

TRƯỜNG ĐẠI HỌC BÁCH KHOA
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



CÔNG NGHỆ PHẦN MỀM (CO3001)

BÀI TẬP LỚN

*Hệ thống hỗ trợ Tutor tại
Trường Đại học Bách khoa
Đại học Quốc gia - TP.HCM*

Giáo viên hướng dẫn: Phan Trung Hiếu, CSE-HCMUT

Sinh viên - Nhóm 3: Nguyễn Tấn Phát - 2352888 (CN01)
Vũ Hà Như Ngọc - 2352818 (CN01)
Lê Diệu Quỳnh - 2353036 (CN01)
Mã Nhật Tiến - 2353178 (CN01)
Bùi Phan Khánh Duy - 2352170 (CN01)
Lương Đức Huy - 2352384 (CN01)
Nguyễn Ngọc Phát - 2352887 (CN01)
Văn Bá Trọng Khiêm - 2352546 (CN01)

THÀNH PHỐ HỒ CHÍ MINH, THÁNG 9 NĂM 2025

Mục lục

Danh sách Ký hiệu	2
Danh sách Từ viết tắt	2
Danh sách Hình ảnh	4
Danh sách Bảng	4
Danh sách thành viên & khối lượng công việc	4
4. Triển khai hệ thống	5
4.1. Sơ đồ triển khai (Deployment View)	5
4.1.1 Giới thiệu	5
4.1.2. Sơ đồ triển khai hệ thống	5
4.1.3. Mô tả các thành phần	6
4.1.4. Luồng hoạt động	6
4.2. Sơ đồ phát triển (Development View)	7
4.2.1 Giới thiệu	7
4.2.2. Sơ đồ gói (Package Diagram)	8
4.2.3. Mô tả các gói	8
4.2.4. Luồng xử lý dữ liệu qua các lớp	9
4.3. Sơ đồ lớp chi tiết (Detailed Class Diagram)	9
4.3.1. Giới thiệu	9
4.3.2. Sơ đồ lớp chi tiết của hệ thống	10
4.3.3. Mô tả cụm chức năng	10
4.4. Mô tả các lớp chi tiết và phương thức	11
4.4.1. Cụm Quản lý Người dùng (User Management)	11
4.4.2. Cụm Chương trình và Đăng ký (Program & Registration)	13
4.4.3. Cụm Lịch trình và Buổi học (Scheduling & Session)	14
4.4.4. Cụm Đánh giá và Phản hồi (Feedback & Support)	16



Danh sách Ký hiệu

Danh sách Từ viết tắt

Bảng 1: Danh sách Từ viết tắt

STT	Chữ viết tắt	Chữ viết đầy đủ
1	AI	Trí tuệ nhân tạo (Artificial Intelligence)
2	APP	Application (Ứng dụng)
3	BM	Bộ môn
4	DB	Database (Cơ sở dữ liệu)
5	DOCX	Microsoft Word Document (Tài liệu Microsoft Word)
6	DRP	Disaster Recovery Plan (Kế hoạch khôi phục sau thảm họa)
7	Excel	Microsoft Excel (Phần mềm bảng tính)
8	GPA	Điểm trung bình học tập (Grade Point Average)
9	HCMUT_DATACORE	Lỗi dữ liệu Đại học Bách khoa
10	HCMUT_LIBRARY	Thư viện Đại học Bách khoa
11	HTTPS	Hypertext Transfer Protocol Secure (Giao thức truyền tải siêu văn bản an toàn)
12	ID	Identifier (Mã định danh)
13	iOS	iPhone Operating System (Hệ điều hành của iPhone)
14	MB	Megabyte
15	MSSV	Mã số sinh viên
16	MVC	Model-View-Controller (Một mẫu kiến trúc phần mềm)
17	NCS	Nghiên cứu sinh
18	OTP	One-Time Password (Mật khẩu dùng một lần)
19	PCTSV	Phòng Công tác Sinh viên
20	PDF	Portable Document Format (Định dạng tài liệu di động)
21	PDT	Phòng Đào tạo
22	PPT	PowerPoint Presentation (Bài trình chiếu PowerPoint)
23	SDT	Số điện thoại
24	SMS	Short Message Service (Dịch vụ tin nhắn ngắn)
25	SV	Sinh viên
26	TLS	Transport Layer Security (Bảo mật tầng truyền tải)
27	UC	Use Case (Ca sử dụng)
28	US	User Story (Câu chuyện người dùng)
29	2FA	Two-Factor Authentication (Xác thực hai yếu tố)



Danh sách Hình ảnh

1	Sơ đồ triển khai hệ thống hỗ trợ Tutor	5
2	Sơ đồ tổ chức các gói của hệ thống	8
3	Sơ đồ lớp chi tiết của hệ thống	10

Danh sách Bảng

1	Danh sách Từ viết tắt	2
2	Danh sách thành viên & khối lượng công việc	4
3	Các thành phần trong sơ đồ triển khai hệ thống hỗ trợ Tutor	6
4	Chức năng của từng gói (package) trong kiến trúc phần mềm	9



Danh sách thành viên & khối lượng công việc

STT	Họ Tên	MSSV	Vai trò	% Hoàn thành
1	Nguyễn Tấn Phát	2352888	Scrum Master	100%
2	Vũ Hà Như Ngọc	2352818	Business Analyst	100%
3	Lê Diệu Quỳnh	2353036	Product Owner	100%
4	Mã Nhật Tiến	2353178	Quality Controller	100%
5	Bùi Phan Khánh Duy	2352170	Techlead + SysAdmin	100%
6	Lương Đức Huy	2352384	Developer	100%
7	Nguyễn Ngọc Phát	2352887	Developer	100%
8	Văn Bá Trọng Khiêm	2352546	Developer	100%

