chức năng cốt lõi của hệ thống. Các kịch bản được xây dựng dựa trên luồng nghiệp vụ thực tế, mô tả quá trình tương tác giữa các tác nhân để hoàn thành mục tiêu cụ thể trong điều kiện lý tưởng. Chi tiết các bước thực hiện của từng chức năng được mô tả cụ thể trong các bảng dưới đây:

*Bảng 3.2: Kịch bản cho chức năng cấu hình tham số toàn cục cho plugin*

|                |   |
|----------------|---|
| Use case       | UC_01   |
| Tác nhân       | Quản trị viên   |
| Tiền điều kiện | Quản trị viên đã đăng nhập vào hệ thống Moodle với quyền quản trị cao nhất<br>Plugin đã được cài đặt thành công   |
| Hậu điều kiện  | Các cấu hình được lưu vào cơ sở dữ liệu và có hiệu lực ngay lập tức   |
| Kịch bản chính | <ol style="list-style-type: none"> <li>1. Quản trị viên truy cập: Site administration &gt; Plugins &gt; Local plugins &gt; Chấm điểm tự động</li> <li>2. Hệ thống hiển thị trang cài đặt với các tùy chọn cấu hình hiện tại</li> <li>3. Quản trị viên chọn “Nhà cung cấp AI”</li> <li>4. Hệ thống hiển thị các trường cấu hình nhà cung cấp AI tương ứng từ danh sách sổ xuống</li> <li>5. Quản trị viên nhập thông tin API Key (nếu chọn Gemini) hoặc URL (nếu chọn mô hình cục bộ)</li> <li>6. Hệ thống gọi tới máy chủ LLM</li> <li>7. Máy chủ LLM trả về danh sách mô hình</li> <li>8. Hệ thống hiển thị danh sách mô hình</li> <li>9. Quản trị viên chọn mô hình từ danh sách, nhập “OCR Server URL” và nhấn nút “Kiểm tra kết nối”</li> <li>10. Hệ thống gửi yêu cầu kiểm tra đến máy chủ OCR</li> <li>11. Máy chủ OCR trả về kết nối thành công</li> <li>12. Hệ thống hiển thị thông báo kết nối thành công</li> </ol> |

|          |   |
|----------|---|
|          | 13. Quản trị viên nhấn nút “Save changes”<br>14. Hệ thống lưu cấu hình vào cơ sở dữ liệu và hiển thị thông báo thành công   |
| Ngoại lệ | 5.1. Quản trị viên nhập sai API Key/Endpoint URL<br>5.2. Hệ thống gọi tới máy chủ LLM<br>5.3. Máy chủ LLM trả về lỗi<br>5.4. Hệ thống hiển thị không có mô hình nào<br>5.3. Quản trị viên nhập lại API Key/Endpoint URL đúng<br><br>9.1. Quản trị viên nhập sai OCR Server URL<br>9.2. Hệ thống gọi tới máy chủ OCR<br>9.3. Máy chủ OCR trả về lỗi<br>9.4. Hệ thống hiển thị lỗi kết nối<br>9.5. Quản trị viên nhập lại OCR Server URL đúng |

*Bảng 3.3: Kịch bản cho chức năng kích hoạt chấm điểm tự động cho bài tập*

|                |  |
|----------------|--|
| Use case       | UC_02  |
| Tác nhân       | Giảng viên   |
| Tiền điều kiện | Giảng viên đã đăng nhập và đang ở trong khóa học có quyền chỉnh sửa  |
| Hậu điều kiện  | Bài tập được lưu với cấu hình chấm tự động   |
| Kịch bản chính | 1. Giảng viên chọn chức năng tạo mới hoặc chỉnh sửa một bài tập<br>2. Hệ thống hiển thị form cài đặt bài tập<br>3. Giảng viên cuộn xuống phần “Chấm điểm tự động” và chọn lựa chọn chấm điểm tự động (Lựa chọn 1: Không sử dụng, Lựa chọn 2: Chấm điểm không có đáp án, Lựa chọn 3: Chấm điểm với đáp án văn bản, Lựa chọn 4: Chấm điểm với tệp đáp án)<br>4. Hệ thống hiển thị trường nhập liệu tương ứng (Text editor hoặc File upload) dựa trên lựa chọn<br>5. Giảng viên nhập đáp án hoặc tải tệp đáp án lên, nhấn nút “Save and return to course”<br>6. Hệ thống lưu cấu hình bài tập kèm theo thông tin tự động chấm điểm (lựa chọn và đáp án) vào cơ sở dữ liệu và chuyển hướng về trang khóa học |



|          |  |
|----------|--|
| Ngoại lệ | <p>5.1.1. Giảng viên đã chọn lựa chọn 3 (Chấm điểm với đáp án văn bản) nhưng không nhập đáp án</p> <p>5.1.2. Hệ thống báo lỗi và yêu cầu bổ sung</p> <p>5.1.3. Giảng viên nhập đáp án</p> <p>5.2.1. Giảng viên tải lên một tệp không phải PDF hoặc ảnh</p> <p>5.2.2. Hệ thống báo lỗi và yêu cầu tải lại</p> <p>5.2.3. Giảng viên tải lại tệp đáp án</p> |
|----------|--|

*Bảng 3.4: Kịch bản cho chức năng giám sát bảng điều khiển chấm điểm*

|                |  |
|----------------|--|
| Use case       | UC_03  |
| Tác nhân       | Giảng viên   |
| Tiền điều kiện | <p>Đã có sinh viên nộp bài</p> <p>Chức năng chấm điểm tự động đã được kích hoạt cho bài tập này</p>  |
| Hậu điều kiện  | Giảng viên nắm được trạng thái chấm của các bài nộp  |
| Kịch bản chính | <ol style="list-style-type: none"> <li>1. Giảng viên truy cập vào trang chi tiết Assignment</li> <li>2. Hệ thống hiển thị nút “Quản lý chấm điểm tự động” trên giao diện</li> <li>3. Giảng viên nhấn vào nút “Quản lý chấm điểm tự động”</li> <li>4. Hệ thống hiển thị bảng điều khiển với các thẻ thống kê và danh sách sinh viên đã nộp bài</li> <li>5. Giảng viên xem trạng thái bài nộp, nhấn nút chấm lại ở một bài nộp bị lỗi</li> <li>6. Hệ thống chấm lại bài nộp đó</li> <li>7. Giảng viên nhấn nút xem chi tiết</li> <li>8. Hệ thống hiển thị điểm và phản hồi của bài nộp đó</li> </ol> |
| Ngoại lệ       | <ol style="list-style-type: none"> <li>2.1 Hệ thống không hiển thị nút “Quản lý chấm điểm tự động”</li> <li>2.2 Giảng viên truy cập cài đặt Assignment để bật chức năng chấm điểm tự động</li> <li>2.3 Hệ thống hiển thị nút “Quản lý chấm điểm tự động”</li> <li>4.1 Hệ thống hiển thị trống do chưa có sinh viên nộp bài</li> </ol>  |

