

**ĐẠI HỌC PHENIKAA**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN PHENIKAA**

\*\*\*\*\*



**BÁO CÁO**  
**PHÂN TÍCH VÀ THIẾT KẾ PHẦN MỀM**  
**XÂY DỰNG HỆ THỐNG QUẢN LÝ SINH VIÊN VỚI ĐIỂM DANH BẰNG KHUÔN**  
**MẶT**  
*(Nhóm 06)*

Sinh viên:

Mã số sinh viên:

Lớp học phần: Phân tích và thiết kế phần mềm (N06)

Giảng viên hướng dẫn: TS. Trịnh Thanh Bình  
ThS. Vũ Quang Dũng

*Hà Nội, tháng 10 năm 2025*

## LỜI CẢM ƠN

Để hoàn thành được đề tài này, em xin tỏ lòng biết ơn sâu sắc đến thầy Vũ Quang Dũng và thầy Trịnh Thanh Bình đã tận tình hướng dẫn và tạo điều kiện thuận lợi cho chúng em trong suốt quá trình thực hiện đề tài. Em xin chân thành cảm ơn quý thầy cô trong khoa đã tận tình chỉ dạy cho em tạo nền tảng kiến thức vững chắc để hoàn thành đề tài này.

Trong quá trình thực hiện đề tài không thể tránh khỏi những sai sót kính mong các quý thầy cô góp ý và chỉ dẫn để em có thể hoàn thiện hơn.

Em xin chân thành cảm ơn!

**BẢNG PHÂN CHIA CÔNG VIỆC**

| Họ và tên       | MSSV     | Nhiệm vụ  | Mức độ hoàn thành |
|-----------------|----------|---|-------------------|
| Lương Hồng Quân | 23010423 | Thực hiện triển khai mô hình trí tuệ nhân tạo (AI) vào hệ thống điểm danh bằng nhận diện khuôn mặt, bao gồm việc xây dựng pipeline xử lý ảnh, phát hiện – căn chỉnh khuôn mặt, trích xuất đặc trưng, và so khớp để nhận dạng. | 100%              |
| Đàm Anh Pháp    | 23010360 | Thiết kế giao diện, cơ sở dữ liệu và thực hiện các chức năng cơ bản của hệ thống (Backend), sử dụng mô hình AI để tiến hành điểm danh cho hệ thống  | 100%              |
| Hoàng Minh Quân | 23010315 | Thiết kế giao diện web có sử dụng figma. Thiết kế giao diện cho các chức năng chính(Login, Dashboard, Student, ...). Thiết kế cho các components để có thể tái sử dụng  | 100%              |
| Lê Thị Dương    | 23010351 | Phân tích yêu cầu, xác định chức năng và trực quan hóa các luồng hoạt động và trình tự hoạt động, CRUD cho Auth, cours và cours class, tích hợp Middleware phân quyền   | 100%              |
| Phạm Hồng Đức   | 23010338 | Thiết kế giao diện web có sử dụng figma Thiết kế giao diện cho các chức năng(Login, Dashboard, Student, Teacher, Display Course, Display Program, Schedule, Enrollment, Attendance)   | 100%              |

|   |           |
|---|-----------|
| <b>CHƯƠNG I: ĐẶT VẤN ĐỀ.....</b>  | <b>3</b>  |
| 1.1 Đặt Vấn Đề và Cơ Sở Lý Luận.....  | 3         |
| 1.2 Mục Tiêu và Phạm Vi Dự Án.....  | 3         |
| 1.3 Phương Pháp Phát Triển và Kế hoạch Thực hiện (10 Tuần).....   | 4         |
| <b>CHƯƠNG II: Phân Tích Yêu Cầu và Mô Hình Hóa.....</b>   | <b>4</b>  |
| 2.1 Các Vai trò (Actors) trong Hệ Thống.....  | 5         |
| 2.2 Yêu Cầu Chức Năng (Functional Requirements) và Yêu Cầu Phi Chức Năng (Non-Functional Requirements)..... | 5         |
| 2.3 Mô Hình Hóa Yêu Cầu (UML Diagrams).....   | 8         |
| 2.4. Thiết kế giao diện.....  | 11        |
| <b>CHƯƠNG III: Thiết Kế Hệ Thống.....</b>   | <b>12</b> |
| 3.1 Thiết Kế Kiến Trúc Hệ Thống (N-Tier/Microservices).....   | 12        |
| 3.2 Thiết Kế Chi Tiết UML và Cơ Sở Dữ Liệu.....   | 14        |
| 3.3 Thiết Kế Module Hệ Thống (Package Diagram).....   | 16        |
| <b>CHƯƠNG IV: TRIỂN KHAI VÀ KIỂM THỬ.....</b>   | <b>17</b> |
| 4.1 Chi Tiết Công Nghệ và Lý do Lựa chọn.....   | 17        |
| 4.2 Chi Tiết Kịch Bản Kiểm Thử (Testing Scenarios).....   | 18        |
| <b>CHƯƠNG 5: TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN.....</b>  | <b>18</b> |
| 5.1 Đánh Giá Dự Án.....   | 18        |
| 5.2 Hướng Phát Triển Tương Lai.....   | 18        |
| <b>CHƯƠNG VI. KIẾN NGHỊ.....</b>  | <b>19</b> |

## CHƯƠNG I: ĐẶT VẤN ĐỀ

---

### 1.1 Đặt Vấn Đề và Cơ Sở Lý Luận

- Trong bối cảnh chuyển đổi số mạnh mẽ của ngành giáo dục, các phương pháp quản lý sinh viên và điểm danh truyền thống tại các cơ sở đào tạo đã bộc lộ nhiều hạn chế đáng kể. Việc ghi chép thủ công tốn thời gian, dễ xảy ra sai sót, và khó kiểm soát được tình trạng điểm danh hộ, gây ảnh hưởng đến tính chính xác của dữ liệu học tập và đánh giá sinh viên. Nhu cầu về một hệ thống quản lý hiện đại, tự động hóa và đáng tin cậy là vô cùng cấp thiết.
- Nhận thấy những thách thức này, đề tài "Xây dựng Hệ thống Quản lý Sinh viên Với Tích hợp Điểm danh Bằng Khuôn mặt" ra đời nhằm mục đích ứng dụng các thành tựu của công nghệ thông tin và trí tuệ nhân tạo (AI) để giải quyết các vấn đề trên. Hệ thống này không chỉ đơn thuần là một công cụ quản lý thông tin mà còn là một giải pháp tự động hóa quy trình điểm danh, nâng cao hiệu quả và minh bạch trong môi trường giáo dục.