Bảng 2: Danh sách thành viên & khối lượng công việc

4. Triển khai hệ thống

4.1. Sơ đồ triển khai (Deployment View)

4.1.1 Giới thiệu

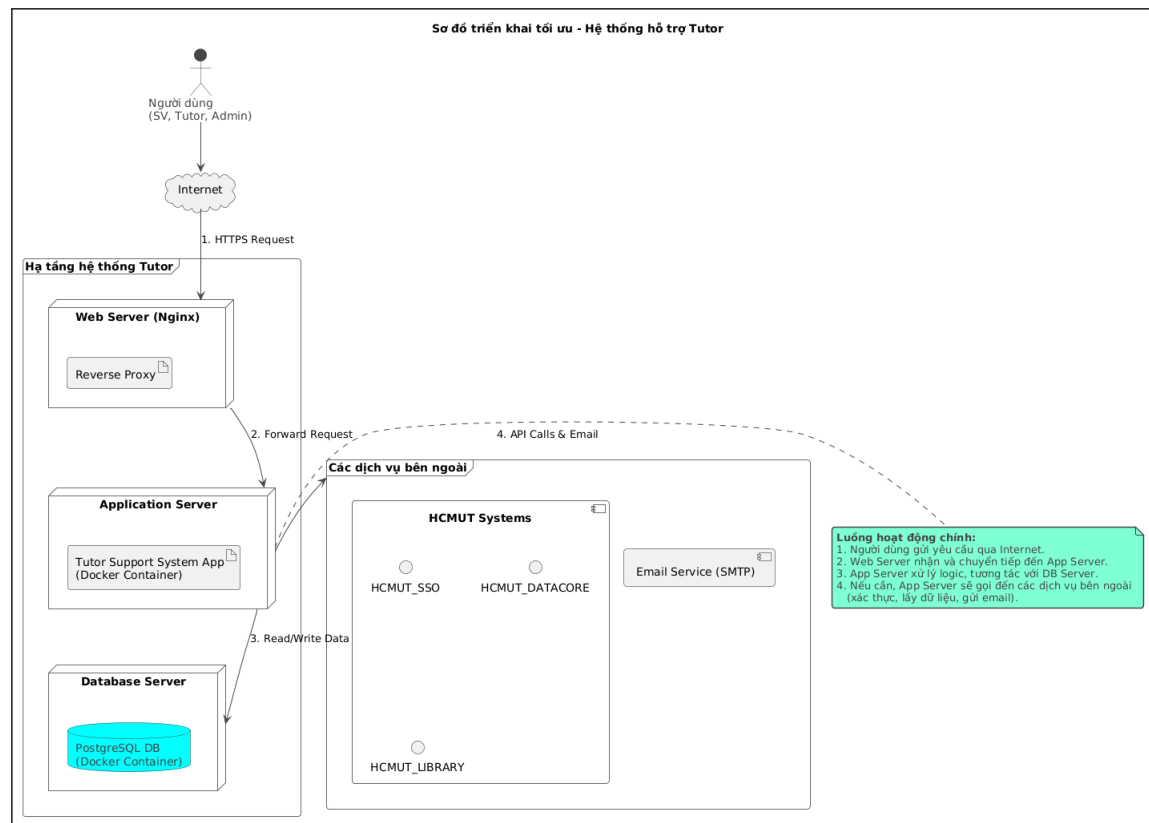
Sơ đồ triển khai (Deployment Diagram) mô tả kiến trúc vật lý của "Hệ thống hỗ trợ Tutor", thể hiện cách các thành phần phần mềm được phân bổ và vận hành trên các nút (node) phần cứng. Sơ đồ này cung cấp một cái nhìn tổng quan về môi trường thực thi của hệ thống, bao gồm các máy chủ, cơ sở dữ liệu, và sự tương tác giữa chúng cũng như với các hệ thống bên ngoài. Kiến trúc được lựa chọn là mô hình **client-server ba lớp (3-tier)** hiện đại, bao gồm:

- **Presentation Tier (Client):** Giao diện người dùng trên trình duyệt web.
- **Application Tier (Server):** Máy chủ ứng dụng xử lý logic nghiệp vụ.
- **Data Tier (Database):** Máy chủ cơ sở dữ liệu để lưu trữ và quản lý dữ liệu.

Mô hình này đảm bảo tính linh hoạt, khả năng mở rộng và bảo mật cho hệ thống.

4.1.2. Sơ đồ triển khai hệ thống

Sơ đồ dưới đây được tạo bằng PlantUML, mô tả chi tiết các thành phần và luồng tương tác.



Hình 1: Sơ đồ triển khai hệ thống hỗ trợ Tutor

4.1.3. Mô tả các thành phần

STT	Tên thành phần	Mô tả	Công nghệ/Phần mềm
1	Người dùng	Các tác nhân (Sinh viên, Tutor, Admin) tương tác với hệ thống thông qua trình duyệt web	Trình duyệt Web (Chrome, Firefox, Safari)
2	Web Server (Nginx)	Đóng vai trò là Reverse Proxy, tiếp nhận yêu cầu từ Internet và chuyển tiếp đến Application Server. Tăng cường bảo mật và cân bằng tải.	Nginx
3	Application Server	Chứa toàn bộ logic nghiệp vụ của hệ thống, được đóng gói trong Docker container để dễ dàng triển khai và quản lý.	Java Spring Boot / Node.js, Docker
4	Database Server	Chịu trách nhiệm lưu trữ và quản lý toàn bộ dữ liệu của hệ thống, được triển khai trong Docker container.	PostgreSQL / MySQL, Docker
5	Hệ thống của HCMUT	Các dịch vụ công nghệ thông tin tập trung của trường mà hệ thống cần tích hợp, bao gồm: HCMUT_SSO, HCMUT_DATACORE, HCMUT_LIBRARY.	API (REST/SOAP)
6	Email Service	Dịch vụ bên ngoài chịu trách nhiệm gửi các thông báo và nhắc nhở qua email cho người dùng.	SMTP Server (ví dụ: SendGrid, AWS SES)

