



BÀI TẬP LỚN

HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

(Mã học phần: IT3103)

Đề tài số 6:

QUẢN LÝ HỆ THỐNG TRƯỜNG HỌC

Sinh viên thực hiện – Nhóm 6.2

STT	Họ và tên	Mã sinh viên
1	Trần Quang Huy	20226109
2	Nguyễn Lương Hoàng Tùng	20226129
3	Ngô Mạnh Hùng	20226083
4	Nguyễn Trung Hiếu	20226082
5	Trịnh Quang Minh	20226092

Giảng viên hướng dẫn: PGS.TS Lê Đức Hậu

Hà Nội, tháng 12 năm 2024

CHƯƠNG 1. KHẢO SÁT, ĐẶC TẢ YÊU CẦU BÀI TOÁN

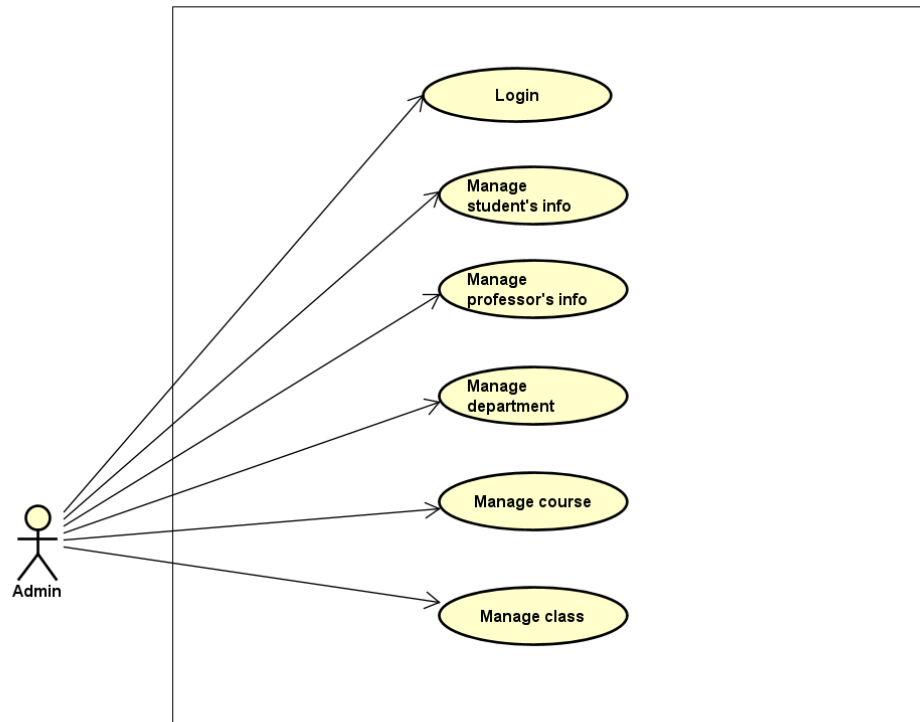
1.1. Mô tả yêu cầu bài toán

Hệ thống quản lý trường học được xây dựng trên nền tảng **Java console** nhằm cung cấp một công cụ đơn giản, hiệu quả và linh hoạt để quản lý các nguồn tài nguyên của trường đại học. Hệ thống này được thiết kế để hỗ trợ các công việc liên quan đến **quản lý giảng viên, sinh viên, lớp học và khóa học**, đồng thời giúp giảm thiểu công việc thủ công, nâng cao tính chính xác và hiệu quả trong quy trình quản lý.