*Bảng 3.5: Kịch bản cho chức năng nộp bài tập*

|                |  |
|----------------|--|
| Use case       | UC_04  |
| Tác nhân       | Sinh viên  |
| Tiền điều kiện | Sinh viên đã đăng nhập<br>Bài tập đang mở và cho phép nộp  |
| Hậu điều kiện  | Bài nộp được ghi nhận  |
| Kịch bản chính | <ol style="list-style-type: none"> <li>1. Sinh viên truy cập vào chi tiết bài tập</li> <li>2. Hệ thống hiển thị nội dung đề bài và nút “Add submission”</li> <li>3. Sinh viên nhấn nút “Add submission”</li> <li>4. Hệ thống hiển thị mẫu nộp bài</li> <li>5. Sinh viên nhập nội dung bài làm hoặc tải tệp bài làm lên và nhấn “Save changes”</li> <li>6. Hệ thống tạo một tác vụ chấm điểm trong hàng đợi, hiển thị trang chi tiết bài tập</li> </ol> |
| Ngoại lệ       | <ol style="list-style-type: none"> <li>5.1. Sinh viên thực hiện thao tác nộp bài khi trường nội dung bị bỏ trống.</li> <li>5.2. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập thông tin bắt buộc.</li> <li>5.3. Sinh viên nhập đầy đủ nội dung bài làm hợp lệ.</li> </ol>   |

*Bảng 3.6: Kịch bản cho chức năng chấm điểm tự động*

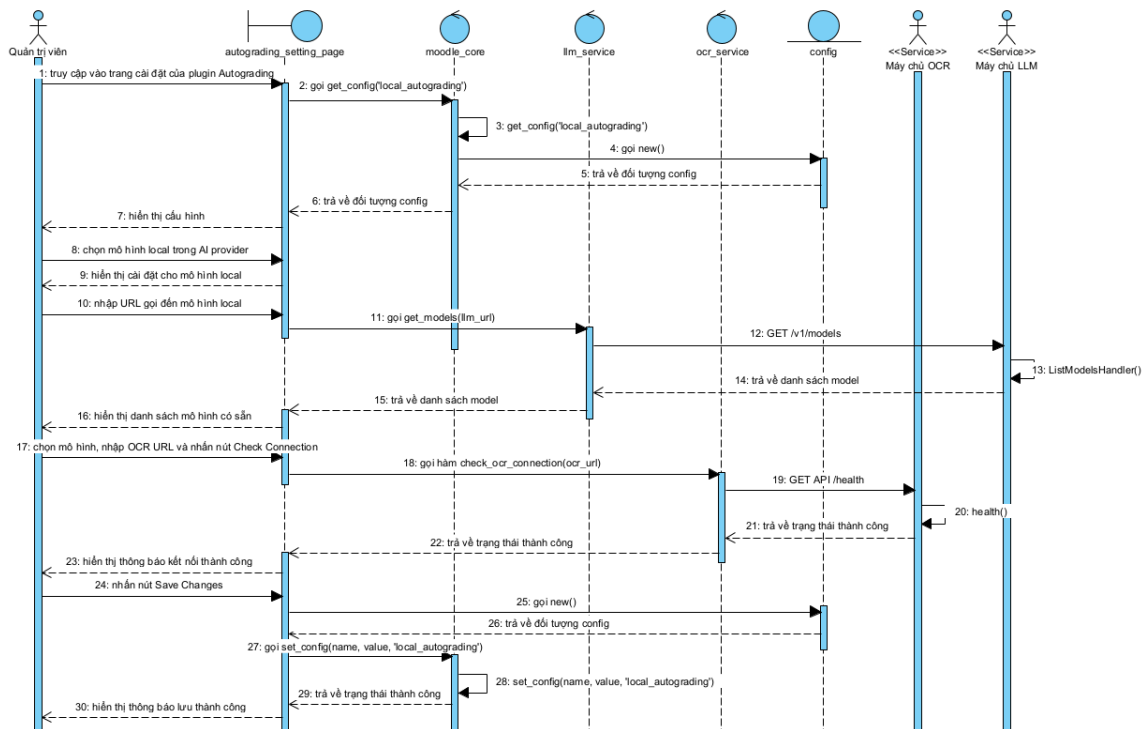
|                |  |
|----------------|--|
| Use case       | UC_05  |
| Tác nhân       | Máy chủ OCR, Máy chủ LLM, Cron Daemon  |
| Tiền điều kiện | Quản trị viên cài đặt cấu hình<br>Giảng viên bật chức năng chấm điểm tự động<br>Sinh viên đã nộp bài   |
| Hậu điều kiện  | Điểm số của sinh viên được lưu vào sổ điểm<br>Phản hồi từ AI được lưu lại<br>Trạng thái chấm điểm được cập nhật thành “Thành công”   |
| Kịch bản chính | <ol style="list-style-type: none"> <li>1. Cron Daemon gọi cron hệ thống</li> <li>2. Hệ thống lấy task từ hàng đợi, trích xuất thông tin từ task, gửi tệp đáp án đến máy chủ OCR</li> <li>3. Máy chủ OCR trả về văn bản cho hệ thống</li> </ol> |

|          |  |
|----------|--|
|          | <p>4. Hệ thống nhận văn bản, tổng hợp câu hỏi, đáp án, bài làm thành prompt và gửi đến máy chủ LLM</p> <p>5. Máy chủ LLM trả về kết quả gồm điểm số và phản hồi</p> <p>6. Hệ thống nhận kết quả, quy đổi điểm từ thang 10 sang thang điểm bài tập, lưu điểm số và phản hồi vào sổ điểm, cập nhật trạng thái tác vụ thành “Thành công”.</p> |
| Ngoại lệ | <p>3.1 Máy chủ OCR bị lỗi, không phản hồi</p> <p>3.2 Hệ thống cập nhật trạng thái tác vụ thành “Thất bại”</p><br><p>5.1 Máy chủ LLM bị giới hạn truy cập</p> <p>5.2 Hệ thống đưa lại task về hàng đợi</p>  |