Bảng 3: Các thành phần trong sơ đồ triển khai hệ thống hỗ trợ Tutor

4.1.4. Luồng hoạt động

Để làm rõ hơn sự tương tác giữa các thành phần, dưới đây là mô tả luồng hoạt động của hai kịch bản tiêu biểu:

A. Kịch bản 1: Người dùng đăng nhập vào hệ thống

- A.1. Người dùng mở trình duyệt, truy cập vào địa chỉ của hệ thống và nhấn nút đăng nhập thông qua **HCMUT_SSO**.
- A.2. Yêu cầu (*HTTPS Request*) được gửi qua Internet đến **Web Server (Nginx)**.
- A.3. **Nginx** giải mã SSL và chuyển tiếp yêu cầu đến **Application Server**.
- A.4. **Application Server** nhận yêu cầu, chuyển hướng người dùng đến trang đăng nhập của **HCMUT_SSO**.
- A.5. Sau khi người dùng đăng nhập thành công trên **HCMUT_SSO**, dịch vụ này sẽ trả về một *token* xác thực cho **Application Server**.
- A.6. **Application Server** sử dụng *token* này để gọi API đến **HCMUT_DATACORE**, lấy và đồng bộ thông tin cơ bản của người dùng (họ tên, MSSV, khoa, vai trò) vào **Database Server**.
- A.7. Cuối cùng, **Application Server** trả về một phiên làm việc (*session*) và giao diện trang chủ cho người dùng.

B. Kịch bản 2: Tutor tạo lịch rảnh mới

- B.1. **Tutor** sau khi đăng nhập, truy cập chức năng “Tạo lịch rảnh” và điền thông tin (ngày, giờ).
- B.2. Yêu cầu tạo lịch được gửi qua Internet, đi qua **Web Server** và đến **Application Server**.

- B.3. **Application Server** xác thực quyền của Tutor, kiểm tra tính hợp lệ của dữ liệu (ví dụ: không trùng lịch đã có).
- B.4. Nếu hợp lệ, **Application Server** sẽ thực hiện một câu lệnh ghi (**INSERT**) để lưu thông tin lịch rảnh mới vào bảng **Schedules** trong **Database Server**.
- B.5. **Database Server** xác nhận ghi thành công.
- B.6. **Application Server** gửi lại một thông báo “Tạo lịch thành công” cho giao diện của Tutor.

C. Kịch bản 3: Sinh viên đặt lịch học cố định với Tutor

- C.1. Sinh viên, sau khi đăng nhập, chọn môn học và xem các khung giờ rảnh (*availability*) mà Tutor đã thiết lập. Sinh viên chọn một khung giờ phù hợp và nhấn “Đặt lịch”.
- C.2. Yêu cầu đặt lịch (*HTTPS Request*) được gửi qua Internet tới **Web Server (Nginx)**.
- C.3. **Nginx** chuyển tiếp yêu cầu đến **Application Server**.
- C.4. **Application Server** nhận yêu cầu và thực hiện các bước xác thực nghiệp vụ:
 - **Bước 4a:** Truy vấn **Database Server** để kiểm tra lại xem khung giờ mà sinh viên chọn có còn trống không và có nằm trong lịch rảnh hợp lệ của Tutor không.
 - **Bước 4b:** Truy vấn **Database Server** để kiểm tra lịch trình của chính sinh viên đó, đảm bảo không bị trùng với các lịch học/lịch thi khác đã đăng ký.
- C.5. Nếu tất cả các điều kiện đều hợp lệ, **Application Server** sẽ tạo một bản ghi lịch học cố định mới trong **Database Server**, liên kết sinh viên, Tutor, và môn học với khung giờ đã chọn. Trạng thái của lịch được đặt là “Đã xác nhận” (*Confirmed*).
- C.6. Sau khi **Database Server** xác nhận lưu thành công, **Application Server** thực hiện hai hành động song song:
 - **Bước 6a:** Gửi một phản hồi thành công về cho trình duyệt của sinh viên, hiển thị thông báo “Bạn đã đặt lịch thành công”.
 - **Bước 6b:** Kết nối đến **Email Service (SMTP Server)** để gửi email thông báo xác nhận lịch học cho cả sinh viên và Tutor.

4.2. Sơ đồ phát triển (Development View)

4.2.1 Giới thiệu

Sơ đồ phát triển, hay **Sơ đồ gói (Package Diagram)**, mô tả cấu trúc tĩnh và cách tổ chức mã nguồn của hệ thống từ góc nhìn của đội ngũ phát triển. Mục tiêu của sơ đồ này là trình bày một kiến trúc phần mềm có tính module hóa cao, rõ ràng và dễ bảo trì, tuân thủ nguyên tắc *tách biệt các mối quan tâm (Separation of Concerns)*.

Hệ thống được thiết kế theo kiến trúc **phân lớp (Layered Architecture)**, một mô hình phổ biến trong các ứng dụng web hiện đại. Các lớp chính bao gồm:

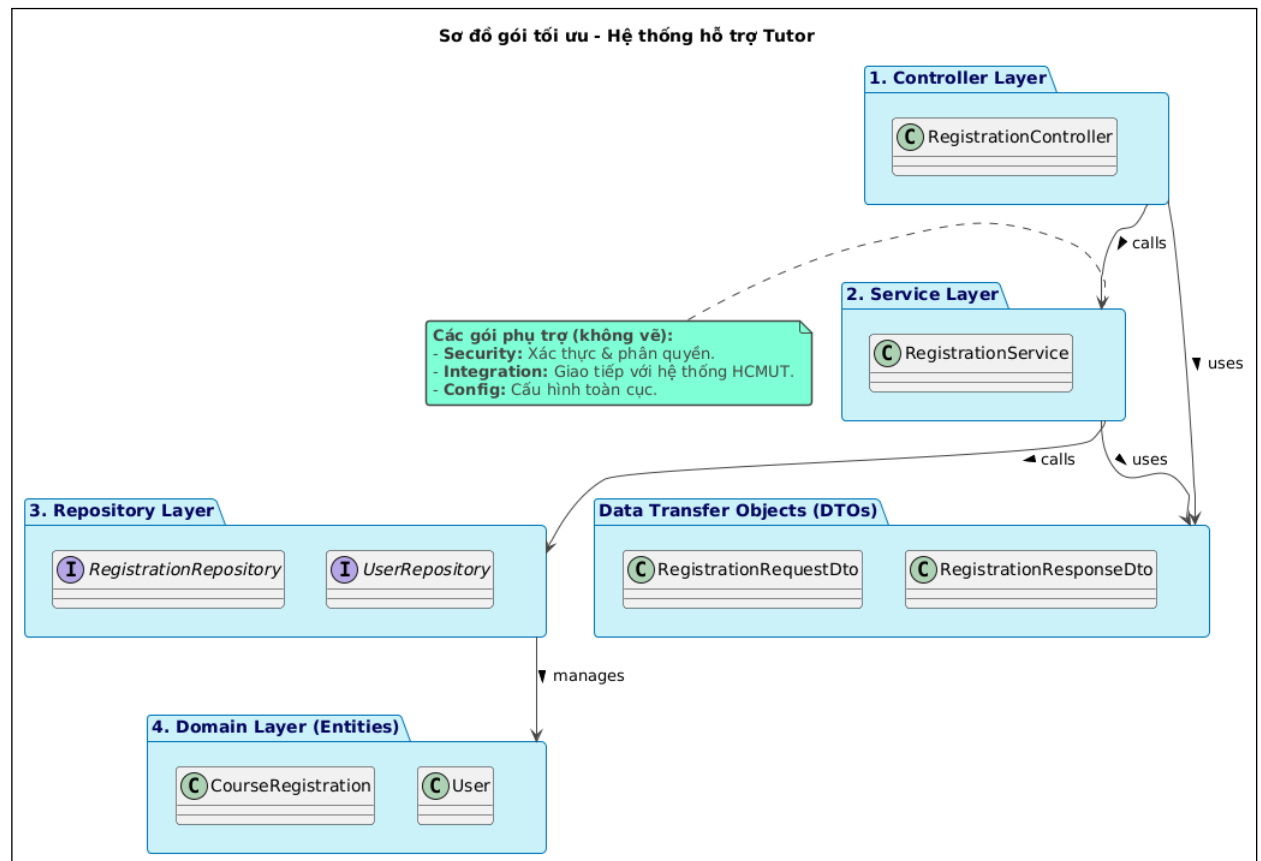
- **Controller Layer:** Tiếp nhận và xử lý các yêu cầu HTTP từ người dùng.
- **Service Layer:** Chứa đựng toàn bộ logic nghiệp vụ cốt lõi của hệ thống.
- **Repository Layer:** Chịu trách nhiệm truy cập và tương tác với cơ sở dữ liệu.
- **Domain/Model Layer:** Định nghĩa các đối tượng và thực thể dữ liệu.

Ngoài ra, hệ thống còn có các gói hỗ trợ cho các chức năng xuyên suốt như:

- **Security:** Quản lý xác thực, phân quyền và bảo mật truy cập.
- **Configuration:** Cấu hình hệ thống, môi trường và thông số hoạt động.
- **Integration:** Tích hợp với các hệ thống bên ngoài như HCMUT_SSO, HCMUT_DATACORE, Email Service,...

4.2.2. Sơ đồ gói (Package Diagram)

Sơ đồ dưới đây minh họa cách mã nguồn được tổ chức thành các gói logic và mối quan hệ phụ thuộc giữa chúng.



Hình 2: Sơ đồ tổ chức các gói của hệ thống

4.2.3. Mô tả các gói

Bảng dưới đây giải thích chức năng của từng gói (package) trong kiến trúc phần mềm.

STT	Tên gói	Chức năng chính
1	Controller Layer	Chứa các lớp chịu trách nhiệm tiếp nhận yêu cầu HTTP từ client, gọi đến các Service tương ứng để xử lý và trả về phản hồi (response).
2	Service Layer	Chứa toàn bộ logic nghiệp vụ của ứng dụng. Đây là nơi các quy trình như ghép cặp Tutor, xử lý đăng ký, tạo báo cáo được thực thi. Gói này điều phối hoạt động giữa Repositories và các thành phần khác.
3	Repository Layer	Chứa các interface/lớp định nghĩa các phương thức để truy cập dữ liệu. Gói này trừu tượng hóa lớp truy cập dữ liệu, giúp Service không cần biết chi tiết về cách dữ liệu được lưu trữ.
4	Domain (Entities)	Chứa các lớp thực thể (Entity) ánh xạ trực tiếp tới các bảng trong cơ sở dữ liệu, định nghĩa cấu trúc dữ liệu cốt lõi của hệ thống.
5	DTOs (Data Transfer Objects)	Chứa các lớp dùng để truyền dữ liệu giữa các lớp, đặc biệt là giữa Controller và Service. Việc sử dụng DTO giúp che giấu cấu trúc của Domain và chỉ truyền đi những dữ liệu cần thiết.
–	Các gói phụ trợ	Bao gồm Security, Integration, Config. Các gói này cung cấp các chức năng xuyên suốt và được sử dụng chủ yếu bởi Service Layer nhưng không được vẽ ra để giữ cho sơ đồ đơn giản.

