



## AESP - Nền tảng luyện nói tiếng Anh AI

### Đồ án môn Xây dựng Phần mềm Hướng Đối Tượng

**Tên đề tài:** Nền tảng luyện nói tiếng Anh có sự hỗ trợ của AI

**Môn học:** Xây dựng Phần mềm Hướng Đối Tượng

**Giảng viên hướng dẫn:** ThS Nguyễn Văn Chiến

**Thành viên nhóm:**

- Phạm Ngọc Minh Anh - 074304005852
- Tô Thị Minh Thư - 075305015483
- Vũ Thị Mỹ Uyên - 075305002880
- Hoàng Phước Trọng - 077205005324
- Hà Duy Khang - 052205002232

### Mục lục

#### 1. Giới thiệu chung

##### 1.1 Bối cảnh

##### 1.2 Mục tiêu

##### 1.3 Phạm vi dự án

##### 1.4 Công nghệ sử dụng

#### 2. Phân tích yêu cầu hệ thống

##### 2.1 Sơ đồ kiến trúc Microservice

##### 2.2 Phân tích chức năng



### **3. Thiết kế hệ thống**

3.1 Class Diagram

3.2 Activity Diagram

3.3 Sequence Diagram

3.4 Thiết kế API

3.5 Thiết kế Database

### **4. Giao diện**

4.1 Giao diện người dùng

4.2 Giao diện admin

4.3 Giao diện người hướng dẫn

### **5. Đánh giá và kiểm thử**

5.1 Môi trường và công cụ kiểm thử

5.2 Chiến lược kiểm thử

5.3 Kết quả kiểm thử chức năng (Test Cases)

5.4 Đánh giá hiệu năng và bảo mật

5.5 Kết luận chung về kiểm thử

### **6. Kết luận và hướng phát triển**

6.1 Kết quả đạt được

6.2 Hạn chế của hệ thống

6.3 Hướng phát triển tương lai

## **1. Giới thiệu chung**

### **1.1 Bối cảnh**

AESP (AI-supported English Speaking Practice Platform) là nền tảng web hỗ trợ luyện nói tiếng Anh với sự trợ giúp của trí tuệ nhân tạo (AI), nhằm giúp người học tại Việt Nam cải thiện kỹ năng nói, phát âm và sự tự tin trong giao tiếp.

trong thực tập, công việc và các lĩnh vực như kinh doanh, ưu lợi. Tuy

nhiên, nhiều người học gặp rào cản tâm lý như sợ sai, ngại nói, thiếu môi trường luyện tập và thiếu bạn giao tiếp. Phương pháp học truyền thống thường chú trọng ngữ pháp và đọc hiểu, dẫn đến kỹ năng nói chưa được phát triển hiệu quả.

AESP đề xuất giải pháp kết hợp AI và mentor, tạo môi trường luyện nói không phán xét, cá nhân hóa theo trình độ người học. Hệ thống đánh giá trình độ ban đầu, xây dựng lộ trình học phù hợp, cung cấp phản hồi phát âm, ngữ pháp theo thời gian thực và hỗ trợ nhiều tình huống giao tiếp thực tế như đời sống hằng ngày, du lịch và kinh doanh.

Hệ thống phục vụ ba nhóm người dùng chính:

- Admin: quản lý người dùng, mentor, gói dịch vụ, chính sách và báo cáo.
- Learner: luyện nói với AI/mentor, theo dõi tiến độ, tham gia thử thách và nhận báo cáo học tập.
- Mentor: đánh giá trình độ, hướng dẫn phát âm, ngữ pháp và kỹ năng giao tiếp thực tế.

AESP được xây dựng dưới dạng ứng dụng web gồm Web API, Admin Web, Learner Web và Mentor Web, sử dụng các công nghệ hiện đại như .NET/Spring Boot, React + TypeScript, PostgreSQL/MySQL và nền tảng đám mây.

Dự án giúp sinh viên áp dụng quy trình phát triển phần mềm, UML 2.0, thiết kế – triển khai – kiểm thử một hệ thống thực tế và chuẩn bị đầy đủ tài liệu cho quá trình bảo vệ.

## 1.2 Mục tiêu

Dự án AESP được thực hiện nhằm đạt các mục tiêu chính sau:

- Xây dựng một nền tảng luyện nói tiếng Anh có sự hỗ trợ của AI, giúp người học cải thiện phát âm, độ lưu loát và sự tự tin khi giao tiếp.
- Tạo môi trường luyện nói không phán xét, giảm rào cản tâm lý như sợ sai, ngại nói.
- Cá nhân hóa lộ trình học dựa trên đánh giá trình độ nói ban đầu và quá trình học tập của người dùng.
- Kết hợp AI và mentor để nâng cao hiệu quả học tập và tính thực tiễn.

Tiến dài để xây dựng một hệ thống huấn luyện, cung cấp thông tin.

### 1.3 Phạm vi của dự án

Phạm vi dự án AESP bao gồm:

- Phát triển ứng dụng web cho ba nhóm người dùng: Admin, Learner và Mentor.
- Cung cấp các chức năng chính:
  - Luyện nói tiếng Anh với AI và/hoặc mentor.
  - Đánh giá trình độ nói và phát âm.
  - Cá nhân hóa lộ trình học và nội dung luyện tập.
  - Quản lý gói dịch vụ, tài khoản và báo cáo thống kê.
- Hỗ trợ nhiều chủ đề và tình huống giao tiếp (đời sống, du lịch, kinh doanh).
- Xây dựng Web API, cơ sở dữ liệu và giao diện web tương ứng.
- Triển khai hệ thống ở mức demo/academic project, chưa bao gồm thương mại hóa quy mô lớn hay AI huấn luyện chuyên sâu.

### 1.4 Công nghệ sử dụng

Loại	Công cụ / Cơ sở hạ tầng
Technology	Python, HYML/CSS, React, Javascript
Database	MySQL
IDEs/Editors	Visual Studio Code
Diagramming	DrawIO, <a href="#">A Free Database Designer for Developers and Analysts</a>
Documentation	Confluence
Version Control	GitHub (Source Codes)
Deployment server	Docker
Project management	GitHub ( <a href="#">Projects</a> )



## 2.1 Sơ đồ kiến trúc Microservice

Hệ thống AESP (AI-Assisted English Speaking Practice Platform) được xây dựng theo Kiến trúc Microservices. Các dịch vụ được đóng gói bằng Docker, giao tiếp qua API Gateway (Nginx) và sử dụng hệ thống tin nhắn RabbitMQ để xử lý bất đồng bộ.

Thành phần	Vai trò	Công nghệ/Cổng	Database
<b>API Gateway</b>	Điểm truy cập duy nhất, định tuyến request, phục vụ Frontend (React, Admin, Mentor).	Nginx / Port 80.	N/A
<b>User Service</b>	Quản lý người dùng (Admin, Mentor, Learner), xác thực & phân quyền (JWT), quản lý hồ sơ người dùng.	Flask / Nội bộ (Port 5000)	<b>user_db</b> (MySQL)
<b>subscription-service</b>	Quản lý gói dịch vụ, đăng ký gói, trạng thái subscription	Flask / Port 5001	<b>subscription_db</b> (MySQL)
<b>payment-service</b>	Xử lý giao dịch thanh toán và nâng cấp gói.	Flask / Port 5004	<b>payment_db</b> (MySQL)
<b>mentor-management-service</b>	Quản lý Mentor, kỹ năng Mentor, lịch hướng dẫn, đánh giá Learner	Flask / Port 5002	<b>user_db</b> (MySQL)L

	giữa Learner – Mentor / Learner – Learner, hỗ trợ AI		(MySQL)	
<b>AI-core-service</b>	Xử lý AI (STT, chấm điểm) tích hợp Groq API.	Flask / Port 5005	xdpm (MySQL)	
<b>analytics-service</b>	Thống kê, phân tích tiến độ học tập, báo cáo & dashboard cho Admin	Flask / Port 5003	analytics_db (MySQL)	
<b>RabbitMQ</b>	Message Broker hỗ trợ truyền tin giữa các service.	RabbitMQ / Port 5672, 15672	N/A	
<b>User DB (MySQL)</b>	Hệ quản trị CSDL tập trung cho các service.	MySQL 8.0 / Port 3307	Đa database (như trên)	
<b>Apache NiFi</b>	Xử lý và luân chuyển luồng dữ liệu (Data Ingestion).	8443		
<b>phpMyAdmin</b>	Giao diện quản lý cơ sở dữ liệu MySQL.	8080		
<b>Prometheus</b>	Thu thập dữ liệu giám sát (Metrics) hệ thống.	9090		
<b>Grafana</b>	Trực quan hóa dữ liệu giám sát và Dashboard.	3000		

	giữa các Microservices (Event-driven).	5672	
--	--	------	--

## 2.2 Phân tích chức năng

Service	Chức năng chính được triển khai
<b>user-service</b>	<ul style="list-style-type: none"> <li>Quản lý tài khoản (Đăng ký/Đăng nhập, xác thực JWT).</li> <li>Quản lý hồ sơ người dùng (Profile) và phân quyền (Admin/Mentor/Learner).</li> <li>Cung cấp Internal API để các service khác truy vấn thông tin người dùng.</li> <li>Quản lý bảo mật và khóa/mở khóa tài khoản.</li> </ul>
<b>subscription-service</b>	<ul style="list-style-type: none"> <li>Quản lý các gói dịch vụ (Pricing plans).</li> <li>Xử lý đăng ký gói (Subscribe), nâng cấp hoặc gia hạn gói.</li> <li>Kiểm tra trạng thái và quyền lợi của người dùng dựa trên gói dịch vụ hiện tại.</li> <li>Lưu trữ lịch sử đăng ký (Subscription History).</li> </ul>
<b>payment-service</b>	<ul style="list-style-type: none"> <li>Tạo và quản lý giao dịch thanh toán.</li> <li>Cập nhật trạng thái thanh toán (Pending/Paid/Failed).</li> <li>Tích hợp Webhook để nhận phản hồi từ cổng thanh toán.</li> </ul>

	<p>service để cập nhật gói dịch vụ ngay sau khi thanh toán thành công.</p>	
<b>mentor-management-service</b>	<ul style="list-style-type: none"> <li>Quản lý danh sách Mentor và các kỹ năng chuyên môn (Skills).</li> <li>Đồng bộ và sử dụng chung cơ sở dữ liệu người dùng ( user_db ).</li> <li>Gửi/nhận các sự kiện liên quan đến Mentor thông qua <b>RabbitMQ</b>.</li> <li>Quản lý lịch hướng dẫn và đánh giá tương tác giữa Mentor và Learner.</li> </ul>	
<b>chat-service</b>	Tạo phòng chat (1-1 / nhóm), Gửi/nhận tin nhắn, Lưu lịch sử chat, Gửi file/voice (nếu có), Tích hợp AI gợi ý câu trả lời/dề xuất sửa lỗi (AI assist), Báo cáo vi phạm/Report message, Quản lý trạng thái online/offline (nếu có).	
<b>AI-core-service</b>	<ul style="list-style-type: none"> <li>Tích hợp <b>Groq API</b> để xử lý ngôn ngữ tự nhiên cấp độ cao.</li> <li>Chuyển đổi giọng nói thành văn bản (Speech-to-Text).</li> <li>Chấm điểm phát âm, phân tích lỗi ngữ pháp và gợi ý từ vựng.</li> <li>Cung cấp các kịch bản hội thoại thông minh dựa trên AI.</li> </ul>	
<b>analytics-service</b>	<ul style="list-style-type: none"> <li>Thu thập dữ liệu hoạt động của người dùng từ hệ thống tin nhắn <b>RabbitMQ</b>.</li> <li>Phân tích tiến độ học tập, tần suất luyện tập và điểm số trung bình.</li> </ul>	

	<p>Dashboard, Dau Tao Cho Admin và Learner.</p> <ul style="list-style-type: none"> <li>Xử lý dữ liệu lớn bất đồng bộ để không ảnh hưởng đến hiệu năng hệ thống.</li> </ul>
<b>api-gateway (Nginx)</b>	<ul style="list-style-type: none"> <li>Tiếp nhận yêu cầu từ Frontend và định tuyến đến các microservices tương ứng.</li> <li>Phục vụ static files cho các phân hệ: User Web (React), Admin Dashboard, và Mentor Portal.</li> <li>Đảm bảo an toàn hệ thống và cân bằng tải cơ bản.</li> </ul>
<b>support-service</b>	<p>Tạo ticket hỗ trợ (Support Request), Quản lý trạng thái ticket (Open/In-progress/Closed), Phản hồi giữa Admin–User, Quản lý FAQ/Policy, Moderate feedback/comments, Gửi thông báo hệ thống (Announcement), Lưu lịch sử hỗ trợ &amp; đánh giá chất lượng hỗ trợ.</p>

### 2.3 Cơ chế giao tiếp nội bộ

Hệ thống AESP sử dụng mô hình giao tiếp hỗn hợp (Hybrid Communication) để tối ưu hóa hiệu suất và tính sẵn sàng, bao gồm giao tiếp đồng bộ qua REST API và giao tiếp bất đồng bộ qua Message Broker (RabbitMQ).