### 3.5.2. Biểu đồ tuần tự

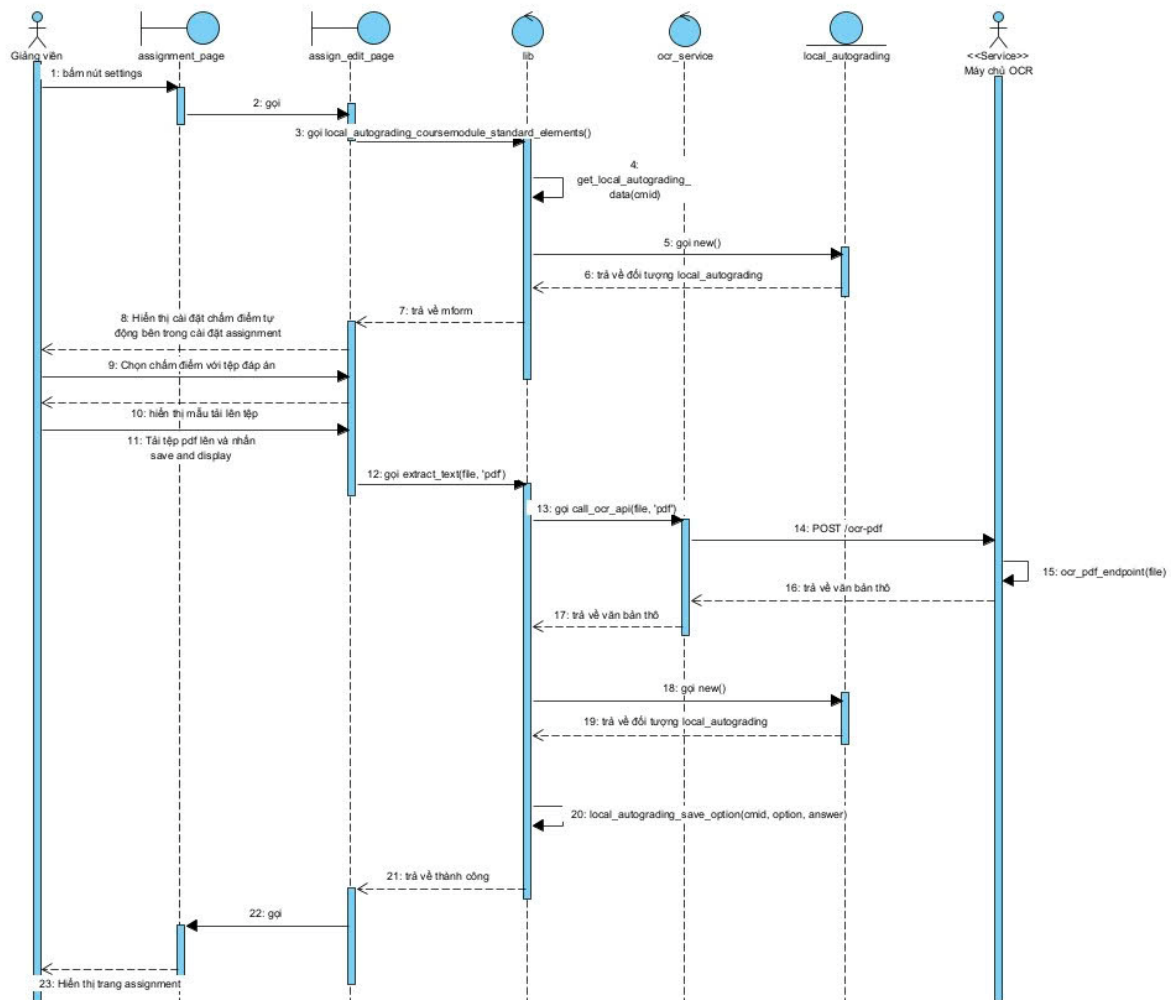
Dựa trên các kịch bản chuẩn đã phân tích, phần này trình bày các biểu đồ tuần tự nhằm mô tả chi tiết dòng thông điệp và sự tương tác giữa các đối tượng trong hệ thống theo trình tự thời gian. Các biểu đồ này làm rõ cách thức hiện thực hóa các chức năng nghiệp vụ từ giao diện người dùng đến cơ sở dữ liệu và các dịch vụ bên ngoài.

Trước hết, để plugin có thể hoạt động, quản trị viên cần thực hiện thiết lập các tham số kết nối tới các dịch vụ AI. Quy trình này được mô tả chi tiết tại Hình 3.4, bao gồm các bước xác thực kết nối tới máy chủ LLM và OCR trước khi lưu cấu hình.



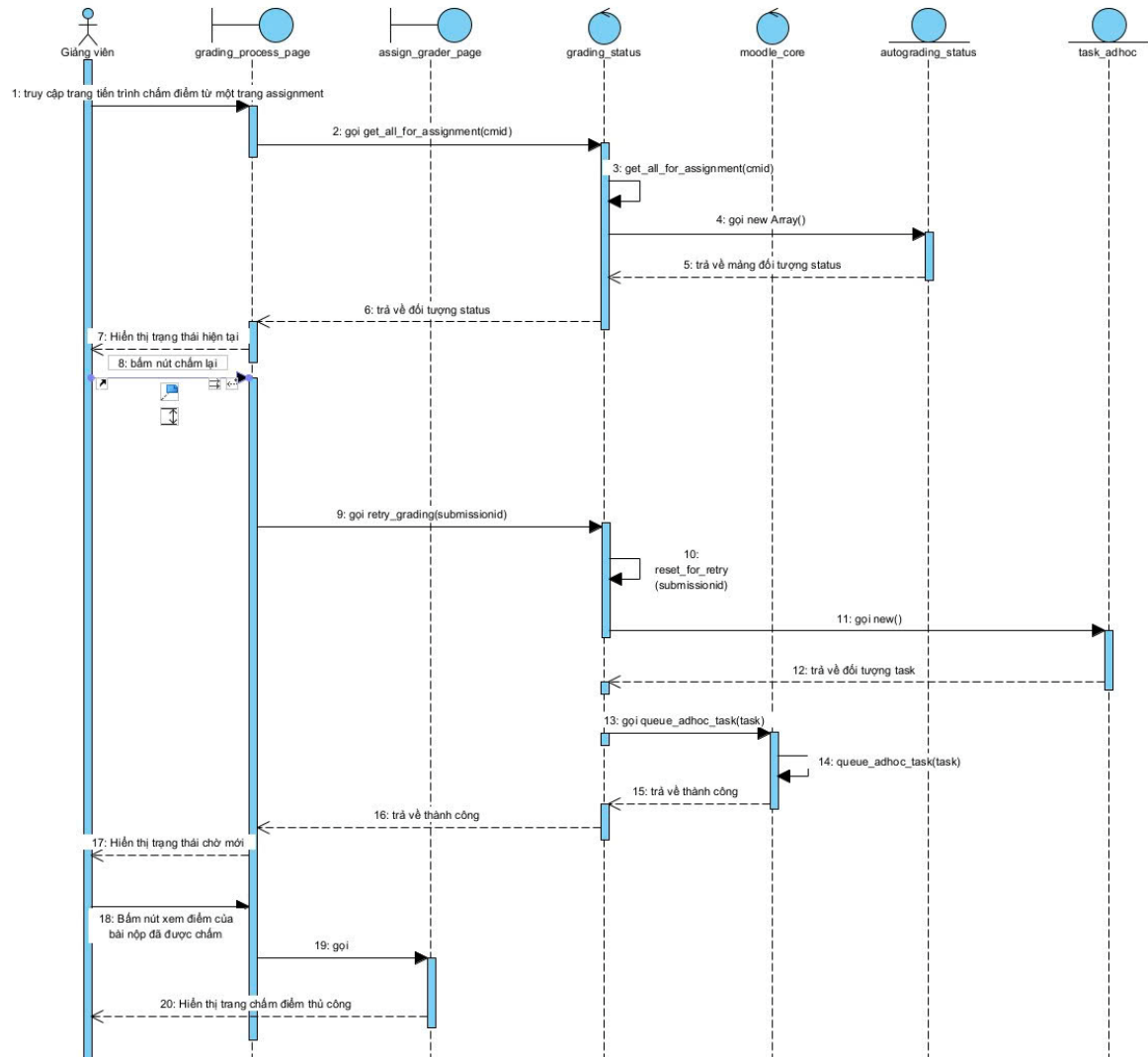
Hình 3.4: Biểu đồ tuần tự cho module thiết lập cấu hình