Bảng 4: Chức năng của từng gói (package) trong kiến trúc phần mềm

4.2.4. Luồng xử lý dữ liệu qua các lớp

Để minh họa cách các gói tương tác với nhau, luồng xử lý cho chức năng “Sinh viên đăng ký môn học” (UC-04):

- **Request:** Sinh viên gửi yêu cầu đăng ký môn học từ giao diện người dùng. Yêu cầu này được gửi đến một endpoint trong Controller Layer.
- **Controller Layer:** Lớp `RegistrationController` nhận yêu cầu. Nó xác thực dữ liệu đầu vào (đóng gói trong một đối tượng `RegistrationRequestDto` từ gói DTOs) và gọi phương thức trong Service Layer.
- **Service Layer:** Lớp `RegistrationService` thực thi logic nghiệp vụ. Nó sử dụng các lớp trong Repository Layer để kiểm tra các quy tắc nghiệp vụ (ví dụ: giới hạn số môn đăng ký).
- **Repository Layer:** Các lớp như `UserRepository` và `RegistrationRepository` tương tác với cơ sở dữ liệu. Chúng làm việc với các đối tượng từ Domain Layer (ví dụ: `User`, `CourseRegistration`).
- **Response:** Sau khi xử lý xong, Service Layer trả về kết quả (thường dưới dạng một DTO khác) cho Controller Layer, và Controller sẽ tạo phản hồi HTTP gửi về cho client.

4.3. Sơ đồ lớp chi tiết (Detailed Class Diagram)

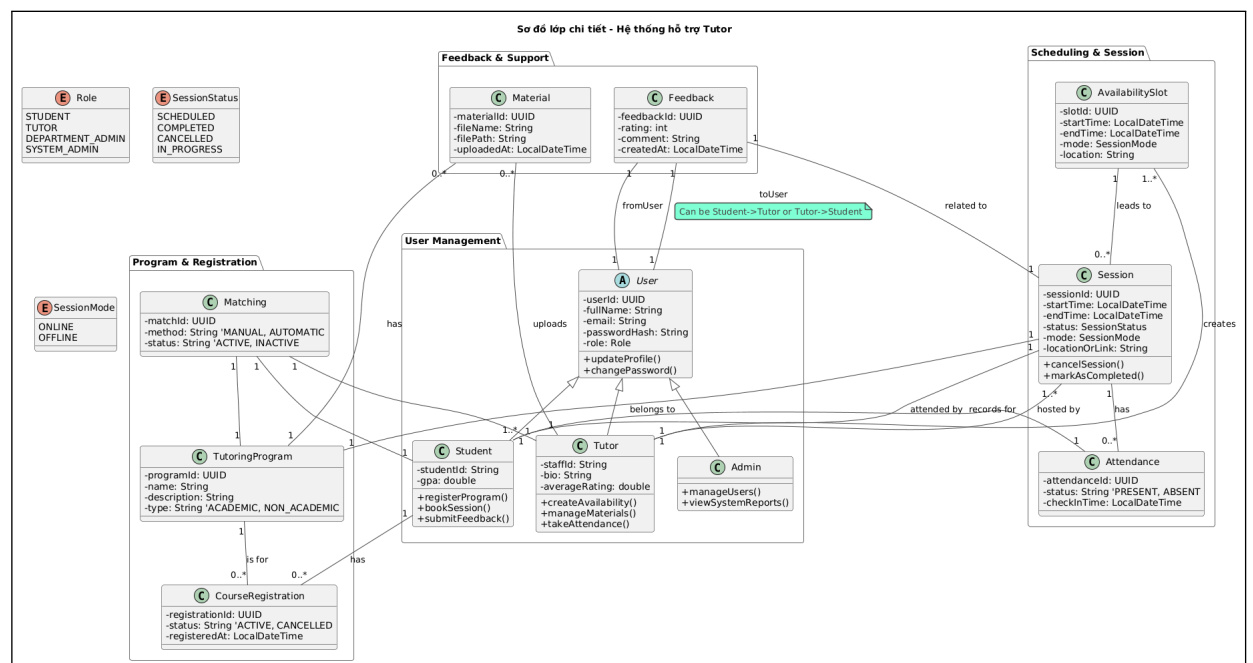
4.3.1. Giới thiệu

Sơ đồ lớp là một sơ đồ cấu trúc tĩnh trong UML, mô tả chi tiết các lớp (*class*), thuộc tính (*attributes*), phương thức (*methods*) và mối quan hệ giữa các lớp đó trong **Hệ thống hỗ trợ Tutor**. Sơ đồ này đóng vai trò là bản thiết kế chi tiết cho việc triển khai mã nguồn, đảm bảo hệ thống được xây dựng một cách nhất quán, có cấu trúc và dễ dàng bảo trì, mở rộng.

Sơ đồ được tổ chức thành các cụm chức năng chính để phản ánh cấu trúc module của hệ thống:

- **Cụm Quản lý Người dùng (User Management):** Định nghĩa các loại người dùng và vai trò.
- **Cụm Chương trình và Đăng ký (Program & Registration):** Quản lý các chương trình học và việc đăng ký của sinh viên.
- **Cụm Lịch trình và Buổi học (Scheduling & Session):** Xử lý việc tạo lịch, đặt lịch và quản lý các buổi học.
- **Cụm Đánh giá và Phản hồi (Feedback & Support):** Quản lý các chức năng hỗ trợ như đánh giá, tài liệu.

4.3.2. Sơ đồ lớp chi tiết của hệ thống



Hình 3: Sơ đồ lớp chi tiết của hệ thống

4.3.3. Mô tả cụm chức năng

- **Cụm Quản lý Người dùng (User Management):**
 - Lớp trừu tượng User là lớp cha, chứa các thuộc tính và phương thức chung cho mọi người dùng.
 - Các lớp Student, Tutor, và Admin kế thừa từ User, bổ sung các thuộc tính và hành vi đặc thù cho từng vai trò. Thiết kế này giúp tái sử dụng mã nguồn và thể hiện rõ mối quan hệ "is-a" (là một).
- **Cụm Chương trình và Đăng ký (Program & Registration)**
 - TutoringProgram đại diện cho một chương trình học (ví dụ: "Phụ đạo môn Cấu trúc dữ liệu").
 - CourseRegistration là một lớp liên kết (association class) thể hiện việc một Student đăng ký một TutoringProgram.