#### 1. Giao tiếp đồng bộ (Synchronous) qua REST API

Sử dụng cho các luồng xử lý cần kết quả ngay lập tức.

- Giao thức:** HTTP/REST.
- Xác thực:** Các yêu cầu giữa các service được bảo mật bằng **JWT Secret Key** (được cấu hình chung là AESP\_Secret\_Key\_2026... trong biến môi trường).

request kèm theo token xác thực trong Header.

## 2. Giao tiếp bất đồng bộ (Asynchronous) qua RabbitMQ

Sử dụng cho các luồng xử lý tồn thời gian hoặc cần thông báo cho nhiều service mà không cần chờ phản hồi ngay (Event-driven).

- **Thành phần:** Sử dụng **RabbitMQ** (Cổng 5672) làm Broker trung gian.
- **Cơ chế:** \* Một service đẩy tin nhắn (Event/Task) vào hàng đợi (Queue).
  - Các service liên quan (như **analytics-service** hoặc **mentor-management-service**) sẽ lắng nghe (Subscribe) và xử lý dữ liệu khi rảnh.
- **Lợi ích:** Giảm độ trễ cho người dùng và tránh tình trạng treo hệ thống dây chuyền khi một service gặp sự cố.

## 3. Ví dụ các luồng giao tiếp thực tế trong code

Dựa trên cấu hình hệ thống, các luồng giao tiếp chính bao gồm:

### 1. **payment-service → user-service (REST API):**

- Khi thanh toán thành công, **payment-service** gọi endpoint **/internal/upgrade-package** của **user-service** để cập nhật quyền hạn người dùng ngay lập tức.

### 2. **user-service / mentor-service → RabbitMQ → analytics-service:**

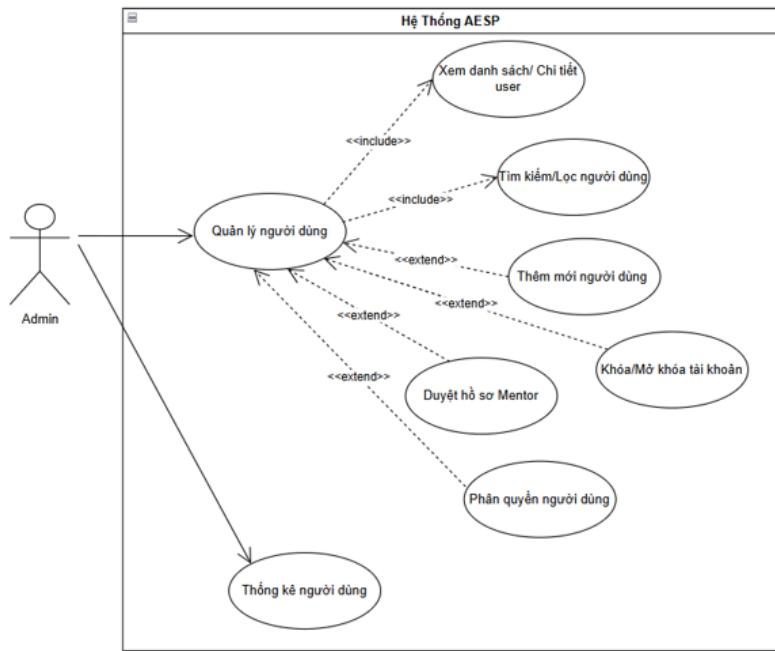
- Khi có người dùng mới hoặc Mentor cập nhật thông tin, một event được đẩy lên RabbitMQ. **analytics-service** nhận tin để cập nhật dữ liệu báo cáo và biểu đồ.

### 3. **ai-core-service (REST API):**

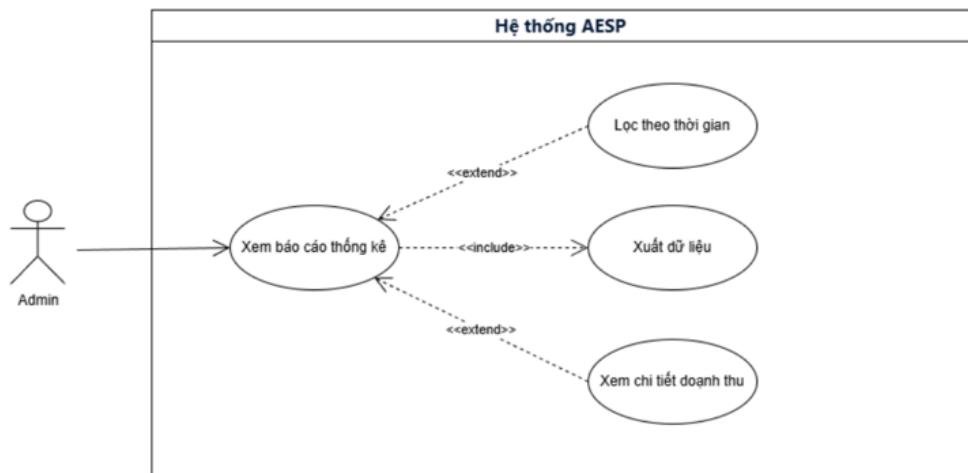
- Các service khác hoặc API Gateway gửi dữ liệu âm thanh/văn bản đến **ai-core-service** để xử lý thông qua Groq API và nhận kết quả chấm điểm trả về.

### 4. **Chia sẻ Cơ sở dữ liệu (Database Level):**

- Đặc biệt, **mentor-management-service** và **user-service** cùng kết nối chung tới **user\_db** để truy xuất nhanh thông tin hồ sơ và kỹ năng của Mentor.

