

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

🞠◊🞠◊🞠



**BÀI BÁO CÁO**

**Đề tài: xây dựng phần mềm thi trắc nhiệm**

**(Môn học: Lập trình web)**

**SINH VIÊN THỰC HIỆN:**

Lê Cao Liêm 16110374

Nguyễn Quang Phú 16110416

Phạm Hoàng Phúc 16110419

Trần Văn Nam 16110588

Ngô Tuấn Lĩnh 16110373

**TP. HỒ CHÍ MINH – 12/2018**

**MỤC LỤC**

[**1. Bảng mô tả phân công công việc 4**](#_Toc533165825)

[**2. Thiết kế giao diện 4**](#_Toc533165826)

[**3. Đặc tả phần mềm (bài toán) 10**](#_Toc533165827)

[**4.​ Các mẫu dùng cho thiết kế code 15**](#_Toc533165828)

[**4.1. Mẫu bảng mô tả các phương thức trong lớp CauHoi 18**](#_Toc533165829)

[**4.2. Mẫu bảng mô tả các phương thức trong lớp MonHoc 19**](#_Toc533165830)

[**4.3. Mẫu bảng mô tả các phương thức trong lớp DeThi 20**](#_Toc533165831)

[**4.4. Mẫu bảng mô tả các phương thức trong lớp Lop 20**](#_Toc533165832)

[**4.5. Mẫu bảng mô tả các phương thức trong lớp ThanhVien 21**](#_Toc533165833)

[**4.6. Mẫu bảng mô tả các phương thức trong lớp ChiTietDeThi 22**](#_Toc533165834)

[**4.7. Mẫu bảng mô tả các phương thức trong lớp DiemThi 23**](#_Toc533165835)

[**4.8. Mẫu bảng mô tả các phương thức trong lớp DBUtils (quanlycauhoi) 24**](#_Toc533165836)

[**4.9. Mẫu bảng mô tả các phương thức trong lớp MySQLConnUtils 28**](#_Toc533165837)

[**4.10. Mẫu bảng mô tả các phương thức trong lớp QuestionListServlet 29**](#_Toc533165838)

[**4.11. Mẫu bảng mô tả các phương thức trong lớp ThemCauHoi 29**](#_Toc533165839)

[**4.12. Mẫu bảng mô tả các phương thức trong lớp SuaCauHoi 31**](#_Toc533165840)

[**4.13. Mẫu bảng mô tả các phương thức trong lớp XoaCauHoi 33**](#_Toc533165841)

[**4.14. Mẫu bảng mô tả các phương thức trong lớp ChiTietCauHoi 33**](#_Toc533165842)

[**4.15. Mẫu bảng mô tả các phương thức trong lớp ThongTinTaiKhoan 34**](#_Toc533165843)

[**4.16. Mẫu bảng mô tả các phương thức trong lớp SoanDeThi 34**](#_Toc533165844)

[**4.17. Mẫu bảng mô tả các phương thức trong lớp ChinhSuaDeThi 37**](#_Toc533165845)

[**4.18. Mẫu bảng mô tả các phương thức trong lớp ChiTietDeThi 39**](#_Toc533165846)

[**4.19. Mẫu bảng mô tả các phương thức trong lớp DangXuatQLDT 39**](#_Toc533165847)

[**4.20. Mẫu bảng mô tả các phương thức trong lớp XoaDeThi 39**](#_Toc533165848)

[**4.21. Mẫu bảng mô tả các phương thức trong lớp NguoiQuanLiDeThi 40**](#_Toc533165849)

[**4.22. Mẫu bảng mô tả các phương thức trong lớp DBUtils (quanlidethi.utils) 40**](#_Toc533165850)

[**4.23. Mẫu bảng mô tả các phương thức trong lớp DBUtils (public.utils) 44**](#_Toc533165851)

[**4.24. Mẫu bảng mô tả các phương thức trong lớp DanhSachThanhVien 44**](#_Toc533165852)

[**4.25. Mẫu bảng mô tả các phương thức trong lớp DanhSachLop 45**](#_Toc533165853)

[**4.26. Mẫu bảng mô tả các phương thức trong lớp DanhSachThanhVienLop 45**](#_Toc533165854)

[**4.27. Mẫu bảng mô tả các phương thức trong lớp SuaLop 46**](#_Toc533165855)

[**4.28. Mẫu bảng mô tả các phương thức trong lớp SuaThanhVien 47**](#_Toc533165856)

[**4.29. Mẫu bảng mô tả các phương thức trong lớp ThemLop 48**](#_Toc533165857)

[**4.30. Mẫu bảng mô tả các phương thức trong lớp ThongTinTaiKhoan (quanlythisinh) 49**](#_Toc533165858)

[**4.31. Mẫu bảng mô tả các phương thức trong lớp ThongTinThanhVien 49**](#_Toc533165859)

[**4.32. Mẫu bảng mô tả các phương thức trong lớp XoaLop 50**](#_Toc533165860)

[**4.33. Mẫu bảng mô tả các phương thức trong lớp XoaThanhVien 50**](#_Toc533165861)

[**4.34. Mẫu bảng mô tả các phương thức trong lớp XoaThanhVienLop 51**](#_Toc533165862)

[**4.35. Mẫu bảng mô tả các phương thức trong lớp DBUtils (quanlithisinh.utils) 51**](#_Toc533165863)

[**4.36. Mẫu bảng mô tả các phương thức trong lớp ChiTietDeThiThiSinh 56**](#_Toc533165864)

[**4.37. Mẫu bảng mô tả các phương thức trong lớp DangXuatTS 58**](#_Toc533165865)

[**4.38. Mẫu bảng mô tả các phương thức trong lớp DanhSachDeThiThiSinh 59**](#_Toc533165866)

[**4.39. Mẫu bảng mô tả các phương thức trong lớp DanhSachMonHoc 60**](#_Toc533165867)

[**4.40. Mẫu bảng mô tả các phương thức trong lớp ThongTinThiSinh 60**](#_Toc533165868)

[**4.41. Mẫu bảng mô tả các phương thức trong lớp DBUtils (thisinh.utils) 61**](#_Toc533165869)