- Matching ghi lại kết quả ghép cặp giữa Student và Tutor cho một chương trình cụ thể.

- **Cụm Lịch trình và Buổi học (Scheduling & Session):**

- AvailabilitySlot là các khung giờ rảnh mà Tutor đã tạo ra.
- Session là một buổi học cụ thể được tạo ra khi một Student đặt lịch trong một AvailabilitySlot. Đây là lớp trung tâm, liên kết Tutor, Student, và TutoringProgram tại một thời điểm cụ thể.
- Session là một buổi học cụ thể được tạo ra khi một Student đặt lịch trong một AvailabilitySlot. Đây là lớp trung tâm, liên kết Tutor, Student, và TutoringProgram tại một thời điểm cụ thể.

- **Cụm Đánh giá và Phản hồi (Feedback & Support):**

- Session là một buổi học cụ thể được tạo ra khi một Student đặt lịch trong một AvailabilitySlot. Đây là lớp trung tâm, liên kết Tutor, Student, và TutoringProgram tại một thời điểm cụ thể.
- Feedback là một lớp linh hoạt, có thể ghi lại đánh giá từ Student đến Tutor hoặc ngược lại, được liên kết với một Session cụ thể để đảm bảo tính xác thực. Mỗi quan hệ giữa Feedback và User được định nghĩa rõ ràng qua hai vai trò "fromUser" và "toUser".

4.4. Mô tả các lớp chi tiết và phương thức

4.4.1. Cụm Quản lý Người dùng (User Management)

Đây là cụm lõi, định nghĩa các đối tượng người dùng trong hệ thống. Việc sử dụng kế thừa từ một lớp trừu tượng **User** giúp tối ưu hóa cấu trúc và tránh lặp lại mã nguồn.

Mô tả: Là lớp cơ sở (base class) cho tất cả các loại người dùng trong hệ thống. Lớp này chứa các thông tin và hành vi chung nhất như thông tin định danh, email, và các chức năng quản lý tài khoản cơ bản. Không thể tạo đối tượng trực tiếp từ lớp này.

Thuộc tính (Attributes):

Tên thuộc tính	Kiểu dữ liệu	Mô tả
- userId	UUID	Khóa chính, một mã định danh duy nhất cho mỗi người dùng trong hệ thống.
- fullName	String	Họ và tên đầy đủ của người dùng.
- email	String	Địa chỉ email của người dùng, được sử dụng để đăng nhập và nhận thông báo.
- passwordHash	String	Chuỗi đã được băm (hash) từ mật khẩu của người dùng. Lưu ý: Hệ thống không bao giờ lưu mật khẩu ở dạng văn bản gốc.
- role	Role	Vai trò của người dùng trong hệ thống (STUDENT, TUTOR, ADMIN), quyết định các quyền truy cập chức năng.

Phương thức (Methods):

Tên phương thức	Tham số	Kiểu trả về	Mô tả
+ updateProfile()	profileData: Map	void	Cập nhật thông tin cá nhân chung như họ tên, email.
+ changePassword()	oldPass: String, newPass: String	boolean	Thay đổi mật khẩu của người dùng sau khi xác thực mật khẩu cũ.

Mô tả: Đại diện cho một người dùng sinh viên. Lớp này kế thừa tất cả các thuộc tính và phương thức từ lớp **User**, đồng thời bổ sung các thông tin và hành vi đặc thù liên quan đến việc học tập và tham gia chương trình Tutor.

Thuộc tính (Attributes):

Tên thuộc tính	Kiểu dữ liệu	Mô tả
- studentId	String	Mã số sinh viên (MSSV).
- gpa	double	Điểm trung bình tích lũy của sinh viên, có thể được dùng làm một tiêu chí gợi ý Tutor.

Phương thức (Methods):

Tên phương thức	Tham số	Kiểu trả về	Mô tả (Liên kết với Use Case)
+ registerProgram()	program: TutoringProgram	boolean	Thực hiện chức năng đăng ký tham gia một chương trình học (UC-04, UC-24, UC-25).
+ bookSession()	slot: AvailabilitySlot	Session	Thực hiện chức năng đặt một buổi học cố định dựa trên lịch rảnh của Tutor (UC-09).
+ submitFeedback()	feedback: Feedback	boolean	Gửi đánh giá và phản hồi về chất lượng của một buổi học hoặc một Tutor (UC-17).

Mô tả: Đại diện cho một người hướng dẫn (Tutor), có thể là giảng viên, sinh viên năm trên, hoặc nghiên cứu sinh. Lớp này kế thừa từ **User** và có thêm các thuộc tính về chuyên môn cũng như các hành vi liên quan đến việc giảng dạy.

Thuộc tính (Attributes):

Tên thuộc tính	Kiểu dữ liệu	Mô tả
- staffId	String	Mã số định danh của Tutor (có thể là mã nhân viên hoặc MSSV nếu là sinh viên).
- bio	String	Một đoạn giới thiệu ngắn về chuyên môn, kinh nghiệm và thành tích của Tutor.
- averageRating	double	Điểm đánh giá trung bình của Tutor, được tự động tính toán từ các Feedback của sinh viên.

Phương thức (Methods):

Tên phương thức	Tham số	Kiểu trả về	Mô tả (Liên kết với Use Case)
+ createAvailability()	slotDetails: Map	AvailabilitySlot	Tạo ra các khung giờ rảnh để sinh viên có thể đặt lịch học (UC-08).
+ manageMaterials()	material: Material, action: String	boolean	Quản lý tài liệu học tập: tải lên, chỉnh sửa, hoặc xóa tài liệu (UC-15).
+ takeAttendance()	session: Session, student: Student	void	Thực hiện điểm danh sinh viên trong một buổi học (UC-13).

Mô tả: Đại diện cho người dùng quản trị, bao gồm quản trị viên của Khoa/Bộ môn và quản

trì viên hệ thống. Lớp này kế thừa từ **User** và có các quyền cao nhất để giám sát và quản lý toàn bộ hoạt động của hệ thống.