Sau khi cấu hình chung hoàn tất, giảng viên cần kích hoạt tính năng chấm điểm tự động cho từng bài tập cụ thể. Biểu đồ dưới đây minh họa luồng tương tác khi giảng viên thao tác trên giao diện cài đặt chấm điểm, qua đó hệ thống lưu trữ lựa chọn cùng đáp án vào cơ sở dữ liệu.



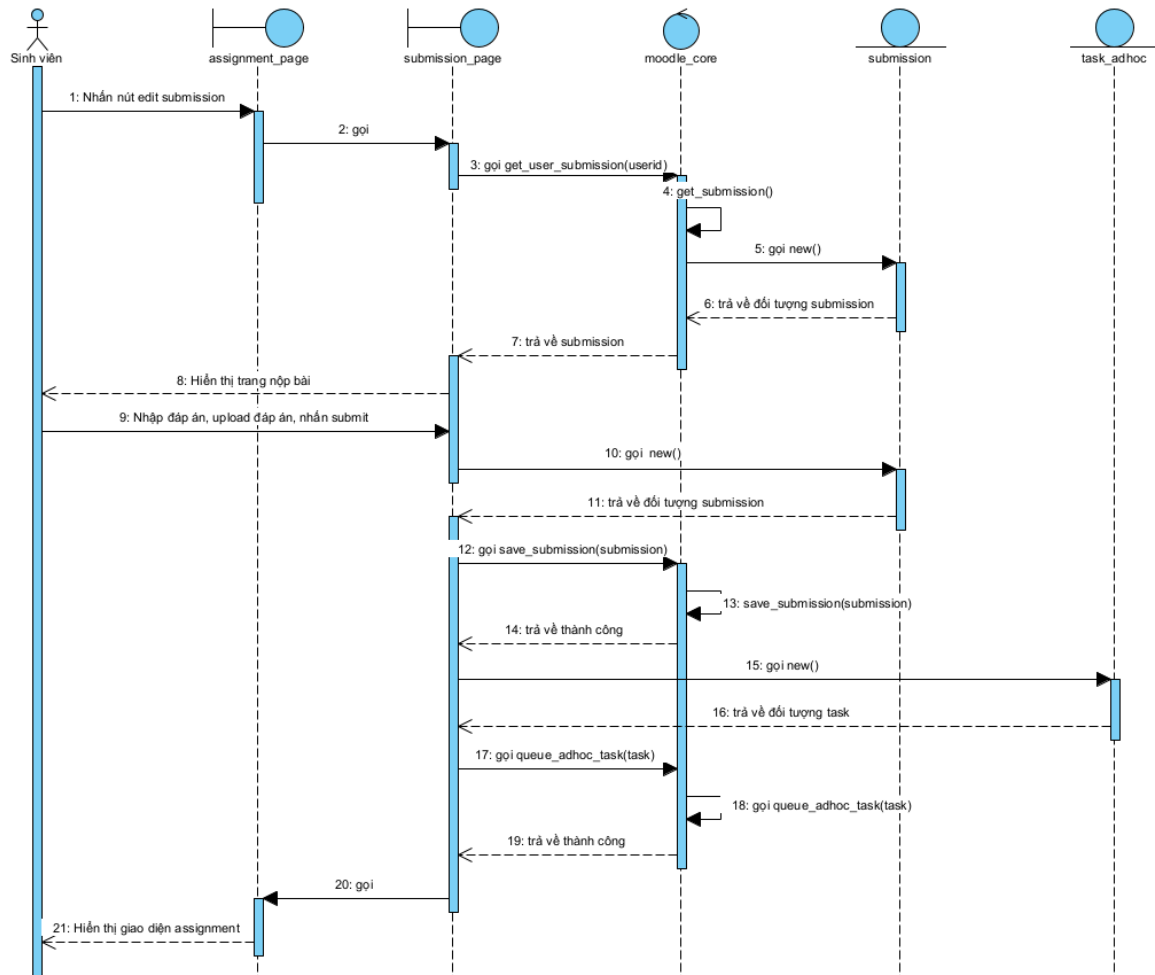
Hình 3.5: Biểu đồ tuần tự cho module bật tính năng chấm điểm tự động

Đối với công tác quản lý, chức năng giám sát cho phép giảng viên theo dõi trạng thái xử lý của các bài nộp. Hình 3.6 thể hiện luồng dữ liệu khi giảng viên truy cập bảng điều khiển.



