|  |  |  |
| --- | --- | --- |
| **Stakeholder** | **Vai trò** | **Nguồn yêu cầu** |
| **Học viên (Student)** | Người dùng cuối của hệ thống, đăng ký tài khoản, mua và học khóa học. | Muốn trải nghiệm học tập dễ dùng, truy cập nhanh, hỗ trợ đa nền tảng. |
| **Giảng viên (Instructor)** | Cung cấp nội dung khóa học, đăng bài giảng, cập nhật tài liệu, theo dõi học viên. | Cần hệ thống dễ đăng tải bài, quản lý học viên, thống kê doanh thu. |
| **Quản trị viên (Admin)** | Quản lý toàn bộ hệ thống: tài khoản, khóa học, thanh toán, nội dung, báo cáo. | Cần giao diện quản trị mạnh mẽ, bảo mật, dễ theo dõi thống kê. |
| **Bộ phận kỹ thuật (IT Support)** | Bảo trì, vận hành và khắc phục sự cố hệ thống. | Cần dữ liệu log, quyền truy cập hệ thống và cảnh báo lỗi. |
| **Bộ phận thanh toán (Payment Provider)** | Cung cấp cổng thanh toán online (VNPay, Momo, Paypal, v.v.). | Cần API tích hợp an toàn, hỗ trợ giao dịch nhanh và bảo mật. |
| **Bộ phận marketing** | Quảng bá khóa học, theo dõi hành vi người dùng. | Cần hệ thống tracking, gửi email/sms marketing, thống kê doanh thu. |
| **Cơ quan quản lý giáo dục (nếu có)** | Kiểm định và giám sát nội dung giáo dục. | Yêu cầu tuân thủ pháp luật và bản quyền nội dung. |

Bài3:

Bài 4:

Bài **tổng hợp kiến thức hệ thống thông tin** dựa trên mô tả tình huống doanh nghiệp logistics muốn xây dựng hệ thống quản lý giao hàng online

**1. Mô tả nghiệp vụ**

Doanh nghiệp logistics cần một **hệ thống quản lý giao hàng online** có các chức năng:

* **Cập nhật đơn hàng:** Nhập, quản lý, và xử lý các đơn hàng mới.
* **Theo dõi trạng thái:** Giám sát vị trí và tình trạng đơn hàng trong suốt quá trình vận chuyển.
* **Phân tích tuyến đường:** Hỗ trợ tối ưu hóa tuyến giao hàng, tiết kiệm thời gian và chi phí vận hành.

2**. Loại hệ thống thông tin phù hợp**  
Hệ thống quản lý giao hàng online của doanh nghiệp logistics nên là **hệ thống tích hợp gồm TPS + MIS + DSS**, trong đó:

* TPS phục vụ giao dịch hằng ngày,
* MIS phục vụ báo cáo quản lý,
* DSS phục vụ tối ưu hóa tuyến và ra quyết định chiến lược.

**3 Mô hình phát triển phần mềm phù hợp** **Kết luận:**  
Với hệ thống **quản lý giao hàng online** có tính năng cập nhật liên tục, yêu cầu thay đổi linh hoạt và tích hợp nhiều module (TPS, MIS, DSS):  
 **Mô hình phát triển phù hợp nhất là *Agile* (hoặc *Spiral*).**

**4. Tóm tắt đề xuất**

|  |  |  |
| --- | --- | --- |
| **Khía cạnh** | **Lựa chọn đề xuất** | **Lý do** |
| **Loại hệ thống** | TPS + MIS + DSS | Đảm bảo xử lý giao dịch, tổng hợp báo cáo và hỗ trợ ra quyết định. |
| **Mô hình phát triển** | Agile (Scrum) | Linh hoạt, phù hợp hệ thống có yêu cầu thay đổi và mở rộng nhanh. |

**Kết luận cuối cùng:**

Hệ thống quản lý giao hàng online là **hệ thống tích hợp đa tầng thông tin (TPS–MIS–DSS)**, nên được phát triển theo **mô hình Agile** để đảm bảo khả năng mở rộng, tối ưu hóa và cập nhật liên tục theo nhu cầu thực tế doanh nghiệp.

