**TRƯỜNG ĐẠI HỌC PHENIKAA**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO BÀI TẬP LỚN**

**MÔN KỸ THUẬT PHẦN MỀM**

ĐỀ TÀI

HỆ THỐNG QUẢN LÝ ĐIỂM DANH HỌC SINH

|  |  |  |
| --- | --- | --- |
| Giáo viên hướng dẫn: | Vũ Quang Dũng |  |
| Sinh viên thực hiện: | Phùng Tiến Đạt | 23010410 |
|  | Nguyễn Đỗ Phi Hùng |  |

*Hà Nội, 14/06/2025*

**MỤC LỤC**

[BẢNG PHÂN CHIA CÔNG VIỆC CHI TIẾT THEO TỪNG TUẦN 1](#_Toc200884864)

[CHƯƠNG 1: GIỚI THIỆU CHUNG VÀ PHÂN TÍCH SƠ BỘ 1](#_Toc200884865)

[1.1. Đặt vấn đề bài toán 1](#_Toc200884866)

[1.2. Đối tượng và nhóm phục vụ của hệ thống 2](#_Toc200884867)

[1.3. Phân loại các nhóm 2](#_Toc200884868)

[CHƯƠNG 2: PHÂN TÍCH YÊU CẦU 4](#_Toc200884869)

[2.1. Các chức năng chính (phân rã chức năng) 4](#_Toc200884870)

[CHƯƠNG 3: THIẾT KẾ CHƯƠNG TRÌNH 8](#_Toc200884871)

[3.1. Giải thích luồng hoạt động 8](#_Toc200884872)

[3.2. Thiết kế cơ sở dữ liệu 11](#_Toc200884873)

[3.3. Thiết kế hướng đối tượng (UML Class Diagram) 14](#_Toc200884874)

[3.4. Thiết kế luồng dữ liệu (data flow diagram) 15](#_Toc200884875)

[3.4.1. Sơ đồ ngữ cảnh (context diagram – level 0) 15](#_Toc200884876)

[3.4.2. Sơ đồ luồng dữ liệu mức 1 (DFD lever 1) 16](#_Toc200884877)

[3.5. Thiết kế chức năng chi tiết (UML Activity Diagram) 17](#_Toc200884878)

[CHƯƠNG 4: GIAO DIỆN 19](#_Toc200884879)

[CHƯƠNG 5. CÁC CÔNG NGHỆ ĐƯỢC SỬ DỤNG VÀ ĐÁNH GIÁ HỆ THỐNG 21](#_Toc200884880)

[5.1. Ngôn ngữ và Framework 21](#_Toc200884881)

[5.2. Cơ sở dữ liệu 22](#_Toc200884882)

[5.3. Kiểm thử và xác minh 22](#_Toc200884883)

[5.4. Môi trường phát triển 22](#_Toc200884884)

[5.5. Đánh giá hệ thống 22](#_Toc200884885)

# BẢNG PHÂN CHIA CÔNG VIỆC CHI TIẾT THEO TỪNG TUẦN

|  |  |  |  |
| --- | --- | --- | --- |
| **Tuần** | **Nội dung chính** | **Đạt (Code chính)** | **Hùng (Hỗ trợ tài liệu)** |
| 1 | Xác định đề tài & phân tích yêu cầu | Thiết lập Repo Github, khởi tạo dự án Laravel | Ghi chú đề tài, mô tả sơ bộ yêu cầu người dùng |
| 2 | Phân tích hệ thống & sơ đồ UML | Thiết lập Database schema, viết file route, phân quyền và xử lý logic roles (Admin/Teacher) | Vẽ sơ đồ Use Case đơn giản, mô tả chức năng |
| 3 | Thiết kế giao diện & UI flow | Tạo view: đăng nhập, dashboard, trang điểm danh (Livewire component) | Vẽ mockup UI bằng draw.io, hỗ trợ mô tả các màn hình |
| 4 | Điểm danh & lọc dữ liệu | Code chức năng điểm danh, lọc theo tháng/lớp, tạo modal nhập lý do bằng Livewire | Phác hoạ sơ đồ trình tự (Sequence Diagram), mô tả quy trình điểm danh |
| 5 | Viết API mô phỏng & kiểm thử | Tạo route export API, viết Unit test và Feature test cho chức năng điểm danh và export | Viết mô tả test case cơ bản, checklist đánh giá |
| 6 | Tổng hợp tài liệu & chuẩn bị báo cáo | Hoàn thiện tính năng, sửa bug, thêm chức năng phụ (search, pagination, báo lỗi UI) | Liệt kê ưu điểm – hạn chế dự án |
| 7 | Trình bày và demo hệ thống | Chuẩn bị phần demo toàn bộ hệ thống (điểm danh + báo cáo), soạn báo cáo |  |

# CHƯƠNG 1: GIỚI THIỆU CHUNG VÀ PHÂN TÍCH SƠ BỘ

* 1. **Đặt vấn đề bài toán**

Trong bối cảnh giáo dục hiện đại, việc quản lý thông tin học sinh, đặc biệt là theo dõi tình hình chuyên cần, đóng vai trò vô cùng quan trọng trong việc đánh giá sự tiến bộ và hành vi học tập của học sinh. Tuy nhiên, tại nhiều cơ sở giáo dục hiện nay, quy trình điểm danh học sinh vẫn còn tồn tại nhiều hạn chế và bất cập, chủ yếu dựa vào các phương pháp thủ công truyền thống hoặc sử dụng những công cụ đơn giản chưa được tối ưu hóa.

Các khó khăn khi sử dụng phương pháp điểm danh truyền thống bao gồm:

* Tốn thời gian và công sức của giáo viên khi phải điểm danh thủ công mỗi ngày.
* Dễ xảy ra sai sót, nhầm lẫn khi nhập liệu thủ công.
* Khó khăn trong việc tổng hợp và truy xuất dữ liệu khi cần thiết.
* Thiếu khả năng cung cấp các báo cáo chuyên sâu và nhanh chóng.
* Không hỗ trợ quản lý thông tin chuyên cần một cách hiệu quả.