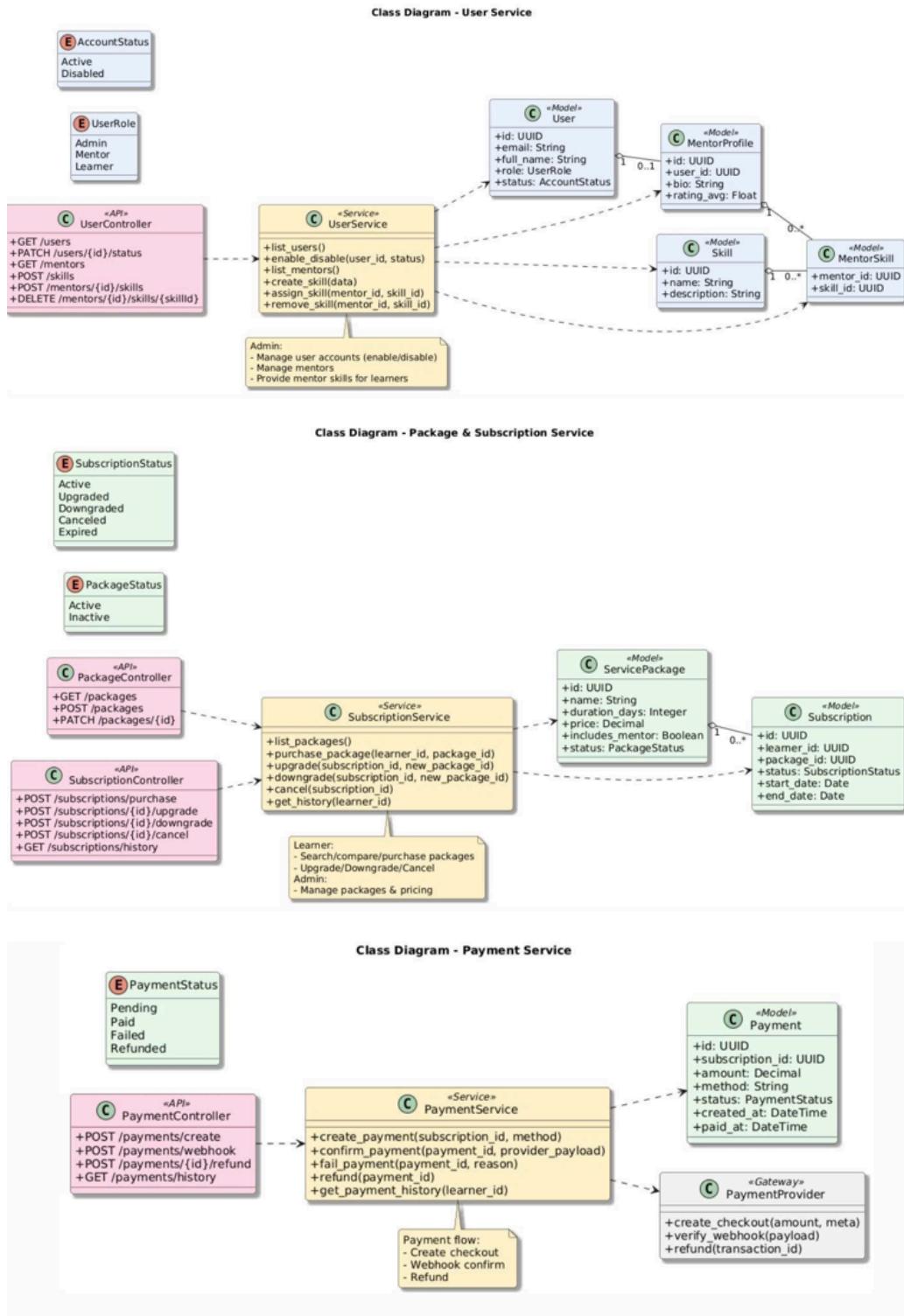


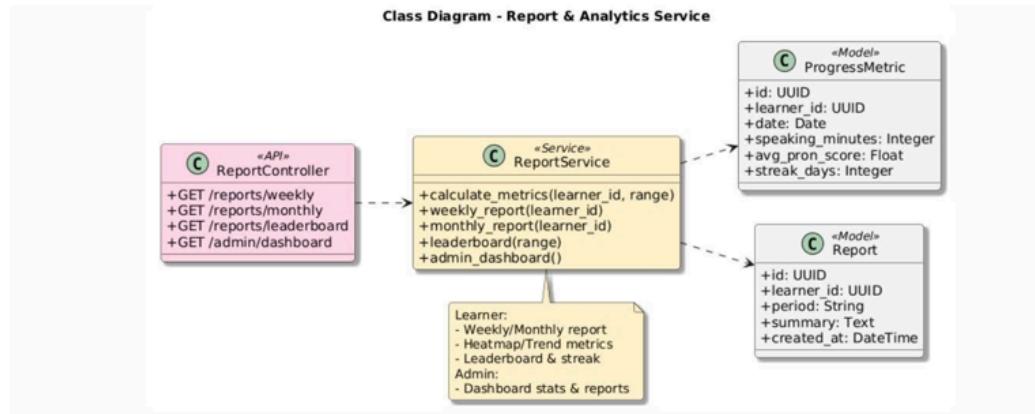
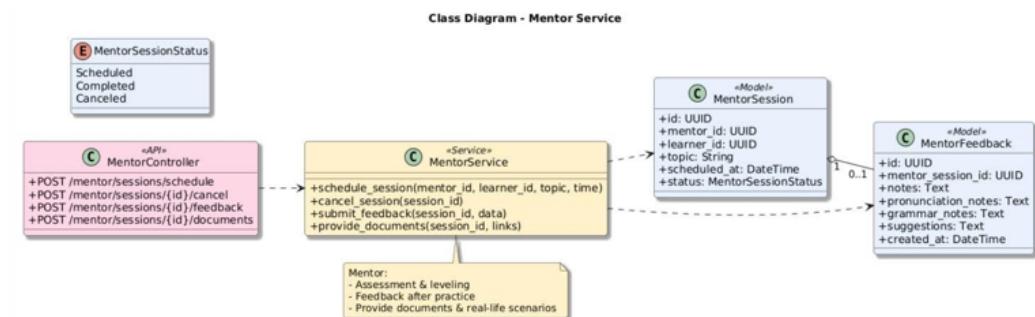
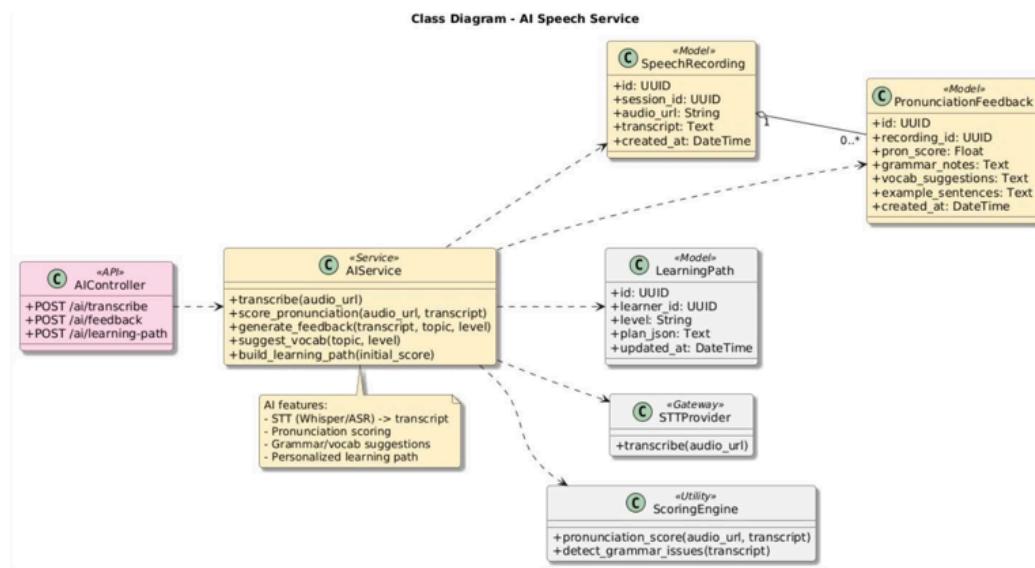
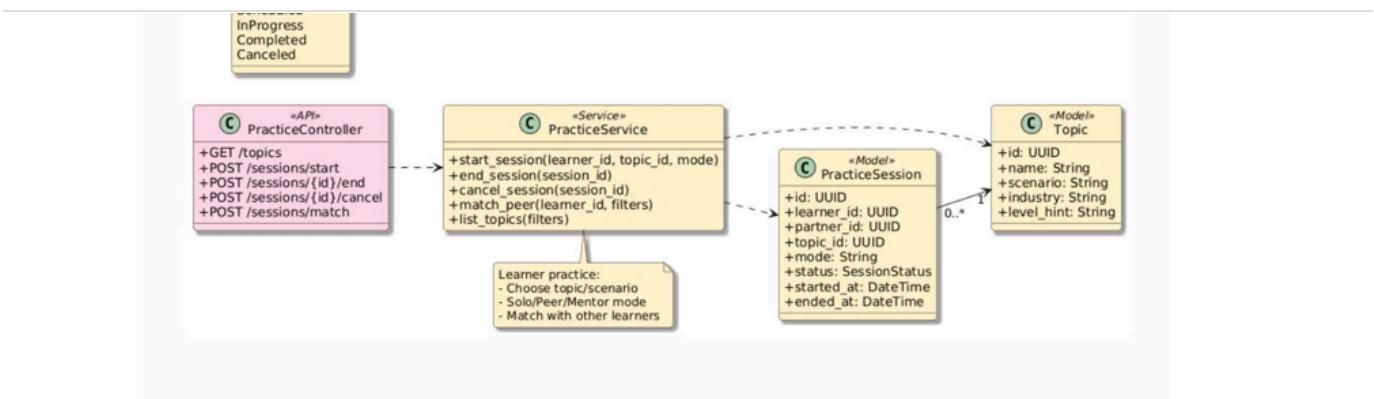
UC quản lý người dùng

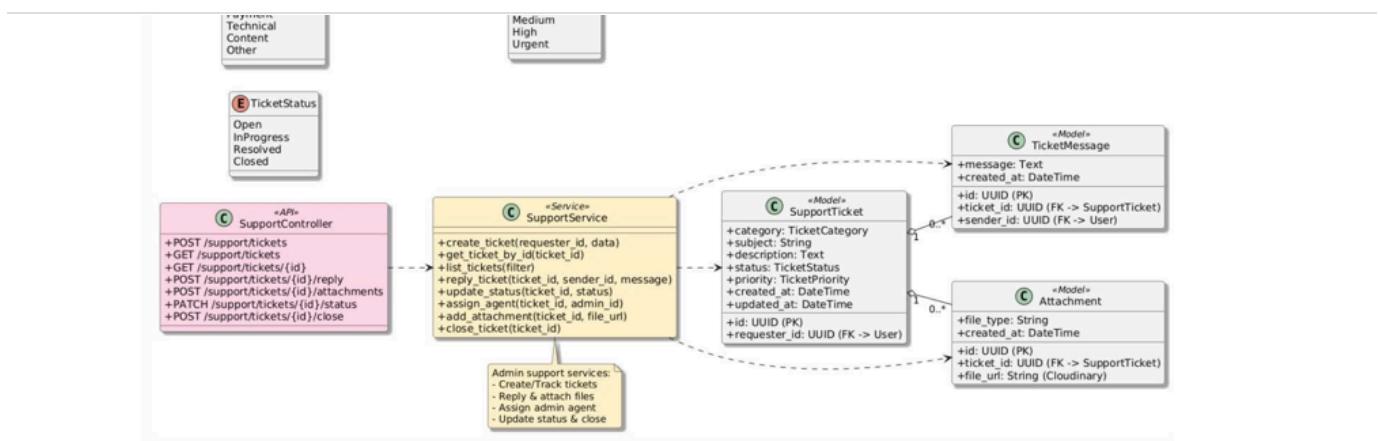


UC xem báo cáo thống kê

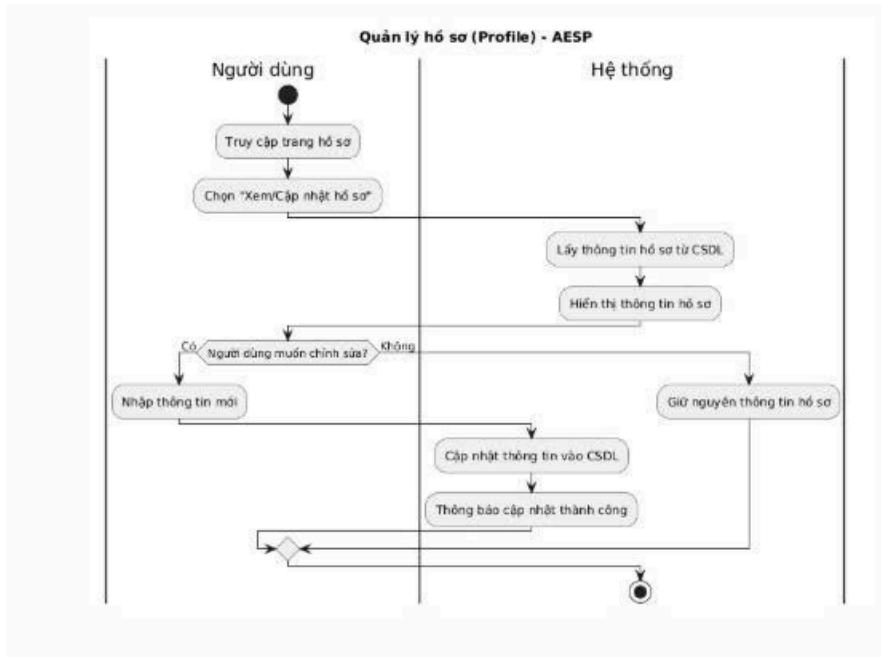
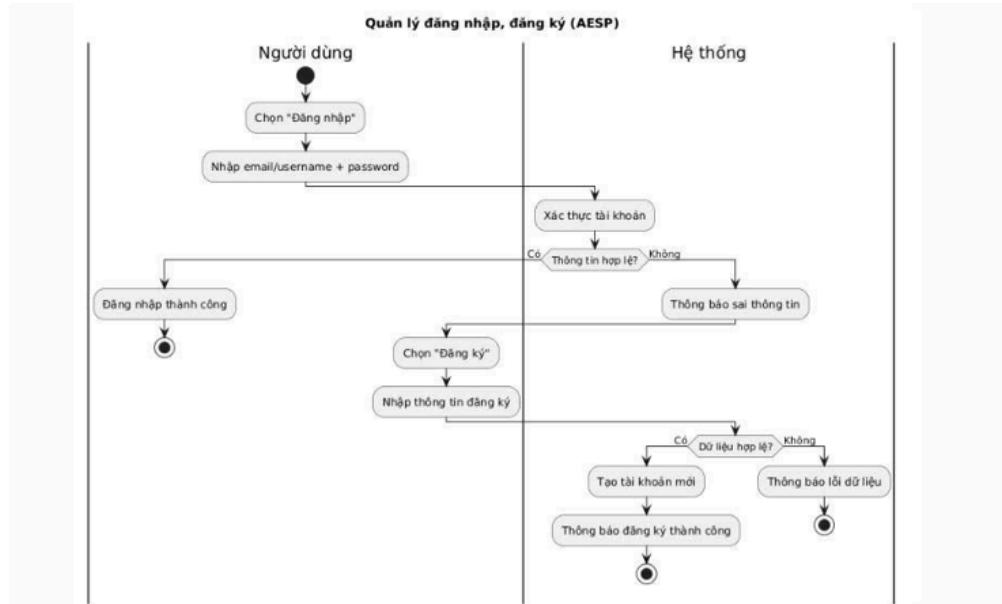
### 3.1 Class Diagram

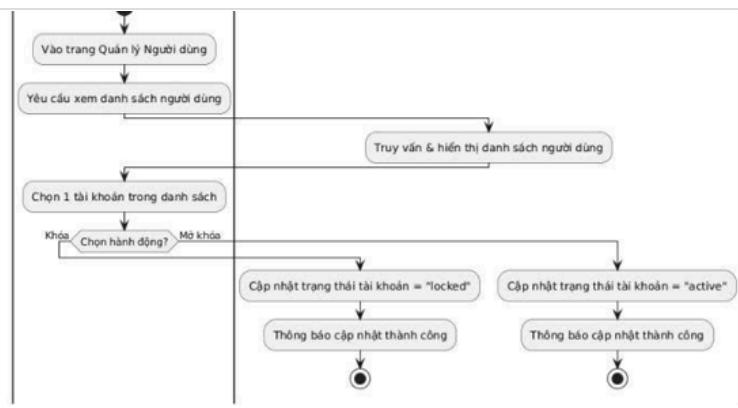
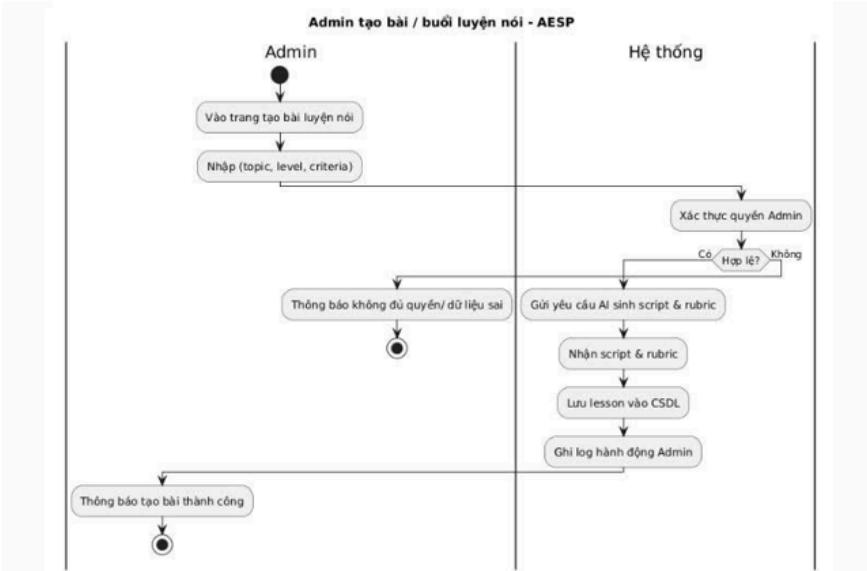
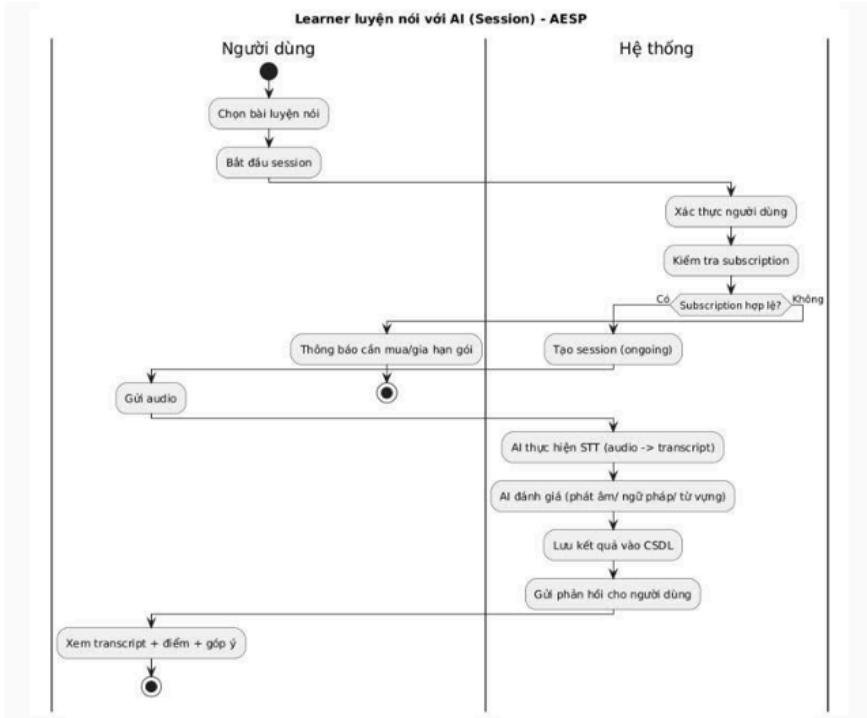