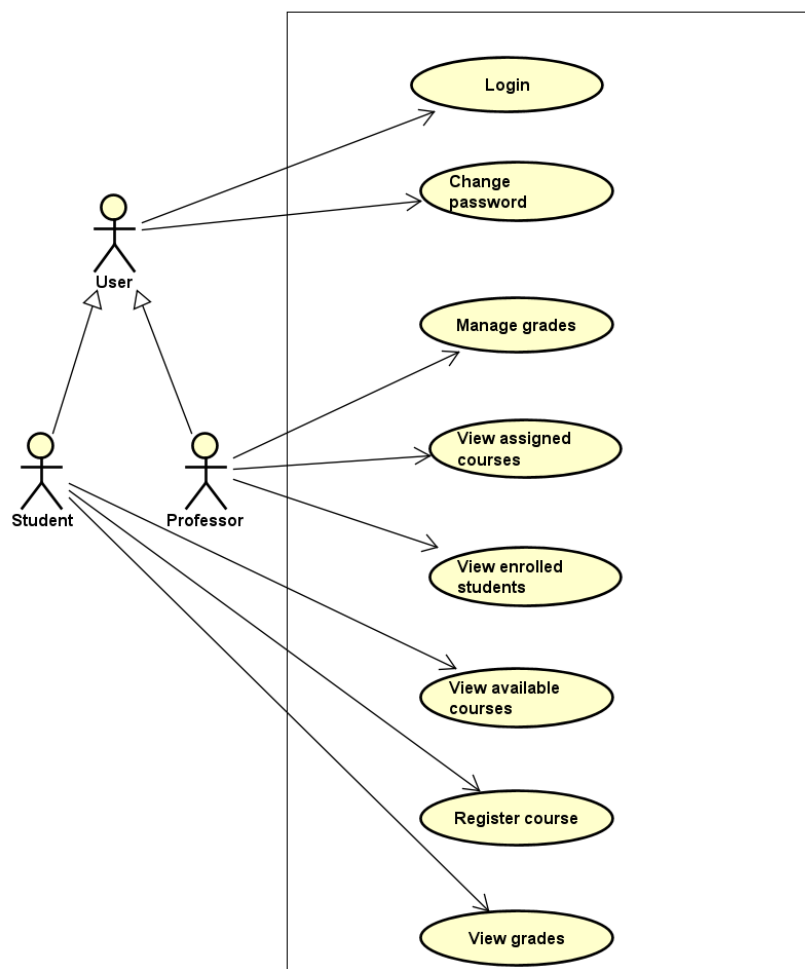
Mục tiêu của Hệ thống

1. **Quản lý các nguồn tài nguyên chính của trường đại học:**
 - Hệ thống tập trung vào việc lưu trữ và tổ chức thông tin liên quan đến **giảng viên, sinh viên, lớp học và khóa học**.
 - Dữ liệu sẽ được tổ chức rõ ràng và có mối quan hệ chặt chẽ để dễ dàng truy xuất và cập nhật.
2. **Hỗ trợ các tác vụ quản lý thông thường:**
 - Cung cấp các tính năng quản lý như **thêm, sửa, xóa và tìm kiếm thông tin** của giảng viên, sinh viên, lớp học và khóa học.
 - Giảm thiểu các sai sót trong quá trình lưu trữ và xử lý dữ liệu.
3. **Cải thiện quy trình đăng ký và quản lý lớp học:**
 - Sinh viên có thể **đăng ký lớp học** một cách dễ dàng.
 - Giảng viên có thể quản lý lớp học của mình và xem danh sách sinh viên đã đăng ký.
4. **Hỗ trợ tính năng chấm điểm và quản lý kết quả học tập:**
 - Hệ thống cho phép giảng viên thực hiện chức năng **chấm điểm** và lưu trữ kết quả học tập của sinh viên.
 - Sinh viên có thể xem được điểm số của mình theo từng khóa học cụ thể.