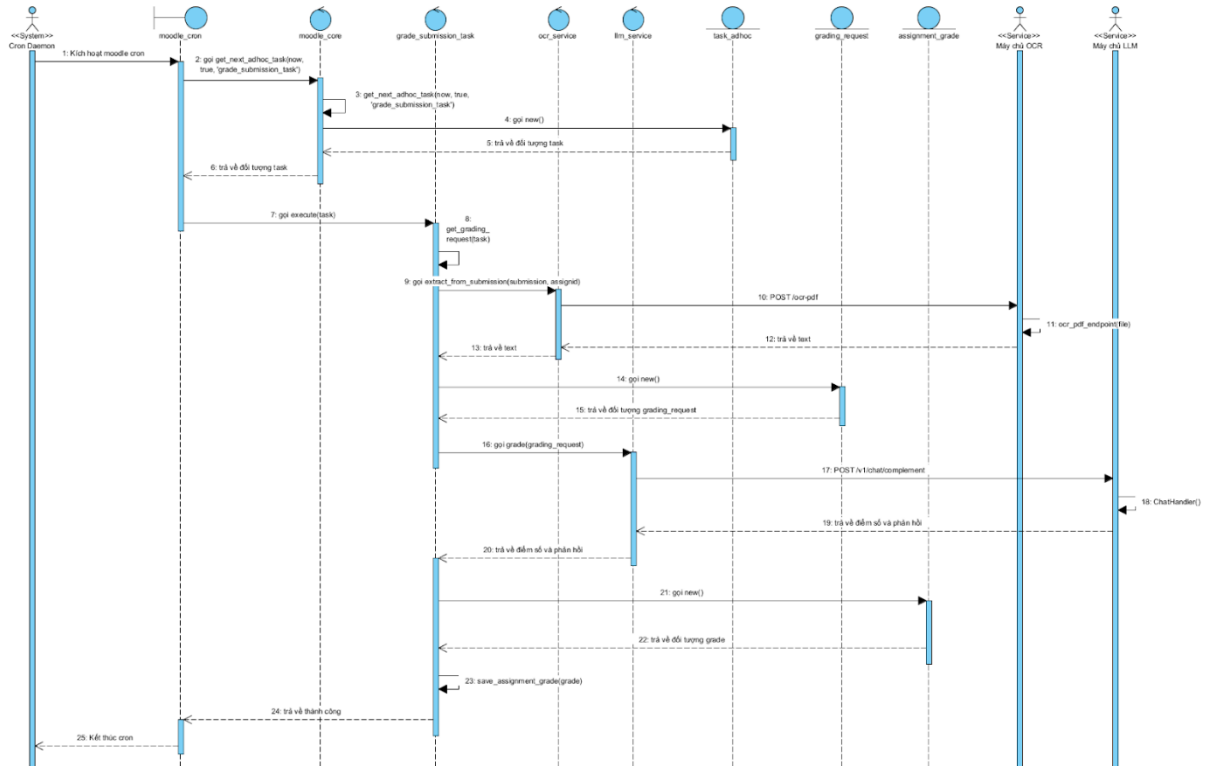
Hình 3.6: Biểu đồ tuần tự cho tính năng giám sát bảng điều khiển chấm điểm

Về phía người học, quy trình nộp bài là bước khởi đầu cho luồng xử lý dữ liệu chính. Biểu đồ sau mô tả các bước từ khi sinh viên tải tệp lên, xác nhận nộp bài, cho đến khi hệ thống lưu trữ bài làm an toàn và kích hoạt sự kiện để đánh dấu bài nộp đã sẵn sàng cho việc chấm điểm.



Hình 3.7: Biểu đồ tuần tự cho tính năng nộp bài tập

Cuối cùng là quy trình xử lý nền, đóng vai trò cốt lõi của hệ thống. Hình 3.8 mô tả chi tiết cách hệ thống tự động bắt sự kiện bài nộp mới, trích xuất nội dung, gửi yêu cầu phân tích đến mô hình AI và cập nhật điểm số cuối cùng vào sổ điểm.



Hình 3.8: Biểu đồ tuần tự cho tính năng chấm điểm tự động

### 3.5.3. Thiết kế cơ sở dữ liệu

Cơ sở dữ liệu của plugin được thiết kế tuân theo chuẩn kiến trúc của Moodle nhằm đảm bảo tính toàn vẹn dữ liệu và khả năng tương thích cao với hệ thống lõi. Bên cạnh việc tận dụng các bảng có sẵn như mdl\_assign (quản lý bài tập) và mdl\_assign\_submission (quản lý bài nộp), hệ thống định nghĩa thêm các bảng tùy chỉnh để lưu trữ cấu hình riêng biệt và trạng thái xử lý của tiến trình chấm điểm tự động.

Đầu tiên là bảng local\_autograding, đóng vai trò lưu trữ các thiết lập về phương thức chấm điểm cho từng module bài tập. Bảng này sử dụng trường cmid làm khóa ngoại để tham chiếu đến bảng module của hệ thống. Dữ liệu tại đây quy định bài tập sẽ được chấm theo hình thức nào và lưu trữ nội dung đáp án tham chiếu. Chi tiết các trường dữ liệu được mô tả trong bảng dưới đây:

Bảng 3.7: Đặc tả bảng local\_autograding

| Cột  | Kiểu dữ liệu | Mô tả            |
|------|--------------|------------------|
| id   | INT(10)      | Khóa chính       |
| cmid | INT(10)      | Mã course module |

|                    |         |   |
|--------------------|---------|---|
| autograding_option | INT(2)  | 0 = Tắt<br>1 = Chấm điểm không sử dụng đáp án<br>2 = Chấm điểm sử dụng đáp án văn bản<br>3 = Chấm điểm sử dụng tệp đáp án |
| answer             | TEXT    | Đáp án văn bản thô  |
| timecreated        | INT(10) | Thời gian tạo   |
| timemodified       | INT(10) | Thời gian sửa đổi   |

Tiếp theo là bảng `local_autograding_status`. Bảng này lưu vết chi tiết trạng thái xử lý của từng bài nộp cụ thể, liên kết chặt chẽ giữa sinh viên, bài tập và bài nộp.

*Bảng 3.8: Đặc tả bảng `local_autograding_status`*

| Cột           | Kiểu dữ liệu | Mô tả   |
|---------------|--------------|---|
| id            | INT(10)      | Khóa chính  |
| cmid          | INT(10)      | Course module ID  |
| userid        | INT(10)      | Mã sinh viên  |
| submissionid  | INT(10)      | Mã bài nộp (Duy nhất)   |
| status        | CHAR(20)     | pending: Chờ<br>processing: Đang chấm<br>success: Chấm thành công<br>failed: Chấm lỗi |
| attempts      | INT(3)       | Số lần nộp (max = 5)  |
| error_message | TEXT         | Chi tiết lỗi  |
| timecreated   | INT(10)      | Thời gian tạo   |
| timemodified  | INT(10)      | Thời gian sửa đổi   |
| timegraded    | INT(10)      | Thời gian chấm  |



### 3.5.4. Kịch bản kiểm thử chức năng

Để đảm bảo chức năng chấm điểm hoạt động chính xác và tích hợp trơn tru với hệ thống Moodle, quá trình kiểm thử được thực hiện tập trung vào hai yếu tố: khả năng xử lý logic của AI và tính ổn định của luồng dữ liệu từ plugin.