### 1.2 Mục Tiêu và Phạm Vi Dự Án

1. Mục tiêu chính: Xây dựng một hệ thống phần mềm hoàn chỉnh, cho phép:
  - o Quản lý toàn diện thông tin về sinh viên, giáo viên, khóa học, lớp học và lịch học.
  - o Tự động hóa quá trình điểm danh bằng công nghệ nhận diện khuôn mặt, giúp tiết kiệm thời gian và tăng cường độ chính xác.
  - o Cung cấp các báo cáo và thống kê về tình hình điểm danh, hỗ trợ việc đánh giá và quản lý.
  - o Đảm bảo hệ thống hoạt động an toàn, hiệu quả và có giao diện thân thiện với người dùng.
2. Phạm vi dự án: Dự án tập trung vào các chức năng cốt lõi sau:
  - o Quản lý thông tin cơ bản: Bao gồm thêm, sửa, xóa, tìm kiếm thông tin về Sinh viên, Giáo viên, Khóa học, Lớp học, Phòng học.
  - o Quản lý lịch học: Cung cấp giao diện để tạo, cập nhật, xem lịch học cho các lớp và giáo viên.
  - o Hệ thống Điểm danh tự động: Phát triển mô-đun nhận diện khuôn mặt sử dụng camera để tự động ghi nhận trạng thái điểm danh của sinh viên.
  - o Hệ thống báo cáo và thống kê: Cho phép xuất báo cáo điểm danh theo ngày, theo lớp, theo sinh viên.
  - o Nền tảng triển khai: Ứng dụng Web Responsive, truy cập được trên các thiết bị desktop và mobile (với các chức năng cơ bản).

### 1.3 Phương Pháp Phát Triển và Kế hoạch Thực hiện (10 Tuần)

- Để đảm bảo tính linh hoạt, khả năng thích ứng với các yêu cầu thay đổi và hiệu quả trong việc quản lý tiến độ, chúng em đã lựa chọn mô hình phát triển Agile

Scrum. Mô hình này cho phép chia dự án thành các chu kỳ ngắn (Sprints), dễ dàng kiểm soát rủi ro và nhận phản hồi sớm để điều chỉnh.

- Kế hoạch thực hiện (Tổng cộng 10 Tuần):
- Chúng em phân chia dự án thành 6 Sprint, mỗi Sprint có mục tiêu cụ thể và đầu ra rõ ràng:

| Sprint | Nội dung chính                                 | Thời gian (Tuần) | Mục tiêu chi tiết  |
|--------|--|------------------|--|
| S1     | Phân tích yêu cầu & Thiết kế CSDL              | 2                | Hoàn thành các sơ đồ Use Case, Activity, Class Diagram. Thiết kế chi tiết cơ sở dữ liệu.                   |
| S2     | Phát triển API Backend Core                    | 2                | Xây dựng các API cho quản lý thông tin (Sinh viên, Giáo viên, Khóa học, Lớp học).                          |
| S3     | Phát triển Frontend & Tích hợp                 | 2                | Hoàn thiện giao diện người dùng cho các vai trò (Admin, Giáo viên, Sinh viên) và tích hợp các API quản lý. |
| S4     | Triển khai Face Recognition Service & Tích hợp | 2                | Xây dựng và tích hợp mô-đun AI nhận diện khuôn mặt. Triển khai luồng điểm danh tự động.                    |
| S5     | Tối ưu hóa hiệu năng & Kiểm thử toàn diện      | 1                | Chạy kiểm thử hiệu năng cho API và AI. Tối ưu hóa các dịch vụ, xử lý lỗi và cải thiện tốc độ.              |
| S6     | Tổng kết, Báo cáo & Hoàn thiện                 | 1                | Viết báo cáo cuối kỳ, chuẩn bị trình bày, và khắc phục các lỗi nhỏ cuối cùng.                              |

## CHƯƠNG II: Phân Tích Yêu Cầu và Mô Hình Hóa

### 2.1 Các Vai trò (Actors) trong Hệ Thống

- Hệ thống được thiết kế để phục vụ ba vai trò chính, mỗi vai trò được định nghĩa rõ ràng về quyền hạn và các tác vụ có thể thực hiện:
  - Admin (Quản trị viên): Là vai trò có quyền cao nhất trong hệ thống, chịu trách nhiệm quản lý toàn bộ cấu trúc và dữ liệu. Các chức năng chính bao gồm:
    - o Quản lý phòng học: Thêm, sửa, xóa thông tin phòng học, đảm bảo dữ liệu phòng học luôn chính xác.
    - o Quản lý người dùng: Quản lý thông tin chi tiết của tất cả các tài khoản (Admin, Giảng viên, Sinh viên), bao gồm thêm mới, chỉnh sửa thông tin cá nhân, cấp quyền và xóa tài khoản. Đặc biệt, Admin chịu trách nhiệm đăng ký dữ liệu khuôn mặt ban đầu cho sinh viên để hệ thống có thể nhận diện.

- o Quản lý và tạo thời khóa biểu: Tạo mới, xem, chỉnh sửa và phân công thời khóa biểu cho các lớp học và Giảng viên, đảm bảo không có xung đột lịch trình.
- o Quản lý môn học: Thêm, sửa, xóa các môn học được giảng dạy trong trường.
- o Quản lý điểm danh: Xem tổng quan và chi tiết kết quả điểm danh của tất cả các lớp học, hỗ trợ việc theo dõi và đánh giá chung.
- Giảng viên (Teacher): Là vai trò trực tiếp tương tác với sinh viên và thực hiện các nghiệp vụ giảng dạy. Các chức năng chính bao gồm:
  - o Xem lịch dạy: Tra cứu thời khóa biểu cá nhân của mình, biết được lịch giảng dạy các môn học, lớp học và phòng học cụ thể.
  - o Thực hiện điểm danh: Kích hoạt chức năng điểm danh tự động bằng khuôn mặt cho lớp học đang diễn ra.
  - o Xem kết quả điểm danh: Xem lại kết quả điểm danh chi tiết của các buổi học đã diễn ra.
  - o Xuất bảng điểm (chuyên cần): Xuất báo cáo tổng kết chuyên cần của sinh viên trong môn học mình phụ trách.
- Sinh viên (Student): Là đối tượng chính của hệ thống, sử dụng ứng dụng để theo dõi thông tin học tập. Các chức năng chính bao gồm:
  - o Xem lịch học: Tra cứu thời khóa biểu cá nhân, bao gồm các môn học, thời gian, địa điểm và giảng viên.
  - o Xem điểm: Xem điểm chuyên cần của bản thân trong các môn học.

## 2.2 Yêu Cầu Chức Năng (Functional Requirements) và Yêu Cầu Phi Chức Năng (Non-Functional Requirements)

### 2.2.1 Yêu cầu Chức năng (YCC)

- Các YCC được xác định dựa trên phân tích các vai trò và nghiệp vụ cốt lõi:
1. Quản lý Phòng học:
    - o Admin có thể thêm thông tin phòng học mới (tên, sức chứa, trạng thái).
    - o Admin có thể chỉnh sửa thông tin phòng học hiện có.
    - o Admin có thể xóa phòng học.
    - o Admin có thể tìm kiếm phòng học theo tên hoặc trạng thái.
  2. Quản lý Người dùng:
    - o Admin có thể thêm tài khoản Giảng viên/Sinh viên mới.
    - o Admin có thể chỉnh sửa thông tin cá nhân của Giảng viên/Sinh viên.
    - o Admin có thể xóa tài khoản Giảng viên/Sinh viên.
    - o Admin có thể đăng ký/cập nhật dữ liệu khuôn mặt cho sinh viên.
    - o Người dùng (Admin, Giảng viên, Sinh viên) có thể đăng nhập/đăng xuất.
    - o Người dùng có thể đổi mật khẩu.
  3. Quản lý Môn học:
    - o Admin có thể thêm môn học mới (mã môn, tên môn, số tín chỉ).