1.2. Biểu đồ phân rã mức 1

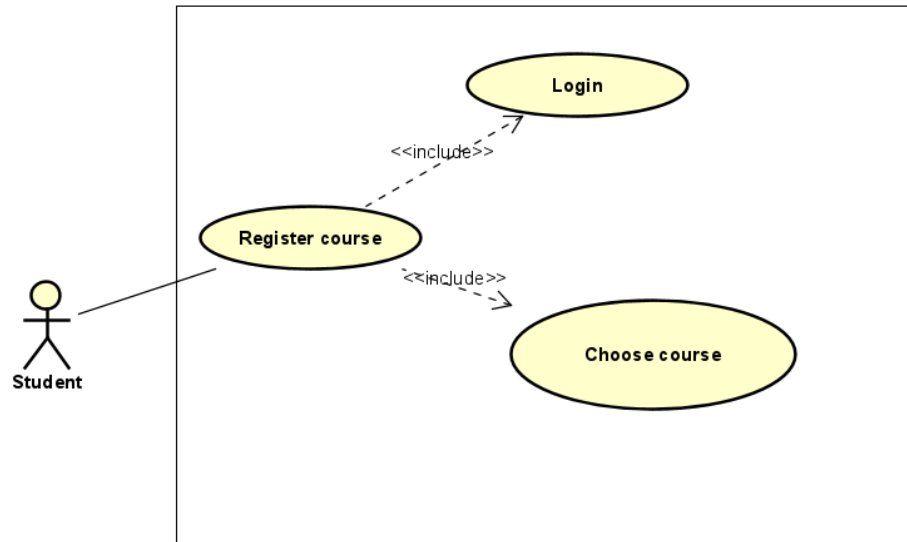


Admin view

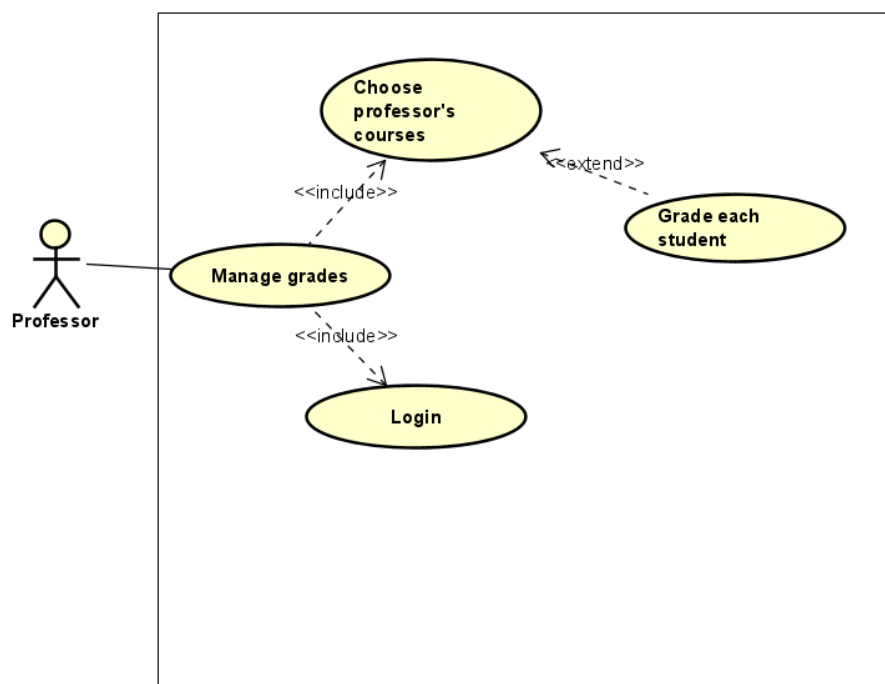


Professor and student's view

1.3. Biểu đồ phân rã mức 2



Phân rã mức 2 usecase Register course



Phân rã mức 2 usecase Manage grades

1.4. Đặc tả Use Case

Use Case “Register course”

1. Mã use case

UC001

2. Mô tả ngắn gọn

Trường hợp sử dụng này mô tả sự tương tác giữa sinh viên và hệ thống khi sinh viên muốn đăng ký môn học.

3. Tác nhân Student

4. Tiền điều kiện: Student đã đăng nhập.

5. Luồng sự kiện cơ sở

Sinh viên chọn chức năng “Register courses”

Hệ thống hiển thị yêu cầu nhập vào CourseID muốn đăng kí.

Sinh viên nhập CourseID tương ứng

Hệ thống ghi nhận và đưa ra thông báo đã đăng kí lớp học thành công

Kết thúc UC

6. Luồng sự kiện thay thế

Bảng N-Các luồng sự kiện thay thế cho thứ tự UC Place

No	Vị trí	Điều kiện	Hành động	Vị trí quay lui
1	Ở bước 3	Nếu sinh viên nhập CourseID đã trùng hoặc CourseID không tồn tại	Hệ thống không cập nhật, quay lại console để nhập lại	Kết thúc UC
2	Ở bước 3	Nếu sinh viên nhập CourseID = -1	Hệ thống hiển thị toàn bộ các lớp có thể, và sinh viên nhập CourseID tương ứng để đăng kí lớp	Bước 2
3	Ở bước 3	Nếu Course đã đầy	Hệ thống thông báo Course đã đầy	Kết thúc UC

Use Case “Grade Student”

1. Mã use case

UC003

2. Mô tả ngắn gọn

Trường hợp sử dụng này mô tả sự tương tác giữa giảng viên và hệ thống khi giảng viên muốn cập nhật điểm số cho sinh viên.

3. Tác nhân**Professor**

4. Tiền điều kiện: Professor đã đăng nhập và có quyền truy cập vào danh sách sinh viên.

5. Luồng sự kiện cơ sở

1. Giảng viên chọn chức năng “Add Course grades”
2. Hệ thống hiển thị khóa học muốn chấm điểm
3. Giảng viên nhập CourseID
4. Hệ thống yêu cầu nhập điểm lớn nhất
5. Giảng viên nhập điểm lớn nhất(10 hoặc 100)
6. Hệ thống hiển thị lớp học và sinh viên để Giảng viên chấm điểm
7. Giảng viên chấm điểm cho sinh viên
8. Hệ thống lưu thông tin mới và hiển thị thông báo thành công
9. Kết thúc UC