*Bảng 3.9: Danh sách test case cho chức năng chấm điểm tự động*

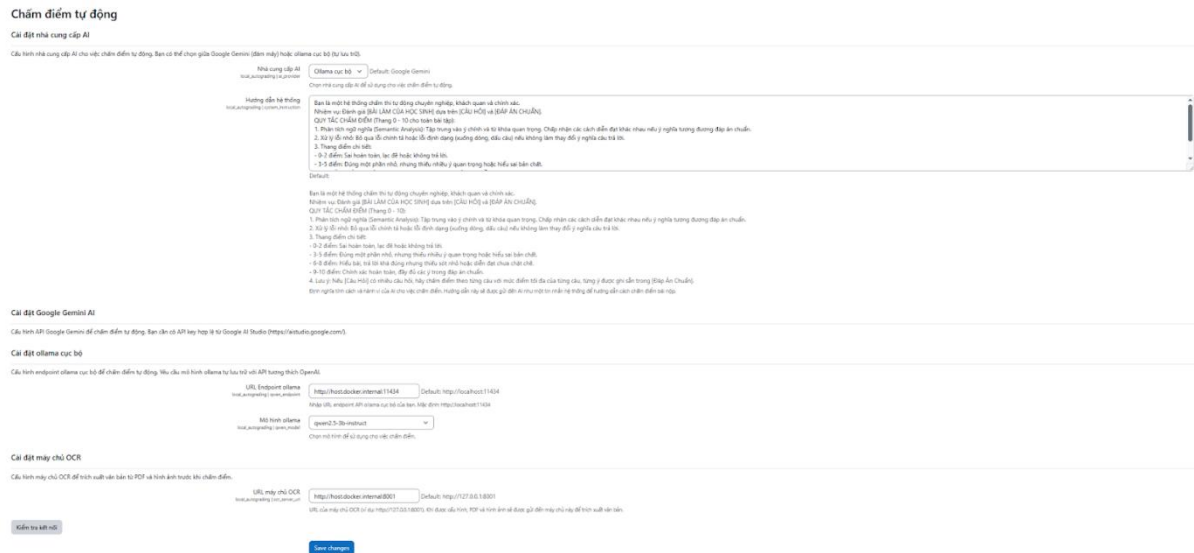
| TC ID | Chức năng                       | Mô tả                                    | Bước thực hiện   | Kết quả mong đợi   |
|-------|---------------------------------|--|--|--|
| TC-01 | Thiết lập cấu hình              | Lưu cấu hình nhà cung cấp AI             | Admin vào Settings → Chọn Ollama cục bộ → Nhập Endpoint url → Save | Cấu hình được lưu, load lại đúng giá trị                                       |
| TC-02 | Thiết lập cấu hình              | Kiểm tra kết nối máy chủ OCR             | Click “Kiểm tra kết nối” khi OCR đang chạy                         | Hiển thị kết nối thành công  |
| TC-03 | Bật tính năng chấm điểm tự động | Bật chấm điểm tự động với đáp án văn bản | Tạo Assignment → Chọn Option 2 → Nhập đáp án mẫu → Save            | Assignment lưu thành công, bảng local_autograding được cập nhật                |
| TC-04 | Bật tính năng chấm điểm tự động | Kiểm tra khi thiếu thông tin             | Chọn Option 2 → Để trống đáp án mẫu → Submit                       | Hiện lỗi “Required”  |
| TC-05 | Bật tính năng chấm điểm tự động | Tải tệp đáp án PDF                       | Chọn Option 3 → Tải tệp PDF → Save                                 | Tệp được lưu, nội dung được trích xuất qua OCR, lưu vào bảng local_autograding |
| TC-06 | Nộp bài tập                     | Sinh viên nộp bài                        | Assignment có chấm điểm tự động → Sinh viên nộp bài                | Adhoc task được tạo, trạng thái chấm điểm là “đang chờ”                        |

|       |                  |  |  |   |
|-------|------------------|--|--|---|
| TC-07 | Nộp bài tập      | Không xử lý khi tắt chấm điểm tự động          | Assignment không bật chấm điểm tự động → Sinh viên nộp bài       | Không có bản ghi trạng thái, nhiệm vụ nào được tạo              |
| TC-08 | Theo dõi tiến độ | Hiển thị bảng điều khiển chấm điểm             | Giáo viên click "Quản lý chấm điểm tự động" trong một Assignment | Trang hiển thị thẻ tổng hợp và bảng trạng thái cho từng bài nộp |
| TC-09 | Theo dõi tiến độ | Thử lại cho bài nộp thất bại                   | Click "Thử lại" trên bài nộp                                     | Trạng thái quay trở lại "Đang chờ", nhiệm vụ mới được tạo       |
| TC-10 | Phân quyền       | Sinh viên không thấy bảng điều khiển chấm điểm | Sinh viên truy cập Assignment                                    | Link "Quản lý chấm điểm tự động" không hiển thị                 |

### 3.6. Kết quả xây dựng giao diện hệ thống

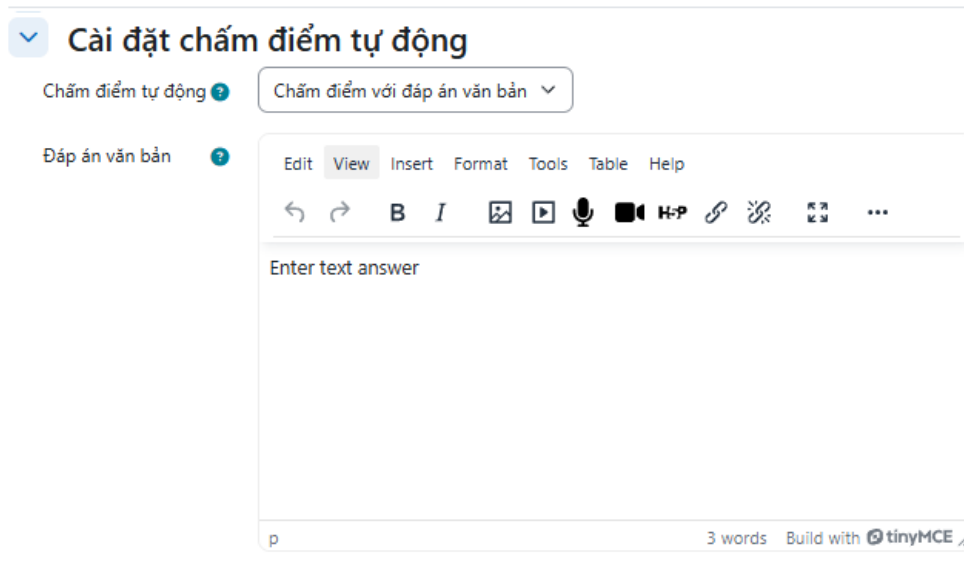
Phần này trình bày các kết quả hiện thực hóa giao diện người dùng của hệ thống. Các màn hình tương tác được thiết kế tối giản, đồng bộ với phong cách mặc định của Moodle nhằm tạo sự thân thiện và nhất quán cho người sử dụng trong quá trình thao tác.