Phương thức (Methods):

Tên phương thức	Tham số	Kiểu trả về	Mô tả (Liên kết với Use Case)
+ manageUsers()	user: User, action: String	boolean	Quản lý tài khoản người dùng: kích hoạt, khóa, hoặc phân quyền.
+ viewSystemReports()	reportType: String	Report	Xem các báo cáo tổng hợp về hoạt động của hệ thống, ví dụ như báo cáo chất lượng Tutor, báo cáo học tập của sinh viên (UC-19, UC-22).

4.4.2. Cụm Chương trình và Đăng ký (Program & Registration)

Cụm này định nghĩa các lớp liên quan đến việc tổ chức các chương trình học và quản lý quá trình đăng ký của sinh viên, cũng như kết quả ghép cặp giữa sinh viên và Tutor.

Mô tả: Đại diện cho một chương trình học thuật hoặc phi học thuật cụ thể được cung cấp trên hệ thống. Ví dụ: “Phụ đạo môn Giải tích 1”, “Luyện thi cuối kỳ môn Lập trình nâng cao”, hoặc “Workshop kỹ năng thuyết trình”.

Thuộc tính (Attributes):

Tên thuộc tính	Kiểu dữ liệu	Mô tả
- programId	UUID	Khóa chính, mã định danh duy nhất cho mỗi chương trình.
- name	String	Tên của chương trình (ví dụ: “Cấu trúc dữ liệu và giải thuật”).
- description	String	Mô tả chi tiết về nội dung, mục tiêu và đối tượng của chương trình.
- type	String	Phân loại chương trình là “ACADEMIC” (học thuật) hay “NON_ACADEMIC” (phi học thuật).

Phương thức (Methods):

Tên phương thức	Tham số	Kiểu trả về	Mô tả
+ getRegisteredStudents()	(none)	List<Student>	Lấy danh sách tất cả sinh viên đã đăng ký tham gia chương trình này.
+ getAvailableTutors()	(none)	List<Tutor>	Lấy danh sách các Tutor có chuyên môn và đang nhận hướng dẫn cho chương trình này.

Mô tả: Đây là một lớp liên kết, ghi lại thông tin về việc một **Student** đã đăng ký tham gia một **TutoringProgram**. Mỗi đối tượng của lớp này là một bản ghi đăng ký duy nhất.

Thuộc tính (Attributes):

Tên thuộc tính	Kiểu dữ liệu	Mô tả
- registrationId	UUID	Khóa chính, mã định danh duy nhất cho mỗi lượt đăng ký.
- status	String	Trạng thái của việc đăng ký. Có thể là “ACTIVE” (đang hoạt động) hoặc “CANCELLED” (đã hủy).
- registeredAt	LocalDateTime	Dấu thời gian ghi lại thời điểm sinh viên thực hiện đăng ký.

Phương thức (Methods):

Tên phương thức	Tham số	Kiểu trả về	Mô tả (Liên kết với Use Case)
+ cancelRegistration()	(none)	void	Thay đổi trạng thái của đơn đăng ký thành “CANCELLED”. Tương ứng với chức năng Hủy đăng ký môn học (UC-05).

Mô tả: Ghi lại kết quả của việc ghép cặp thành công giữa một **Student** và một **Tutor** cho một **TutoringProgram** cụ thể. Lớp này giúp hệ thống quản lý các cặp Tutor-Mentee đang hoạt động.

Thuộc tính (Attributes):

Tên thuộc tính	Kiểu dữ liệu	Mô tả
- matchId	UUID	Khóa chính, mã định danh duy nhất cho mỗi cặp ghép.
- method	String	Phương thức ghép cặp. Có thể là “MANUAL” (sinh viên tự chọn) hoặc “AUTOMATIC” (hệ thống tự động gợi ý).
- status	String	Trạng thái của cặp ghép. Có thể là “ACTIVE” (đang hoạt động) hoặc “INACTIVE” (đã kết thúc hoặc bị hủy).

Phương thức (Methods):

Tên phương thức	Tham số	Kiểu trả về	Mô tả (Liên kết với Use Case)
+ deactivateMatch()	(none)	void	Chuyển trạng thái của cặp ghép thành “INACTIVE”, ví dụ như khi môn học kết thúc hoặc sinh viên hủy ghép cặp.

4.4.3. Cụm Lịch trình và Buổi học (Scheduling & Session)

Cụm này chịu trách nhiệm quản lý thời gian của Tutor, quá trình đặt lịch của sinh viên, và thông tin chi tiết của từng buổi học. Đây là trung tâm của các hoạt động tương tác trong hệ thống.

Mô tả: Đại diện cho một khung thời gian cụ thể mà một **Tutor** đã đăng ký là rảnh và sẵn sàng để nhận lịch dạy. Đây là “nguyên liệu” đầu vào để sinh viên có thể đặt lịch.

Thuộc tính (Attributes):



Tên thuộc tính	Kiểu dữ liệu	Mô tả
- slotId	UUID	Khóa chính, mã định danh duy nhất cho mỗi khung giờ rảnh.
- startTime	LocalDateTime	Thời điểm bắt đầu của khung giờ rảnh.
- endTime	LocalDateTime	Thời điểm kết thúc của khung giờ rảnh.
- mode	SessionMode	Hình thức học trong khung giờ này: ONLINE hoặc OFFLINE.
- location	String	Địa điểm (nếu là OFFLINE) hoặc có thể để trống.

Phương thức (Methods):

Tên phương thức	Tham số	Kiểu trả về	Mô tả
+ isSlotAvailable()	(none)	boolean	Kiểm tra xem khung giờ này còn chỗ trống để sinh viên đặt lịch hay không (dựa trên số lượng Session đã được tạo).
+ updateSlot()	slotDetails: Map	void	Cho phép Tutor chỉnh sửa thông tin của khung giờ rảnh (ví dụ: thay đổi thời gian, hình thức).