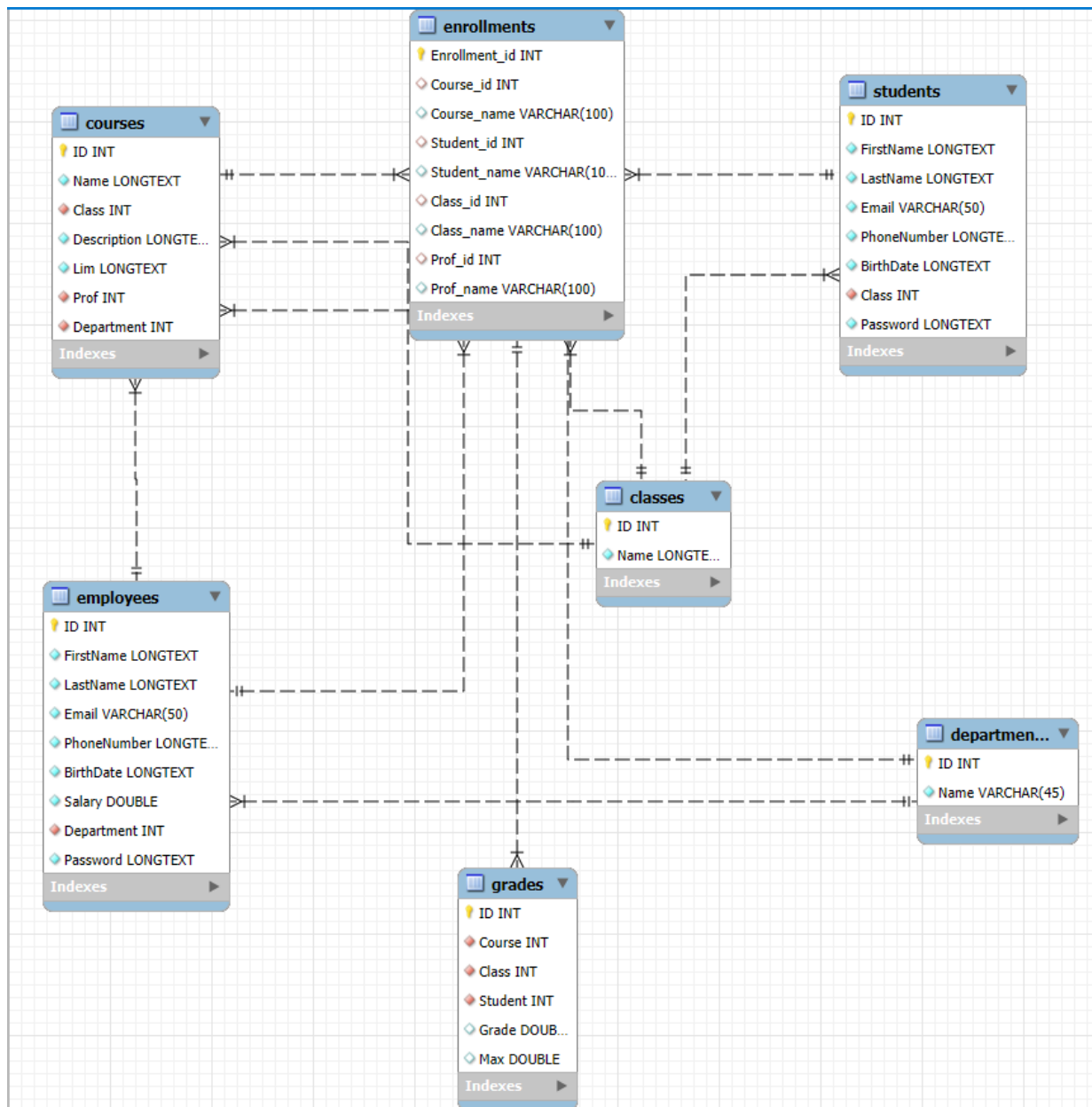
6. Luồng sự kiện thay thế

No	Vị trí	Điều kiện	Hành động	Vị trí quay lui
1.	Ở bước 3	Nếu giảng viên nhập CourseID không tồn tại	Hệ thống không cập nhật, quay lại console để nhập lại	Kết thúc UC
2.	Ở bước 3	Nếu sinh viên nhập CourseID = -1	Hệ thống hiển thị toàn bộ các lớp của giảng viên	Bước 3

CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ BÀI TOÁN

2.1. Thiết kế Cơ sở dữ liệu

2.1.1 Sơ đồ ERD



2.1.2 Phân tích cơ sở dữ liệu

a. Classes

- Đặc điểm chính:

- **ID**: Mã định danh duy nhất của lớp học (khóa chính).
- **Name**: Tên lớp học (dạng longtext), ví dụ: "CNTT Việt Pháp K67", "Kỹ thuật Hóa học K67".
- **Vai trò**: Bảng classes lưu trữ thông tin về các lớp học trong trường. Một lớp học đại diện cho một nhóm sinh viên thuộc một ngành cụ thể, khóa học cụ thể. Đây là thông tin

quan trọng để phân nhóm sinh viên và liên kết dữ liệu với các bảng khác như courses (khóa học), students (sinh viên), và grades (điểm số).

b. Bảng courses

- **Đặc điểm chính:**

- *ID*: Mã định danh duy nhất của khóa học (khóa chính).
- *Name*: Tên khóa học, ví dụ: "Nhập môn lập trình", "Cấu trúc dữ liệu và giải thuật".
- *Class*: Mã lớp học mà khóa học được tổ chức (khóa ngoại tham chiếu

đến classes(ID)).