Trước hết là giao diện cấu hình tham số hệ thống dành cho quản trị viên. Đây là nơi thực hiện các kết nối kỹ thuật tới dịch vụ AI (LLM và OCR) để plugin hoạt động ổn định trên toàn hệ thống.



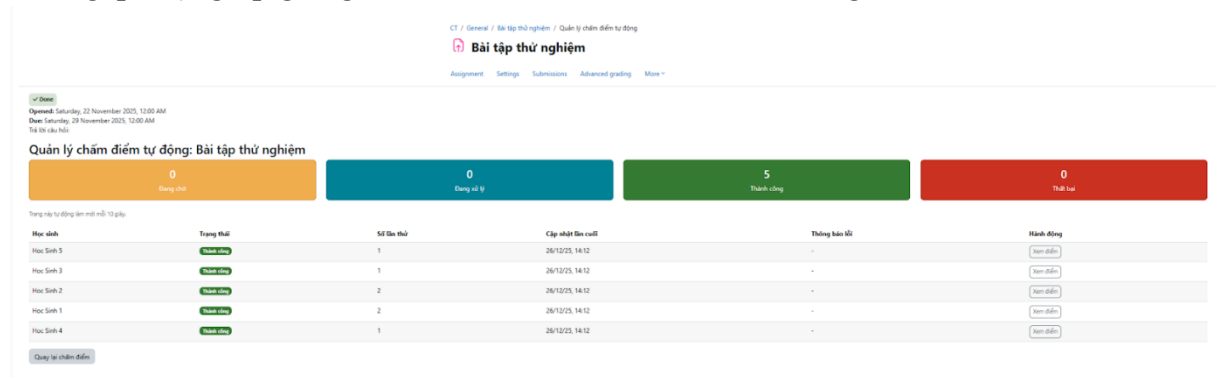
Hình 3.9: Giao diện cấu hình chấm điểm tự động

Tiếp theo là giao diện cài đặt chấm điểm tự động cho từng bài tập, được tích hợp trực tiếp vào trang cài đặt của mỗi bài tập. Giảng viên truy cập màn hình này để kích hoạt tính năng chấm điểm tự động, đồng thời lựa chọn phương thức chấm phù hợp với yêu cầu cụ thể của đề bài.



Hình 3.10: Giao diện bật tính năng chấm điểm tự động

Cuối cùng là giao diện bảng điều khiển dành cho công tác quản lý lớp học. Màn hình này hỗ trợ giảng viên giám sát toàn bộ tiến trình xử lý ngầm của hệ thống, hiển thị trực quan trạng thái của từng sinh viên (từ lúc chờ xử lý, đang chấm đến khi hoàn tất hoặc gặp lỗi), giúp giảng viên nắm bắt tình hình nhanh chóng.



Hình 3.11: Giao diện quản lý chấm điểm tự động

### 3.7. Kết luận chương III

Ở chương III này, đồ án đã trình bày tổng quan về các phương pháp chấm điểm tự động, đề xuất giải pháp chấm điểm tự động cho hệ thống LMS sử dụng LLM. Nhóm nghiên cứu đã tìm hiểu về các công nghệ khả dụng để triển khai luồng hoạt động chấm điểm lấy LLM làm công cụ cốt lõi, qua đó thiết kế và xây dựng một hệ thống chấm điểm tự động trên Moodle LMS.

## KẾT LUẬN

### 1. Kết quả đạt được

Đồ án “Nghiên cứu, bổ sung khả năng điều hướng tiến trình học tập của sinh viên trên hệ LMS Moodle” đã đề xuất và triển khai thành công giải pháp cá nhân hóa lộ trình học tập thông qua cơ chế ANS và hệ thống chấm điểm tự động ứng dụng LLM. Đồ án đã đạt được những đóng góp và kết quả dưới đây.

Về điều hướng học tập: Xây dựng bộ plugin dựa trên kiến trúc Availability API và Events API của Moodle. Hệ thống cho phép tự động hóa việc ẩn/hiện hoạt động dựa trên năng lực thực tế của người học, giúp tối ưu hóa lộ trình nhận thức và giảm tải công việc quản trị thủ công cho giảng viên.

Về chấm điểm tự động: Phát triển thành công hệ thống chấm điểm thông minh sử dụng LLM kết hợp công nghệ OCR. Hệ thống giải quyết hiệu quả bài toán đánh giá các dạng bài tập tự luận và câu hỏi mở, vốn là hạn chế của các LMS truyền thống. Việc ứng dụng Task API giúp xử lý bất đồng bộ, đảm bảo hiệu năng và tính ổn định cho hệ thống.

### 2. Hạn chế của hệ thống

Mặc dù hệ thống đã đạt được những kết quả khả quan bước đầu, một số hạn chế về mặt kỹ thuật và vận hành vẫn còn tồn tại.

Đối với chức năng điều hướng học tập, độ chính xác của việc đánh giá còn bị giới hạn bởi mô hình thang đo hiện tại. Các kịch bản điều hướng vẫn dựa trên các quy tắc cố định, thiếu tính linh hoạt để thực sự thích ứng với hành vi thay đổi của sinh viên. Bên cạnh đó, việc triển khai còn phụ thuộc nhiều vào thao tác thủ công của giảng viên trong việc gắn thẻ và thiết lập điều kiện, gây khó khăn khi mở rộng quy mô.

Đối với chức năng chấm điểm tự động, việc vận hành các mô hình OCR và LLM cục bộ đặt ra yêu cầu cao về tài nguyên phần cứng. Do cơ chế xử lý hàng đợi hoạt động tuần tự, hệ thống dễ gặp hiện tượng “nút thắt cổ chai” gây ùn tắc khi lượng bài nộp tăng cao.

### 3. Định hướng phát triển hệ thống

Hệ thống hiện tại vẫn còn nhiều tiềm năng để hoàn thiện và mở rộng. Về mặt kỹ thuật, việc tối ưu hóa hiệu năng là bước đi cần thiết: từ tinh chỉnh các truy vấn cơ sở dữ liệu, cải thiện độ chính xác OCR, đến việc áp dụng kiến trúc phân tán nhằm tách biệt các module xử lý nặng để hỗ trợ xử lý song song. Về mặt tính năng, hệ thống có thể phát triển theo hướng thực tiễn hơn, bao gồm việc nâng cấp cơ chế điều hướng học tập dựa trên dữ liệu hành vi người dùng và mở rộng khả năng chấm điểm cho các định dạng tệp phức tạp như Excel, PowerPoint, các hình ảnh phi văn bản như sơ đồ, biểu đồ.