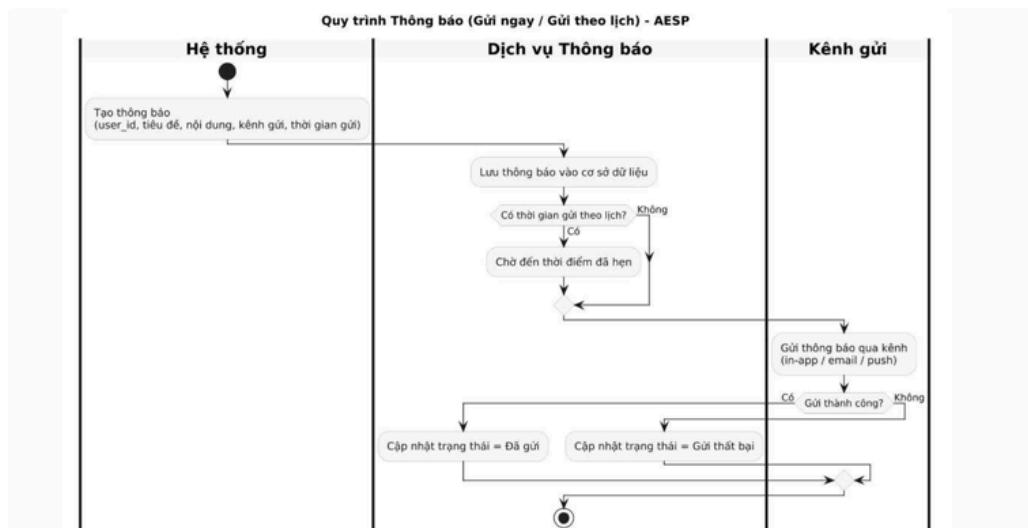
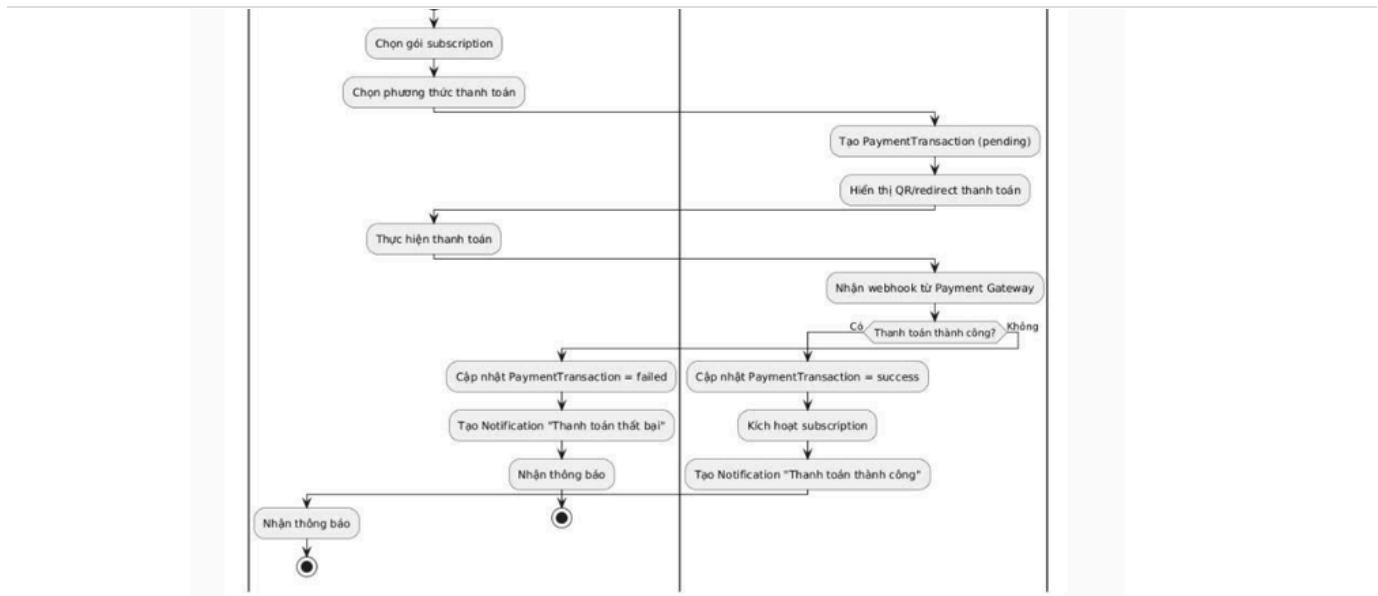




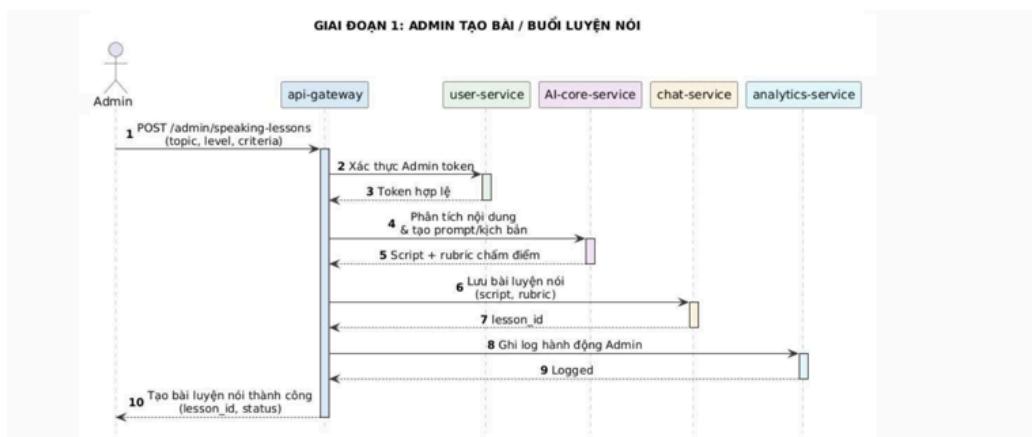
### 3.2 Activity Diagram

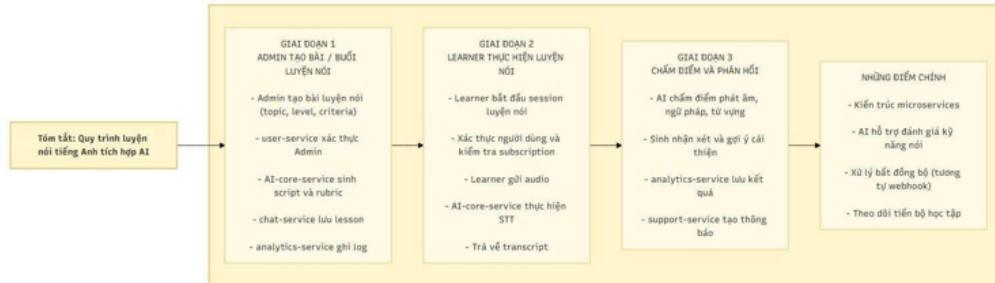
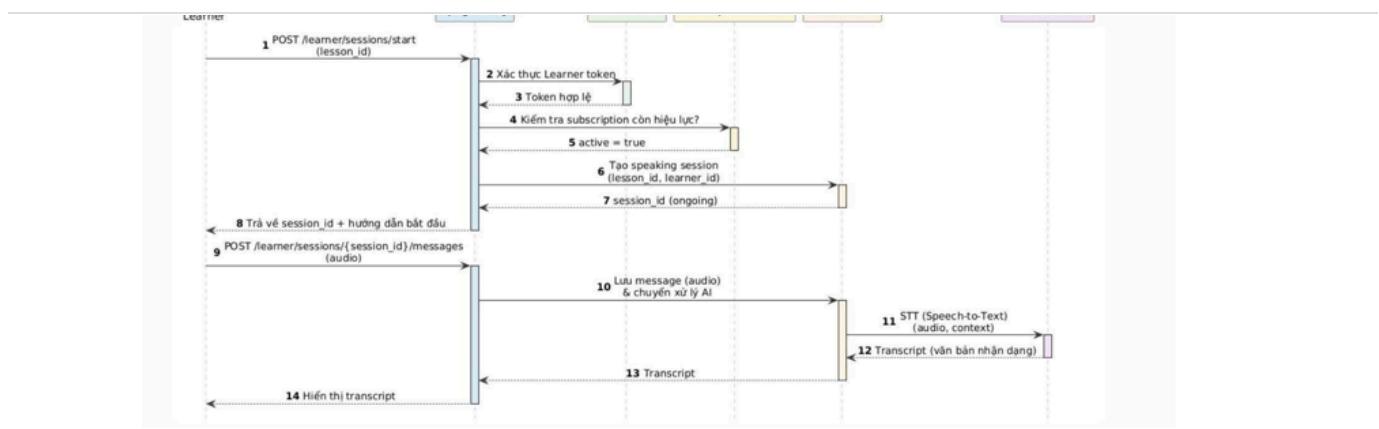


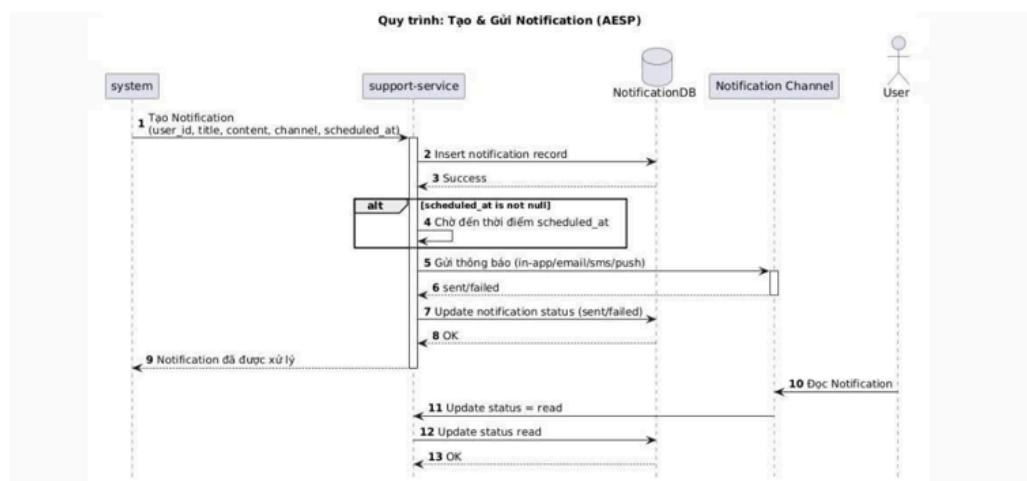
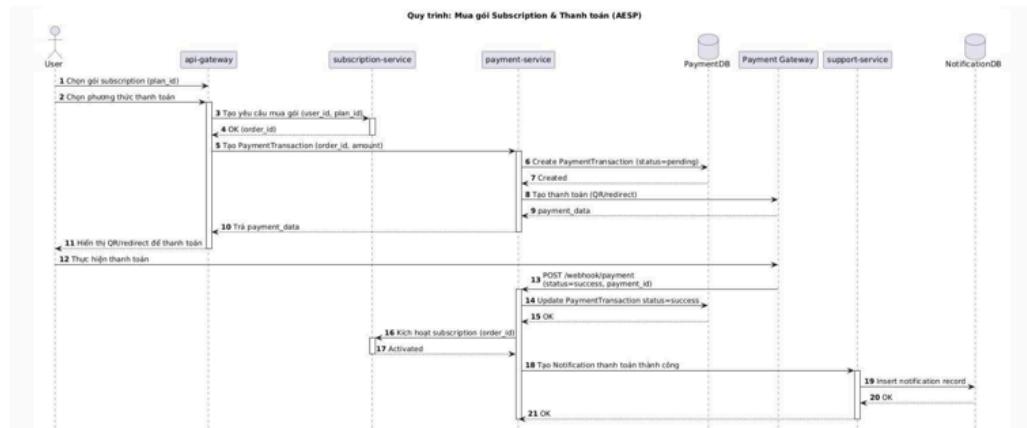
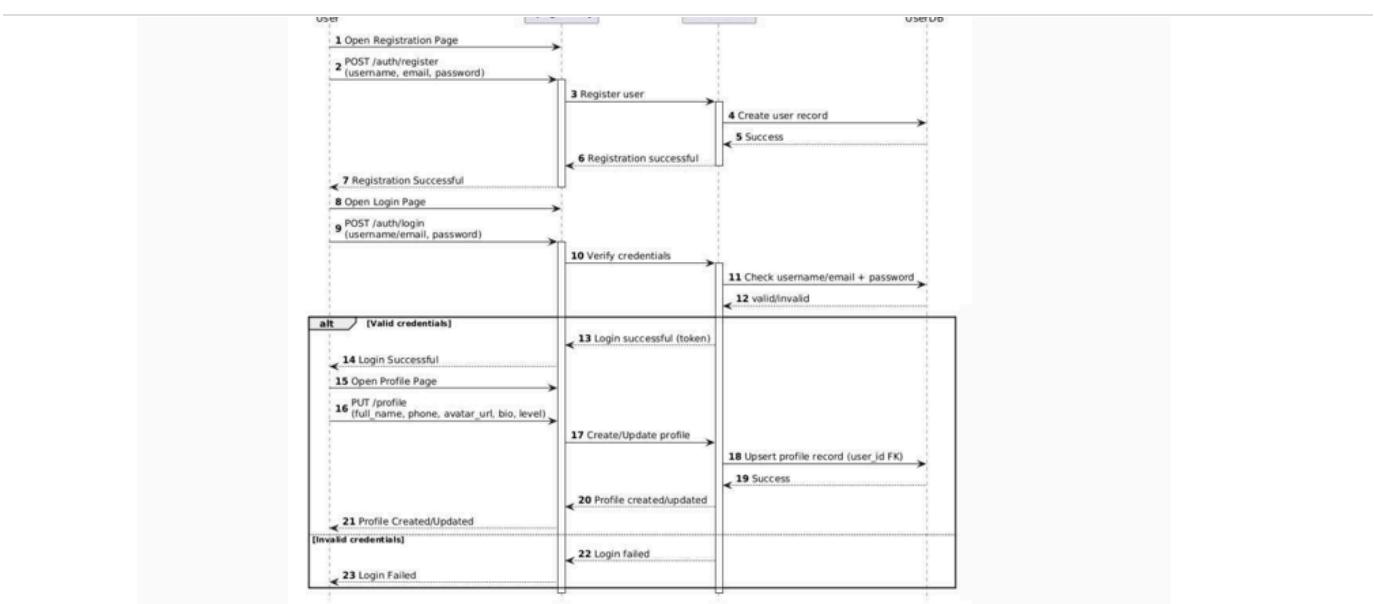
**Admin tạo bài / buổi luyện nói - AESP****Learner luyện nói với AI (Session) - AESP**