# ***1. Bảng mô tả phân công công việc***

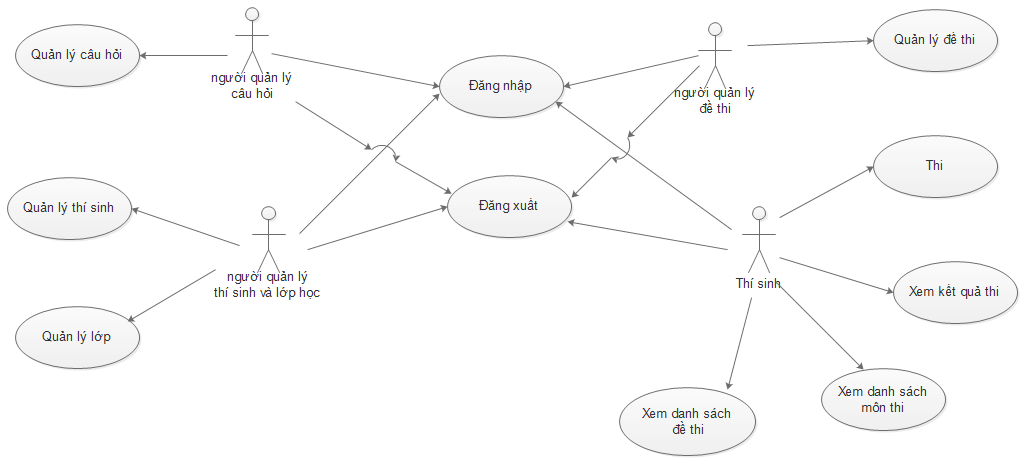
|  |  |  |
| --- | --- | --- |
| Tên SV | Đánh giá chung phần trăm đóng góp | Mô tả khái quát mảng công việc SV thực hiện trong đồ án. |
| Phạm Hoàng Phúc | 20% | Viết giao diện website của chức năng người quản lý đề thi. Viết các trang chức năng của người quản lý đề thi. Viết phần đăng nhập và đăng xuất cho mỗi chức năng |
| Nguyễn Quang Phú | 20% | Thiết kế giao diện cho người quản lý câu hỏi và cơ sở dữ liệu cho phần mềm; Viết trang chức năng cho người quản lý câu hỏi |
| Lê Cao Liêm | 25% | Thiết kế giao diện cho trang chủ, trang đăng nhập và website cho thí sinh vào thi; Viết trang chức năng cho thí sinh |
| Trần Văn Nam | 17,5% | Thiết kế giao diên website cho chức năng quản lý thí sinh và lớp học   * Nam: viết trang chức năng cho người quản lý thí sinh * Lĩnh: viết chức năng cho người quản lý lớp gồm nhiều thí sinh |
| Ngô Tuấn Lĩnh | 17,5% |

SV hãy tự đánh giá mức độ hoàn thành được mục tiêu. Các khó khăn gặp phải, khắc phục thế nào, ý tưởng phát triển đồ án. Những khuyết điểm của đồ án. Những ưu điểm.

# ***2. Thiết kế giao diện***

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Màn hình/Cửa sổ/Dialog  (Tên màn hình trên 1 dòng; dưới dòng này là hình chụp màn hình) | Người thiết kế & giải thích ngắn gọn các quyết định thiết kế khi thiết kế màn hình (Người thiết kế ở trên 1 dòng. Sau dòng đó là phần giải thích) | Mục đích chính của màn hình |
| 1 | Trang chủ website | Lê Cao Liêm Trang sẽ được hiển thị đầu tiên khi truy cập vào trang chủ website | Trang hiển thị hình ảnh minh họa và nút ấn để chuyển đến trang đăng nhập của phần mềm |
| 2 | Trang đăng nhập | Lê Cao Liêm | Trang hiển thị các yêu cầu thông tin cần có để đăng nhập vào phần mềm. có các nút chọn để chọn chức vụ đúng |
| 3 | Trang danh sách môn thi của thí sinh | Lê Cao Liêm | Trang hiển thị danh sách các môn thi của thí sinh dưới dạng nút ấn. ấn vào môn nào sẽ chuyển đến danh sách đề thi theo môn của thí sinh đó |
| 4 | Trang danh sách đề thi của thí sinh | Lê Cao Liêm | Trang hiển thị danh sách các đề thi theo môn học mà thí sinh đã chọn trước đó. Nhấn nút vào thi thì bài thi sẽ hiện ra và thí sinh bắt đầu làm |
| 5 | Trang thi của thí sinh | Lê Cao Liêm | Trang hiển thị nội dung câu hỏi của đề thi mà thí sinh đã nhấn nút vào thi trước đó |
| 6 | Trang kết quả thi của thí sinh | Lê Cao Liêm | Trang hiển thị kết quả làm bài thi của thí sinh và điềm trung bình của thí sinh với đề thi đó |
| 7 | Trang thông tin tài khoản của thí sinh | Lê Cao Liêm | Hiển thị nhật thông tin tài khoản của thí sinh |
| 8 | Trang chủ quản lý câu hỏi | Nguyễn Quang Phu | Hiển thị thanh các chức năng của quản lý câu hỏi |
| 9 | Trang danh sách câu hỏi | Nguyễn Quang Phú | Hiển thị danh sách các câu hỏi |
| 10 | Trang thêm câu hỏi | Nguyễn Quang Phú | Thực hiện chức năng thêm một câu hỏi mới |
| 11 | Trang sửa câu hỏi | Nguyễn Quang Phú | Thực hiện chức năng sửa một câu hỏi trong danh sách câu hỏi |
| 12 | Trang thông tin chi tiết câu hỏi | Nguyễn Quang Phú | Xem nội dung của câu hỏi đó |
| 13 | Trang thông tin người quản lý câu hỏi | Nguyễn Quang Phú | Xem thông tin của người quản lý câu hỏi |
| 14 | Trang chủ quản lý thí sinh và lớp học | Trần Văn Nam, Ngô Tuấn Lĩnh | Hiển thị thanh các chức năng của quản lý thí sinh và lớp học |
| 15 | Trang danh sách thí sinh | Trần Văn Nam, Ngô Tuấn Lĩnh | Hiển thị danh sách các thí sinh |
| 16 | Trang thêm thí sinh | Trần Văn Nam, Ngô Tuấn Lĩnh | Thực hiện chức năng thêm một thí sinh mới |
| 17 | Trang sửa thí sinh | Trần Văn Nam, Ngô Tuấn Lĩnh | Thực hiện chức năng sửa một thí sinh trong danh sách thí sinh |
| 18 | Trang xem thông tin chi tiết thí sinh | Trần Văn Nam, Ngô Tuấn Lĩnh | Xem thông tin chi tiết của một thí sinh trong danh sách thí sinh |
| 19 | Trang danh sách lớp | Trần Văn Nam, Ngô Tuấn Lĩnh | Hiển thị danh sách các lớp gồm nhiều thí sinh |
| 20 | Trang thêm lớp | Trần Văn Nam, Ngô Tuấn Lĩnh | Thực hiện chức năng tạo một lớp và thêm các thí sinh vào lớp đó |
| 21 | Trang sửa lớp | Trần Văn Nam, Ngô Tuấn Lĩnh | Thực hiện chức năng sửa một lớp trong danh sách lớp và có quyền bổ sung thêm thành viên từ danh sách thành viên |
| 22 | Trang danh sách thí sinh của một lớp | Trần Văn Nam, Ngô Tuấn Lĩnh | Hiển thị danh sách thí sinh của một lớp trong danh sách lớp |
| 23 | Trang thông tin người quản lý thí sinh | Trần Văn Nam, Ngô Tuấn Lĩnh | Hiển thị thông tin của người quản lý thí sinh và lớp học |
| 24 | Trang chủ quản lý đề thi | Phạm Hoàng Phúc | Hiển thị thanh các chức năng của quản lý đề thi |
| 25 | Trang danh sách đề thi | Phạm Hoàng Phúc | Hiển thị danh sách các đề thi |
| 26 | Trang thêm đề thi | Phạm Hoàng Phúc | Thực hiện chức năng tạo ra một đề thi mới và chọn các lớp trong danh sách lớp để thi đề thi này |
| 27 | Trang sửa đề thi | Phạm Hoàng Phúc | Thực hiện chức năng sửa thông tin chi tiết của đề thi |
| 28 | Trang chi tiết đề thi | Phạm Hoàng Phúc | Hiển thị thông tin chi tiết của đề thi, danh sách lớp thi đề thi này và nội dung câu hỏi của đề thi |
| 29 | Trang thông tin người quản lý đề thi | Phạm Hoàng Phúc | Hiển thị thông tin người quản lý đề thi |