Để giải quyết những thách thức này, việc xây dựng một hệ thống quản lý điểm danh học sinh tự động, hiệu quả và đáng tin cậy là một yêu cầu cấp thiết. Hệ thống được đề xuất không chỉ giúp tối ưu hóa quy trình điểm danh mà còn cung cấp các công cụ quản lý dữ liệu hiệu quả cùng khả năng tạo báo cáo chuyên sâu. Điều này góp phần nâng cao chất lượng quản lý giáo dục, đồng thời hỗ trợ tốt hơn cho quá trình dạy và học của học sinh và giáo viên.

* 1. **Đối tượng và nhóm phục vụ của hệ thống**

**Đối tượng phục vụ:**

Hệ thống “Quản lý điểm danh (student-attendance-tracking-system)” được thiết kế nhằm phục vụ các cơ sở giáo dục như:

* Trường học (tiểu học, trung học, đại học…)
* Trung tâm đào tạo
* Các tổ chức giáo duc cần theo dõi chuyên cần

**Mục đích và vai trò của hệ thống:**

Đây là một nền tảng web giúp quản lý điểm danh và thông tin học sinh một cách hiệu quả, tập trung vào việc tự động hoá quy trình điểm danh, thay thế cách làm thủ công truyền thống.

**Các nhóm người dùng chính:**

* **Giáo viên:**
  + Thực hiện điểm danh hang ngày
  + Quản lý thông tin lớp học, học sinh
  + Xem báo cáo chuyên cần chi tiết
* **Quản trị viên (Admin):**
  + Quản lý tài khoản người dùng, lớp học, học sinh
  + Theo dõi và xuất báo cáo tổng quan cho toàn trường

**Nhóm khách hàng mục tiêu:**

* Các trường học, trung tâm giáo dục có nhu cầu số hoá quy trình điểm danh
* Ban giám hiệu muốn theo dõi tình hình chuyên cần để đánh giá học sinh và giáo viên

**Nhóm sản phẩm:**

* Phần mềm quản lý giáo dục
* Hệ thống theo dõi chuyên cần
* Ứng dụng hỗ trợ số hoá nghiệp vụ hành chính giáo dục
  1. **Phân loại các nhóm**

Trong hệ thống này, chúng ta xác định hai tác nhân chính sẽ tương tác trực tiếp với ứng dụng:

* Admin (Người quản trị hệ thống)
* Teacher (Giáo viên)

Mỗi tác nhân đại diện cho một vai trò người dùng cụ thể với các quyền hạn và trách nghiệm riêng biệt trong hệ thống.

**Đặc điểm người dùng:**

* **Admin**:
  + **Mô tả:** Là người quản trị và vận hành tổng thể hệ thống. Có thể là cán bộ công nghệ thông tin hoặc nhân viên hành chính được giao trách nhiệm quản lý dữ liệu cấp cao
  + **Kinh nghiệm:** Thường có kiến thức tốt về quản lý hệ thống phần mềm, có khả năng xử lý các tác vụ phức tạp và hiểu biết cấu trúc dữ liệu.
  + **Mục tiêu sử dụng:** Đảm bảo hệ thống hoạt động ổn định, quản lý người dùng, cấu hình lớp học và học sinh ở quy mô toàn trường, đồng thời theo dõi các báo cáo và thống kê tổng quát.
  + **Yêu cầu giao diện:** Yêu cầu giao diện quản trị mạnh mẽ, cung cấp đầy đủ chức năng nhưng vẫn đảm bảo tính trực quan để dễ dàng thao tác với lượng lớn dữ liệu và tuỳ trọn cấu hình.
* **Teacher (Giáo viên):**
  + **Mô tả:** Là người trực tiếp giảng dạy và theo dõi tình hình chuyên cần của học sinh trong các lớp được phân công.
  + **Kinh nghiệm:** Có kỹ năng sử dụng máy tính và trình duyệt web ở mực cơ bản. Tập trung vào các tác vụ liên quan đến lớp học và học sinh của mình.
  + **Mục tiêu sử dụng:** Thực hiện điểm danh nhanh chóng, chính xác, cập nhật thông tin học sinh và lớp học trong phạm vi được giao, cũng như xem các báo cáo chuyên cần của các lớp mình phụ trách.
  + **Yêu cầu giao diện:** Yêu cầu giao diện thân thiện, dễ sử dụng, tập trung vào các chức năng cốt lõi liên quan đến điểm danh và quản lý học sinh/lớp học. Các thao tác phải đơn giản và trực quan để không làm mất nhiều thời gian của giáo viên.

**Vai trò và quyền hạn của từng nhóm quản trị/quản lý:**

Hệ thống áp dụng cơ chế phân quyền chặt chẽ để đảm bảo mỗi người dùng chỉ có thể truy cập và thực hiện các tác vụ phù hợp với vai trò của họ.

* **Admin:**
  + **Quản lý người dùng:** Thêm, sửa, xoá tài khoản người dùng (đặc biệt là tài khoản giáo viên), gán thay đổi vai trò người dùng.
  + **Quản lý lớp học:** Thực hiện đầy đủ các thao tác CRUD (Tạo, Đọc, Cập nhật, Xoá) thông tin lớp học trên toàn hệ thống.
  + **Quản lý học sinh:** Thực hiện đầy đủ các thao tác CRUD thông tin học sinh trên toàn bộ hệ thống, bao gồm gán học sinh vào lớp.
  + **Báo cáo và thống kê:** Truy cập và xem tất cả các báo cáo, biểu đồ thống kê về tình hình điểm danh của toàn trường và từng lớp.
* **Teacher (Giáo viên):**
  + **Quản lý học sinh:** Có quyền CRUD thông tin học sinh, nhưng chỉ giới hạn trong các lớp học mà họ phân công phụ trách.
  + **Quản lý lớp học:** Có quyền xem danh sách các lớp học, và có thể chỉnh sửa các thông tin cơ bản của lớp mình phụ trách (nếu được phép bới Admin).
  + **Điểm danh:** Là tác nhân chính thực hiện vệc điểm danh học sinh hàng ngày cho các lớp được phân công. Có quyền đánh dấu trạng thái điểm danh và nhập lý do.
  + **Báo cáo và thống kê:** Có quyền truy cập và xem các báo cáo, biểu đồ thống kê điểm danh nhưng chỉ giới hạn cho các lớp học sinh và họ được phân công phụ trách.