- o Admin có thể chỉnh sửa thông tin môn học.
  - o Admin có thể xóa môn học.
4. Quản lý và Tạo Thời khóa biểu:
- o Admin có thể tạo thời khóa biểu cho một lớp/môn học cụ thể (chọn Giảng viên, Phòng học, thời gian).
  - o Admin có thể chỉnh sửa thời khóa biểu đã tạo.
  - o Admin có thể xem tổng quan thời khóa biểu của toàn trường.
5. Quản lý Điểm danh:
- o Giảng viên có thể thực hiện điểm danh tự động bằng khuôn mặt trong buổi học.
  - o Hệ thống tự động nhận diện sinh viên có mặt.
  - o Giảng viên/Admin có thể xem kết quả điểm danh chi tiết theo lớp/buổi học.
6. Xem thông tin:
- o Sinh viên có thể xem lịch học cá nhân.
  - o Giảng viên có thể xem lịch dạy cá nhân.
  - o Sinh viên có thể xem kết quả điểm danh của bản thân.
- 2.2.2 Yêu cầu Phi Chức năng (YCPC)
- Các YCPC đóng vai trò quan trọng trong việc đảm bảo chất lượng, hiệu suất và tính bền vững của hệ thống:
1. Hiệu năng (Performance):
- o Thời gian phản hồi API: Mọi yêu cầu API (trừ các tác vụ AI nặng) phải có thời gian phản hồi dưới 2 giây.
  - o Thời gian tải giao diện: Giao diện người dùng phải được tải hoàn tất trong vòng 3 giây trên mạng ổn định.
  - o Tốc độ nhận diện khuôn mặt: Toàn bộ quy trình nhận diện khuôn mặt (từ khi chụp ảnh đến khi có kết quả) phải hoàn tất trong vòng 5 giây để không làm gián đoạn buổi học.
2. Bảo mật (Security):
- o Xác thực và phân quyền: Sử dụng JSON Web Token (JWT) cho xác thực người dùng và triển khai cơ chế phân quyền dựa trên vai trò (Role-Based Access Control - RBAC) để đảm bảo Admin, Giảng viên, Sinh viên chỉ có thể truy cập các chức năng được phép.
  - o Mã hóa mật khẩu: Mật khẩu người dùng phải được mã hóa mạnh (ví dụ: sử dụng bcrypt) trước khi lưu trữ vào cơ sở dữ liệu.
  - o Bảo vệ dữ liệu sinh trắc học: Dữ liệu khuôn mặt của sinh viên sẽ được chuyển đổi thành vector đặc trưng (face embeddings) và mã hóa trước khi lưu trữ. Không lưu trữ ảnh khuôn mặt gốc.
  - o Phòng chống tấn công: Hệ thống cần có các biện pháp chống lại các cuộc tấn công phổ biến như SQL Injection, Cross-Site Scripting (XSS), CSRF.
3. Khả năng sử dụng (Usability):

- o Giao diện trực quan: Giao diện người dùng phải được thiết kế đơn giản, rõ ràng, dễ hiểu và dễ thao tác cho tất cả các đối tượng người dùng.
- o Tương thích đa thiết bị (Responsive Design): Giao diện phải tự động điều chỉnh và hiển thị tối ưu trên các loại thiết bị khác nhau (máy tính bàn, laptop, máy tính bảng).
- o Hệ thống thông báo: Cung cấp các thông báo rõ ràng về trạng thái hệ thống, kết quả thao tác cho người dùng.

#### 4. Khả năng mở rộng (Scalability):

- o Kiến trúc Microservices: Thiết kế hệ thống theo hướng microservices-ready, đặc biệt là tách dịch vụ nhận diện khuôn mặt, cho phép dễ dàng mở rộng từng thành phần độc lập khi số lượng người dùng hoặc dữ liệu tăng lên.
- o Công nghệ linh hoạt: Lựa chọn các công nghệ có khả năng mở rộng tốt (FastAPI, PostgreSQL).

#### 5. Độ tin cậy (Reliability):

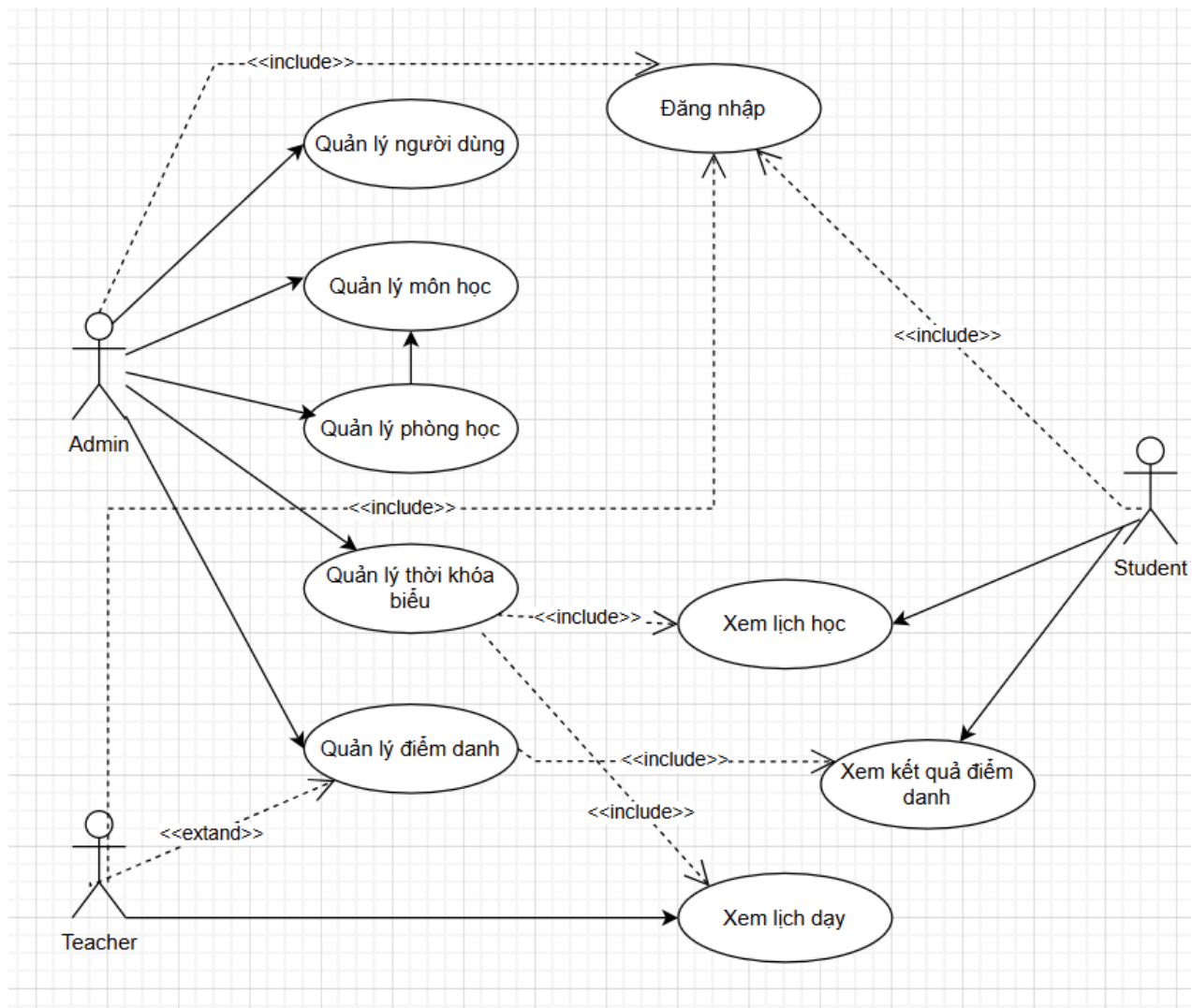
- o Thời gian hoạt động (Uptime): Hệ thống phải đảm bảo thời gian hoạt động liên tục tối thiểu 99%.
- o Xử lý lỗi: Hệ thống phải có cơ chế xử lý lỗi mạnh mẽ, ghi log lỗi chi tiết để hỗ trợ khắc phục sự cố nhanh chóng.

### 2.3 Mô Hình Hóa Yêu Cầu (UML Diagrams)

Phần này sẽ trực quan hóa các yêu cầu đã phân tích bằng các sơ đồ UML, cung cấp cái nhìn tổng quan và chi tiết về cấu trúc hệ thống.

#### 2.3.1 Use Case Diagram (Tổng quan chức năng)

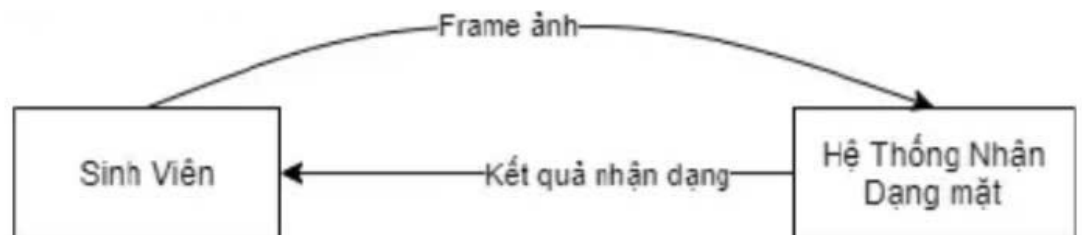
Sơ đồ này là nền tảng cho việc xác định phạm vi chức năng của dự án, minh họa các chức năng cốt lõi và các Actors (tác nhân) tương tác với hệ thống.

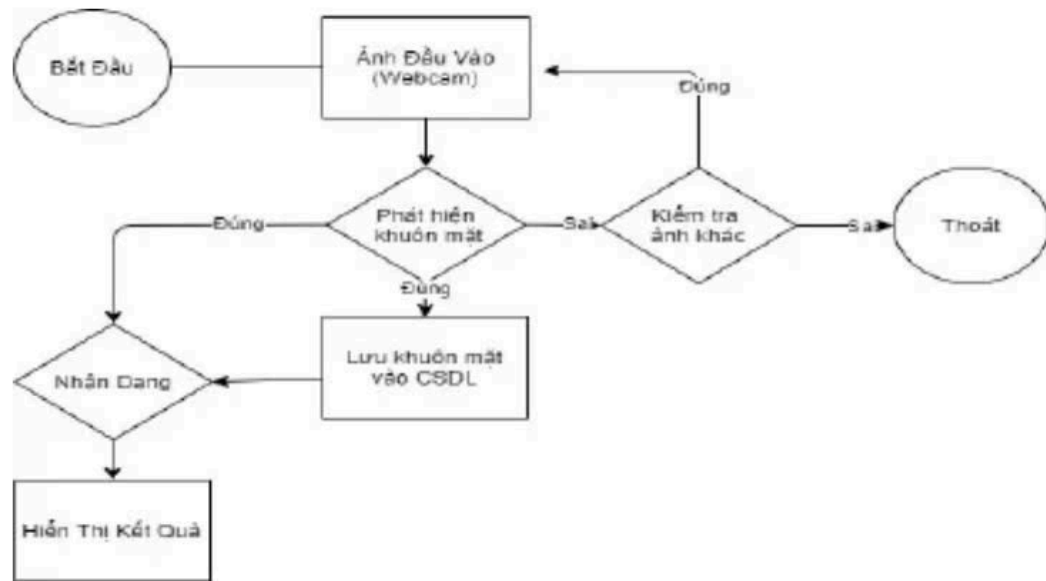


Danh sách Use Case và mô tả

| Use Case                      | Actor tham gia        | Mô tả ngắn  |
|-------------------------------|-----------------------|---|
| Quản lý phòng học             | Admin                 | Thêm/sửa/xóa thông tin phòng học.                           |
| Quản lý người dùng            | Admin                 | Tạo tài khoản, phân quyền người dùng.                       |
| Quản lý và tạo thời khóa biểu | Admin                 | Lập và cập nhật thời khóa biểu cho sinh viên và giảng viên. |
| Quản lý môn học               | Admin                 | Quản lý danh sách môn học.                                  |
| Quản lý điểm danh             | Admin, Giảng viên     | Ghi nhận và cập nhật trạng thái điểm danh của sinh viên.    |
| Xem kết quả điểm danh         | Admin, Giảng viên, SV | Xem kết quả điểm danh theo lớp hoặc cá nhân.                |
| Xem lịch học                  | Admin, Giảng viên, SV | Hiển thị lịch học đã được tạo trong hệ thống.               |
| Xem lịch dạy                  | Admin, Giảng viên     | Giảng viên xem lịch dạy của mình.                           |

### 2.3.2 Activity Diagram (Luồng Điểm Danh Tự động)





- Giải thích: Sơ đồ này mô tả chi tiết các bước và quyết định trong quá trình điểm danh tự động. Bắt đầu từ khi Giáo viên kích hoạt điểm danh, hệ thống sẽ liên tục chụp ảnh, gửi ảnh đến dịch vụ AI, AI xử lý và nhận diện, sau đó gửi kết quả về Backend API để cập nhật trạng thái điểm danh. Luồng sẽ bao gồm các nút rẽ nhánh (decision node) như "Nhận diện thành công?" để xác định các nhánh hoạt động tiếp theo. **Mục tiêu:** mô tả các bước hoạt động khi diễn ra điểm danh bằng khuôn mặt.

- Các bước chính:

Giáo viên mở buổi học → kích hoạt “Điểm danh tự động”.

Camera bắt đầu chụp khuôn mặt sinh viên.

Ảnh được gửi đến AI Face Recognition Service.

AI phân tích ảnh → trích xuất “vector khuôn mặt” (embedding).

Hệ thống kiểm tra:

Nếu khớp với dữ liệu đã đăng ký → đánh dấu “Có mặt”.

Nếu không khớp → đánh dấu “Vắng”.

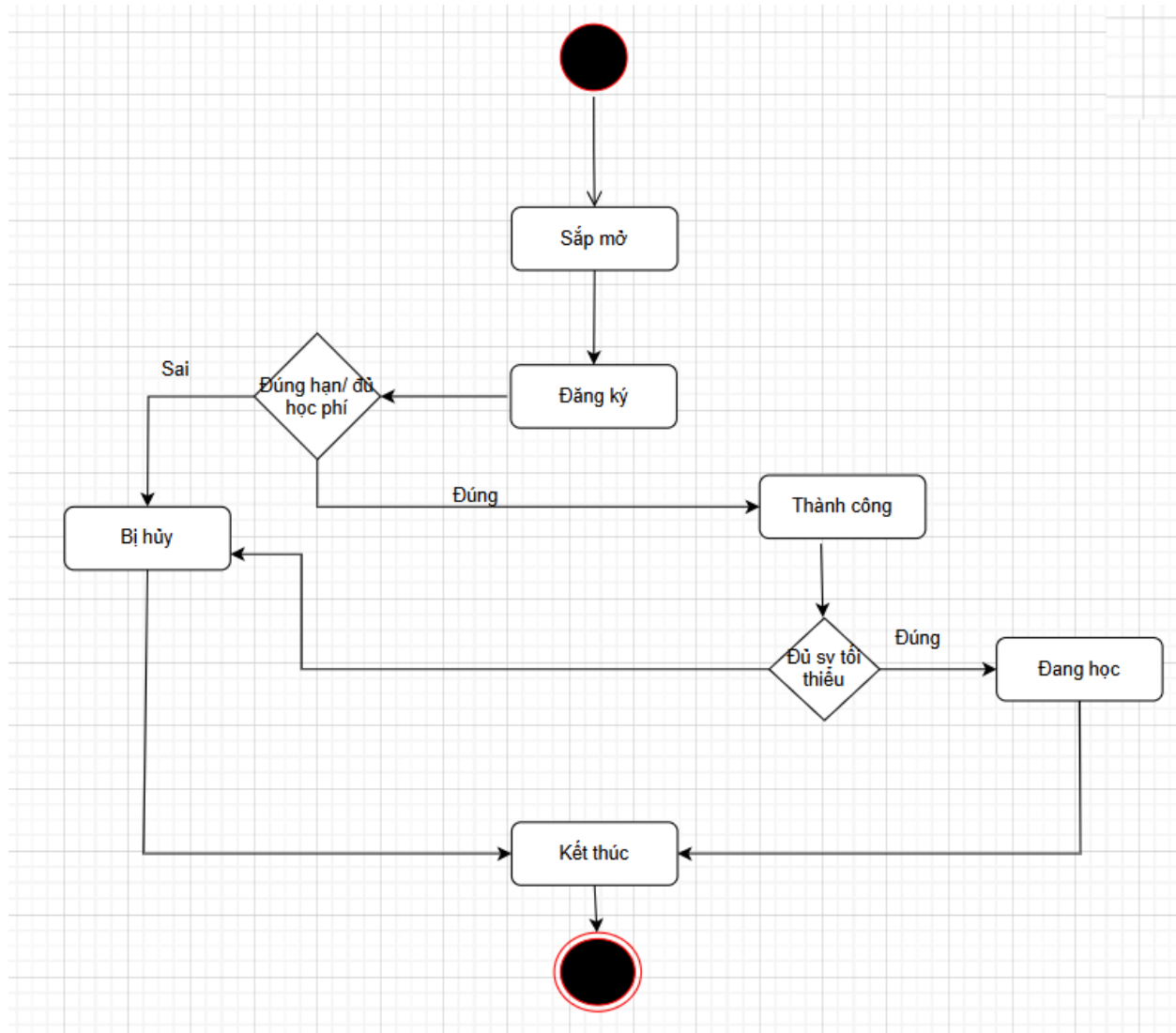
Kết quả trả về Backend → lưu vào bảng Attendance.

Frontend cập nhật danh sách sinh viên có mặt theo thời gian thực.

- Ý nghĩa:

Cho thấy hệ thống hoạt động tự động, liên tục và khép kín

### 2.3.2 Activity Diagram (Luồng đăng ký học)



1. Sinh viên đăng nhập → chọn môn học mong muốn.
2. Hệ thống kiểm tra điều kiện (môn tiên quyết, lịch trùng, số lượng tối đa).
3. Nếu hợp lệ → ghi nhận vào bảng **enrollments**.
4. Sinh viên xem lại danh sách lớp đã đăng ký.

#### Ý nghĩa:

Đảm bảo logic “ràng buộc dữ liệu” và tính chính xác trong quá trình học vụ.

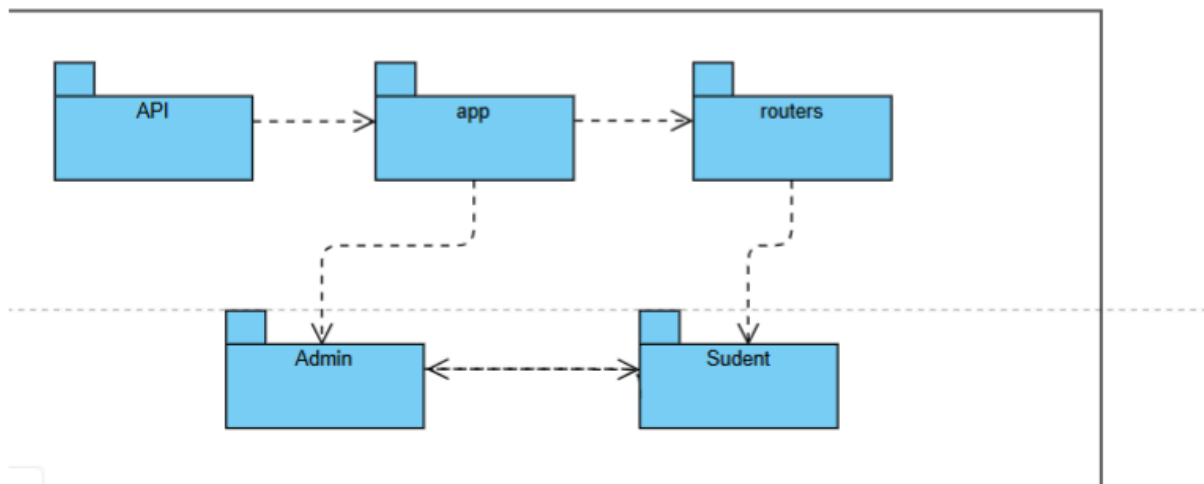
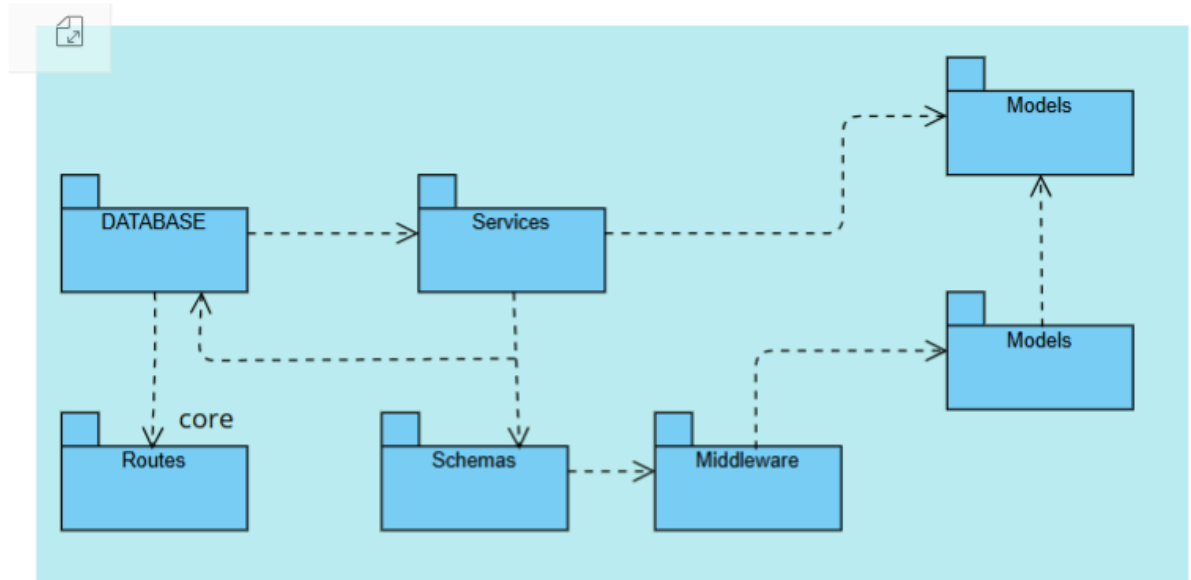
#### 2.4. Thiết kế giao diện

<https://www.figma.com/design/vUdY9ou3E42lFqP1vElGNK/PTTK-PM?node-id=229-98&t=fXDFHse2gZpaKQ2e-0>

### CHƯƠNG III: Thiết Kế Hệ Thống

### 3.1 Thiết Kế Kiến Trúc Hệ Thống (N-Tier/Microservices)

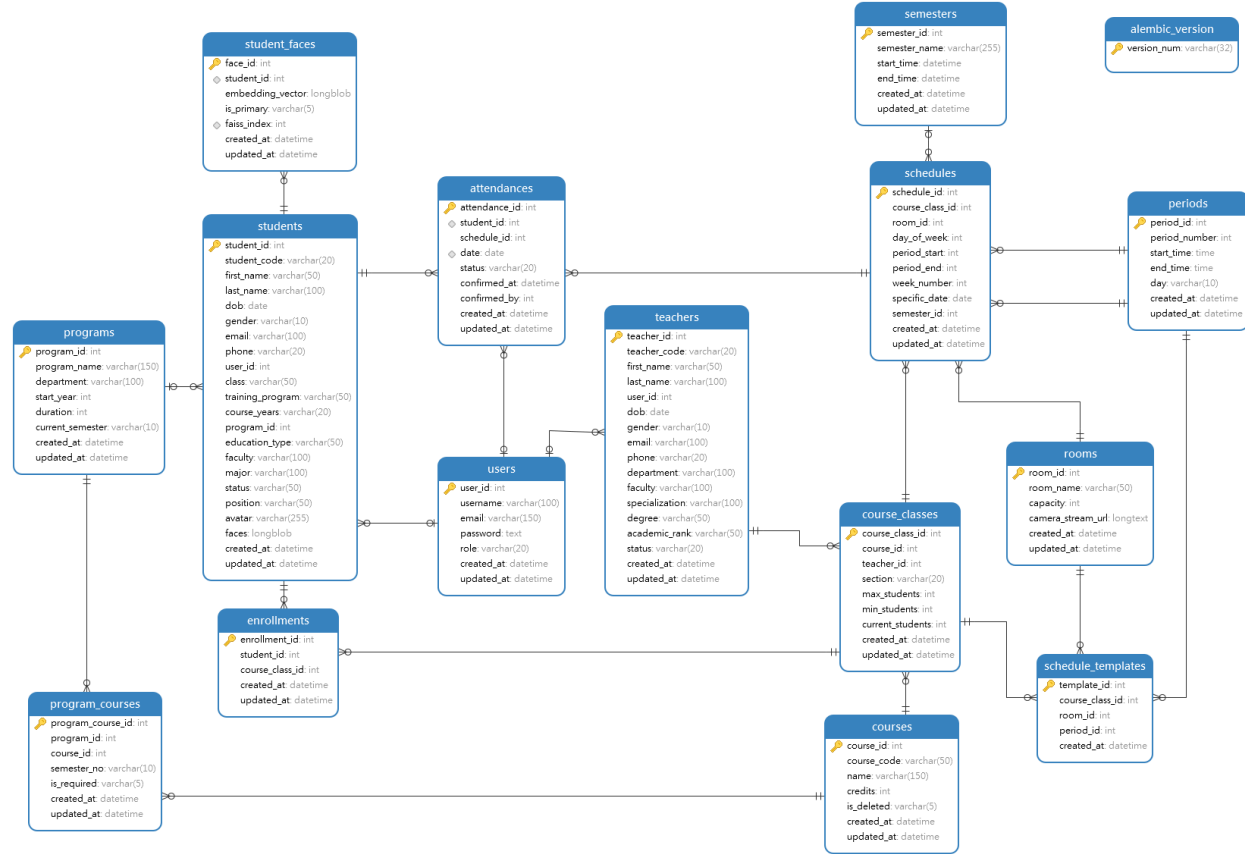
- Hệ thống được thiết kế dựa trên kiến trúc N-Tier Architecture (Kiến trúc đa tầng) kết hợp với phương pháp tiếp cận Microservices Approach. Điều này mang lại sự linh hoạt, khả năng mở rộng và dễ dàng bảo trì.
- Lý do lựa chọn:
  - Phân tách rõ ràng: Mỗi tầng có trách nhiệm riêng biệt, giúp code dễ hiểu, dễ phát triển và giảm thiểu sự phụ thuộc.
  - Mở rộng độc lập: Các microservice, đặc biệt là Face Recognition Service, có thể được triển khai và mở rộng hiệu năng độc lập (ví dụ: sử dụng GPU acceleration) mà không ảnh hưởng đến các dịch vụ khác.
  - Linh hoạt về công nghệ: Cho phép sử dụng các công nghệ tối ưu nhất cho từng phần (ví dụ: Python cho AI, JavaScript/Vue.js cho Frontend).
- Các thành phần chính trong kiến trúc:
  - Presentation Layer (Frontend): Xây dựng bằng Nuxt.js (Vue 3), chịu trách nhiệm hiển thị giao diện người dùng và tương tác với người dùng.
  - Application Layer (Backend API): Xây dựng bằng FastAPI (Python), chứa toàn bộ logic nghiệp vụ, quản lý phiên, xác thực và ủy quyền.
  - Data/Service Layer (PostgreSQL & Face Recognition Service):
    - PostgreSQL: Cơ sở dữ liệu quan hệ chính, lưu trữ thông tin về người dùng, khóa học, lịch học, và lịch sử điểm danh.
    - Face Recognition Service: Một microservice chuyên biệt, phát triển bằng Python, chịu trách nhiệm xử lý các tác vụ liên quan đến nhận diện khuôn mặt.
- Sơ đồ Kiến trúc Tổng thể:



- o Giải thích: Sơ đồ này minh họa cách các tầng và dịch vụ tương tác với nhau. Người dùng tương tác với Frontend (Nuxt.js). Frontend gửi yêu cầu qua API Gateway (hoặc Load Balancer) đến Backend API (FastAPI). Backend API tương tác với PostgreSQL để lưu trữ/truy xuất dữ liệu và gọi đến Face Recognition Service khi cần thực hiện điểm danh.