- *Description*: Mô tả nội dung khóa học.
- *Lim*: Số lượng sinh viên tối đa được phép tham gia khóa học.
- *Prof*: Mã định danh giáo viên phụ trách khóa học (khóa ngoại tham chiếu

đến employees(ID)).

- *Department*: Mã khoa quản lý khóa học (khóa ngoại tham chiếu

đến departments(ID)).

- **Vai trò:** Bảng courses quản lý các khóa học được giảng dạy trong trường, bao gồm thông tin về nội dung khóa học, số lượng sinh viên tối đa, lớp học nào tham gia, giáo viên phụ trách, và khoa quản lý. Đây là bảng trung tâm liên kết với nhiều bảng khác như employees (giáo viên), classes (lớp học), và departments (khoa).

c. Bảng departments

- **Đặc điểm chính:**

- *ID*: Mã định danh duy nhất của khoa (khóa chính).
- *Name*: Tên khoa, ví dụ: "Công nghệ thông tin", "Kỹ thuật Hóa học".

- **Vai trò:** Bảng departments lưu trữ thông tin về các khoa trong trường. Mỗi khoa quản lý một nhóm ngành học hoặc bộ môn cụ thể, chịu trách nhiệm quản lý giáo viên, khóa học, và các hoạt động liên quan. Đây là bảng được tham chiếu bởi các bảng courses và employees.

d. employees

- **Đặc điểm chính:**

- *ID*: Mã định danh duy nhất của giáo viên/nhân viên (khóa chính).
- *FirstName, LastName*: Họ và tên giáo viên/nhân viên.
- *Email*: Email liên hệ (dữ liệu dạng unique, đảm bảo không trùng lặp).
- *PhoneNumber*: Số điện thoại liên hệ.
- *BirthDate*: Ngày sinh.
- *Salary*: Lương của giáo viên/nhân viên.
- *Department*: Khoa mà giáo viên/nhân viên làm việc (khóa ngoại tham chiếu

đến departments(ID)).

- *Password*: Mật khẩu dùng để đăng nhập hệ thống (chưa mã hóa trong dữ liệu mẫu).

- **Vai trò:** Bảng employees quản lý thông tin chi tiết của giáo viên và nhân viên. Dữ liệu từ bảng này liên quan chặt chẽ đến bảng courses (chỉ định giáo viên phụ trách khóa học) và departments (xác định khoa quản lý). Giáo viên có vai trò trung tâm trong việc giảng dạy và tổ chức các khóa học.