# CHƯƠNG 2: PHÂN TÍCH YÊU CẦU

* 1. **Các chức năng chính (phân rã chức năng)**

Các chức năng của hệ thống được phân rã thành các nhóm chính, nhằm đảm bảo quản lý toàn diện quá trình điểm danh và thông tin liên quan. Mỗi nhóm chức năng phục vụ một mục tiêu cụ thể và được thiết kế để hỗ trợ các tác nhân thực hiện công việc của họ một cách hiệu quả.

*Hình 2.1. Sơ đồ use case*

**FR1: Quản lý xác thực và tài khoản (User roles & Authentication)**

* **FR1.1: Đăng ký và Đăng nhập người dùng**
* **Mô tả:** Hệ thống cho phép người dùng đăng ký tài khoản mới và đăng nhập vào hệ thống
* **Actor:** Người dùng mới, Admin, Teacher.
* **Luồng chính:**
* **Đăng ký:** Người dùng cung cấp Email, mật khẩu và các thông tin cần thiết khác (ví dụ: Tên). Hệ thống kiểm tra tích hợp của dữ liệu, hash mật khẩu và lưu tài khoản. (Admin có thể tạo tài khoản cho teacher).
* **Đăng nhập:** Người dùng nhập Email và mật khẩu. Hệ thống xác thực thông tin đăng nhập.
* **Nếu xác thực** thành công, hệ thống sẽ đièu hướng người dùng đến giao diện Dashboard phù hợp với vai trò của họ.
* **Dữ liệu đầu vào:** Email, Password.
* **Dữ liệu đầu ra:** Thông báo đăng nhập thành công hoặc thất bại.
* **Luồng sự kiện (ví dụ Đăng nhập):**
* Người dùng vào trang đăng nhập.
* Người dùng nhập mật khẩu email, password và submit.
* Hệ thống kiểm tra thông tin, redirect nếu hợp lệ, thông báo lỗi nếu sai.
* **Trường hợp ngoại lệ:**
  + - **FR1.1.1:** Sai thông tin đăng nhập: Hệ thống thông báo lỗi.
    - **FR1.1.2:** Không nhập đủ thông tin: Hệ thống yêu cầu nhập đầy đủ thông tin.
* **FR1.2: Phân quyền Admin và Teacher**
* **Mô tả:** Hệ thống phân biệt và áp dụng quyền hạn khác nhau cho Admin và Teacher.
* **Actor:** Hệ thống.
* **Quy tắc nghiệp vụ:**
  + - Admin có toàn quyền truy cập và quản lý tất cả các chức năng và dữ liệu của hệ thống.
    - Teacher chỉ có quyền truy cập vào các chức năng liên quan đến điểm danh và quản lý học sinh/lớp học mà họ được phân công.

**FR2: Quản lý học sinh và lớp học**

* **FR2.1: CRUD thông tin học sinh**
* **Mô tả:** Cho phép tạo, đọc, cập nhật và xoá thông tin của học sinh.
* **Actor:** Admin, Teacher (trong phạm vi lớp được phân công).
* **Đầu vào:** Tên học sinh, tuổi, lớp học.
* **Quy tắc nghiệp vụ:** Mã học sinh phải là duy nhất.
* **FR2.2: CRUD thông tinh lớp học**
* **Mô tả:** Cho phép tạo, đọc, cập nhật và xoá thông tin của các lớp học.
* **Actor:** Admin.
* **Đầu vào:** Tên lớp học.
* **Quy tắc nghiệp vụ:** Tên lớp học phải là duy nhất.
* **FR2.3: Gán học sinh vào lớp học**
* **Mô tả:** Cho phép liên kết một học sinh với một lớp học cụ thể.
* **Actor:** Admin, Teacher (trong phạm vi lớp được phân công).
* **Quy tắc nghiệp vụ:** Một hoc sinh chỉ thuộc một lớp tại một thời điểm. Việc gán học sinh vào một lớp sẽ cập nhật thông tin lớp hiện tại của học sinh đó.

**FR3: Điểm danh học sinh**

* **FR3.1: Điểm danh hang ngày**
* **Mô tả:** Giáo viên có thể điểm danh học sinh của một lớp vào mỗi ngày.
* **Actor:** Teacher.
* **Input:** Lớp học, ngày điểm danh, trạng thái điểm danh cho từng học sinh.
* **Output:** Bản ghi điểm danh được lưu vào cơ sở dữ liệu.
* **Các trạng thái điểm danh:** Có mặt (present), vắng mặt (absent), ốm (sick), khác (other).
* **FR3.2: Nhập lý do cho trạng thái “other”**
* **Mô tả:** Khi giáo viên chọn trạng thái “other”, một cửa sổ modal sẽ hiện lên để nhập lý do cụ thể.
* **Actor:** Teacher.
* **Input:** Chuỗi văn bản lý do.

**FR4: Lọc và thống kê điểm danh**

* **FR4.1: Lọc điểm danh**
* **Mô tả:** Cho phép người dùng lọc các bản ghi điểm danh.
* **Actor:** Admin, Teacher.
* **Input:** Lựa chọn tháng , lựa chọn lớp học.
* **Output:** Bảng dữ liệu điểm danh được cập nhật theo tiêu chi lọc.
* **FR4.2: Hiển thị dữ liệu dưới dạng bảng**
* **Mô tả:** Dữ liệu điểm danh được hiển thị dưới dạng bảng để dễ dàng theo dõi.
* **Actor:** Admin, Teacher.
* **Output:** Bảng hiển thị thông tin chi tiết của các bản ghi điểm danh (ví dụ: tên học sinh, lớp, ngày , trạng thái, lý do).
* **Tính năng bổ sung:** Bảng có thể bao gồm các tuỳ chọn hành động (action dropdown) để chỉnh sửa hoặc xem chi tiết bàn ghi điểm danh.