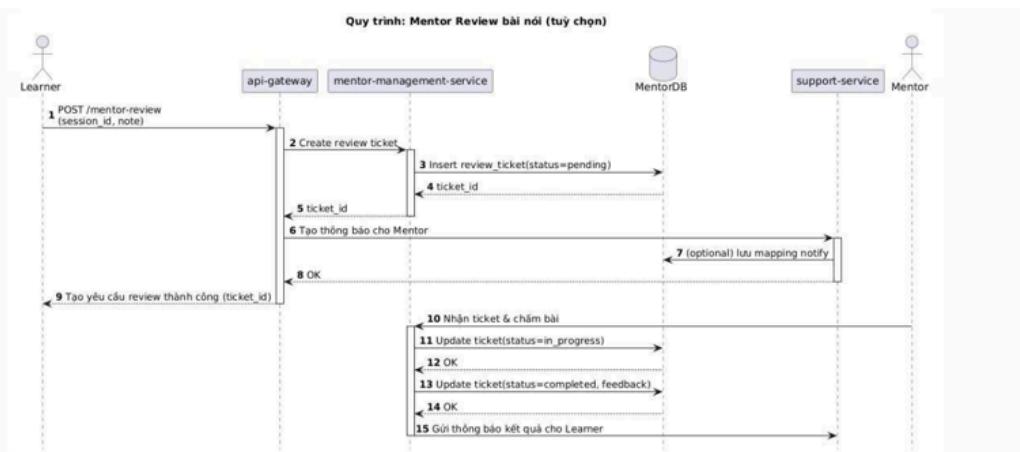
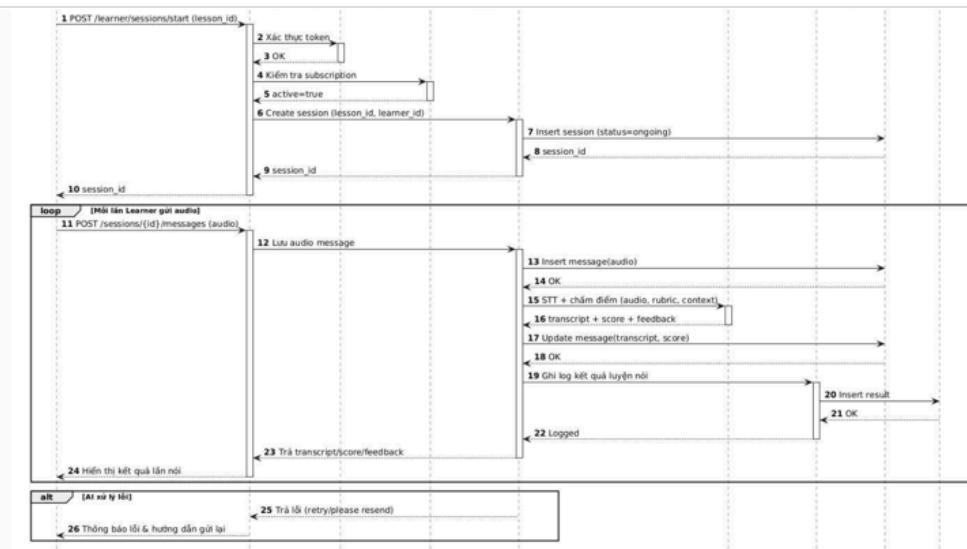


### 3.3 Sequence Diagram









### 3.4 Thiết kế API

#### Thiết kế API (API Documentation)

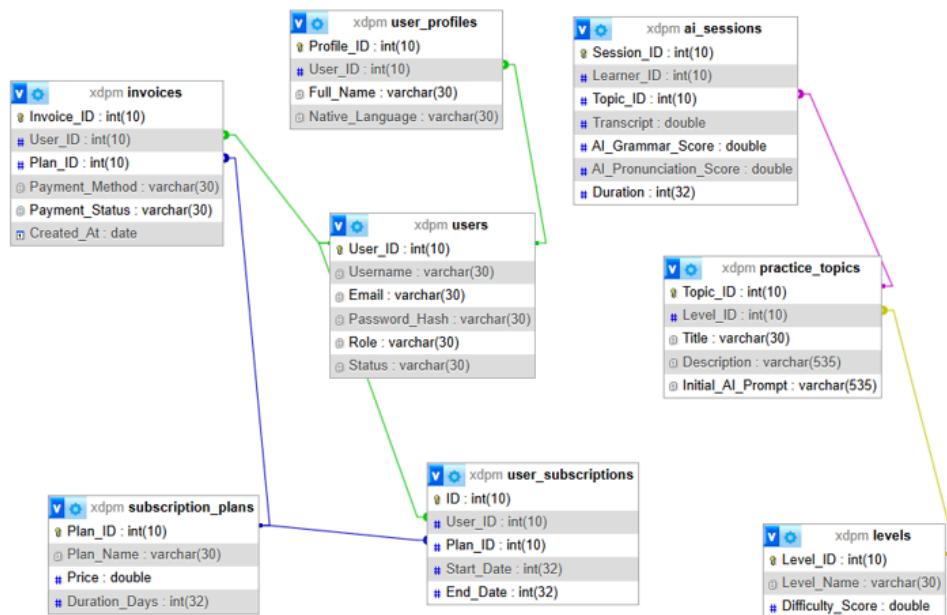
Hệ thống AESP triển khai thiết kế API theo kiến trúc Microservice, nơi mỗi dịch vụ quản lý một tập hợp các Endpoint riêng biệt. API Gateway (Nginx) đóng vai trò là điểm đầu nối duy nhất (Entry Point) tại cổng **80**, chịu trách nhiệm điều hướng request đến đúng service bên trong mạng nội bộ.

Service	Mô tả	Endpoint tiêu biểu	Role yêu cầu
<b>user-service</b>	Quản lý tài khoản, xác thực JWT và phân quyền hệ thống.	POST /api/auth/login	Public, Learner, Mentor, Admin

		/api/users /profile	
<b>subscription-service</b>	Quản lý thông tin các gói dịch vụ và đăng ký của người dùng.	GET  /api/subscriptions/palans  POST  /api/subscriptions/subscribe	Learner, Admin
<b>payment-service</b>	Xử lý tạo hóa đơn, thanh toán và nhận phản hồi từ cổng thanh toán.	POST  /api/payments  POST  /api/payments/webhook	Learner, Public (Webhook)
<b>AI-core-service</b>	Xử lý AI (Groq), chấm điểm phát âm và phân tích văn bản.	POST  /api/ai/stt-analyze  GET  /api/ai/recommendations	Learner
<b>chat-service</b>	Chat luyện nói & AI assist	POST  /api/chats/messages  GET	learner, mentor

		<code>/history</code>	
<b>mentor-management-service</b>	Quản lý danh sách mentor, kỹ năng và lịch hướng dẫn.	GET <code>/api/mentors/all</code>  POST <code>/api/mentors/feedback</code>	Learner, Mentor, Admin
<b>analytics-service</b>	Cung cấp dữ liệu thống kê tiến độ học tập và báo cáo hệ thống.	GET <code>/api/analytics/statistics</code>  GET <code>/api/analytics/dashboard</code>	Learner, Admin
<b>support-service</b>	Ticket & thông báo	POST <code>/api/support/tickets</code>	learner, admin

## Mô hình cấu trúc dữ liệu



## Thiết kế CSDL

Hệ thống tuân thủ mô hình **Database per Service** để đảm bảo tính độc lập. Tuy nhiên, theo cấu hình thực tế, **user-service** và **mentor-management-service** đang chia sẻ chung cơ sở dữ liệu **user\_db** để tối ưu hóa việc truy xuất dữ liệu định danh.

Service	Database Name	Bảng chính	Các trường quan trọng
<b>User &amp; Mentor Service</b>	user_db	<b>users</b> <b>profiles</b> <b>mentors</b>	<b>user_id</b> (PK), <b>email</b> , <b>password_h</b> <b>ash</b> , <b>role</b> (ADMIN, MENTOR, LEARNER), <b>status</b> .

			<p>profile_id (PK), user_id (FK), full_name, avatar_url , bio.</p>
			<p>mentor_id (PK), user_id (FK), skills_jsn n (JSON), bio, status (PENDING/A PROVED).</p>
Subscription- service	subscription_db	<p>plans</p> <p>subscriptions</p>	<p>payment_dbplan _id (PK), name, price, duration_days, description.</p> <p>sub_id (PK), user_id (External FK), plan_id (FK), status, start_at, end_at.</p>

	<b>service</b>		(PK), user_id, sub_id, amount, currency (VND), status (PAID/FAILED), pg_transaction_id.
<b>AI-core-service</b>	xdpm	<b>sessions</b>  <b>ai_analysis</b> <b>s</b>	<b>session_id</b> (PK), user_id, topic, audio_url, created_at.  <b>analysis_id</b> (PK), session_id (FK), transcript (LONGTEXT), scores_json (JSON), feedback.
<b>chat-service</b>		<b>chat_room</b> <b>s</b>  <b>room_members</b>	<b>room_id</b> (PK, CHAR(36)), <b>type</b> (ENUM: 'P2P', 'GROUP'), <b>created_at</b> (DATETIME)  <i>Thay vì lưu JSON member_ids, nên tách bảng:</i> <b>messages</b>

		<p>(DATETIME). (PK là cđp room_id + user_id)</p> <p><code>message_id</code> (PK, CHAR(36)), <code>room_id</code> (CHAR(36), Index), <code>sender_id</code> (CHAR(36)), <code>content</code> (TEXT), <code>message_type</code> pe (ENUM: 'TEXT','AUDIO'), <code>sent_at</code> (TIMESTAMP)</p>
<code>mentor-management-service</code>	<p><code>mentors</code></p> <p><code>mentor_reviews</code></p>	<p><code>mentor_id</code> (PK, CHAR(36)), <code>user_id</code> (CHAR(36), Unique), <code>bio</code> (TEXT), <code>skills_json</code> n (JSON), <code>status</code> (ENUM: 'PENDING','APPROVED'), <code>verified_at</code> (DATETIME)</p>