# ***3. Đặc tả phần mềm (bài toán)***

Hình vẽ use case diagram 

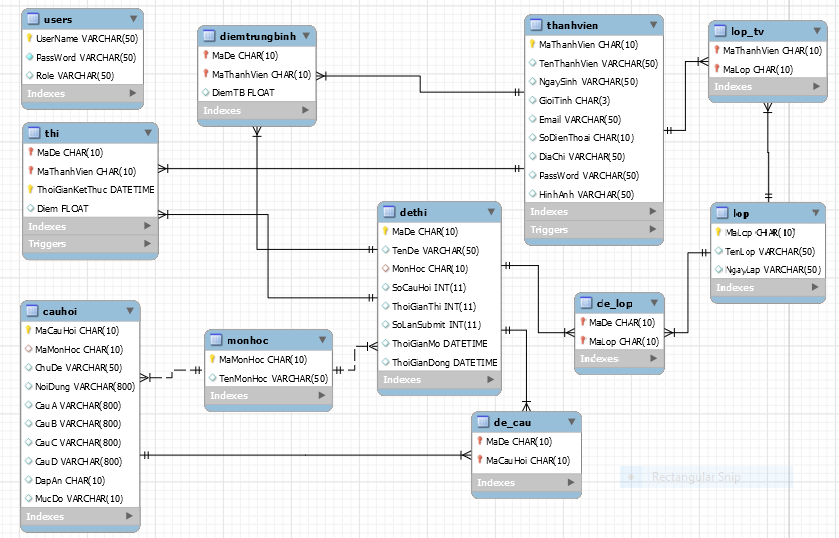
Bảng mô tả các use case

|  |  |  |
| --- | --- | --- |
| TT | Tên use case | Ý nghĩa |
| 1 | Wesite Thi trắc nhiệm | Mô tả các chức năng chính của website. |

Bảng mô tả các actor

|  |  |  |
| --- | --- | --- |
| TT | Tên Actor | Ý nghĩa |
| 1 | Người quản lý câu hỏi | Có thể xem danh sách các câu hỏi đã tạo, có thể tạo, sửa và xóa câu hỏi |
| 2 | Người quản lý thí sinh và lớp học | Có thể xem danh sách các thí sinh, thêm sửa xóa các thí sinh. Có thể tạo, sửa và xóa một lớp học gồm nhiều thí sinh hay các thí sinh trong lớp học đó |
| 3 | Người quản lý đề thi | Có thể xem danh sách các đề thi, có thể tạo đề thi mới, thêm các câu hỏi cho nội dung đề thi và gán đề thi cho lớp học. Có thể sửa lại thông tin đề thi và xóa dề thi |
| 4 | Thí sinh | Có thể xem danh sách các đề thi còn hạn theo môn học của mình, có thể vào thi và làm bài. Sau khi nộp bài có thể xem kết quả hiển thị ngay sau đó |

**Sơ đồ cơ sở dữ liệu**



Mẫu bảng mô tả các bảng trong CSDL

|  |  |  |
| --- | --- | --- |
| TT | Tên bảng | Mục đích của bảng trong chương trình |
| 1 | Câu hỏi | Chứa các câu hỏi |
| 2 | Đề câu | Chứa câu hỏi thuộc một đề |
| 3 | Đề lớp | Chứa các lớp thi một đề |
| 4 | Đề thi | Chứa các đề thi |
| 5 | Điểm trung bình | Chứa các điểm trung bình |
| 6 | Lớp | Chứa các lớp học |
| 7 | Lớp thành viên | Chứa các thành viên thuộc một lớp |
| 8 | Môn học | Chứa các môn học |
| 9 | Thành viên | Chứa các thành viên |
| 10 | Thi | Chứa các điểm của thành viên sau một lần thi |
| 11 | Users | Chứa thông tin đăng nhập tài khoản |

Bảng mô tả các trường trong bảng CauHoi

|  |  |  |
| --- | --- | --- |
| TT | Tên trường | Mục đích của trường trong chương trình |
| 1 | MaCauHoi | Khóa chính của bảng CauHoi |
| 2 | MaMonHoc | Khóa ngoại từ bảng MonHoc |
| 3 | NoiDung | Nội dung câu hỏi |
| 4 | CauA | Nội dung đáp án A |
| 5 | CauB | Nội dung đáp án B |
| 6 | CauC | Nội dung đáp án C |
| 7 | CauD | Nội dung đáp án D |
| 8 | DapAn | Đáp án đúng |
| 9 | MucDo | Mức độ câu hỏi |
| 10 | ChuDe | Chủ đề câu hỏi |

Bảng mô tả các trường trong bảng DeThi

|  |  |  |
| --- | --- | --- |
| TT | Tên trường | Mục đích của trường trong chương trình |
| 1 | MaDe | Khóa chính của bảng DeThi |
| 2 | TenDe | Tên đề thi |
| 3 | MonHoc | Khóa ngoại của bảng MonHoc |
| 4 | SoCauHoi | Số câu hỏi trong đề thi |
| 5 | ThoiGianThi | Thời gian làm bài |
| 6 | SoLanSubmit | Số lần làm bài thi |
| 7 | ThoiGianMo | Thời gian mở để thi |
| 8 | ThoiGianDong | Thời gian hết hạn thi |

Bảng mô tả các trường trong bảng DiemTrungBinh

|  |  |  |
| --- | --- | --- |
| TT | Tên trường | Mục đích của trường trong chương trình |
| 1 | MaDe | Khóa chính của bảng DiemTrungBinh tham chiếu đến MaDe của bảng DeThi |
| 2 | MaThanhVien | Khóa chính của bảng DiemTrungBinh tham chiếu đến MaThanhVien của bảng ThanhVien |
| 3 | DiemTB | Điểm trung bình của thí sinh |

Bảng mô tả các trường trong bảng Lop

|  |  |  |
| --- | --- | --- |
| TT | Tên trường | Mục đích của trường trong chương trình |
| 1 | MaLop | Khóa chính của bảng Lop |
| 2 | TenLop | Tên lớp học |
| 3 | NgayLap | Ngày tạo ra lớp học |

Bảng mô tả các trường trong bảng ThanhVien

|  |  |  |
| --- | --- | --- |
| TT | Tên trường | Mục đích của trường trong chương trình |
| 1 | MaThanhVien | Khóa chính của bảng ThanhVien |
| 2 | TenThanhVien | Tên thí sinh |
| 3 | NgaySinh | Ngày sinh |
| 4 | GioiTinh | Giới tính |
| 5 | Email | Email |
| 6 | SoDienThoai | Số điện thoại |
| 7 | DiaChi | Địa chỉ |
| 8 | PassWord | Mật khẩu thí sinh để đăng nhập vào phần mềm |
| 9 | HinhAnh | Hình ảnh thí sinh |