Bài 5:

**Các lớp chính trong hệ thống**

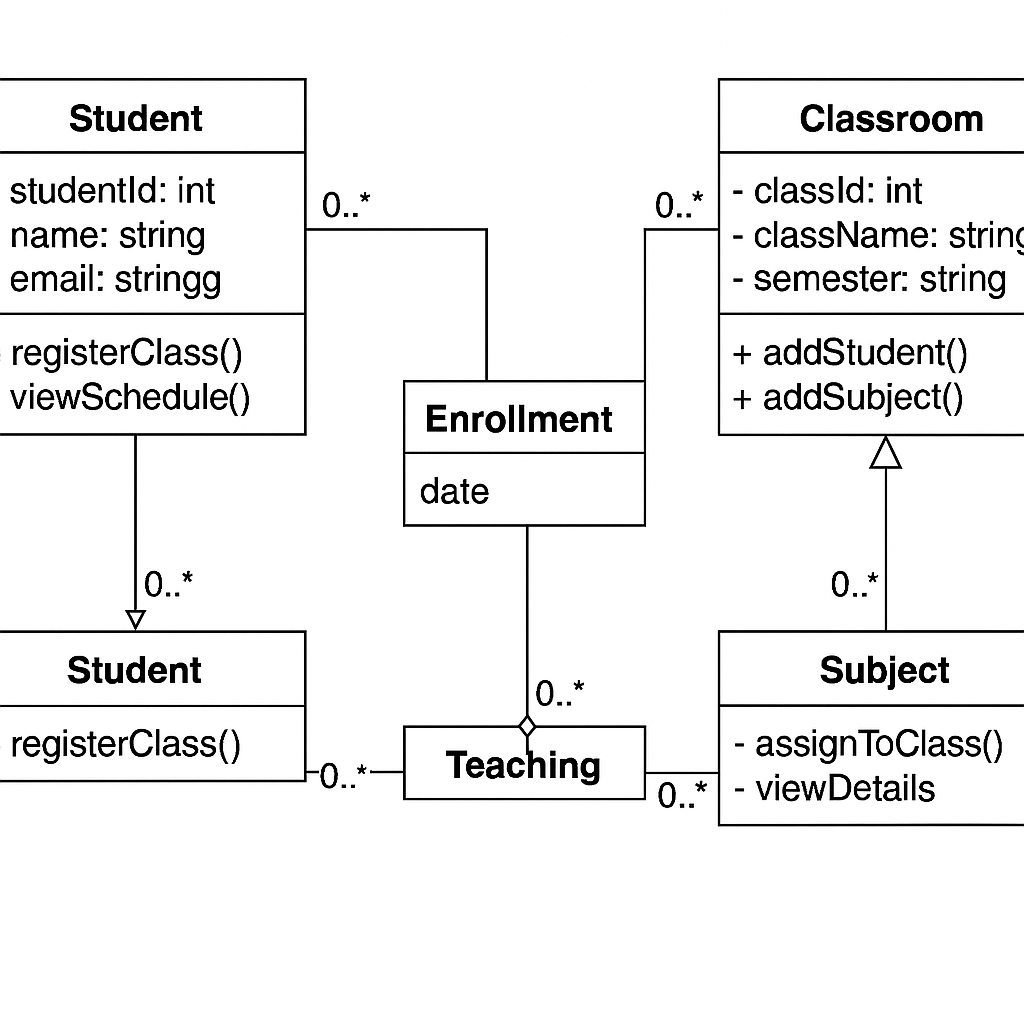
| **Lớp** | **Thuộc tính (Attributes)** | **Phương thức (Methods)** |
| --- | --- | --- |
| **Student** *(Sinh viên)* | - studentId : int - name : string - email : string | + registerClass() + viewSchedule() |
| **Classroom** *(Lớp học)* | - classId : int - className : string - semester : string | + addStudent() + addSubject() + viewList() |
| **Subject** *(Môn học)* | - subjectId : int - subjectName : string - credit : int | + assignToClass() + viewDetails() |

**Phân tích mối quan hệ & Multiplicity**

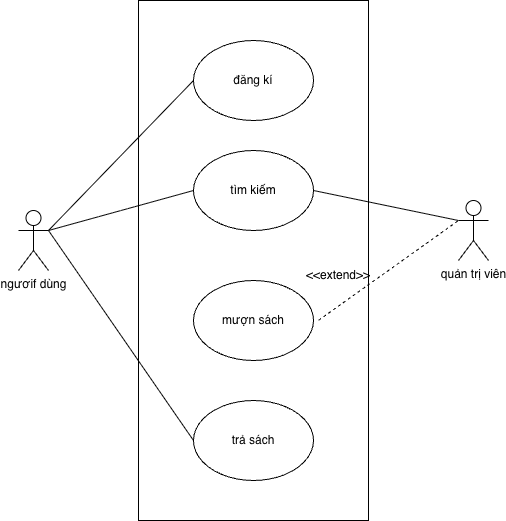
| **Quan hệ** | **Loại quan hệ** | **Multiplicity** | **Giải thích** |
| --- | --- | --- | --- |
| **Student – Classroom** | Nhiều – Nhiều (n–n) | Student 0..\* ↔ 0..\* Classroom | Một sinh viên có thể học nhiều lớp, và mỗi lớp có nhiều sinh viên. |
| **Classroom – Subject** | Nhiều – Nhiều (n–n) | Classroom 0..\* ↔ 0..\* Subject | Một lớp học có thể giảng nhiều môn, và một môn có thể dạy ở nhiều lớp. |