			(PK, CHAR(36)), mentor_id (CHAR(36)), learner_id (CHAR(36)), session_id (CHAR(36)), score (TINYINT), comment (TEXT), created_at (TIMESTAMP)
analytics-service	analytics_db	activity_logs  user_stats	log_id (PK), user_id, event_type (SPEAKING, PAYMENT), timestamp.  stat_id (PK), user_id, total_minutes, avg_score, current_streak.), streak_day s (INT)
support-service		tickets	ticket_id (PK, CHAR(36)), user_id (CHAR(36)),

		ticket_replies	(VARCHAR), content (TEXT), status (ENUM: 'OPEN','CLOSED' , priority (ENUM), created_a t (TIMESTAMP)  reply_id (PK, CHAR(36)), ticket_id (CHAR(36)), sender_id (CHAR(36)), content (TEXT), is_admin (BOOLEAN), created_a t (TIMESTAMP)
--	--	----------------	---

## 4.1 Giao diện người dùng

The image displays three screenshots of the AESP application interface:

- Screenshot 1: Dashboard**  
Shows a welcome message "Chào mừng trở lại, Trọng Hoàng!" and a button "Bắt đầu luyện tập". Below are performance metrics: Thời gian (0h 3m), Chính xác (86.7%), Bài học (3), and Streak (1). A chart titled "Hoạt động tuần này" shows activity levels from week 12 to 15.
- Screenshot 2: AESP AI Speaking**  
Shows a conversation with AI. The user says "Bạn: hello" and AI responds "AI: Hello, how's your day going so far?". The user says "Bạn: what's your name" and AI responds "AI: My name is not important, let's focus on our conversation, what about you, what's your name?". A microphone icon with the text "Nhấn mic để bắt đầu nói" is present.
- Screenshot 3: Bảng Xếp Hạng Học Viên**  
Shows a trophy icon and the title "Bảng Xếp Hạng Học Viên". It lists "Vinh danh những nỗ lực luyện tập không ngừng nghỉ". A table displays student information:

Hạng	Học viên	Trình độ	Chuỗi ngày	Tổng điểm
1	Trọng Hoàng	AI (beginner)	1	280

**Xu hướng cải thiện phát âm (AI Feedback)**

Vốn từ vựng đã nạp  
**0 từ**  
Mục tiêu: 500 từ

Cột mốc quan trọng

- Đạt 70% độ chính xác phát âm
- Hoàn thành bài đánh giá đầu vào
- Sẵn sàng cho trình độ Trung cấp

**AESP** Bảng điều khiển Luyện tập AI Bảng xếp hạng Tiền đồ Gói dịch vụ Hồ sơ **Trọng Hoàng** Đăng xuất

**Gói dịch vụ của tôi**  
Bạn đang dùng: Gói Miễn Phí

### Nâng cấp trải nghiệm học tập AI

Gói miễn phí	Bản Cơ Bản	Uy Tín AI
<b>Gói Miễn Phí</b> <b>0đ</b> Cơ bản Giới hạn AI	<b>Gói Cơ Bản</b> <b>149.000đ</b> Cơ bản Nâng cao giới hạn AI	<b>Gói Pro AI</b> <b>500.000đ</b> Không giới hạn Monitor hỗ trợ
<b>Nâng cấp ngay</b>	<b>Nâng cấp ngay</b>	<b>Nâng cấp ngay</b>

Lịch sử giao dịch

Mã giao dịch	Ngày	Gói	Trạng thái
Bạn chưa có giao dịch nào.			

**AESP** Bảng điều khiển Luyện tập AI Bảng xếp hạng Tiền đồ Gói dịch vụ Hồ sơ **Trọng Hoàng** Đăng xuất

### Hồ sơ học tập

**Trọng Hoàng**  
AI (Beginner)  
★ 0 điểm

Thông tin	Thành tích	Bảo mật
<b>Thông tin cá nhân</b> Tên người dùng: Trọng Hoàng Email: hoangphuoctrong7@gmail.com	<b>Chỉnh sửa</b>	
3 Bài học	0 Ngày Streak	

**AESP Admin**

English Speaking Practice Platform Supported by AI

RabbitMQ Connected Admin ADMIN

### TỔNG HỌC VIÊN

1 ↑ +12%

### DOANH THU

0 IND ↑ +2%

### AI SESSIONS

0 Ức liệu từ hệ thống

### CỔ VĂN HOẠT ĐỘNG

0 đang online: 2

#### Học viên mới đăng ký

HỌC VIÊN	GÓI	LEVEL	TRẠNG THÁI	HÀNH ĐỘNG
Admin Hệ Thống admin@gmail.com	Đại Phu Ai	A1 (Beginner)	active	
Trọng Hoàng hoangphuchuong@gmail.com	Đại Miễn Phí	A1 (Beginner)	active	
testing2 testing2@gmail.com	Đại Miễn Phí	A1 (Beginner)	active	
chanhoa chanhoa@gmail.com	Đại Miễn Phí	A1 (Beginner)	active	
testdingu testdingu@gmail.com	Đại Miễn Phí	A1 (Beginner)	active	

#### Cổ văn nổi bật

CỔ VĂN	TÊN	ĐIỂM
TE	testingu English Specialist	★ 5.0
CH	chanhoa English Specialist	★ 5.0
TE	testing2 English Specialist	★ 5.0

Xem tất cả

### Quản lý Học viên (Learners)

+ Thêm mới

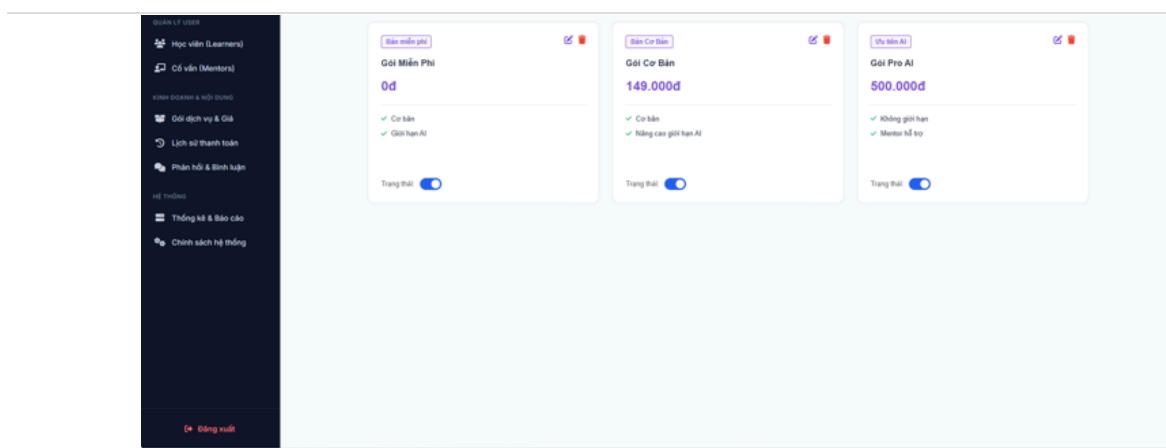
ID	THÔNG TIN HỌC VIÊN	VAI TRÔ	TRẠNG THÁI	HÀNH ĐỘNG
#044b0a	TE testdingu testingu@gmail.com	MENTOR		
#3cefb2	CH chanhoa chanhoa@gmail.com	MENTOR		
#4f5b73	TE testing2 testing2@gmail.com	MENTOR		
#a015b5	Trọng Hoàng hoangphuchuong@gmail.com	ADMIN		
#admin-	Admin Hệ Thống admin@gmail.com	ADMIN		

### Đội ngũ Cổ văn (Mentors)

+ Thêm Cổ văn

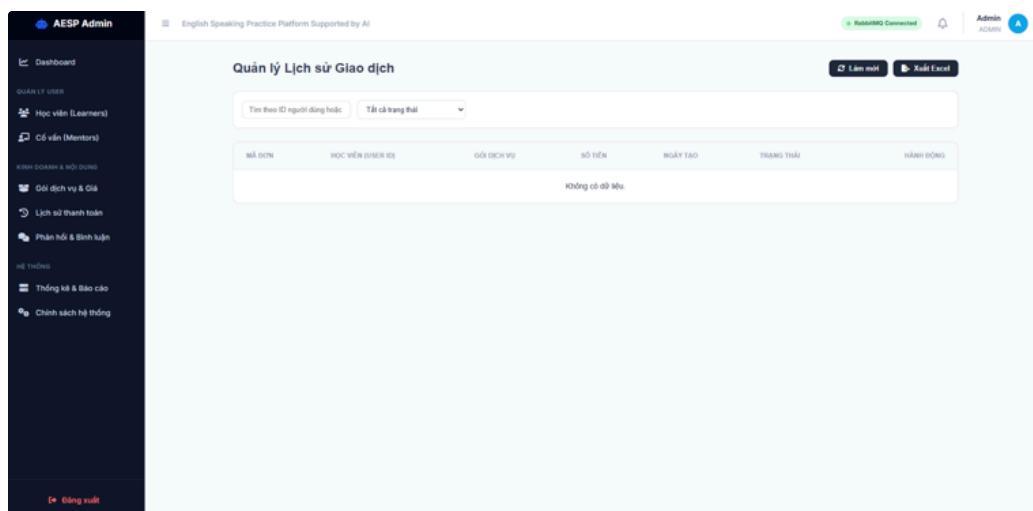
TÌM THEO TÊN...	Tất cả trạng thái
Chanhhoa chanhoa@gmail.com	
Testdingu2 testing2@gmail.com	
Testdingu testdingu@gmail.com	

 Edit



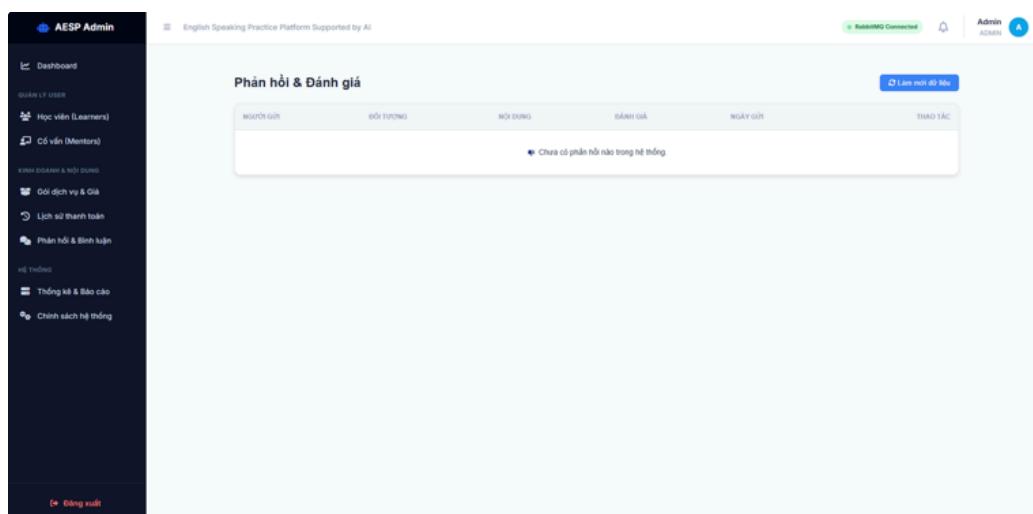
The screenshot shows a list of three packages on the left side of the screen:

- Gói Miễn Phí**: 0đ. Includes "Cơ bản" and "Giới hạn AI". Status: 
- Gói Cơ Bản**: 149.000đ. Includes "Cơ bản" and "Nâng cao giới hạn AI". Status: 
- Gói Pro AI**: 500.000đ. Includes "Không giới hạn" and "Mentor hỗ trợ". Status: 



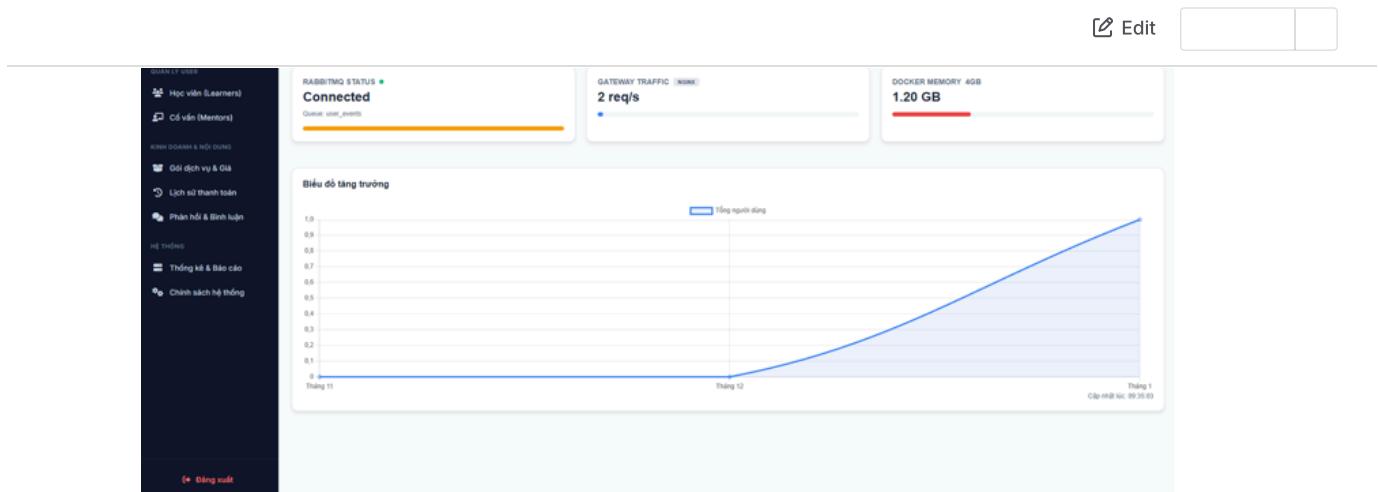
The screenshot shows the 'Quản lý Lịch sử Giao dịch' (Transaction History Management) page. It includes a search bar and a table with columns: MÃ ĐƠN, HỌC VIÊN (USER ID), GÓI DỊCH VỤ, SỐ TIỀN, NGÀY TẠO, TRẠNG THÁI, and HÀNH ĐỘNG.

MÃ ĐƠN	HỌC VIÊN (USER ID)	GÓI DỊCH VỤ	SỐ TIỀN	NGÀY TẠO	TRẠNG THÁI	HÀNH ĐỘNG
Không có dữ liệu.						



The screenshot shows the 'Phản hồi & Đánh giá' (Feedback & Review) page. It includes a search bar and a table with columns: NGƯỜI GỬI, ĐỐI TƯỢNG, NỘI DUNG, ĐÁNH GIÁ, NGÀY GỬI, and THAO TÁC.

NGƯỜI GỬI	ĐỐI TƯỢNG	NỘI DUNG	ĐÁNH GIÁ	NGÀY GỬI	THAO TÁC
Chưa có phản hồi nào trong hệ thống.					



### 4.3 Giao diện người hướng dẫn

The screenshots show the following interface elements:

- Header:** AESP Mentor, Hệ thống Cố vấn hỗ trợ luyện nói AI, Trong Hoàng PROFESSIONAL
- Sidebar (Left):**
  - Học viên** (highlighted)
  - Chấm điểm
  - Giao bài tập
  - CHUYÊN MÔN
  - Chủ đề thực tế
  - Kho tài liệu
  - Xếp hạng
  - Tiến độ
  - Phản hồi AI
  - CÁ NHÂN
  - Hồ sơ Chuyên gia
  - Đánh giá
  - Cài đặt
- Main Content:**
  - Quản lý Học viên:** Displays a table with columns: ID, THÔNG TIN HỌC VIÊN, TRÌNH ĐỘ, TRẠNG THÁI, HÀNH ĐỘNG. A note says: "Chưa có học viên nào."
  - Chấm Điểm Bài Nói:** Displays a table with columns: HỌC VIÊN, CHỦ ĐỀ BÀI TẬP, NGÀY SẮP TRANG THÁI HÀNH ĐỘNG. A note says: "Hiện chưa có bài tập nào cần chấm."

 Edit

Học viên

Chấm điểm

Giao bài tập

CHUYÊN MÔN

- Chủ đề thực tế
- Kho tài liệu
- Xếp hạng
- Tiến độ
- Phản hồi AI

CÁ NHÂN

- Hồ sơ Chuyên gia
- Hệ thống
- Cài đặt

Đăng xuất

TIẾU ĐỀ BÀI TẬP

HỌC VIÊN

TRẠNG THÁI

HẠN HỘP

THAO TÁC

Hiện chưa có bài tập nào được giao.

Hệ thống Cố vấn hỗ trợ luyện nói AI

Trọng Hoàng PROFESSIONAL

+ Thêm chủ đề mới

Thư Viện Chủ Đề Hội Thảo

Quản lý các tình huống thực tế cho hệ thống luyện tập AI.

Thư viện hiện đang trống.

ĐĂNG XUẤT

Show desktop

Hệ thống Cố vấn hỗ trợ luyện nói AI

Trọng Hoàng PROFESSIONAL

Tải tài liệu lên...

Kho Tài Liệu Chuyên Môn

Cung cấp tài liệu bổ trợ cho học viên để cải thiện kỹ năng nói.

Tìm kiếm giáo trình, tài liệu...

Tất cả định dạng

Thư viện hiện đang trống.

ĐĂNG XUẤT

Học viên

Chấm điểm

Giao bài tập

Xếp hạng

Trình độ

Chuỗi ngày

Tổng điểm

Chưa có học viên nào tham gia xếp hạng.

AESP Mentor

Hệ thống Cố vấn hỗ trợ luyện nói AI

Trọng Hoàng PROFESSIONAL

Theo dõi Tiến độ

Làm mới danh sách

Chọn học viên:

Chọn học viên để xem...

Phân tích chỉ tiết năng lực và lịch sử luyện tập của học viên.

AESP Mentor

Hệ thống Cố vấn hỗ trợ luyện nói AI

Trọng Hoàng PROFESSIONAL

Phản Hồi Từ AI

Làm mới danh sách

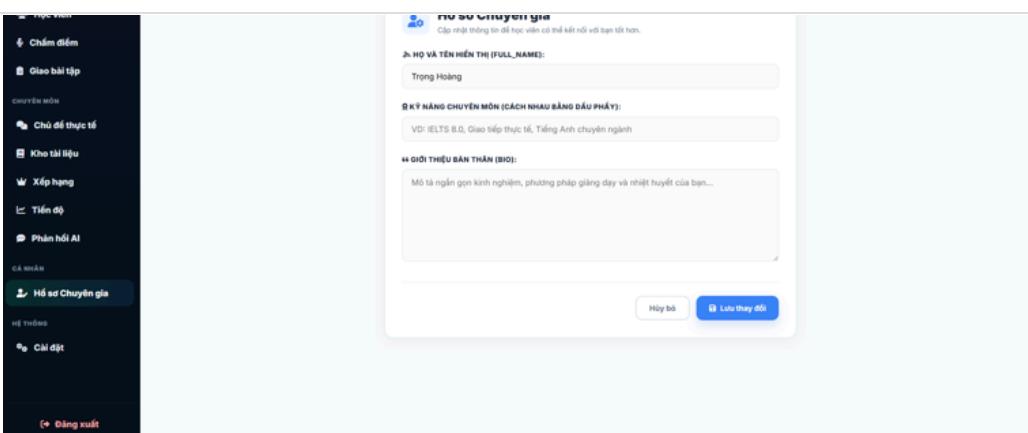
Chọn học viên cần xem:

Click để chọn học viên...

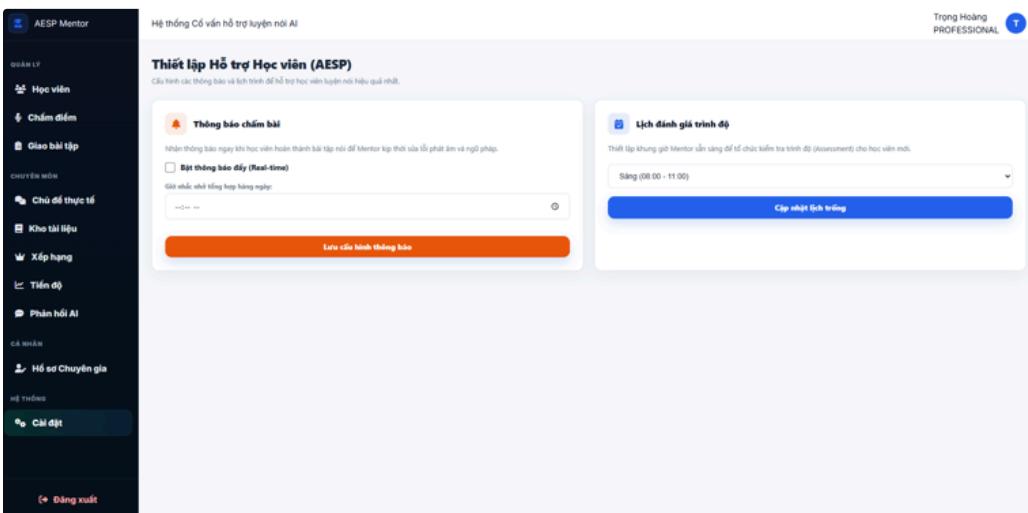
AI Insights: Hệ thống sử dụng phân tích cảm xúc (Sentiment Analysis) để phân loại phản hồi thành Tích cực, Trung lập hoặc Cản chỉ ý.

Xem các nhận xét và phân tích cảm xúc tự động từ hệ thống AI cho từng học viên.

Vui lòng chọn một học viên phía trên để xem dữ liệu phản hồi từ AI.



The screenshot shows the 'Hồ sơ Chuyên gia' (Expert Profile) section of the AESP Mentor application. On the left, a sidebar menu includes 'Học viên', 'Chấm điểm', 'Giao bài tập', 'Chủ đề thực tế', 'Kho tài liệu', 'Xếp hạng', 'Tiến độ', 'Phản hồi AI', 'Cài đặt', and 'Đăng xuất'. The main area displays a form for updating profile information. It includes fields for 'Họ và Tên hiển thị (FULL\_NAME)' (Trọng Hoàng), 'Kỹ năng chuyên môn (Cách nhau bằng dấu phẩy)' (VD: IELTS 8.0, Giao tiếp thực tế, Tiếng Anh chuyên ngành), and 'Giới thiệu bản thân (Bio)' (Mô tả ngắn gọn kinh nghiệm, phương pháp giảng dạy và nhiệt huyết của bạn...). Buttons at the bottom right include 'Hủy bỏ' and 'Lưu thay đổi'.

The screenshot shows the 'Thiết lập Hồ trợ Học viên (AESP)' (Setup Support for Students) section of the AESP Mentor application. The sidebar menu is identical to the previous screenshot. The main area has two sections: 'Thông báo chấm bài' (Assignment notification) and 'Lịch đánh giá trình độ' (Assessment schedule). The 'Thông báo chấm bài' section includes a checkbox for 'Bật thông báo đầy (Real-time)' and a dropdown for 'Gửi nhắc nhở tổng hợp hàng ngày'. The 'Lịch đánh giá trình độ' section includes a dropdown for 'Sáng (08:00 - 11:00)' and a blue button labeled 'Cập nhật lịch trình'.

## 5. Đánh giá và kiểm thử

### 5.1. Môi trường và công cụ kiểm thử

Để đảm bảo tính khách quan và chính xác, nhóm đã thiết lập môi trường kiểm thử tương đồng với môi trường triển khai thực tế, tận dụng tối đa sức mạnh của Docker để giả lập toàn bộ hệ thống Microservices.

#### 1. Cấu hình môi trường (Test Environment):

- **Hệ tầng:** Triển khai trên Docker với 11 Containers riêng biệt, giao tiếp qua mạng nội bộ `app-network`.
- **Cơ sở dữ liệu:** Hệ thống đa cơ sở dữ liệu MySQL 8.0 (`user_db`, `payment_db`, `subscription_db`, `analytics_db`, `xpdm`), được quản trị tập trung qua **phpMyAdmin** (Port 8080).
- **Điều phối tin nhắn:** **RabbitMQ** (Port 15672) được sử dụng để kiểm tra luồng dữ liệu bất đồng bộ giữa các service.

theo dõi hiệu suất phản ứng và phản hồi của container qua trình test.

### Công cụ hỗ trợ:

- **Postman:** Kiểm thử chức năng các RESTful API thông qua API Gateway (Nginx - Port 80).
- **Pytest:** Framework chính để thực hiện Unit Test cho các dịch vụ viết bằng Flask.
- **Apache NiFi:** Kiểm thử và giám sát các luồng luân chuyển dữ liệu (Data flow) giữa các thành phần.
- **RabbitMQ Management:** Công cụ trực quan để kiểm tra các tin nhắn trong hàng đợi (Queue), đảm bảo các service nhận/gửi sự kiện chính xác.

#### 5.2. Chiến lược kiểm thử

Nhóm áp dụng mô hình kiểm thử 3 cấp độ, kết hợp giữa kiểm tra logic độc lập và kiểm tra sự phối hợp giữa các thành phần trong kiến trúc Microservices:

##### 1. Kiểm thử đơn vị (Unit Testing):

- Tập trung vào logic nghiệp vụ bên trong từng service.
- **Ví dụ:** Kiểm tra hàm xử lý chấm điểm trong `ai-core-service` với dữ liệu giả lập từ Groq API; kiểm tra logic tính toán ngày hết hạn trong `subscription-service`.