Bảng mô tả các trường trong bảng MonHoc

|  |  |  |
| --- | --- | --- |
| TT | Tên trường | Mục đích của trường trong chương trình |
| 1 | MaMonHoc | Khóa chính của bảng MonHoc |
| 2 | TenMonHoc | Tên môn học |

Bảng mô tả các trường trong bảng Thi

|  |  |  |
| --- | --- | --- |
| TT | Tên trường | Mục đích của trường trong chương trình |
| 1 | MaDe | Khóa chính của bảng Thi tham chiếu đến MaDe của bảng DeThi |
| 2 | MaThanhVien | Khóa chính của bảng Thi tham chiếu đến MaThanhVien của bảng ThanhVien |
| 3 | ThoiGianKetThuc | Thời gian thí sinh nộp bài |
| 4 | Diem | Điểm của thí sinh trong lần thi đó |

Bảng mô tả các trường trong bảng Users

|  |  |  |
| --- | --- | --- |
| TT | Tên trường | Mục đích của trường trong chương trình |
| 1 | UserName | Tên đăng nhập |
| 2 | PassWord | Mật khẩu |
| 3 | Role | Quyền đăng nhập |

# ***4.​ Các mẫu dùng cho thiết kế code***

Mẫu bảng danh mục các lớp được sử dụng trong chương trình

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên lớp | Tên các SV phụ trách viết | Mục đích chính của lớp trong chương trình |
| 1 | CauHoi | Nguyễn Quang Phú | Chứa các câu hỏi sau khi truy vấn từ CSDL hoặc khi xử lý thêm, sửa, xóa câu hỏi |
| 2 | DeThi | Phạm Hoàng Phúc | Chứa các đề thi sau khi truy vấn từ CSDL hoặc khi xử lý thêm, sửa, xóa đề thi |
| 3 | Lop | Ngô Tuấn Lĩnh | Chứa các lớp sau khi truy vấn từ CSDL hoặc khi xử lý thêm, sửa, xóa lớp |
| 4 | MonHoc | Nguyễn Quang Phú | Chứa môn học sau khi truy vấn từ CSDL |
| 5 | ThanhVien | Trần Văn Nam | Chứa các thành viên sau khi truy vấn từ CSDL hoặc khi xử lý thêm, sửa, xóa thành viên |
| 6 | ChiTietDeThi | Lê Cao Liêm | Chứa các câu hỏi của một đề thi sau khi truy vấn từ CSDL |
| 7 | Diem | Lê Cao Liêm | Chứa điểm của thí sinh theo đề thi và thời gian thí sinh nộp bài khi truy vấn vào cơ sở dữ liệu |
| 6 | ConnectionUtils | Nguyễn Quang Phú | Gọi lớp MySQLConnUtils |
| 7 | MySQLConnUtils | Nguyễn Quang Phú | Kết nối đến cơ sở dữ liệu |
| 8 | ChiTietCauHoi | Nguyễn Quang Phú | Xử lý xem chi tiết câu hỏi |
| 9 | QuestionListServlet | Nguyễn Quang Phú | Xử lý xem danh sách câu hỏi |
| 10 | SuaCauHoi | Nguyễn Quang Phú | Xử lý sửa câu hỏi |
| 11 | ThemCauHoi | Nguyễn Quang Phú | Xử lý thêm câu hỏi |
| 12 | ThongTinTaiKhoan | Nguyễn Quang Phú | Xử lý xem thông tin tài khoản người quản lý câu hỏi |
| 13 | XoaCauHoi | Nguyễn Quang Phú | Xử lý xóa câu hỏi |
| 14 | DBUtils(quanlycauhoi) | Nguyễn Quang Phú | Thêm, xóa, sửa, lấy danh sách và lấy thông tin câu hỏi |
| 15 | Login | Phạm Hoàng Phúc | Xử lí đăng nhập |
| 16 | DanhSachDeThi | Phạm Hoàng Phúc | Truy vấn CSDL lấy lên danh sách đề thi |
| 17 | SoạnDeThi | Phạm Hoàng Phúc | Tạo một đề thi và lưu xuống CSDL |
| 18 | ChinhSuaDeThi | PhạmHoàngPhúc | Sửa thông tin của đề thi và lưu xuống CSDL |
| 19 | ChiTietDeThi | Phạm Hoàng Phúc | Truy vấn CSDL lấy lên tất cả thông tin của một đề thi |
| 20 | DangXuatQLDT | Phạm Hoàng Phúc | Xử lí đăng xuất |
| 21 | XoaDeThi | Phạm Hoàng Phúc | Xóa đề thi khỏi CSDL |
| 22 | NguoiQuanLiDeThi | Phạm Hoàng Phúc | Hiển thị thông tin người QLDT |
| 23 | DBUtils (quanlidethi.utils) | Phạm Hoàng Phúc | Thực hiện các truy vấn đến CSDL |
| 24 | DanhSachThanhVien | Trần Văn Nam | Xử lý hiển thị danh sách thành viên |
| 25 | ThongTinThanhVien | Trần Văn Nam | Xử lý hiển thị thông tin thanh vien |
| 26 | SuaThanhVien | Trần Văn Nam | Xử lý sửa thành viên |
| 27 | XoaThanhVien | Trần Văn Nam | Xử lý xóa thành viên |
| 28 | ThemThanhVien | Trần Văn Nam | Xử lý thêm thành viên |
| 29 | DanhSachLop | Ngô Tuấn Lĩnh | Xử lý hiển thị danh sách lớp |
| 30 | DanhSachThanhVienLop | Ngô Tuấn Lĩnh | Xử lý hiển thị danh sách thành viên lớp. |
| 31 | XoaThanhVienLop | Ngô Tuấn Lĩnh | Xử lý xóa thành viên lớp |
| 32 | SuaLop | Ngô Tuấn Lĩnh | Xử lý sửa lớp |
| 33 | XoaLop | Ngô Tuấn Lĩnh | Xử lý xóa lớp |
| 34 | ThemLop | Ngô Tuấn Lĩnh | Xử lý thêm lớp |
| 35 | DBUtils |  | Thực hiện các truy vấn đến CSDL |
| 37 | ChiTietDeThiThiSinh | Lê Cao Liêm | Xử lý hiển thị nội dung đề thi cho thí sinh thi, tính điểm và xuất điểm cho thí sinh |
| 38 | DangXuatTS | Lê Cao Liêm | Xử lý đăng xuất cho thí sinh |
| 39 | DanhSachDeThiThiSinh | Lê Cao Liêm | Xử lý hiển thị danh sách các đề thi của thí sinh theo môn học |
| 40 | DanhSachMonHoc | Lê Cao Liêm | Xử lý hiển thị danh sách môn thi |
| 41 | ThongTinThiSinh | Lê Cao Liêm | Xử lý hiển thị thông tin thí sinh |

Mỗi lớp, mô tả các phương thức (được sử dụng trong chương trình) của 1 lớp.

## 4.1. Mẫu bảng mô tả các phương thức trong lớp CauHoi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV | Mục đích chính của phương thức | Code chương trình |
| 1 | Constructor CauHoi Các phương thức Getter và Setter | Nguyễn Quang Phú | Lấy và đặt các giá trị cho các thuộc tính của đối tượng câu hỏi | **public** CauHoi(String maCauHoi, String monHoc, String noiDung, String cauA, String cauB, String cauC, String cauD,  String dapAn) {  **super**();  **this**.maCauHoi = maCauHoi;  **this**.monHoc = monHoc;  **this**.noiDung = noiDung;  **this**.cauA = cauA;  **this**.cauB = cauB;  **this**.cauC = cauC;  **this**.cauD = cauD;  **this**.dapAn = dapAn;  }  **public** String getMaCauHoi() {  **return** maCauHoi;  }  **public** **void** setMaCauHoi(String maCauHoi) {  **this**.maCauHoi = maCauHoi;  }  **public** String getMonHoc() {  **return** monHoc;  }  **public** **void** setMonHoc(String monHoc) {  **this**.monHoc = monHoc;  }  **public** String getNoiDung() {  **return** noiDung;  } |

## 4.2. Mẫu bảng mô tả các phương thức trong lớp MonHoc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết | Mục đích chính của phương thức | Code chương trình |
| 1 | Constructor MonHoc  Các phương thức Getter và  Setter | Nguyễn Quang Phu | Lấy và đặt các giá trị cho các thuộc tính của đối tượng môn học | **public** **class** MonHoc {  **private** String MaMonHoc;  **private** String TenMonHoc;    **public** MonHoc()  {    }  **public** MonHoc(String MaMonHoc, String TenMonHoc)  {  **this**.MaMonHoc=MaMonHoc;  **this**.TenMonHoc=TenMonHoc;  }  **public** String getMaMonHoc() {  **return** MaMonHoc;  }  **public** **void** setMaMonHoc(String maMonHoc) {  MaMonHoc = maMonHoc;  }  **public** String getTenMonHoc() {  **return** TenMonHoc;  }  **public** **void** setTenMonHoc(String tenMonHoc) {  TenMonHoc = tenMonHoc;  } |

## 4.3. Mẫu bảng mô tả các phương thức trong lớp DeThi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết | Mục đích chính của phương thức | Code chương trình |
| 1 | Constructor DeThi  Các phương thức Getter và  Setter | Phạm Hoàng Phúc | Lấy và đặt các giá trị cho các thuộc tính của đối tượng đề thi | **public** DeThi(String maDe, String tenDe, String monHoc, **int** soCauHoi, **int** thoiGianThi, String thoiGianMo,String thoiGianDong, **int** soLanSubmit) {  **super**();  **this**.maDe = maDe;  **this**.tenDe = tenDe;  **this**.monHoc = monHoc;  **this**.soCauHoi = soCauHoi;  **this**.thoiGianThi = thoiGianThi;  **this**.thoiGianMo = thoiGianMo;  **this**.thoiGianDong = thoiGianDong;  **this**.soLanSubmit = soLanSubmit;  }  **public** String getMaDe() {  **return** maDe;  }  **public** **void** setMaDe(String maDe) {  **this**.maDe = maDe;  }  **public** String getTenDe() {  **return** tenDe;  }  **public** **void** setTenDe(String tenDe) {  **this**.tenDe = tenDe;  } |

## 4.4. Mẫu bảng mô tả các phương thức trong lớp Lop

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết | Mục đích chính của phương thức | Code chương trình |
| 1 | Constructor Lop  Các phương thức Getter và  Setter | Ngô Tuấn Lĩnh | Lấy và đặt các giá trị cho các thuộc tính của đối tượng lớp | **public** Lop(String Malop, String TenLop,String NgayLap)  {  **this**.MaLop=Malop;  **this**.TenLop=TenLop;  **this**.NgayLap=NgayLap;  }  **public** String getMaLop() {  **return** MaLop;  }  **public** **void** setMaLop(String maLop) {  MaLop = maLop;  }  **public** String getTenLop() {  **return** TenLop;  }  **public** **void** setTenLop(String tenLop) {  TenLop = tenLop;  }  **public** String getNgayLap() {  **return** NgayLap;  }  **public** **void** setNgayLap(String ngayLap) {  NgayLap = ngayLap;  } |

## 4.5. Mẫu bảng mô tả các phương thức trong lớp ThanhVien

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết | Mục đích chính của phương thức | Code chương trình |
| 1 | Constructor ThanhVien  Các phương thức Getter và  Setter | TranVanNam | Lấy và đặt các giá trị cho các thuộc tính của đối tượng ThanhVien | **public** ThanhVien(String MaThanhVien, String TenThanhVien, String NgaySinh,String GioiTinh, String Email, String SoDienThoai, String DiaChi, String PassWord, String HinhAnh)  {  **super**();  **this**.MaThanhVien= MaThanhVien;  **this**.TenThanhVien= TenThanhVien;  **this**.NgaySinh = NgaySinh;  **this**.GioiTinh = GioiTinh;  **this**.Email= Email;  **this**.SoDienThoai= SoDienThoai;  **this**.DiaChi= DiaChi;  **this**.PassWord= PassWord;  **this**.HinhAnh= HinhAnh;  }  **public** ThanhVien() {    }  **public** String getMaThanhVien() {  **return** MaThanhVien;  }  **public** **void** setMaThanhVien(String maThanhVien) {  **this**.MaThanhVien = maThanhVien;  }  **public** String getTenThanhVien() {  **return** TenThanhVien;  }  **public** **void** setTenThanhVien(String tenThanhVien) {  **this**.TenThanhVien = tenThanhVien;  } |

## 4.6. Mẫu bảng mô tả các phương thức trong lớp ChiTietDeThi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết | Mục đích chính của phương thức | Code chương trình |
| 1 | Constructor ChiTietDeThi  Các phương thức Getter và  Setter | Lê Cao Liêm | Lấy và đặt các giá trị cho các thuộc tính cần có của nội dung đề thi | **public** ChiTietDeThi( String tenDe, String thoiGianThi, String maCauHoi, String noiDung, String cauA,String cauB, String cauC, String cauD, String dapAn) {  **super**();  TenDe = tenDe;  ThoiGianThi = thoiGianThi;  MaCauHoi = maCauHoi;  NoiDung = noiDung;  CauA = cauA;  CauB = cauB;  CauC = cauC;  CauD = cauD;  DapAn = dapAn;  }    **public** String getTenDe() {  **return** TenDe;  }    **public** **void** setTenDe(String tenDe) {  TenDe = tenDe;  }    **public** String getThoiGianThi() {  **return** ThoiGianThi;  }    **public** **void** setThoiGianThi(String thoiGianThi) {  ThoiGianThi = thoiGianThi;  } |

## 4.7. Mẫu bảng mô tả các phương thức trong lớp DiemThi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết | Mục đích chính của phương thức trong chương trình | Code chương trình |
| 1 | Constructor DiemThi  Các phương thức Getter và  Setter | TranVanNam | Lấy và đặt các giá trị cho các thuộc tính của đối tượng DiemThi | **public** DiemThi(String maDe, String maThanhVien, String thoiGianKT, **float** diem) {  **super**();  MaDe = maDe;  MaThanhVien = maThanhVien;  ThoiGianKT = thoiGianKT;  Diem = diem;  }  **public** String getMaDe() {  **return** MaDe;  }  **public** **void** setMaDe(String maDe) {  MaDe = maDe;  }  **public** String getMaThanhVien() {  **return** MaThanhVien;  }  **public** **void** setMaThanhVien(String maThanhVien) {  MaThanhVien = maThanhVien;  } |

## 4.8. Mẫu bảng mô tả các phương thức trong lớp DBUtils (quanlycauhoi)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết | Mục đích chính của phương thức | Code chương trình |
| 1 | Find Question | Nguyễn Quang Phú | Lấy câu hỏi theo mã câu hỏi được yêu cầu | **public** **static** Question findQuestion(Connection conn, String MaCauHoi) **throws** SQLException {  String sql = "SELECT MaCauHoi,TenMonHoc,ChuDe,NoiDung,CauA,CauB,CauC,CauD,DapAn,MucDo " + "FROM CauHoi LEFT JOIN MonHoc ON MonHoc.MaMonHoc = CauHoi.MaMonHoc where MaCauHoi=?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, MaCauHoi);  ResultSet rs = pstm.executeQuery();  **while** (rs.next()) {  String TenMonHoc = rs.getString("TenMonHoc");  String ChuDe = rs.getString("ChuDe");  String NoiDung =rs.getString("NoiDung");  String CauA =rs.getString("CauA");  String CauB =rs.getString("CauB");  String CauC =rs.getString("CauC");  String CauD =rs.getString("CauD");  String DapAn =rs.getString("DapAn");  String MucDo =rs.getString("MucDo");  Question qt = **new** Question(MaCauHoi,TenMonHoc,ChuDe,NoiDung,CauA,CauB,CauC,CauD,DapAn,MucDo);  **return** qt;  }  **return** **null**;  } |
| 2 | Insert Question | Nguyễn Quang Phú | Thêm câu hỏi | **public** **static** **void** insertQuestion(Connection conn, Question question) **throws** SQLException {  String sql = "INSERT INTO CauHoi( MaCauHoi,MaMonHoc,ChuDe,NoiDung,CauA,CauB,CauC,CauD,DapAn,MucDo) VALUES(?,?,?,?,?,?,?,?,?,?)";    PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, question.getMaCauHoi());  pstm.setString(2, question.getMonHoc());  pstm.setString(3, question.getChuDe());  pstm.setString(4, question.getNoiDung());  pstm.setString(5, question.getCauA());  pstm.setString(6, question.getCauB());  pstm.setString(7, question.getCauC());  pstm.setString(8, question.getCauD());  pstm.setString(9, question.getDapAn());  pstm.setString(10, question.getMucDo());  pstm.executeUpdate();  } |
| 3 | Update Question | Nguyễn Quang Phú | Sửa câu hỏi | **public** **static** **void** updateQuestion(Connection conn, Question question) **throws** SQLException {  String sql = "UPDATE CauHoi SET MaMonHoc = ?, ChuDe = ?, NoiDung = ?,CauA=?,CauB=?,CauC=?,CauD=?,DapAn=?,MucDo=? "  + "WHERE MaCauHoi=? ";    PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, question.getMonHoc());  pstm.setString(2, question.getChuDe());  pstm.setString(3, question.getNoiDung());  pstm.setString(4, question.getCauA());  pstm.setString(5, question.getCauB());  pstm.setString(6, question.getCauC());  pstm.setString(7, question.getCauD());  pstm.setString(8, question.getDapAn());  pstm.setString(9, question.getMucDo());  pstm.setString(10, question.getMaCauHoi());    pstm.executeUpdate();  } |
| 4 | Delete Question | Nguyễn Quang Phú | Xóa câu hỏi | **public** **static** **void** deleteQuestion(Connection conn, String MaCauHoi) **throws** SQLException {  String sql = "DELETE FROM cauhoi WHERE MaCauHoi = ?";    PreparedStatement pstm = conn.prepareStatement(sql);    pstm.setString(1, MaCauHoi);    pstm.executeUpdate();  } |
| 5 | Query Question | Nguyễn Quang Phú | Lấy toàn bộ câu hỏi | **public** **static** List<Question> queryQuestion(Connection conn) **throws** SQLException {  String sql = "SELECT MaCauHoi,TenMonHoc,ChuDe,NoiDung,CauA,CauB,CauC,CauD,DapAn,MucDo FROM CauHoi "  + "LEFT JOIN MonHoc ON MonHoc.MaMonHoc = CauHoi.MaMonHoc";    PreparedStatement pstm = conn.prepareStatement(sql);    ResultSet rs = pstm.executeQuery();    List<Question> list = **new** ArrayList<Question>();    **while** (rs.next()) {  String TenMonHoc = rs.getString("TenMonHoc");  String ChuDe = rs.getString("ChuDe");  String MaCauHoi = rs.getString("MaCauHoi");  String NoiDung =rs.getString("NoiDung");  String CauA =rs.getString("CauA");  String CauB =rs.getString("CauB");  String CauC =rs.getString("CauC");  String CauD =rs.getString("CauD");  String DapAn =rs.getString("DapAn");  String MucDo =rs.getString("MucDo");    Question question = **new** Question();    question.setMonHoc(TenMonHoc);  question.setChuDe(ChuDe);  question.setMaCauHoi(MaCauHoi);  question.setNoiDung(NoiDung);  question.setCauA(CauA);  question.setCauB(CauB);  question.setCauC(CauC);  question.setCauD(CauD);  question.setDapAn(DapAn);  question.setMucDo(MucDo);    list.add(question);  }  **return** list;  } |
| 6 | Query MonHoc | Nguyễn Quang Phú | Lấy toàn bộ môn học | **public** **static** List<MonHoc> queryMonHoc(Connection conn) **throws** SQLException {  String sql = "SELECT \* FROM MonHoc";    PreparedStatement pstm = conn.prepareStatement(sql);    ResultSet rs = pstm.executeQuery();    List<MonHoc> list = **new** ArrayList<MonHoc>();    **while** (rs.next()) {  String TenMonHoc = rs.getString("TenMonHoc");  String MaMonHoc = rs.getString("MaMonHoc");  MonHoc monhoc =**new** MonHoc();    monhoc.setTenMonHoc(TenMonHoc);  monhoc.setMaMonHoc(MaMonHoc);      list.add(monhoc);  }  **return** list;  } |

## 4.9. Mẫu bảng mô tả các phương thức trong lớp MySQLConnUtils

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết | Mục đích chính của phương thức | Code chương trình |
| 1 | getMySQLConnection | Nguyễn Quang Phú | Tạo driver kết nối CSDL | **public** **static** Connection getMySQLConnection()  **throws** ClassNotFoundException, SQLException {  String hostName = "localhost";  String dbName = "thitracnhiem";  String userName = "root";  String password = "14111998";  **return** *getMySQLConnection*(hostName, dbName, userName, password);  }    **public** **static** Connection getMySQLConnection(String hostName, String dbName,  String userName, String password) **throws** SQLException,  ClassNotFoundException {    Class.*forName*("com.mysql.jdbc.Driver");    Connection con=DriverManager.*getConnection*(  "jdbc:mysql://localhost:3307/thitracnhiem?useUnicode=true&characterEncoding=UTF-8","root","14111998");    **return** con;  } |