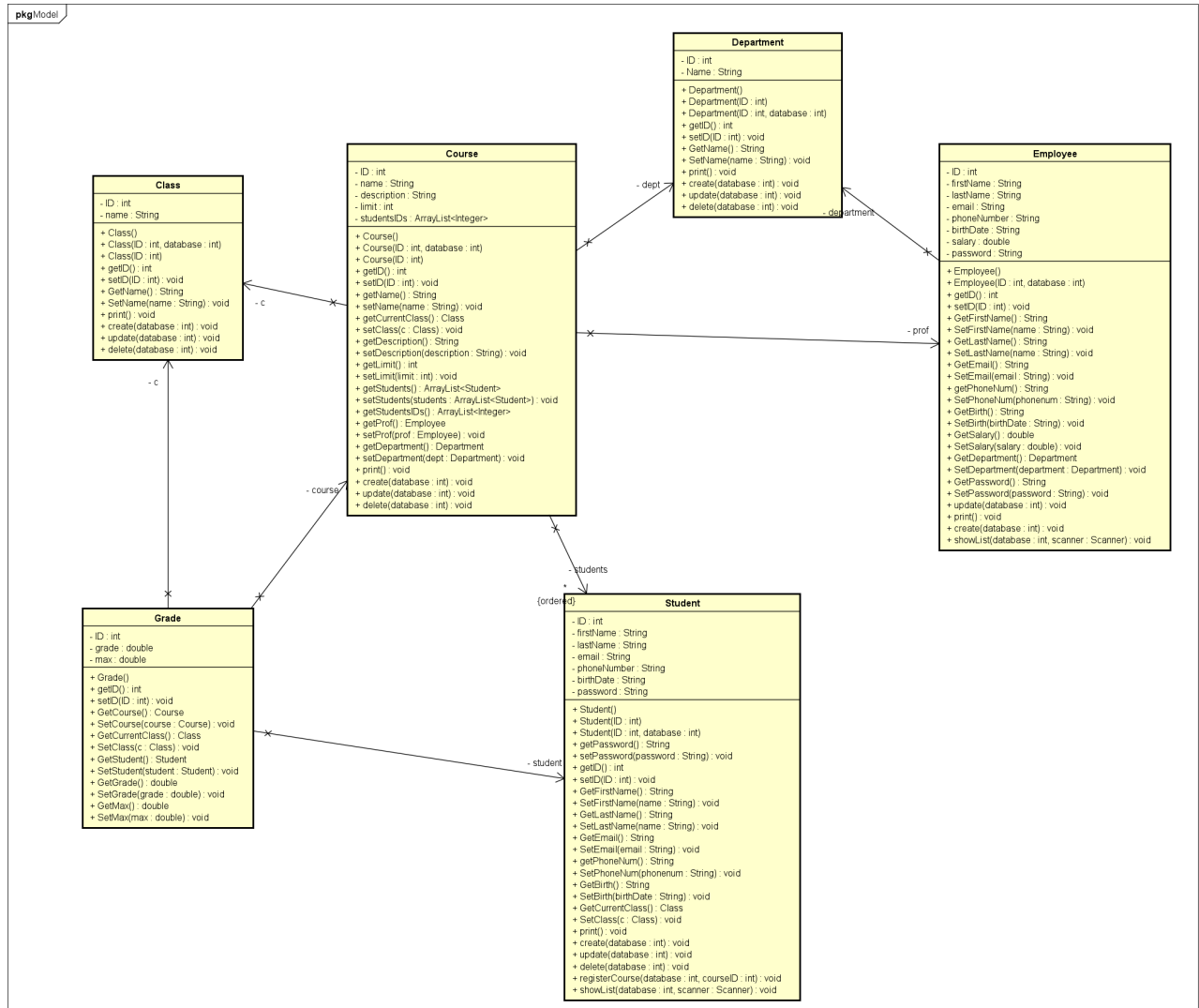
e. students

- **Đặc điểm chính:**

- *ID*: Mã định danh duy nhất của sinh viên (khóa chính).

- *FirstName, LastName*: Họ và tên sinh viên.
 - *Email*: Email liên hệ của sinh viên (dữ liệu unique).
 - *PhoneNumber*: Số điện thoại liên hệ.
 - *BirthDate*: Ngày sinh.
 - *Class*: Lớp mà sinh viên tham gia (khóa ngoại tham chiếu đến *classes(ID)*).
 - *Password*: Mật khẩu dùng để đăng nhập hệ thống.
- **Vai trò:** Bảng *students* lưu trữ thông tin cá nhân của sinh viên, bao gồm thông tin liên lạc, lớp học, và mật khẩu. Đây là bảng quan trọng để quản lý các hoạt động học tập và đăng ký học của sinh viên. Dữ liệu từ bảng này được liên kết với bảng *enrollments* để quản lý việc đăng ký học và bảng *grades* để lưu trữ điểm số.
- f. enrollments**
- **Đặc điểm chính:**
- *Enrollment_id*: Mã định danh duy nhất của việc đăng ký (khóa chính).
 - *Course_id*: Khóa học mà sinh viên đăng ký (khóa ngoại tham chiếu đến *courses(ID)*).
 - *Student_id*: Mã sinh viên tham gia khóa học (khóa ngoại tham chiếu đến *students(ID)*).
 - *Class_id*: Mã lớp học mà sinh viên thuộc về (khóa ngoại tham chiếu đến *classes(ID)*).
 - *Prof_id*: Mã giáo viên giảng dạy khóa học (khóa ngoại tham chiếu đến *employees(ID)*).
- **Vai trò:** Bảng *enrollments* lưu trữ thông tin về việc sinh viên đăng ký học các khóa học. Đây là bảng trung gian liên kết giữa các bảng *students*, *courses*, và *classes*, đồng thời ghi nhận giáo viên phụ trách khóa học. Nó hỗ trợ theo dõi ai đã đăng ký học và thuộc lớp nào.
- g. grades**
- **Đặc điểm chính:**
- *ID*: Mã định danh duy nhất của bản ghi điểm (khóa chính).
 - *Course*: Khóa học mà sinh viên tham gia (khóa ngoại tham chiếu đến *enrollments(Course_id)*).
 - *Class*: Lớp mà sinh viên thuộc về (khóa ngoại tham chiếu đến *enrollments(Class_id)*).
 - *Student*: Mã sinh viên (khóa ngoại tham chiếu đến *enrollments(Student_id)*).
 - *Grade*: Điểm số mà sinh viên đạt được trong khóa học.
 - *Max*: Điểm tối đa của khóa học.
- **Vai trò:** Bảng *grades* lưu trữ điểm số của sinh viên trong các khóa học. Đây là nơi để theo dõi thành tích học tập, liên kết dữ liệu giữa sinh viên, lớp học, và khóa học mà họ tham gia.

2.2. Biểu đồ lớp



CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH MINH HOẠ

3.1. Kết quả chương trình minh họa

Chương trình được đặt tên là CollegeManagementSystem, chạy trên console log của Eclipse, phục vụ việc quản lý sinh viên 1 trường học.

Chương trình được xây dựng đáp ứng đầy đủ tất cả các yêu cầu ban đầu của đề tài về chức năng

3.2. Giao diện chương trình

3.2.1. Giao diện admin tổng

```
Welcome to University Management System
1. Staff
2. Student
1
Enter email:
admin0@gmail.com
Enter password:
0000
|

-----
01. Add new Department
02. Show all Departments
03. Edit Department
04. Delete Department
05. Add New Class
06. Show all classes
07. Edit Class
08. Delete Class
09. Add new Course
10. Show all Courses
11. Edit Course
12. Delete Course
13. Add new Employee
14. Show all Employees
15. Edit Employee
16. Delete Employee
17. Add new Student
18. Show all Students
19. Edit Student
20. Delete Student
21. Change Password
-----
```

3.2.2. Giao diện thêm cán bộ mới

- Giao diện nhập thông tin cán bộ

```
13
Enter First Name:
Prof
Enter Last Name:
1
Enter Email:
prof1@gmail.com
Enter Phone Number:
12345678
Enter Birth Date:
01-01-2000
Enter Salary (double):
1000
Enter Department ID (-1 to show all departments):
-1
ID:      0
Name:    Management

ID:      1
Name:    Công nghệ thông tin

ID:      2
Name:    Kỹ thuật Hóa học

ID:      3
Name:    Ngôn ngữ Anh

ID:      4
Name:    Kinh tế

ID:      5
Name:    Điện tử Viễn thông

Enter Department ID (-1 to show all departments):
4
Enter Password:
1234
Confirm Password:
1234
Employee added successfully
```

- Giao diện hiện danh sách cán bộ

```
14
ID: 0
Name: Admin 0
Email: admin0@gmail.com
Phone Number: 1234567890
Birth Date: 2000-01-01
Salary: 0.0
Department: Management

ID: 1
Name: Nguyễn Văn A
Email: nguyenvana@gmail.com
Phone Number: 0123456789
Birth Date: 1980-05-10
Salary: 60000.0
Department: Công nghệ thông tin

ID: 2
Name: Trần Thị B
Email: tranthib@gmail.com
Phone Number: 0987654321
Birth Date: 1978-08-15
Salary: 65000.0
Department: Kỹ thuật Hóa học

ID: 3
Name: Lê Văn C
Email: levanc@gmail.com
Phone Number: 0169852357
Birth Date: 1975-03-22
Salary: 70000.0
Department: Ngôn ngữ Anh

ID: 4
Name: Phạm Thị D
Email: phamthid@gmail.com
Phone Number: 0963258741
Birth Date: 1982-07-05
Salary: 75000.0
Department: Kinh tế

ID: 5
Name: Hồ Văn E
Email: hovane@gmail.com
Phone Number: 0357896541
Birth Date: 1979-11-18
Salary: 80000.0
Department: Điện tử Viễn thông

ID: 6
Name: Prof 1
Email: prof1@gmail.com
Phone Number: 12345678
Birth Date: 01-01-2000
Salary: 1000.0
Department: Kinh tế
```

3.2.3 Giao diện của giáo viên

```
Welcome to University Management System
1. Staff
2. Student
1
Enter email:
nguyenvana@gmail.com
Enter password:
1234
|
```

```
01. Show all Departments
02. Show all Classes
03. Show all Courses
04. Show my Courses
05. Show Course Students
06. Add Course Grades
07. Show Course Grades
08. Edit Course Grades
09. Edit Course Max
10. Delete Course Grades
11. Delete Spec Course Grade
12. Change Password
```

3.2.3 Giao diện của học sinh

```
Welcome to University Management System
1. Staff
2. Student
2
Enter email:
minhtrinh@gmail.com
Enter password:
1234
```

```
01. Register Courses
02. Show my Data
03. Show Available Courses
04. Show my Grades
05. Change Password
```

CHƯƠNG 4: PHÂN TÍCH ĐẶC TÍNH HƯỚNG ĐỐI TƯỢNG

4.1. Tính đóng gói (Encapsulation)

- **Nhận dạng:** Các thuộc tính trong tất cả các lớp đều được khai báo là `private` (biểu diễn bằng dấu "-").
- **Phân tích:**
 - Mỗi lớp đều đóng gói dữ liệu của nó bằng cách ẩn các thuộc tính bên trong và chỉ cung cấp các phương thức `public` (biểu diễn bằng dấu "+") để truy cập và thay đổi dữ liệu.
 - Ví dụ, trong lớp `Student`, các thuộc tính như `ID`, `firstName`, `lastName` đều là `private`. Để truy cập hoặc thay đổi các thuộc tính này, cần sử dụng các phương thức `getter` và `setter` tương ứng (ví dụ `getID()`, `setID()`).
- **Lợi ích:**
 - **Bảo vệ dữ liệu:** Ngăn chặn truy cập trực tiếp và sửa đổi dữ liệu từ bên ngoài, đảm bảo tính toàn vẹn của dữ liệu.
 - **Kiểm soát truy cập:** Kiểm soát cách thức truy cập và thay đổi dữ liệu thông qua các phương thức `public`, tăng tính bảo mật.
 - **Giảm sự phụ thuộc:** Các lớp khác không phụ thuộc vào cấu trúc dữ liệu bên trong, chỉ cần biết cách sử dụng các phương thức `public`.
 - **Dễ bảo trì:** Khi cần thay đổi cấu trúc dữ liệu bên trong, chỉ cần sửa đổi lớp tương ứng mà không ảnh hưởng đến các lớp khác.

4.2. Tính kế thừa (Inheritance)

- **Nhận dạng:** Mỗi quan hệ kế thừa được biểu diễn bằng mũi tên có đầu hở và nét liền.
 - `Student` và `Employee` kế thừa từ `User`.
- **Phân tích:**
 - `User` là lớp cha, chứa các thuộc tính chung cho tất cả người dùng trong hệ thống (ví dụ: `ID`, `firstName`, `lastName`).
 - `Student` và `Employee` là lớp con, kế thừa các thuộc tính từ `User` và bổ sung thêm các thuộc tính riêng biệt. Ví dụ, `Student` có thêm `currentClass`, `Employee` có thêm `salary`, `department`.
- **Lợi ích:**
 - **Tái sử dụng mã:** Tránh lặp lại code, giúp code ngắn gọn và dễ bảo trì hơn.
 - **Mở rộng dễ dàng:** Dễ dàng thêm các loại người dùng mới bằng cách kế thừa từ `User` hoặc các lớp con hiện có.
 - **Tạo hệ thống phân cấp rõ ràng:** Thể hiện mối quan hệ giữa các đối tượng trong hệ thống.

4.3. Tính trừu tượng (Abstraction)

- **Thể hiện:** Interface `Operation` định nghĩa một phương thức trừu tượng `oper()`. Phương thức này không có phần thân (body) cụ thể, chỉ khai báo tên, tham số và kiểu dữ liệu trả về.
- **Ý nghĩa:** `Operation` đóng vai trò như một khuôn mẫu chung, một bản thiết kế cho các thao tác trên dữ liệu. Nó chỉ ra "Cái gì" cần làm (`oper()`), chứ không chỉ rõ "Làm thế nào" để làm.
- **Lợi ích:**
 - **Trừu tượng hóa:** Ẩn giấu chi tiết triển khai cụ thể của từng thao tác (thêm điểm, tạo lớp, xóa sinh viên,...).
 - **Tạo sự thống nhất:** Cung cấp một giao diện chung cho tất cả các controller, giúp code dễ quản lý và hiểu hơn.

4.4. Tính đa hình (Polymorphism)

- **Thể hiện:** Mỗi controller (ví dụ: `AddCourseGrades`, `CreateClass`, `DeleteStudent`) sẽ **triển khai (implement)** interface `Operation` và cung cấp một cách thức thực hiện `oper()` riêng biệt.
- **Ý nghĩa:** Cùng một phương thức `oper()`, nhưng mỗi controller sẽ có một hành vi cụ thể khác nhau.
- **Lợi ích:**
 - **Linh hoạt:** Cho phép xử lý nhiều loại thao tác khác nhau thông qua cùng một interface.
 - **Mở rộng:** Dễ dàng thêm các thao tác mới bằng cách tạo thêm controller và triển khai `Operation`.

Ví dụ cụ thể:

- `AddCourseGrades` sẽ triển khai `oper()` để thêm điểm cho một khóa học, có thể bao gồm các bước: lấy thông tin điểm từ người dùng, kiểm tra tính hợp lệ, cập nhật cơ sở dữ liệu,...
- `CreateClass` sẽ triển khai `oper()` để tạo lớp học mới, bao gồm các bước: lấy thông tin lớp học, tạo đối tượng `Class`, lưu vào cơ sở dữ liệu,...

TÀI LIỆU THAM KHẢO

1. Bài giảng học phần Lập trình hướng đối tượng của thầy Lê Đức Hậu.
2. Lập trình hướng đối tượng với Java – Đoàn Văn Ban – NXB Khoa học và kỹ thuật.
3. Phân tích và thiết kế hệ thống thông tin - Nguyễn Văn Ba.
4. Các video, tài liệu về lập trình hướng đối tượng và lập trình Java trên Youtube.