**FR5: Dashboard và Analytics**

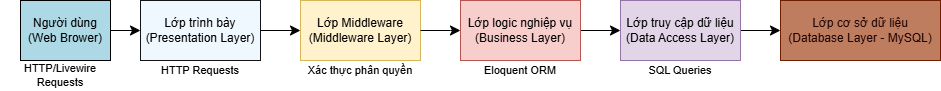
* **FR5.1: Biểu đồ trực quan hoá dữ liệu điểm danh**
* **Mô tả:** Dashboard sẽ hiển thị các biểu đồ trực quan hoá dữ liệu điểm danh, cung cấp cái nhìn tổng quan nhanh chóng.
* **Actor:** Admin, Teacher.
* **Output:** Các biểu đồ hiển thị điểm danh theo trạng thái, theo lớp, hoặc theo tổng thể.
* **Quy tắc nghiệp vụ:**
* Admin có thể xem biểu đồ của tất cả các lớp.
* Teacher chỉ xem được biểu đồ của các lớp mà họ phụ trách.
* **Phi chức năng**

| **Yêu cầu** | **Mô tả** |
| --- | --- |
| Bảo mật (Security) | - Hash mật khẩu bằng **bcrypt** để bảo vệ thông tin đăng nhập.  - Quản lý phiên đăng nhập an toàn để đảm bảo chỉ người dùng được xác thực mới có quyền truy cập. - Thực thi phân quyền dựa trên vai trò để ngăn chặn truy cập trái phép vào các chức năng và dữ liệu. |
| Hiệu suất (Performance) | - Tải trang nhanh, thời gian tải không vượt quá 3 giây.  - Các thao tác xử lý dữ liệu (CRUD, lọc) phải có phản hồi nhanh chóng, dưới 2 giây. |
| Khả dụng  (Usability) | - Giao diện phải trực quan và dễ sử dụng.  - Giao diện phải **responsive**, đảm bảo hiển thị tốt và trải nghiệm người dùng nhất quán trên các kích thước màn hình khác nhau (máy tính, máy tính bảng, điện thoại).  - Cung cấp phản hồi rõ ràng cho người dùng (thông báo lỗi, thành công). |
| Tính tương thích (Compatibility) | - Hệ thống phải hỗ trợ và hoạt động tốt trên các trình duyệt web hiện đại như Chrome, Firefox, và Edge. |
| Khả năng bảo trì (Maintainability) | - Mã nguồn phải được viết theo chuẩn của Laravel, đảm bảo dễ đọc và dễ hiểu.  - Hệ thống phải dễ dàng mở rộng, cho phép thêm các chức năng mới mà không làm ảnh hưởng lớn đến cấu trúc hiện có. |

# CHƯƠNG 3: THIẾT KẾ CHƯƠNG TRÌNH

## 3.1. Giải thích luồng hoạt động

Hệ thống quản lý điểm danh học sinh sẽ tuân theo mô hình kiến trúc **MVC (Model-View-Controller)** trên nền tảng **Client-Server**, cụ thể hơn là kiến trúc **phân lớp (N-Tier Architecture)**, một cách tiếp cận phổ biến và mạnh mẽ cho các ứng dụng web. Kiến trúc này giúp tách biệt các mối quan tâm, tăng tính module hóa, dễ bảo trì và mở rộng.



*Hình 3.1. Sơ đồ kiến trúc hệ thống*

1. **Người dùng (Web Browser):**

* **Hành động**: Người dùng (Admin hoặc giáo viên) tương tác với giao diện web thông qua trình duyệt của họ
* Ví dụ:

+ Một giáo viên muốn điểm danh: họ truy cập màn hình điểm danh.

+ Mộ admin muốn them học sinh mới: họ nhấn vào nút “Thêm học sinh” trên màn hình quản lý học sinh.

+ Người dùng muốn đăng nhập: họ nhập email và mật khẩu vào form đăng nhập.

* **Gửi yêu cầu:** Trình duyệt sẽ gửi một yêu cầu HTTP (có thể là yêu cầu thông thường hoặc Livewire requests/callback) đến server. Yêu cầu này chứa thông tin về hành động mà người dùng muốn thực hiện (ví dụ: gửi form, nhấp nút).

1. **Lớp trình bày (Presentation Layer – Frontend):**

* **Mục đích:** Đây là giao diện hiển thị cho người dùng, được xây dựng bằng Laravel Blade Templates, HTML, CSS (TailwindCSS, Bootstrap) và Laravel Livewire.
* **Tiếp nhận yêu cầu:** Lớp này nhận các yêu cầu HTTP từ trình duyệt.
* **Xử lý phía Client:**

+ Livewire: Các Livewire components (như component điểm danh, form CRUD học sinh/lớp) sẽ xử lý các tương tác cục bộ mà không cần tải lại trang. Khi có dữ liệu cần gửi về server (ví dụ: sau khi chọn trạng thái điểm danh, nhập thông tin học sinh), Livewire sẽ đóng gói dữ liệu và tạo ra một yêu cầu HTTP gửi đến server.

+ Validation cơ bản: Một số validation ban đầu có thể được thực hiện ở phía client (ví dụ: kiểm tra định dạng email) trước khi gửi request.

* **Chuyển tiếp yêu cầu:** Yêu cầu (request) từ lớp trình bày sẽ được gửi đến lớp Middleware.

1. **Lớp Middleware (Middleware Layer):**

* **Mục đích:** Đóng vai trò như một bộ lọc cho các yêu cầu HTTP. Mỗi request đi vào server sẽ phải đi qua một hoặc nhiều Middleware đã được cấu hình.
* **Xử lý chính:**

**+** **Xác thực (Authentication):** Middleware như web hoặc auth của Laravel sẽ kiểm tra xem người dùng đã được đăng nhập hay chưa. Nếu người dùng chưa đăng nhập nhưng cố gắng truy cập một trang yêu cầu xác thực (ví dụ: dashboard), Middleware sẽ chặn yêu cầu và chuyển hướng họ đến trang đăng nhập.

* Ví dụ: Khi giáo viên cố gắng truy cập /attendance mà chưa đăng nhập, Middleware sẽ redirect về /login.

**+ Phân quyền (Authorization):** Middleware có thể kiểm tra vai trò của người dùng (Admin hay Teacher) và các quyền cụ thể của họ để đảm bảo rằng họ có đủ quyền thực hiện hành động được yêu cầu.

* Ví dụ: Nếu một giáo viên cố gắng truy cập trang quản lý người dùng (chỉ dành cho Admin), Middeware sẽ từ chối truy cập và có thể trả về lỗi 403 (Forbidden). Laravel Policies/Gates thường được kiểm tra trong controller, nhưng Middleware cũng có thể được sử dụng để kiểm tra quyền hạn cấp cao hơn cho toàn bộ nhóm route.