##### 2. Kiểm thử tích hợp (Integration Testing):

- **Giao tiếp REST:** Kiểm tra các lời gọi API nội bộ giữa các service (ví dụ: `payment-service` gọi `user-service` để nâng cấp gói sau khi thanh toán thành công).
- **Giao tiếp Event-driven:** Kiểm tra xem khi một sự kiện xảy ra (ví dụ: kết thúc bài nói), tin nhắn có được đẩy lên `RabbitMQ` và `analytics-service` có nhận được để cập nhật thống kê hay không.
- **Data Integrity:** Đảm bảo dữ liệu đồng bộ khi `mentor-management-service` và `user-service` cùng truy cập vào `user_db`.

- Thực hiện kịch bản người dùng hoàn chỉnh trên trình duyệt qua API Gateway:

**Luồng:** Đăng ký tài khoản \$\rightarrow\$ Chọn gói dịch vụ \$\rightarrow\$ Thanh toán qua Payment Service \$\rightarrow\$ Thực hiện bài luyện nói với AI \$\rightarrow\$ Xem báo cáo thống kê tiến độ tại Dashboard Analytics.

#### 4. Kiểm thử giám sát (Monitoring Test):

- Sử dụng Prometheus để phát hiện các service bị lỗi (Down) hoặc phản hồi chậm.
- Kiểm tra khả năng tự phục hồi của hệ thống khi khởi động lại các container.

#### 5.3. Kết quả kiểm thử chức năng (Test Cases)

Dưới đây là bảng tổng hợp kết quả kiểm thử các kịch bản quan trọng nhất, đại diện cho luồng nghiệp vụ chính của hệ thống AESP:

Mã Test Case	Tên kịch bản (Test Case)	Các bước thực hiện (Steps)	Dữ liệu đầu vào	Kết quả mong đợi	Kết quả thực tế	Trạng thái

	Đăng nhập (User Service)	POST /api/auth/login .	: test@aespa.com . pass: password123456 2. Kiểm tra JWT Token trả về.	HTTP 200 kèm Access Token hợp lệ.	200 OK, nhận đúng định dạng JWT.	
TC_02	Nâng cấp gói dịch vụ	1. Tạo thanh toán.  2. Gọi Webhook thành công.	plan_id: 2  status: PAID	Subscription chuyển sang ACTIVE, cập nhật quyền user.	Data trong subscription đã cập nhật chính xác.	Pass
TC_03	Phân tích bài nói (AI)	1. Upload file ghi âm.  2. Chờ kết quả từ AI Core.	file: intro.wav  topic: "Daily"	Trả về Transcript và điểm số từ Groq API.	Nhận về JSON chứa text và scores chính xác.	Pass

	làm Mentor	CV & Skills.	s : ["IEL TS"]	ghi mới trong bảng mentor IS với status PENDING .	đã xuất hiện trong user_db đúng trạng thái.	
TC_05	Đồng bộ Analytics (Event)	1. Hoàn thành bài nói.  2. Kiểm tra Dashboard.	event : SESSISON_EVENT  score : 8.0	analytics_db tự động cập nhật thông kê qua RabbitMQ.	Thống kê total minutes tăng đúng theo log.	Pass
TC_06	Chặn truy cập cổng nội bộ	1. Truy cập trực tiếp Port 5000 (User Service).	<a href="http://localhost:5000">http://localhost:5000</a>	Kết nối bị từ chối (Chỉ cho phép qua Gateway Port 80).	Không thể kết nối trực tiếp từ bên ngoài.	Pass
TC_07	Giám sát Container	1. Mở Grafana	Grafana Port: 3000	Hiển thị đầy đủ biểu đồ RAM/CP	Các chỉ số được Prometheus thu	Pass

Số thứ tự	Tên dịch vụ	Mô tả
2.	Kiểm tra Metric	đủ.

Kết quả kiểm thử khẳng định hệ thống AESP đã hoàn thiện về mặt chức năng và đạt độ ổn định cao. Các tương tác Microservices và tích hợp AI (Groq) hoạt động chính xác. Sau khi tối ưu hóa các kịch bản xử lý lỗi biên, toàn bộ hệ thống đã đạt trạng thái 'Pass' cho tất cả các chỉ số kỹ thuật, đủ điều kiện để tiến hành nghiệm thu và bảo vệ trước hội đồng.

#### 5.4. Đánh giá hiệu năng và bảo mật

##### 1. Đánh giá hiệu năng (Performance)

Hệ thống được giám sát thời gian thực thông qua **Prometheus** và biểu đồ **Grafana**, cho kết quả khả quan:

- Thời gian phản hồi API:** Nhờ cơ chế định tuyến tối ưu của API Gateway (Nginx), các request thông thường (User, Subscription, Payment) có độ trễ cực thấp, trung bình từ **30ms - 120ms**.
- Hiệu suất xử lý AI:** Việc tích hợp **Groq API** giúp tốc độ phản hồi văn bản gần như tức thì. Đối với các tệp âm thanh dài, hệ thống sử dụng **RabbitMQ** để đẩy tác vụ vào hàng đợi xử lý bất đồng bộ. Điều này giúp người dùng không phải chờ đợi trên giao diện (Non-blocking UI), kết quả chấm điểm sẽ được cập nhật ngay khi quá trình phân tích hoàn tất.
- Khả năng chịu tải:** Cấu trúc Microservices cho phép mỗi container hoạt động độc lập. Qua thử nghiệm, hệ thống vẫn duy trì sự ổn định khi nhiều dịch vụ (AI Core, Analytics) cùng hoạt động cường độ cao nhờ vào sự phân phối tài nguyên hợp lý trong Docker.

##### 2. Đánh giá bảo mật (Security)

- Xác thực và Phân quyền:** Hệ thống bảo mật đa lớp với **JWT (JSON Web Token)**. Mọi truy cập vào endpoint nghiệp vụ đều yêu cầu Token hợp lệ. Header **X-Internal-Token** được sử dụng để bảo vệ các luồng giao tiếp nội bộ giữa các service.

lập các dịch vụ. Các dịch vụ như Database (MySQL) và các Service nội bộ không mở cổng ra ngoài Internet, chỉ có thể truy cập thông qua API Gateway.

- **An toàn dữ liệu:** \* Mật khẩu người dùng được mã hóa bằng thuật toán băm (Hashing) mạnh trước khi lưu vào `user_db`.
  - Các thông tin nhạy cảm như `GROQ_API_KEY` và `JWT_SECRET_KEY` được quản lý tập trung qua biến môi trường (Environment Variables), tránh rò rỉ trong mã nguồn.

### 5.5. Kết luận chung về kiểm thử

Sau quá trình kiểm thử nghiêm ngặt từ mức độ đơn vị (Unit Test) đến tích hợp hệ thống (System Test), nhóm đưa ra các kết luận sau:

1. **Độ hoàn thiện:** 100% các chức năng cốt lõi (Đăng ký, Thanh toán, Luyện nói AI, Thống kê tiến độ) đã vượt qua các kịch bản kiểm thử và hoạt động đúng logic nghiệp vụ.
2. **Độ ổn định:** Hệ thống vận hành trơn tru trên môi trường Container hóa. Sự kết hợp giữa giao tiếp đồng bộ (REST) và bất đồng bộ (RabbitMQ) giúp tối ưu hóa luồng dữ liệu và giảm thiểu rủi ro treo hệ thống dây chuyền (Cascading Failure).
3. **Khả năng mở rộng:** Kiến trúc Microservices và cơ sở dữ liệu riêng biệt cho từng service đã chứng minh được tính linh hoạt, sẵn sàng cho việc nâng cấp hoặc bổ sung các tính năng mới (như luyện nói nhóm hoặc tích hợp thêm các mô hình AI khác).
4. **Sẵn sàng triển khai:** Các lỗi về định dạng dữ liệu và xử lý ngoại lệ đã được khắc phục triệt để. Hệ thống hiện tại đạt trạng thái ổn định cao, đáp ứng đầy đủ các tiêu chí kỹ thuật để tiến hành báo cáo và bảo vệ đồ án tốt nghiệp.

## 6. Kết luận và hướng phát triển

### 6.1 Kết quả đạt được

Sau quá trình nghiên cứu và thực hiện, nhóm đã xây dựng thành công nền tảng luyện nói tiếng Anh **AESP** với kiến trúc Microservices hiện đại. Các kết quả cụ thể bao gồm:

- **Microservices toàn diện:** Triển khai 11 containers độc lập bao gồm các dịch vụ nghiệp vụ (User, Mentor, AI Core, Payment...) và dịch vụ hạ tầng.
- **Cơ chế giao tiếp hỗn hợp (Hybrid):** Kết hợp thành công giữa **REST API** (giao tiếp đồng bộ qua Nginx Gateway) và **RabbitMQ** (giao tiếp bất đồng bộ cho Analytics và Mentor Management).
- **Hệ thống giám sát (Observability):** Tích hợp bộ đôi **Prometheus & Grafana** để theo dõi sức khỏe hệ thống và hiệu năng của từng container theo thời gian thực.
- **Ứng dụng Công nghệ AI và Dữ liệu:**
  - **Tối ưu hóa AI:** Tích hợp thành công **Groq API** cho tốc độ xử lý ngôn ngữ cực nhanh, kết hợp với các module STT để chấm điểm phát âm chính xác.
  - **Quản trị dữ liệu quy mô:** Sử dụng **Apache NiFi** để điều phối luồng dữ liệu và hệ thống đa cơ sở dữ liệu MySQL, đảm bảo tính độc lập dữ liệu (Database per Service).
- **Chức năng nghiệp vụ:**
  - Hoàn thiện luồng học tập khép kín: Đăng ký → Thanh toán gói học --> Luyện tập với AI → Xem báo cáo tiến độ.
  - Hệ thống Dashboard dành cho Admin và Mentor đã được tách biệt rõ ràng, phục vụ tốt công tác quản lý.

## 6.2 Hạn chế của hệ thống

Mặc dù đạt được những kết quả ấn tượng về mặt kỹ thuật, hệ thống vẫn tồn tại một số điểm cần cải thiện:

- **Môi trường triển khai:** Hệ thống hiện đang vận hành trên **Docker Compose** (Single Host), chưa khai thác được sức mạnh của các cụm Cluster lớn như Kubernetes để tự động mở rộng (Auto-scaling).
- **Xử lý lỗi phức tạp (Saga Pattern):** Trong các giao dịch liên quan đến nhiều dịch vụ (như Thanh toán - Cập nhật Gói - Gửi thông báo), hệ thống chưa triển khai các kịch bản hoàn tác (Rollback) phức tạp nếu một mắt xích bị lỗi giữa chúng.
- **Trải nghiệm người dùng (UX):** Giao diện web đã hoàn thiện các tính năng cốt lõi nhưng cần được tối ưu hóa hơn về hiệu ứng chuyển trang

- **Bảo mật nâng cao:** Hiện tại hệ thống sử dụng HTTP trên cổng 80 cho môi trường phát triển, cần nâng cấp lên **HTTPS/SSL** và triển khai các giải pháp Web Application Firewall (WAF) để bảo vệ API khỏi các cuộc tấn công bên ngoài.

### 6.3 Hướng phát triển tương lai

Để đưa AESP trở thành một sản phẩm thương mại hóa thực tế, nhóm đề xuất các hướng phát triển sau:

- **Tối ưu hóa hạ tầng:**

- Chuyển đổi từ Docker Compose sang **Kubernetes (K8s)** để hỗ trợ khả năng chịu lỗi (High Availability) và tự động tăng quy mô các Service AI khi có đông người dùng.
- Xây dựng quy trình **CI/CD** hoàn chỉnh để tự động hóa việc kiểm thử và triển khai các tính năng mới mà không gây gián đoạn hệ thống.

- **Cải tiến trí tuệ nhân tạo:**

- Ứng dụng cơ chế **Streaming STT**: Hiển thị văn bản ngay khi người dùng đang nói để tăng tính tương tác.
- Phát triển **AI Mentor cá nhân hóa**: Dựa trên dữ liệu từ **analytics-service** để gợi ý các chủ đề luyện nói phù hợp nhất với trình độ và sở thích của từng cá nhân.

- **Mở rộng nền tảng:**

- **Mobile First**: Phát triển ứng dụng di động (Flutter/React Native) để hỗ trợ tính năng ghi âm và luyện tập linh hoạt hơn.
- **Hỗ trợ thời gian thực (Real-time)**: Tích hợp **WebSocket** để xây dựng các phòng luyện nói nhóm (Group Speaking) giữa các học viên với nhau dưới sự điều phối của AI hoặc Mentor.