## 4.10. Mẫu bảng mô tả các phương thức trong lớp QuestionListServlet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết | Mục đích chính của phương thức | Code chương trình |
| 1 | doGet | Nguyễn Quang Phú | Lấy danh sách các câu hỏi từ lớp query Question và đổ dữ liệu lên QuestionLisView.jsp | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String errorString = **null**;  List<Question> list = **null**;  **try** {  list = DBUtils.*queryQuestion*(conn);  } **catch** (SQLException e) {  e.printStackTrace();  errorString = e.getMessage();  }  request.setAttribute("questionList", list);  RequestDispatcher dispatcher = request.getServletContext()  .getRequestDispatcher("/QuanLiCauHoi/DanhSachCauHoi.jsp");  dispatcher.forward(request, response); |
| 2 | doPost |  |  |  |

## 4.11. Mẫu bảng mô tả các phương thức trong lớp ThemCauHoi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết | Mục đích chính của phương thức | Code chương trình |
| 1 | doGet | Nguyễn Quang Phú | Đưa đến trang ThemCauHoi. Jsp, đổ dữ liệu môn học | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  List<MonHoc> mh = **null**;    String errorString = **null**;    **try** {  mh = DBUtils.*queryMonHoc*(conn);  } **catch** (SQLException e) {  e.printStackTrace();  errorString = e.getMessage();  }      **if** (errorString != **null** && mh == **null**) {  response.sendRedirect(request.getServletPath() + "/DanhSachCauHoi");  **return**;  }    request.setAttribute("errorString", errorString);  request.setAttribute("monhoc", mh);  RequestDispatcher dispatcher = request.getServletContext()  .getRequestDispatcher("/QuanLiCauHoi/ThemCauHoi.jsp");  dispatcher.forward(request, response); |
| 2 | doPost |  | Thêm Câu hỏi mới | request.setCharacterEncoding("utf-8");  Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();    String MaCauHoi = (String) request.getParameter("MaCauHoi");  String MonHoc = (String) request.getParameter("MonHoc");  String ChuDe = (String) request.getParameter("ChuDe");  String NoiDung = (String) request.getParameter("NoiDung");  String CauA = (String) request.getParameter("CauA");  String CauB = (String) request.getParameter("CauB");  String CauC = (String) request.getParameter("CauC");  String CauD = (String) request.getParameter("CauD");  String DapAn = (String) request.getParameter("DapAn");  String MucDo = (String) request.getParameter("MucDo")  Question qt = **new** Question(MaCauHoi, MonHoc, ChuDe, NoiDung, CauA, CauB, CauC, CauD, DapAn, MucDo);  String errorString = **null**;  String regex = "\\w+";  **if** (MaCauHoi == **null** || !MaCauHoi.matches(regex)) {  errorString = "MaCauHoi Code invalid!";  }  **if** (errorString == **null**) {  **try** {  DBUtils.*insertQuestion*(conn, qt);  } **catch** (SQLException e) {  e.printStackTrace();  errorString = e.getMessage();  }  }  request.setAttribute("errorString", errorString);  request.setAttribute("question", qt);      **if** (errorString != **null**) {  RequestDispatcher dispatcher = request.getServletContext() .getRequestDispatcher("/QuanLiCauHoi/ThemCauHoi.jsp");  dispatcher.forward(request, response);  }  **else** { response.sendRedirect(request.getContextPath() + "/DanhSachCauHoi");  } |

## 4.12. Mẫu bảng mô tả các phương thức trong lớp SuaCauHoi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết | Mục đích chính của phương thức | Code chương trình |
| 1 | doGet | Nguyễn Quang Phú | Lấy câu hỏi theo mã câu hỏi từ lớp findQuestion và đổ lên trang SuaCauHoi. jsp | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaCauHoi = (String) request.getParameter("MaCauHoi");  Question qt = **null**;  List<MonHoc> mh= **null**;  String errorString = **null**;  qt = DBUtils.*findQuestion*(conn, MaCauHoi);  mh = DBUtils.*queryMonHoc*(conn);  **if** (qt == **null**) {  response.sendRedirect(request.getServletPath() + "/DanhSachCauHoi");  **return**;  }  request.setAttribute("question", qt);  request.setAttribute("monhoc", mh);  RequestDispatcher dispatcher = request.getServletContext()  .getRequestDispatcher("/QuanLiCauHoi/SuaCauHoi.jsp");  dispatcher.forward(request, response); |
| 2 | doPost |  | Sửa câu hỏi | request.setCharacterEncoding("utf-8");  Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaCauHoi = (String) request.getParameter("MaCauHoi");  String MonHoc = (String) request.getParameter("MonHoc");  String ChuDe = (String) request.getParameter("ChuDe");  String NoiDung = (String) request.getParameter("NoiDung");  String CauA = (String) request.getParameter("CauA");  String CauB = (String) request.getParameter("CauB");  String CauC = (String) request.getParameter("CauC");  String CauD = (String) request.getParameter("CauD");  String DapAn = (String) request.getParameter("DapAn");  String MucDo = (String) request.getParameter("MucDo");  Question qt = **new** Question(MaCauHoi, MonHoc, ChuDe, NoiDung, CauA, CauB, CauC,  CauD, DapAn, MucDo);  DBUtils.*updateQuestion*(conn,qt);  request.setAttribute("question", qt); response.sendRedirect(request.getContextPath() + "/DanhSachCauHoi");  }  } |

## 4.13. Mẫu bảng mô tả các phương thức trong lớp XoaCauHoi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết | Mục đích chính của phương thức trong chương trình | Code chương trình |
| 1 | doGet | Nguyễn Quang Phú | Nhận MaCauHoi và Gọi lớp deleteQuestion | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaCauHoi = (String) request.getParameter("MaCauHoi");    String errorString = **null**;  DBUtils.*deleteQuestion*(conn,MaCauHoi);  response.sendRedirect(request.getContextPath() + "/DanhSachCauHoi"); |
| 2 | doPost |  |  |  |

## 4.14. Mẫu bảng mô tả các phương thức trong lớp ChiTietCauHoi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết | Mục đích chính của phương thức trong chương trình | Code chương trình |
| 1 | doGet | Nguyễn Quang Phú | Nhận MaCauHoi và Gọi lớp findQuestion. Lấy dữ liệu và đổ lên ChiTietQCauHoi.jsp | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaCauHoi = (String) request.getParameter("MaCauHoi");  Question qt = **null**;  qt = DBUtils.*findQuestion*(conn, MaCauHoi); request.setAttribute("errorString", errorString);  request.setAttribute("question", qt);    RequestDispatcher dispatcher = request.getServletContext()  .getRequestDispatcher("/QuanLiCauHoi/ChiTietCauHoi.jsp");  dispatcher.forward(request, response); |
| 2 | doPost |  |  |  |