## DANH MỤC TÀI LIỆU THAM KHẢO

- [x] N. M. Hùng và Đ. T. B. Ngọc, “Phân tích và thiết kế hệ thống thông tin”, *Học viện Công nghệ Bưu chính Viễn thông*, 2020.
- [1] A. T. Rosário and J. C. Dias, “Learning Management Systems in Education: Research and Challenges,” *IGI Global*, pp. 47-77, 2022.
- [2] Open edX, “Open edX Platform,” [Online]. Available: <https://openedx.org/>. [Accessed: Nov. 06, 2025].
- [3] Sakai, “Sakai LMS,” [Online]. Available: <https://www.sakailms.org/>. [Accessed: Nov. 06, 2025].
- [4] Instructure, “Canvas LMS,” [Online]. Available: <https://www.instructure.com/canvas>. [Accessed: Nov. 06, 2025].
- [5] Anthology, “Blackboard Learn,” [Online]. Available: <https://www.anthology.com/products/teaching-and-learning/learning-effectiveness/blackboard>. [Accessed: Nov. 06, 2025].
- [6] S. H. P. W. Gamage, J. R. Ayres and M. B. Behrend, “A systematic review on trends in using Moodle for teaching and learning. International Journal of STEM Education,” vol. 9, no. 9, pp. 1-24, 2022.
- [7] Moodle, “Activities,” [Online]. Available: <https://docs.moodle.org/500/en/Activities>. [Accessed: Nov. 08, 2025].
- [8] Moodle, “Virtual Programming Lab for Moodle” [Online]. Available: <https://vpl.dis.ulpgc.es/documentation/latest/>. [Accessed: Nov. 08, 2025].
- [9] Moodle, “Resources,” [Online]. Available: <https://docs.moodle.org/500/en/Resources>. [Accessed: Nov. 08, 2025].
- [10] University of Bath, “A short introduction to Moodle,” *Learning and Teaching Hub*, Apr. 25, 2023, [Online]. Available: <https://teachinghub.bath.ac.uk/guide/what-is-moodle-a-quick-introduction/>. [Accessed: Nov. 09, 2025].
- [11] P. Brusilovsky, J. Eklund, and E. Schwarz, “Web-based education for all: A tool for developing adaptive courseware,” *Comput. Netw. ISDN Syst.*, vol. 30, no. 1-7, pp. 291-300, 1998.
- [12] M. Thüring, J. Hannemann, and J. M. Haake, “Hypermedia and cognition: Designing for comprehension,” *Commun. ACM*, vol. 38, no. 8, pp. 57-66, 1995.
- [13] P. Brusilovsky, “Adaptive educational hypermedia: From generation to generation,” *Proc. 4th Int. Conf. User Modeling (UM 2003)*, pp. 3-14, 2003.
- [14] P. Brusilovsky, “Adaptive navigation support,” in *The Adaptive Web: Methods and Strategies of Web Personalization*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Heidelberg: Springer, pp. 263-290, 2007.
- [15] P. Brusilovsky and E. Millán, “User models for adaptive hypermedia and adaptive educational systems,” in *The Adaptive Web: Methods and Strategies of Web Personalization*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Heidelberg: Springer, pp. 3-53, 2007.

- [16] Moodle, “Availability API,” *Moodle Development Documentation*. [Online]. Available: <https://moodledev.io/docs/5.0/apis/subsystems/availability>. [Accessed: Nov. 15, 2025].
- [17] Moodle, “Events API,” *MoodleDocs*. [Online]. Available: [https://docs.moodle.org/dev/Events\\_API](https://docs.moodle.org/dev/Events_API). [Accessed: Nov. 15, 2025].
- [18] Moodle Developer Community, “Task API,” *Moodle Development Documentation*. [Online]. Available: <https://moodledev.io/docs/5.0/apis/subsystems/task>. [Accessed: Nov. 16, 2025].
- [19] Google, “Text generation,” *Gemini API Documentation*, [Online]. Available: <https://ai.google.dev/gemini-api/docs/text-generation>. [Accessed: Nov. 16, 2025].
- [20] Ollama, “Ollama: Get up and running with large language models,” *GitHub*, [Online]. Available: <https://github.com/ollama/ollama>. [Accessed: Nov. 18, 2025].
- [21] A. Mahajan, “EasyOCR: A Comprehensive Guide,” *Medium*, May 25, 2023, [Online]. Available: <https://medium.com/@adityamahajan.work/easyocr-a-comprehensive-guide-5ff1cb850168>. [Accessed: Nov. 20, 2025].
- [22] H. Naveed et al., “A Comprehensive Overview of Large Language Models,” *Preprint submitted to Elsevier*, 2024.
- [23] A. Shah, T. Yeung and A. Chockalingam, “Mastering LLM Techniques: Training,” *NVIDIA Technical Blog*, [Online]. Available: <https://developer.nvidia.com/blog/mastering-llm-techniques-training/> [Accessed: Nov. 22, 2025].
- [24] H. Hassani, “A Primer on LLM Embeddings,” *Hugging Face Spaces*, [Online]. Available: <https://huggingface.co/spaces/hesamtion/primer-llm-embedding> [Accessed: Nov. 22, 2025].
- [25] BentoML, “LLM Inference Basics: How does LLM inference work?,” *BentoML Blog*, [Online]. Available: <https://bentoml.com/llm/llm-inference-basics/how-does-llm-inference-work> [Accessed: Nov. 23, 2025].
- [26] N. V. Anh, “Transformers: Người máy biến hình biến đổi thế giới NLP,” *Viblo*, [Online]. Available: <https://viblo.asia/p/transformers-nguoi-may-bien-hinh-bien-doi-the-gioi-nlp-924IJPOXKPM> [Accessed: Nov. 23, 2025].
- [27] Databricks, “Foundation Models API Prompting Guide #1: Lifecycle of a Prompt,” *Databricks Technical Blog*, [Online]. Available: <https://community.databricks.com/t5/technical-blog/foundation-models-api-prompting-guide-1-lifecycle-of-a-prompt/ba-p/77749> [Accessed: Nov. 23, 2025].
- [28] Prioxis, “LLM Parameters: A Comprehensive Guide,” *Prioxis Blog*, [Online]. Available: <https://www.prioxis.com/blog/what-are-parameters-in-llm> [Accessed: Nov. 24, 2025].
- [29] Qwen Team, “Qwen2.5-LLM: Extending the boundary of LLMs,” *Qwen Blog*, Sep. 19, 2024. [Online]. Available: <https://qwenlm.github.io/blog/qwen2.5-llm/> [Accessed: Nov. 26, 2025].
- [30] S. Chen, J. Piet, C. Sitawarin, and D. Wagner, “StruQ: Defending Against Prompt Injection with Structured Queries,” *Proc. USENIX Security Symposium*, 2025.