## 3.2 Thiết Kế Chi Tiết UML và Cơ Sở Dữ Liệu

### 3.2.1 Thiết kế cơ sở dữ liệu

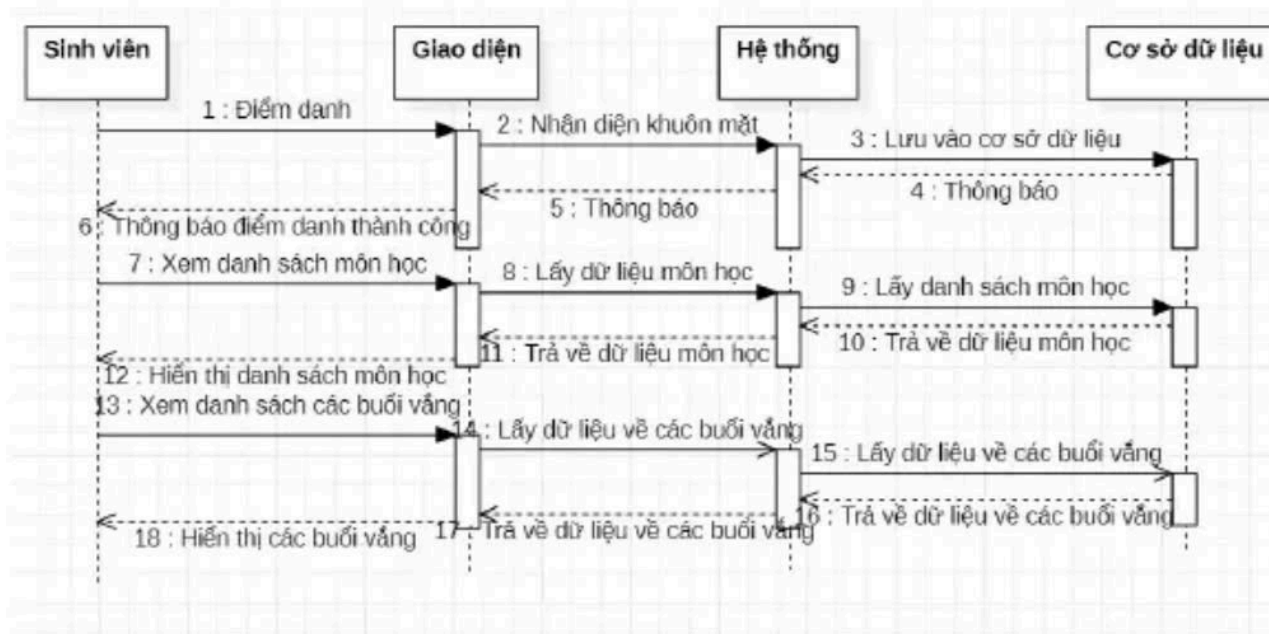


#### Danh sách bảng và mô tả:

- **users** – Lưu thông tin tài khoản đăng nhập, gồm email, mật khẩu, vai trò.
  - Khóa chính: `user_id`.
  - Dùng để xác thực và phân quyền.
- **students** – Thông tin sinh viên, liên kết với tài khoản người dùng và chương trình đào tạo.
  - Khóa chính: `student_id`.
  - Khóa ngoại: `user_id`, `program_id`.
- **teachers** – Thông tin giảng viên, liên kết với bảng `users`.
  - Khóa chính: `teacher_id`.
  - Khóa ngoại: `user_id`.
- **programs** – Quản lý các chương trình đào tạo.
  - Khóa chính: `program_id`.
- **courses** – Danh sách môn học, số tín chỉ, mô tả.
  - Khóa chính: `course_id`.
- **program\_courses** – Bảng trung gian thể hiện quan hệ n-n giữa chương trình đào tạo và môn học.
  - Khóa chính: (`program_id`, `course_id`).

- **course\_classes** – Đại diện cho một lớp học phần cụ thể, có môn học và giảng viên phụ trách.
  - Khóa chính: course\_class\_id.
  - Khóa ngoại: course\_id, teacher\_id.
  - **enrollments** – Ghi nhận sinh viên đăng ký lớp học phần.
  - Khóa chính: enrollment\_id.
  - Khóa ngoại: student\_id, course\_class\_id.
  - **rooms** – Quản lý phòng học và thiết bị nhận diện.
  - Khóa chính: room\_id.
  - **periods** – Định nghĩa các ca học (thời gian bắt đầu, kết thúc).
  - Khóa chính: period\_id.
  - **semesters** – Quản lý học kỳ và năm học.
  - Khóa chính: semester\_id.
  - **schedules** – Lịch học chi tiết, liên kết lớp học phần, phòng, ca học, học kỳ.
  - Khóa chính: schedule\_id.
  - Khóa ngoại: course\_class\_id, room\_id, period\_id, semester\_id.
  - **schedule\_templates** – Mẫu thời khóa biểu phục vụ việc tạo tự động.
  - Khóa chính: template\_id.
  - **student\_faces** – Lưu dữ liệu khuôn mặt của sinh viên cho nhận diện.
  - Khóa chính: face\_id.
  - Khóa ngoại: student\_id.
  - **attendances** – Trạng thái điểm danh của sinh viên (present, absent, late).
  - Khóa chính: attendance\_id.
  - Khóa ngoại: student\_id, schedule\_id.
- Thiết kế này giúp dữ liệu thống nhất, dễ truy vấn theo học kỳ, lớp học, sinh viên hoặc giảng viên. Ngoài ra còn tạo nền tảng cho phân tích và thống kê.

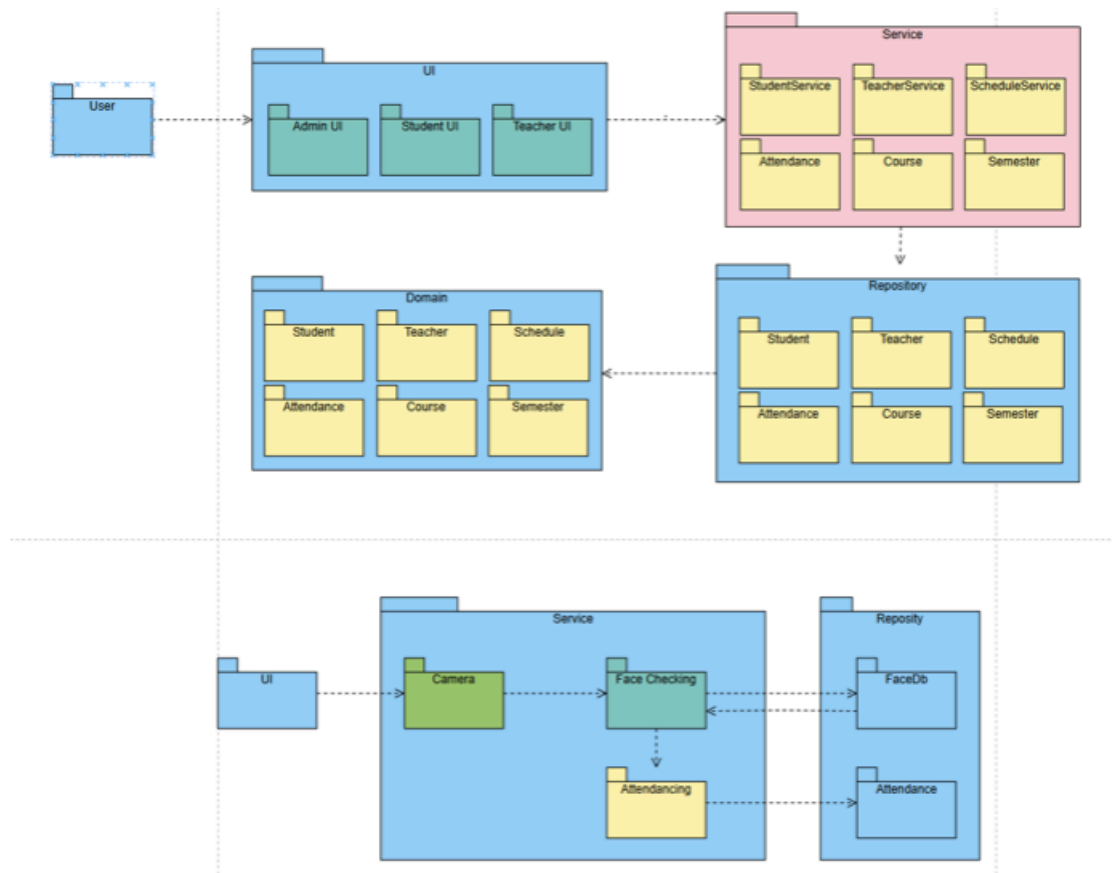
### 3.2.2 Sequence Diagram - Quy trình hoạt động của dự án