Mô tả: Đại diện cho một buổi học thực tế đã được lên lịch, kết nối một **Tutor** với một hoặc nhiều **Student** tại một thời điểm cụ thể. Đây là kết quả của việc sinh viên đặt lịch thành công vào một **AvailabilitySlot**.

Thuộc tính (Attributes):

Tên thuộc tính	Kiểu dữ liệu	Mô tả
- sessionId	UUID	Khóa chính, mã định danh duy nhất cho mỗi buổi học.
- startTime	LocalDateTime	Thời điểm bắt đầu thực tế của buổi học.
- endTime	LocalDateTime	Thời điểm kết thúc thực tế của buổi học.
- status	SessionStatus	Trạng thái của buổi học (SCHEDULED, COMPLETED, CANCELLED, IN_PROGRESS).
- mode	SessionMode	Hình thức của buổi học (ONLINE hoặc OFFLINE).
- locationOrLink	String	Chứa địa điểm phòng học (nếu là OFFLINE) hoặc đường link phòng học trực tuyến (nếu là ONLINE).

Phương thức (Methods):

Tên phương thức	Tham số	Kiểu trả về	Mô tả (Liên kết với Use Case)
+ cancelSession()	reason: String	boolean	Hủy một buổi học đã được lên lịch. Phương thức này sẽ cập nhật status thành CANCELLED (UC-10).
+ markAsCompleted()	(none)	void	Đánh dấu buổi học đã hoàn thành. Phương thức này cập nhật status thành COMPLETED (UC-14).
+ getAttendees()	(none)	List<Student>	Lấy danh sách tất cả sinh viên đã đăng ký tham gia buổi học này.

Mô tả: Ghi lại thông tin điểm danh của một **Student** cụ thể cho một **Session** cụ thể. Lớp này giúp theo dõi sự chuyên cần của sinh viên.

Thuộc tính (Attributes):

Tên thuộc tính	Kiểu dữ liệu	Mô tả
- attendanceId	UUID	Khóa chính, mã định danh duy nhất cho mỗi lượt điểm danh.
- status	String	Trạng thái điểm danh: “PRESENT” (có mặt), “ABSENT” (vắng mặt).
- checkInTime	LocalDateTime	Thời điểm Tutor thực hiện điểm danh.

Phương thức (Methods):

Tên phương thức	Tham số	Kiểu trả về	Mô tả (Liên kết với Use Case)
+ updateStatus()	newStatus: String	void	Cập nhật lại trạng thái điểm danh, ví dụ trong trường hợp cần điều chỉnh sau khi điểm danh ban đầu.

4.4.4. Cụm Đánh giá và Phản hồi (Feedback & Support)

Cụm này bao gồm các lớp hỗ trợ cho quá trình dạy và học, cụ thể là quản lý tài liệu học tập và thu thập các phản hồi, đánh giá từ người dùng để cải thiện chất lượng chương trình.

Mô tả: Đại diện cho một phản hồi hoặc đánh giá mà một người dùng gửi cho một người dùng khác, liên quan đến một buổi học cụ thể. Lớp này được thiết kế linh hoạt để hỗ trợ cả hai chiều: sinh viên đánh giá Tutor và Tutor nhận xét về sinh viên.

Thuộc tính (Attributes):

Tên thuộc tính	Kiểu dữ liệu	Mô tả
- feedbackId	UUID	Khóa chính, mã định danh duy nhất cho mỗi phản hồi.
- rating	int	Điểm số đánh giá (ví dụ: từ 1 đến 5 sao).
- comment	String	Nội dung nhận xét, bình luận chi tiết.
- createdAt	LocalDateTime	Dấu thời gian ghi lại thời điểm phản hồi được tạo.

Mối quan hệ đặc biệt:

Lớp này có hai mối quan hệ tới lớp **User**: **fromUser** (để xác định người gửi) và **toUser** (để xác định người nhận).

Nó cũng có một mối quan hệ tới lớp **Session** để đảm bảo mỗi phản hồi đều gắn liền với một bối cảnh (buổi học) cụ thể.

Phương thức (Methods):

Tên phương thức	Tham số	Kiểu trả về	Mô tả (Liên kết với Use Case)
+ saveFeedback()	(none)	boolean	Lưu thông tin phản hồi vào cơ sở dữ liệu. Được gọi khi sinh viên hoặc Tutor gửi đánh giá (UC-17, UC-18).
+ isValid()	(none)	boolean	Kiểm tra tính hợp lệ của phản hồi (ví dụ: điểm rating phải nằm trong khoảng cho phép).



Mô tả: Đại diện cho một tài liệu học tập (ví dụ: slide bài giảng, bài tập, đề cương) mà **Tutor** tải lên hệ thống để chia sẻ với các sinh viên trong chương trình học của mình.

Thuộc tính (Attributes):

Tên thuộc tính	Kiểu dữ liệu	Mô tả
- materialId	UUID	Khóa chính, mã định danh duy nhất cho mỗi tài liệu.
- fileName	String	Tên gốc của tệp tài liệu.
- filePath	String	Đường dẫn lưu trữ tệp trên máy chủ hoặc dịch vụ lưu trữ đám mây.
- uploadedAt	LocalDateTime	Dấu thời gian ghi lại thời điểm tài liệu được tải lên.

Phương thức (Methods):

Tên phương thức	Tham số	Kiểu trả về	Mô tả (Liên kết với Use Case)
+ download()	(none)	File	Thực hiện chức năng cho phép người dùng (sinh viên) tải tài liệu về máy (UC-16).
+ delete()	(none)	void	Xóa tệp tài liệu khỏi hệ thống. Chỉ người tải lên (Tutor) mới có quyền này (UC-15).
+ updateInfo()	newName: String, newDesc: String	void	Cập nhật thông tin mô tả của tài liệu (UC-15).