**+ Bảo mật:** Middleware Laravel cung cấp các lớp bảo vệ chống lại các cuộc tấn công phổ biến như CSRF.

* **Chuyển tiếp yêu cầu:** Nếu yêu cầu vượt qua tất cả các kiểm tra của Middleware, nó sẽ được chuyển tiếp đến lớp Logic nghiệp vụ.

1. **Lớp logic nghiệp vụ (Business Logic Layer – Laravel Backend):**

* **Mục đích:** Đây là “bộ não” của ứng dụng, nơi chứa các Controllers và Services, xử lý các quy tắc nghiệp vụ cốt lõi.
* **Tiếp nhận yêu cầu:** Một Laravel Controller phù hợp (ví dụ: AttendanceController, StudentController, AuthController) sẽ nhập yêu cầu đã được Middleware cho phép đi qua.
* **Xử lý chính:**

**+ Validation chi tiết:** Controller sẽ thực hiện validation chi tiết hơn về dữ liệu đầu vào (ví dụ: đảm bảo tất cả các trường hợp bắt buộc của học sinh đều được điền, định dạng ngày hợp lệ).

**+ Áp dụng quy tắc nghiệp vụ:** Controller sẽ áp dụng các quy tắc kinh doanh (ví dụ: khi thêm học sinh, kiểm tra xem mã học sinh đã tồn tại chưa).

**+ Điều phối**: Controller điều phối việc thực hiện hành động:

* Ví dụ điểm danh: AttendanceController sẽ nhận dữ liệu điểm danh (Lớp, ngày, trạng thái học sinh), kiểm tra tính hợp lệ, và sau đó yêu cầu lớp truy cập dữ liệu lưu thông tin điểm danh.
* Ví dụ thêm học sinh: StudentController sẽ nhận dữ liệu học sinh, xác thực, và yêu cầu lớp truy cập dữ liệu lưu học sinh mới.
* **Chuyển tiếp yêu cầu:** Controller sẽ gọi lớp truy cập dữ liệu để thực hiện các thao tác với CSDL.

1. **Lớp truy cập dữ liệu (Data Access Layer – Eloquent Models):**

* **Mục đích:** Cung cấp một lóp trừu tượng để tương tác với cơ sở dữ liệu, sử dụng Laravel Eloquent ORM.
* **Tiếp nhận yêu cầu:** Các Eloquent Models (ví dụ: App\Models\Students, App\Models\Attendance, App\Models\Grade) được gọi bởi lớp logic nghiệp vụ.
* **Xử lý chính:**

**+ Chuyển đổi:** Eloquent ORM chuyển đổi các yêu cầu PHP thành các truy vấn SQL tương xứng.

**+ Thao tác CSDL:** Thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên cơ sở dữ liệu.

* Ví dụ: Khi thêm học sinh, Eloquent Model Student sẽ tạo một bản ghi mới trong bảng students.
* Ví dụ; Khi truy vấn danh sách điểm danh, Eloquent sẽ lấy dữ liệu từ bảng attendances.
* **Chuyển tiếp yêu cầu:** Eloquent ORM gửi các truy vấn SQL đến lớp cơ sở dữ liệu.

1. **Lớp cơ sở dữ liệu (Database Layer – MySQL):**

* **Mục đích:** Lưu trữ tất cả dữ liệu của hệ thống một cách bên vững.
* **Tiếp nhận yêu cầu:** MySQL Server nhận các truy vấn SQL từ lớp truy cập dữ liệu.
* **Xử lý chính:** Thực hiện các truy vấn SQL (chèn, cập nhật, xoá đọc) trên các bảng dữ liệu thực tế (ví dụ: users, students, grades, attendances).
* **Trả về kết quả:** MySQL trả về kết quả của truy vấn (dữ liệu hoặc thông báo thành công/thất bại) cho lớp truy cập dữ liệu.

1. **Phản hồi (Response Flow – ngược chiều):**

* Kết quả từ MySQL được chuyển ngược lại cho Eloquent ORM.
* Eloquent trả về các đối tượng PHP (Model) cho lớp logic nghiệp vụ (Controller).
* Controller xử lý kết quả này (ví dụ: tạo thông báo thành công, lấy dữ liệu để hiển thị) và chuẩn bị phản hồi.
* Phản hồi này được gửi qua lớp Middleware (nơi các Middleware “sau” request có thể xử lý, ví dụ: ghi log response).
* Cuối cùng, lớp trình bày nhận phản hồi từ server, và giao diện người dùng dược cập nhật để hiển thị kết quả (ví dụ: thông báo “Điểm danh thành công”, hiển thị danh sách học sinh đã cập nhật, hiển thị biểu đồ mới).

## 3.2. Thiết kế cơ sở dữ liệu

**A diagram of a user

AI-generated content may be incorrect.**

*Hình 3.2. Sơ đồ thực thể ERD*

* **Danh sách bảng và thuộc tính**

**Bảng users**

* Mục đích: Lưu trữ thông tin tài khoản người dùng, bao gồm quản trị viên (admin) và giáo viên (teacher).

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | bigint(20) UNSIGNED | khoá chính, tự động tăng |
| name | varchar(255) | tên người dùng |
| email | varchar(255) | địa chỉ email, duy nhất |
| password | varchar(255) | mật khẩu đã mã hoá (hash) |
| role | enum(‘admin’,’teacher’) | vai trò người dùng |
| remember\_token | varchar(100) | token ghi nhớ phiên đăng nhập (dùng trong chức năng “ghi nhớ đăng nhập”) |
| created\_at | timestamp | thời gian tạo bản ghi |
| update\_at | timestamp | thời gian cập nhật bản ghi |

**Bảng grades**

* Mục đích: Lưu trữ thông tin về các lớp học trong hệ thống.

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | bigint(20) UNSIGNED | khoá chính, tự động tăng |
| name | varchar(255) | tên lớp học, đảm bảo duy nhất |
| created\_at | timestamp NULL | thời điểm tạo bản ghi |
| update\_at | timestamp NULL | thời điểm bản ghi được cập nhật gần nhất |

**Bảng students**

* Mục đích: Lưu trữ thông tin chi tiết của từng học sinh.