- Giải thích: Sơ đồ này mô tả trình tự tương tác chi tiết giữa các đối tượng trong quy trình điểm danh tự động.
  1. Giáo viên kích hoạt điểm danh trên Frontend.
  2. Frontend gửi yêu cầu StartAttendance đến Backend API.
  3. Backend API khởi tạo một phiên điểm danh, có thể thông báo cho Face Service sẵn sàng.
  4. Camera liên tục chụp ảnh và gửi về Face Service.
  5. Face Service xử lý ảnh, trích xuất đặc trưng khuôn mặt (face embeddings), và so sánh với dữ liệu khuôn mặt đã đăng ký trong Database.
  6. Face Service gửi kết quả nhận diện (ví dụ: student\_id và độ tin cậy) đến Backend API.
  7. Backend API ghi nhận thông tin điểm danh vào bảng Attendance trong Database.
  8. Backend API gửi kết quả cập nhật về Frontend để hiển thị (ví dụ: danh sách sinh viên có mặt).

### 3.3 Thiết Kế Module Hệ Thống (Package Diagram)

- Việc phân chia thành các gói (Package) giúp quản lý code dễ dàng hơn, đặc biệt trong một hệ thống có nhiều chức năng và dịch vụ.
- #### 3.3.1 Package Diagram



- Giải thích: Sơ đồ này minh họa cách hệ thống được tổ chức thành các module (packages) logic độc lập, chia hệ thống thành các tầng: ui (giao diện), service (logic nghiệp vụ), repository (truy cập dữ liệu), domain (các lớp đối tượng).

## CHƯƠNG IV: TRIỂN KHAI VÀ KIỂM THỬ

### 4.1 Chi Tiết Công Nghệ và Lý do Lựa chọn

- Backend (FastAPI/Python):
  - Lợi thế: Hiệu năng đồng bộ/bất đồng bộ (async/await) cao, cộng đồng lớn về AI/ML (giúp tích hợp dịch vụ nhận diện dễ dàng).
  - Bảo mật: Sử dụng JWT với Refresh Tokens để đảm bảo tính an toàn cho phiên người dùng.
- Database (PostgreSQL):
  - Lý do: Độ tin cậy cao, hỗ trợ các kiểu dữ liệu phức tạp (như mảng hoặc JSONB), và khả năng mở rộng tốt. Sử dụng SQLAlchemy ORM để chuẩn hóa thao tác CSDL.
- AI/ML (OpenCV/dlib):
  - Thư viện: dlib được sử dụng để phát hiện khuôn mặt và trích xuất face embeddings (vector đặc trưng khuôn mặt). OpenCV được dùng để xử lý hình ảnh đầu vào từ camera.

### 4.2 Chi Tiết Kịch Bản Kiểm Thử (Testing Scenarios)

- Chúng tôi tập trung vào 3 loại kiểm thử chính để đảm bảo chất lượng:

| Loại Kiểm Thử              | Mục tiêu                                   | Kịch bản chi tiết   | Kết quả mong đợi  |
|----------------------------|--|---|---|
| Kiểm thử AI (Độ chính xác) | Đánh giá độ tin cậy của mô hình nhận diện. | 1. Thử nghiệm với các góc độ khuôn mặt khác nhau (trái/phải). 2. Thay đổi điều kiện ánh sáng (tối/sáng quá mức). 3. Che mặt một phần (ví dụ: khẩu trang). | Độ chính xác trên 90%, thời gian xử lý < 5 giây.                  |
| Kiểm thử Hiệu năng API     | Đánh giá khả năng chịu tải của API chính.  | Sử dụng công cụ (ví dụ: Apache JMeter) để tạo 100 request/giây đến API điểm danh.   | API phản hồi ổn định, độ trễ không tăng quá 20% so với tải thấp.  |
| Kiểm thử Chức năng (Admin) | Đảm bảo tính đúng đắn của logic nghiệp vụ. | Admin tạo mới một lớp học và phân lịch dạy cho giáo viên (test phân quyền).   | Dữ liệu được lưu chính xác, giáo viên chỉ thấy lịch dạy của mình. |

## CHƯƠNG 5: TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN

### 5.1 Đánh Giá Dự Án

- Dự án đã hoàn thành các mục tiêu đề ra về mặt chức năng. Điểm mạnh lớn nhất là sự thành công trong việc tích hợp Face Recognition Service vào luồng nghiệp vụ quản lý sinh viên. Hạn chế là cần tối ưu hóa hơn nữa về hiệu năng xử lý của AI trong điều kiện thiếu sáng và cần xây dựng Unit Tests toàn diện.

### 5.2 Hướng Phát Triển Tương Lai

1. Tối ưu hóa AI: Nâng cấp Face Service để hỗ trợ GPU acceleration nhằm giảm thời gian nhận diện xuống dưới 1 giây.
2. Mở rộng tính năng: Thêm hệ thống thông báo tự động (Notification) cho sinh viên vắng mặt và giao diện Thống kê chuyên sâu cho Admin.
3. Triển khai: Chuyển sang mô hình triển khai Containerization (Docker/Kubernetes) để dễ dàng quản lý và mở rộng cho nhiều trường học cùng lúc (đa tenant).

## CHƯƠNG VI. KIẾN NGHỊ

Trong quá trình thực hiện đề tài, nhóm nhận thấy rằng việc triển khai một hệ thống hoàn chỉnh đòi hỏi nhiều thời gian, kiến thức thực tế và sự hướng dẫn chuyên sâu. Do đó, nhóm xin đưa ra một số kiến nghị:

- Mong giảng viên tiếp tục hỗ trợ, góp ý và định hướng để nhóm có thể hoàn thiện sản phẩm ở các phiên bản tiếp theo.

- Đề nghị nhà trường và bộ môn tạo thêm các buổi thực hành hoặc chia sẻ từ doanh nghiệp thực tế để sinh viên có điều kiện cọ xát với nghiệp vụ và công nghệ hiện đại.