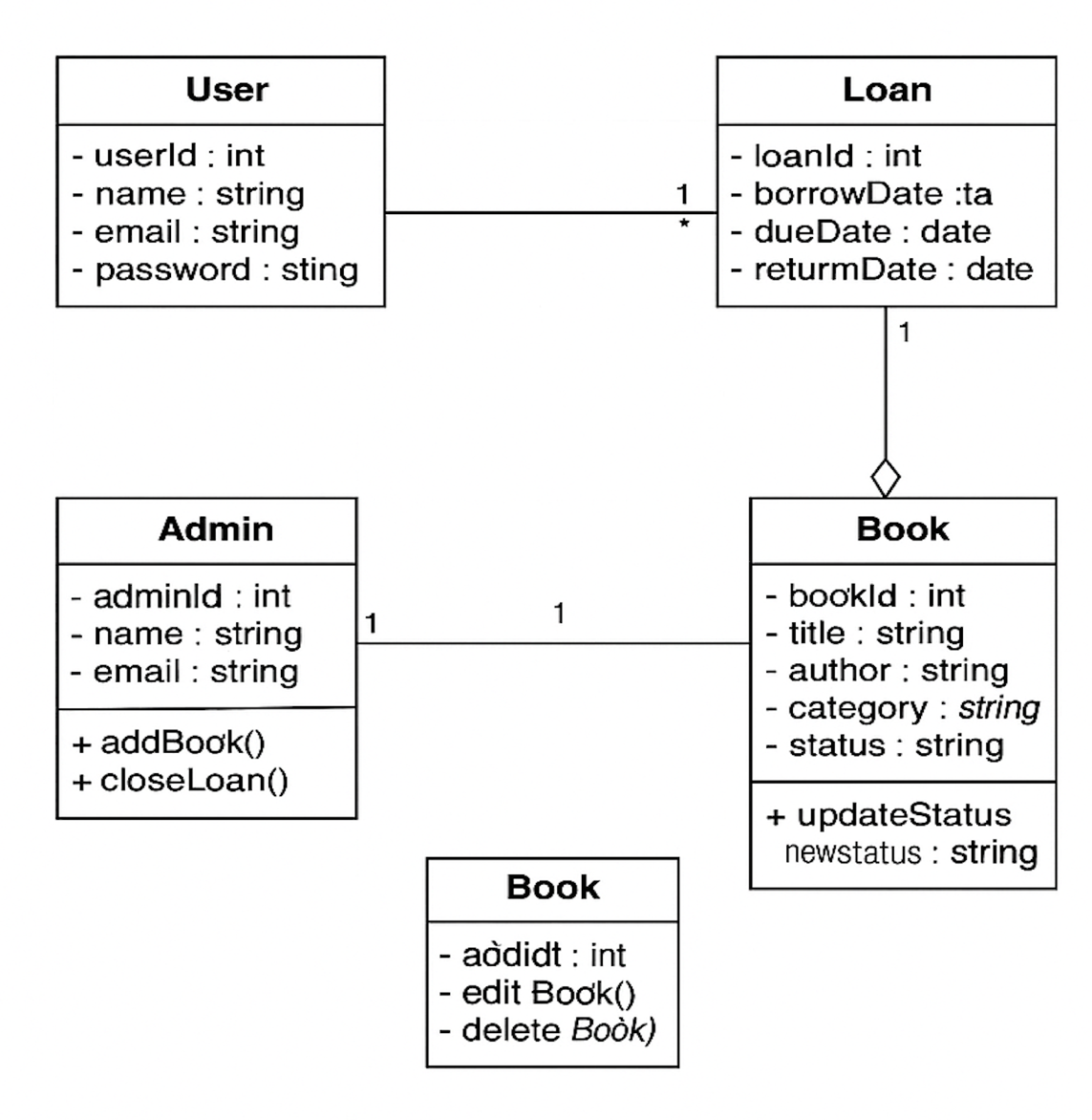
Vì đều là **n–n**, nên ta cần **lớp trung gian (associative class)** để biểu diễn quan hệ này chính xác hơn:

* Enrollment (Ghi danh): giữa **Student** và **Classroom**
* Teaching (Giảng dạy): giữa **Classroom** và **Subject**



Bài6:

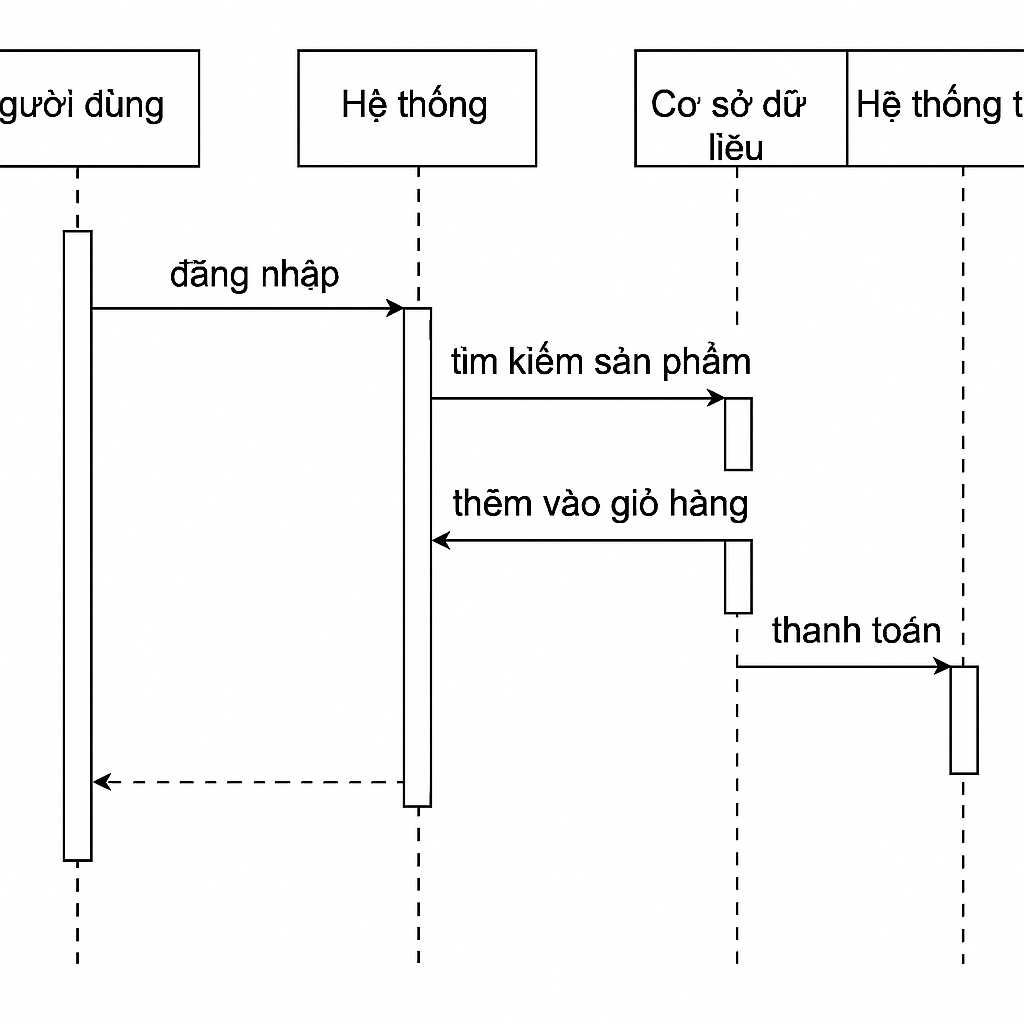




**Giải thích logic chuyển đổi từ Use Case → Class Diagram**

|  |  |  |
| --- | --- | --- |
| **Bước** | **Hành động phân tích** | **Kết quả** |
| 1 | Mỗi **Use Case** đại diện cho **hành động hoặc chức năng chính** của hệ thống. | Xác định được các lớp chịu trách nhiệm cho từng hành vi. |
| 2 | Các **Actor (User, Admin)** trở thành **class hoặc entity** trong hệ thống. | Lớp User, Admin được tạo ra. |
| 3 | Các **đối tượng chính mà hành động tác động lên** (ví dụ: Sách, Mượn sách). | Sinh ra các lớp Book và Loan. |
| 4 | Xác định **mối quan hệ và multiplicity** dựa trên hành vi. | Ví dụ: User có thể mượn nhiều sách → 1–n giữa User và Loan. |
| 5 | Gán các **thuộc tính và phương thức** tương ứng với hành động trong Use Case. | borrowBook(), returnBook(), addBook()… được thêm vào từng class. |

Bai7:



Bài8:

