

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



BÁO CÁO ĐỒ ÁN MÔN HỌC

MÔN HỌC: Đồ Án Tổng Hợp - Công nghệ Phần mềm

Đề tài: Grade Portal for students at HCMUT

HK241

Giảng viên hướng dẫn: Lê Đình Thuận
Lớp, nhóm hiện thực: L07 - Nhóm thầy Thuận
Sinh viên: Trần Đại Việt - 2213951
Phạm Văn Quốc Việt - 2213950
Nguyễn Nhật Khoa - 2211629
Phạm Việt Anh - 2210128
Nguyễn Gia Nguyên - 2212303
Lê Đăng Khoa - 2211599

HO CHI MINH CITY, SEPTEMBER 2024

Mục lục

1	Danh Sách Thành Viên Và Phân Công Nhiệm Vụ	1
2	Tổ chức và quản lý Code trong mã nguồn	2
2.1	Mô tả chi tiết các thành phần chính	3
2.2	Code style	3

1 Danh Sách Thành Viên Và Phân Công Nhiệm Vụ

MSSV	Họ và tên	Nhiệm vụ	Phần trăm công việc
2213951	Trần Đại Việt	PO	100%
2213950	Phạm Văn Quốc Việt	DEV	100%
2211629	Nguyễn Nhật Khoa	DEV	100%
2210128	Phạm Việt Anh	DEV	100%
2212303	Nguyễn Gia Nguyên	DEV	100%
2211599	Lê Đăng Khoa	DEV	100%

Bảng 1: *Danh sách sinh viên và nhiệm vụ*

2 Tổ chức và quản lý Code trong mã nguồn

Cấu trúc mã nguồn của dự án được tổ chức theo dạng cây thư mục với các thư mục chính và chức năng cụ thể như sau:

```
.
|--- .mvn                                % Cấu hình Maven Wrapper
|   |___ wrapper
|--- .vscode                             % Cấu hình cho VSCode
|--- docs                                % Thư mục chứa tài liệu dự án
|   |--- api-document                    % Tài liệu API
|   |--- database-design                 % Thiết kế CSDL
|   |--- deploy-guide                    % Hướng dẫn triển khai
|   |--- git-flow                        % Quy trình làm việc với Git
|   |___ system-architecture             % Kiến trúc hệ thống
|--- reports                             % Báo cáo dự án
|   |--- mainReport                      % Báo cáo chính
|   |___ weeklyReport                    % Báo cáo hàng tuần
|       |--- meeting_minute_report      % Biên bản họp
|       |___ weekly_result               % Kết quả hàng tuần
|           |___ img_10_27_24            % Hình ảnh minh họa
|--- src                                 % Mã nguồn chính
|   |--- main
|       |___ java
|           |___ com
|               |___ hcmut
|                   |___ gradeportal
|                       |--- config        % Cấu hình hệ thống
|                       |--- controller    % Các controller xử lý API
|                           |--- admin
|                           |--- client_auth
|                           |--- hall_of_fame
|                           |--- health_check
|                           |--- init_data
|                           |--- student
|                           |___ teacher
|                       |--- database        % Thao tác cơ sở dữ liệu
|                       |--- dtos           % Data Transfer Objects
|                           |--- admin
|                           |   |--- request
|                           |   |___ response
|                           |--- auth
|                           |--- course
|                           |--- sheetMark
|                           |___ teacher
|                       |--- entities        % Định nghĩa các entity
|                           |--- enums
|                           |___ idsOfEntities
|                       |--- repositories    % Repository làm việc với CSDL
|                       |--- security        % Bảo mật JWT
|                       |--- service         % Logic nghiệp vụ
|                       |--- specification  % Query Specification
|                       |___ validation      % Validation dữ liệu
|       |___ resources
|           |--- baseData                  % Dữ liệu nền
|           |--- static                    % File tĩnh
|           |___ templates                  % Template giao diện
|       |___ test                          % Mã nguồn test
|--- target                                % Thư mục build của Maven
|___ pom.xml                              % File cấu hình Maven
```

2.1 Mô tả chi tiết các thành phần chính

- **docs:** Chứa các tài liệu liên quan đến dự án, bao gồm:
 - **api-document:** Tài liệu mô tả chi tiết các API trong hệ thống.
 - **database-design:** Thiết kế cơ sở dữ liệu, bao gồm lược đồ và quan hệ giữa các bảng.
 - **deploy-guide:** Hướng dẫn triển khai hệ thống.
 - **git-flow:** Quy trình làm việc với Git.
 - **system-architecture:** Mô tả kiến trúc hệ thống.
- **src/main/java:** Thư mục chính chứa mã nguồn của hệ thống:
 - **config:** Chứa các lớp cấu hình hệ thống, như cấu hình bảo mật, cấu hình CSDL hoặc các bean cần thiết.
 - **controller:** Chứa các lớp xử lý yêu cầu API từ người dùng:
 - * Chia thành nhiều module như `admin`, `client_auth`, `student`, và `teacher` để dễ quản lý.
 - * Mỗi lớp controller sẽ đảm nhận việc điều hướng và trả kết quả phù hợp về tầng Presentation.
 - **dtos (Data Transfer Objects):** Chứa các lớp trung gian dùng để chuyển dữ liệu giữa các tầng.
 - * Được chia thành các gói nhỏ như: **admin**, **auth**, **course**, **sheetMark**, **teacher**, v.v.
 - * Mỗi gói chứa các lớp `request` và `response` để định nghĩa dữ liệu đầu vào và đầu ra.
 - **entities:** Định nghĩa các đối tượng ánh xạ với bảng trong cơ sở dữ liệu.
 - * **enums:** Chứa các kiểu dữ liệu liệt kê được sử dụng trong hệ thống.
 - * **idsOfEntities:** Định nghĩa các khóa chính tổng hợp hoặc khóa đặc biệt của các entity.
 - **repositories:** Cung cấp các phương thức làm việc với cơ sở dữ liệu (CRUD). Đây là các interface giúp giảm thiểu việc viết câu lệnh SQL thủ công.
 - **security:** Chứa các lớp hiện thực cơ chế bảo mật cho hệ thống.
 - * **jwt:** Quản lý và xử lý token JWT cho việc xác thực và phân quyền.
 - * **utils:** Các tiện ích hỗ trợ việc mã hóa và kiểm tra bảo mật.
 - **service:** Chứa các lớp xử lý logic nghiệp vụ của hệ thống. Mỗi module (như `admin`, `student`, `course`) sẽ có một lớp service tương ứng.
 - **specification:** Cung cấp các điều kiện để truy vấn linh hoạt khi làm việc với cơ sở dữ liệu.
 - **validation:** Chứa các lớp xác thực dữ liệu đầu vào để đảm bảo tính đúng đắn và toàn vẹn.
- **src/test/java:** Chứa các lớp kiểm thử cho các thành phần của hệ thống. Các lớp test được tổ chức tương tự như cấu trúc mã nguồn chính.
- **resources:** Chứa các tệp cấu hình và dữ liệu cho hệ thống:
 - **baseData:** Dữ liệu nền để khởi tạo hệ thống.
 - **templates:** Các file template phục vụ cho giao diện hoặc email.
 - **static:** Chứa các tài nguyên tĩnh như hình ảnh, CSS hoặc JavaScript.
- **target:** Thư mục được tạo ra sau khi hệ thống được build bằng Maven. Bao gồm các file `.class`, báo cáo test và các tài nguyên được biên dịch.

2.2 Code style

Đoạn mã nguồn tuân thủ các quy tắc code style hiện đại nhằm đảm bảo tính nhất quán, dễ đọc và dễ bảo trì. Các quy tắc chính được áp dụng như sau:

- **Tổ chức import:** Các import được sắp xếp theo nhóm (Java chuẩn, thư viện bên thứ ba, module nội bộ) và loại bỏ các import dư thừa.
- **Đặt tên:** Sử dụng *camelCase* cho tên biến và phương thức (`getAllTeachers`, `updateTeacher`). Tên biến rõ ràng, mô tả đúng chức năng. *Class* sử dụng *PascalCase* (ví dụ: `UserManager`). *Hằng số* sử dụng chữ in hoa và gạch dưới (ví dụ: `MAX_CONNECTIONS`).
- **Tổ chức logic:** Các thao tác CRUD (`createTeacher`, `updateTeacher`, `deleteTeacher`) và các chức năng khác được viết tách biệt theo trách nhiệm.

- **Xử lý null:** Sử dụng `Optional` để tránh lỗi `NullPointerException`, đảm bảo mã nguồn an toàn.
- **Xử lý lỗi:** Sử dụng ngoại lệ như `IllegalArgumentException` với thông báo rõ ràng ("*Teacher not found*").
- **Format mã:** Thụt lề 4 spaces, có khoảng cách giữa các toán tử và từ khóa (`if (condition) { ... }`), sử dụng dòng trắng để phân tách các khối logic.
- **Comment:** Comment block mô tả chức năng chính của phương thức; comment ngắn gọn giải thích từng bước xử lý.
- **Tuân thủ SOLID:** Phân chia rõ trách nhiệm giữa repository và service; mỗi phương thức đảm nhận một chức năng cụ thể.

Những quy tắc trên giúp mã nguồn đảm bảo tính dễ đọc, dễ bảo trì và sẵn sàng mở rộng trong tương lai.