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | bigint(20) UNSIGNED | khoá chính, tự động cập nhật |
| first\_name | varchar(255) | tên của học sinh |
| last\_name | varchar(255) | họ của học sinh |
| age | varchar(255) | tuổi của học sinh |
| Grade\_id | bigint(20) UNSIGNED | khoá ngoại liên kết đến bảng grades – xác định lớp học |
| Created\_at | timestamp NULL | thời điểm tạo bản ghi |
| Updated\_at | timestamp NULL | thời điểm cập nhật bản ghi |

**Bảng attendances**

* Mục đích: Lưu trữ các bản ghi điểm danh hàng ngày của học sinh

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | bigint(20) UNSIGNED | |  | | --- | |  |  |  | | --- | | Khóa chính, tự động tăng. | |
| student\_id | bigint(20) UNSIGNED | Khóa ngoại liên kết đến bảng students – học sinh được điểm danh. |
| grade\_id | bigint(20) UNSIGNED | Khóa ngoại liên kết đến bảng grades – lớp học tại thời điểm điểm danh. |
| date | date | |  | | --- | |  |  |  | | --- | | Ngày điểm danh. | |
| status | enum('present','absent','sick','other') | Trạng thái điểm danh (mặc định là 'present') |
| reason | text | |  | | --- | |  |  |  | | --- | | Lý do vắng mặt nếu trạng thái là 'other'. | |
| created\_at | timestamp NULL | Thời điểm tạo bản ghi. |
| updated\_at | timestamp NULL | Thời điểm cập nhật gần nhất. |

* **Các mối quan hệ dữ liệu**
* **Grades và students (một – nhiều):**
* Mối quan hệ: Một lớp học (grades) có thể có nhiều học sinh (students), nhưng mỗi học sinh chỉ thuộc về một lớp học tại một thời điểm (được chỉ định bởi grade\_id trong bảng students).
* Giải thích: students.grade\_id là khoá ngoại trỏ tới grades.id.
* **Students và attendances (một – nhiều):**
* Mối quan hệ: Một học sinh (students) có thể có nhiều bản ghi điểm danh (attendance), nhưng mỗi bản ghi điểm danh chỉ thuộc về một học sinh duy nhất.
* Giải thích: attendance.student\_id là khoá ngoại trỏ tới students.id.
* **Grades và attendance (một – nhiều):**
* Mối quan hệ: Một lớp học (grades) có thể có nhiều bản ghi điểm danh (attendances), nhưng mỗi bản ghi điểm danh thuộc về một lớp học cụ thể tại thời điểm đó.
* Giải thích: attendances.grade\_id là khoá ngoại trỏ tới grades.id. Mối quan hệ này giúp lọc và tổng hợp điểm danh theo lớp một cách hiệu quả.
* **Users và grades (một – nhiều):**
* Mối quan hệ: Một người dùng (users với role = ‘teacher’) có thể phụ trách nhiều lớp học (grades).
* Giải thích: Nếu một giáo viên có thể phụ trách nhiều lớp và một lớp có thể có nhiều giáo viên, cần một bảng trung gian (grade\_teacher).
* **Users và attendances (một nhiều – đề xuất bổ sung):**
* Mối quan hệ: Một người dùng (users với role = ‘teacher’) có thể tạo nhiều bản ghi điểm danh (attendances), nhưng mỗi bản ghi điểm danh được tạo bởi một người dùng duy nhất.
* Giải thích: Việc thêm cột marked\_by\_user\_id vào bảng attendances (là khoá ngoại trỏ tới users.id) sẽ giúp theo dõi ai là người đã thực hiện điểm danh, rất quan trọng cho việc kiểm soát và truy vết dữ liệu.

## 3.3. Thiết kế hướng đối tượng (UML Class Diagram)

Sơ đồ Diagram dưới đây minh hoạ cấu trúc tĩnh của hệ thống, bao gồm các lớp chính (attributes) và phương thức (methods) của chúng, cùng với các mối quan hệ giữa các lớp. Đây là cơ sở cho việc triển khai mã nguồn sau này.

**3.4. Thiết kế luồng dữ liệu (Data Flow)**

* **Sơ đồ luồng dữ liệu cấp ngữ cảnh (context level DFD)**

Sơ đồ này biểu diễn hệ thống như một hộp đen, thể hiện các thực thể bên ngoài tương tác với hệ thống và các luồng dữ liệu chính đi vào/ra khỏi hệ thống.

*Hình 5. Sơ đồ luồng dữ liệu cấp ngữ cảnh*

*Hình 3.3. Sơ đồ Class Diagram*

**Giải thích sơ đồ:**

* **Hệ thống (system):** Đây là trung tâm của sơ đồ, biểu thị toàn bộ “Students Attendance Tracking System”.
* **Thực thể ngoài (external entities):**
* **Admin:** Tương tác với hệ thống để quản lý người dùng, quản lý lớp học, quản lý học sinh, xem báo cáo tổng quan.
* **Teacher:** Tương tác với hệ thống để điểm danh học sinh, xem báo cáo lớp học.
* **Database** : Lưu trữ và cung cấp dữ liệu hệ thống cho các hoạt động của hệ thống.
* **Luồng dữ liệu (data flows):**
* Thông tin quản lý, yêu cầu báo cáo từ Admin đến hệ thống.
* Thông tin cập nhật, báo cáo quản lý từ hệ thống đến admin.
* Yêu cầu điểm danh, thông tin điểm danh từ Teacher đến hệ thống.
* Báo cáo điểm danh, báo cáo lớp học từ hệ thống đến Teacher.
* Truy vấn dữ liệu, cập nhật dữ liệu giữa hệ thống và Database.

## 3.4. Thiết kế luồng dữ liệu (data flow diagram)

### 3.4.1. Sơ đồ ngữ cảnh (context diagram – level 0)

Sơ đồ mô tả tổng quan hệ thống như một “hộp đen”, chỉ ra các tác nhân tương tác chính và luồng dữ liệu giữa họ với hệ thống.

A diagram of students attendance tracking system

AI-generated content may be incorrect.

*Hình 3.4. Sơ đồ ngữ cành hệ thống*

**Mô tả:**

* Giáo viên: Gửi yêu cầu điểm danh và nhận báo cáo chuyên cần của lớp mình.
* Admin: Quản lý toàn bộ hệ thống, bao gồm người dùng, lớp học, học sinh và xem báo cáo tổng thể.
* Hệ thốn báo cáo (tác nhân phụ): Nhận dữ liệu để xử lý ngoài hệ thống chính (ví dụ: in ra PDF, gửi mail).

**Luồng dữ liệu chính:**

* Từ giáo viên Hệ thống: Yêu cầu điểm danh, quản lý học sinh/lớp.
* Từ hệ thống Giáo viên: Báo cáo chuyên cần.
* Từ admin Hệ thống: Quản trị hệ thống
* Từ hệ thống Hệ thống báo cáo: Dữ liệu điểm danh đã xử lý.

### 3.4.2. Sơ đồ luồng dữ liệu mức 1 (DFD lever 1)

Mục tiêu: Phân rã các tiến trình bên trong hệ thống và thể hiện rõ cách các tác nhân, kho dữ liệu và tiến trình tương tác với nhau.

*Hình 3.5. Sơ đồ luồng dữ liệu mức 1 của hệ thống*

**Các tiến trình chính:**

* **Quản lý người dùng (1.0)** – Xử lý thêm/sửa/xoá tài khoản Admin và Giáo viên.
* **Quản lý lớp học (2.0)** – Tạo mới, cập nhật thông tin lớp.
* **Quản lý học sinh (3.0)** – Tạo mới học sinh, gán lớp.
* **Thực hiện điểm danh (4.0)** – Ghi nhận trạng thái chuyên cần của từng học sinh theo ngày.

**Các kho dữ liệu:**

* D1 – Người dùng
* D2 – Lớp học
* D3 – Học sinh
* D4 – Điểm danh

**Mô tả luồng dữ liệu:**

* Giáo viên tương tác chủ yếu với tiến trình 4.0 để thực hiện điểm danh, sau đó hệ thống ghi dữ liệu vào D4.
* Admin có quyền gửi yêu cầu đến tất cả tiến trình, bao gồm việc tạo và xem báo cáo (5.0).
* Tất cả các tiến trình đều cập nhật hoặc truy xuất dữ liệu từ các kho dữ liệu tương ứng (D1 → D4).

## 3.5. Thiết kế chức năng chi tiết (UML Activity Diagram)

Sơ đồ luồng hoạt động (Activity Diagram) minh hoạ luồng công việc hoặc quy trình nghiệp vụ từ điểm bắt đầu đến điểm kết thúc, bao gồm các hoạt động, điểm quyết định, luồng song song và các luồng điều khiển. Sơ đồ dưới đây mô tả chi tiết quy trình điểm danh học sinh trong hệ thống.

*Hình 3.4. Sơ đồ luồng hoạt động chức năng điểm danh học sinh*

**Phân tích luồng hoạt động chức năng “Điểm danh học sinh”.**

* **Bắt đầu (start node):** Quy trình bắt đầu từ điểm đen.
* **Swimlane:** Sơ đồ sử dụng một swimlane duy nhất cho Teacher, cho thấy tất cả hoạt động được thực hiện bởi vai trò giáo viên.
* **Các hoạt động (activities) và luồng (flows):**

1. **Đăng nhập hệ thống:** Giáo viên phải đăng nhập thành công vào hệ thống
2. **Chọn lớp học:** Sau khi đăng nhập, giáo viên chọn lớp học mà mình muốn điểm danh từ danh sách các lớp được phân công.
3. **Hiển thị danh sách học sinh:** Hệ thống hiển thị danh sách các học sinh thuộc lớp đã chọn.
4. **Chọn học sinh cần điểm danh:** Giáo viên chọn một học sinh từ danh sách để tiến hành điểm danh.
5. **Chọn trạng thái điểm danh (Có mặt/Vắng mặt/Ốm/Khác):** Giáo viên chọn trạng thái chuyên cần cho học sinh đó.
6. **Quyết định (decision node – hình thoi):**

* Trạng thái là ‘khác’: Nếu giáo viên chọn trạng thái là “khác”, luồng hoạt động sẽ đi theo nhánh này.
* Nhập lý do: Giáo viên phải nhập lý do cụ thể cho trạng thái “khác”.
* Lưu điểm danh: Sau khi nhập lý do, hệ thống tiến hành lưu bản ghi điểm danh.
* Trạng thái không phải ‘khác’: Nếu giáo viên chọn trạng thái là “Có mặt”, “Vắng mặt” hoặc “Ốm”, luồng hoạt động sẽ đi theo nhánh này.
* Lưu điểm danh: Hệ thống lưu bản ghi điểm danh với trạng thái đã chọn.

1. **Đã điểm danh hết học sinh trong lớp (decision node):** Sau khi lưu điểm danh cho một học sinh, hệ thống kiểm tra xem tất cả học sinh trong lớp đã được điểm danh chưa.

* Chưa: Nếu còn học sinh chưa điểm danh, luồng quay lại hoạt động chọn học sinh cần điểm danh để tiếp tục quy trình.
* Rồi: Nếu tất cả học sinh đã được điểm danh, quy trình kết thúc.

# CHƯƠNG 4: GIAO DIỆN

A screenshot of a login form

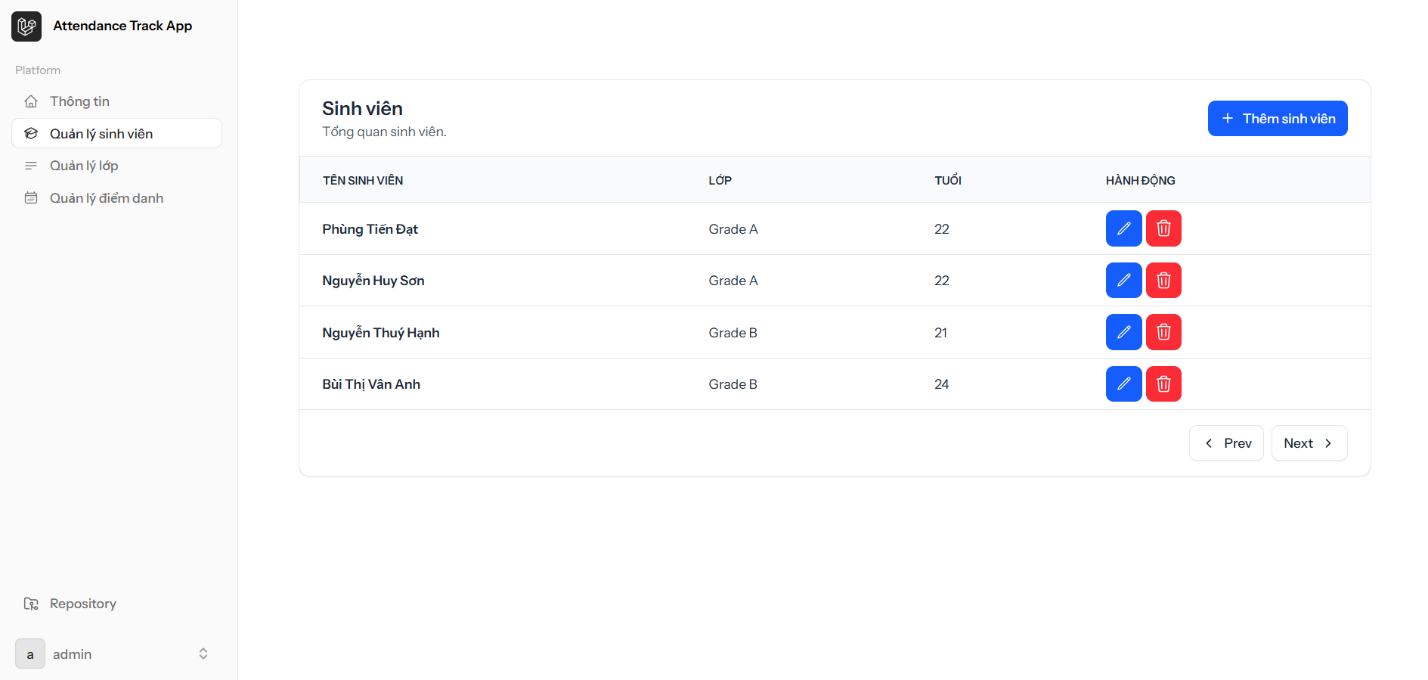
AI-generated content may be incorrect.

*Hình 4.1. Giao diện trang đăng nhập*

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 4.2. Giao diện trang Dashboard*



*Hình 4.3. Giao diện trang quản lý sinh viên*

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 4.4. Giao diện trang quản lý lớp học.*

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 4.5. Giao diện trang quản lý điểm danh*

# CHƯƠNG 5. CÁC CÔNG NGHỆ ĐƯỢC SỬ DỤNG VÀ ĐÁNH GIÁ HỆ THỐNG

Hệ thống “Quản lý điểm danh (Students attendance tracking system)” được xây dựng dựa trên các công nghệ phổ biến, mạnh mẽ và hiện đại, bao gồm:

## 5.1. Ngôn ngữ và Framework

|  |  |
| --- | --- |
| **Công nghệ** | **Vai trò chính** |
| PHP 8.2 | Ngôn ngữ lập trình chính phía server |
| Laravel 10 | Framework PHP dùng để xây dựng backend MVC |
| Livewire | Thư viện hỗ trợ xây dựng giao diện động mà không cần JavaScript |
| Blade | Template engine mặc định trong Laravel |
| HTML5/CSS3 | Xây dựng giao diện frontend cơ bản |
| Tailwind CSS | Thư viện CSS hỗ trợ thiết kế giao diện hiện đại và responsive |

## 5.2. Cơ sở dữ liệu

|  |  |
| --- | --- |
| **Công nghệ** | **Vai trò** |
| MySQL | Lưu trữ dữ liệu học sinh, điểm danh, tài khoản |
| phpMyAdmin | Giao diện quản lý cơ sở dữ liệu |
| Eloquent ORM | Công cụ ORM đi kèm Laravel giúp truy vấn dễ dàng |

## 5.3. Kiểm thử và xác minh

|  |  |
| --- | --- |
| **Công cụ** | **Mục đích sử dụng** |
| Laravel Testing (PHPUnit) | Kiểm thử đơn vị và tích hợp hệ thống |

## 5.4. Môi trường phát triển

|  |  |
| --- | --- |
| **Thành phần** | **Mô tả** |
| XAMPP | Môi trường máy chủ cục bộ (Apache, MySQL, PHP) |
| Visual Studio Code | Trình soạn thảo mã nguồn chính |
| Composer | Trình quản lý thư viện PHP |
| Git/Github | Quản lý mã nguồn, backup và cộng tác nhóm |
| Draw.io | Vẽ sơ đồ Use, class, activity, Data Flow |

## 5.5. Đánh giá hệ thống

Sau quá trình thiết kế và triển khai, hệ thống “Quản lý điểm danh (Students attendance tracking system)” được đánh giá tổng thể như sau:

* **Tính hoàn thiện chức năng**
* Hệ thống đã đáp ứng đầy đủ chức năng cần thiết: quản lý người dùng, lớp học, học sinh, điểm danh, tạo báo cáo và phân quyền.
* Phân quyền rõ ràng giữa Admin và Giáo viên, dễ dàng thao tác.
* **Thết kế cơ sở dữ liệu**
* Các bảng được thiết kế logic, có đầy đủ khoá chính, khoá ngoại, các cột timestamp hỗ trợ tracking dữ liệu.
* Sử dụng enum cho trạng thái điểm danh là phù hợp và tối ưu.
* **Thiết kế giao diện**
* Giao diện đơn giản, dễ sử dụng, tập trung vào trải nghiệm giáo viên.
* Có biểu đồ chuyên cần và chức năng lọc báo cáo rõ ràng.
* Một số phần chưa tối ưu cho thiết bị di động nhưng hoàn toàn có thể mở rộng trong tương lai.
* **UML và sơ đồ hệ thống**
* Đầy đủ sơ đồ hỗ trợ phân tích: Use Case, Class, Activity, Data Flow
* Luồng xử lý phản ánh đúng logic nghiệp vụ, đặc biệt là phần điểm danh có xử lý cho trạng thái đặc biệt (other).
* **Đánh giá hiệu năng và thực tiễn**
* Hệ thống hoạt động ổn định trong mỗi trường localhost với Laravel + MySQL.
* Giao diện thân thiện, có thể triển khai thực tế trong quy mô nhỏ tại các trường học hoặc trung tâm đào tạo.
* **Hạn chế và đề xuất mở rộng**
* Chưa có phân hệ cho phụ huynh và học sinh.
* Chưa có xác thực bảo mật 2 lớp (MFA).
* Chưa có kiểm thử hiệu suất trên dữ liệu lớn.