## 4.15. Mẫu bảng mô tả các phương thức trong lớp ThongTinTaiKhoan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết | Mục đích chính của phương thức trong chương trình | Code chương trình |
| 1 | doGet | Nguyễn Quang Phú | Chuyển tới trang ThongTinTaiKhoan.jsp | RequestDispatcher rs = request.getServletContext().getRequestDispatcher("/QuanLiCauHoi/ThongTinTaiKhoan.jsp");  rs.forward(request, response); |

## 4.16. Mẫu bảng mô tả các phương thức trong lớp SoanDeThi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stt | Tên phương thức | Tên SV phụ trách | Mục đích trong chương trình | Code chương trình |
| 1 | doGet | Phạm Hoàng Phúc | Hiển thị trang soan-de-thi.jsp | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String errorShowLop = **null**;  String errorShowMon = **null**;  List<MonHoc> listMonHoc = **null**;  List<Lop> listLopHoc = **null**;  listLopHoc = DBUtils.*ShowListLopHoc*(conn);  listMonHoc = DBUtils.*ShowListMonHoc*(conn);  request.setAttribute("listLopHoc", listLopHoc);  request.setAttribute("listMonHoc", listMonHoc);  RequestDispatcher dispatcher = request.getServletContext()  .getRequestDispatcher("/QuanLiDeThi/soan-de-thi.jsp");  dispatcher.forward(request, response); |
| 2 | doPost | Phạm Hoàng Phúc | Lưu một đề thi xuống CSDL | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  request.setCharacterEncoding("utf-8");  String maDe = (String) request.getParameter("maDe");  String tenDe = (String) request.getParameter("tenDe");  String soCauHoiSt = (String) request.getParameter("soCauHoi");  String thoiGianLamBaiSt = (String) request.getParameter("thoiGianLamBai");    String thoiGianMoSt = (String) request.getParameter("thoiGianMo");  String thoiGianMo = thoiGianMoSt.replace("T", " ");    String thoiGianDongSt = (String) request.getParameter("thoiGianDong");  String thoiGianDong = thoiGianDongSt.replace("T", " ");    String soLanSubmitSt = (String) request.getParameter("soLanSubmit");  String monHoc = (String) request.getParameter("monHoc");    String soCauDeSt = (String) request.getParameter("soCauDe");  String soCauTrungBinhSt = (String) request.getParameter("soCauTrungBinh");  String soCauKhoSt = (String) request.getParameter("soCauKho");    String[] dsLopHoc = request.getParameterValues("maLop");    **int** soCauHoi = 0;  **int** thoiGianLamBai = 0;  **int** soLanSubmit = 0;  **int** soCauDe = 0;  **int** soCauTrungBinh = 0;  **int** soCauKho = 0;    **try** {  soCauHoi = Integer.*parseInt*(soCauHoiSt);  thoiGianLamBai = Integer.*parseInt*(thoiGianLamBaiSt);  soLanSubmit = Integer.*parseInt*(soLanSubmitSt);  soCauDe = Integer.*parseInt*(soCauDeSt);  soCauTrungBinh = Integer.*parseInt*(soCauTrungBinhSt);  soCauKho = Integer.*parseInt*(soCauKhoSt);    }**catch**(Exception e) {    }  DeThi deThi = **new** DeThi(maDe, tenDe, monHoc, soCauHoi, thoiGianLamBai, thoiGianMo, thoiGianDong, soLanSubmit);    **try** {  DBUtils.*LuuDeThi*(conn, deThi);  } **catch** (SQLException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  errorSoanDe = e.getMessage();  }    List<CauHoi> listCauHoi = **new** ArrayList<CauHoi>();    List<CauHoi> listCauHoiDe = **null**;  List<CauHoi> listCauHoiTrungBinh = **null**;  List<CauHoi> listCauHoiKho = **null**;    **try** {  listCauHoiDe = DBUtils.*ListCauHoi*(conn, soCauDe, monHoc, "De");  listCauHoiTrungBinh = DBUtils.*ListCauHoi*(conn, soCauTrungBinh, monHoc, "Trung binh");  listCauHoiKho = DBUtils.*ListCauHoi*(conn, soCauKho, monHoc, "Kho");  } **catch** (SQLException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  errorSoanDe = e.getMessage();  }    **if**(errorSoanDe == **null**) {  listCauHoi.addAll(listCauHoiDe); listCauHoi.addAll(listCauHoiTrungBinh); listCauHoi.addAll(listCauHoiKho);  **for**(CauHoi i : listCauHoi) { DBUtils.*LuuCauHoi\_De*(conn, deThi, i);  **for**(String i : dsLopHoc) {    DBUtils.*LuuLop\_De*(conn, deThi, i);  }  response.sendRedirect(request.getContextPath() + "/DanhSachDeThi");  }  **else** {  response.sendRedirect(request.getContextPath() + "/SoanDeThi");  }  } |

## 4.17. Mẫu bảng mô tả các phương thức trong lớp ChinhSuaDeThi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stt | Tên phương thức | Tên SV phụ trách | Mục đích trong chương trình | Code chương trình |
| 1 | doGet | Phạm Hoàng Phúc | Kết nối CSDL lấy đề thi cần sửa sau đó chuyển đến trang chinh-sua-de-thi.jsp và hiển thị nội dung đề thi cần cần sửa | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String maDe = (String) request.getParameter("maDe");  DeThi deThi = **null**;  deThi = DBUtils.*ShowDetailDeThi*(conn, maDe);    **if**(deThi == **null**) {  response.sendRedirect(request.getContextPath() + "/DanhSachDeThi");  }  **else** { request.setAttribute("maDe", maDe); request.setAttribute("deThi", deThi);  RequestDispatcher dispatcher = request.getServletContext().getRequestDispatcher("/QuanLiDeThi/chinh-sua-de-thi.jsp");  dispatcher.forward(request, response);  } |
| 2 | doPost | Phạm Hoàng Phúc | Lưu lại nội dung đề thi vừa được sửa xuống CSDL | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  request.setCharacterEncoding("utf-8");  String maDe = (String) request.getParameter("maDe");  String tenDe = (String) request.getParameter("tenDe");  String soLanSubmitSt = (String) request.getParameter("soLanSubmit");  String thoiGianThiSt = (String) request.getParameter("thoiGianThi");  String thoiGianMoSt = (String) request.getParameter("thoiGianMo");  String thoiGianMo = thoiGianMoSt.replace("T", " "); String thoiGianDongSt = (String) request.getParameter("thoiGianDong");  String thoiGianDong = thoiGianDongSt.replace("T", " ")  **int** soLanSubmit = 0;  **int** thoiGianThi = 0;  String errorEditDeThi = **null**;  soLanSubmit = Integer.*parseInt*(soLanSubmitSt);  thoiGianThi = Integer.*parseInt*(thoiGianThiSt);    DeThi deThi = **new** DeThi();  deThi.setMaDe(maDe);  deThi.setTenDe(tenDe);  deThi.setSoLanSubmit(soLanSubmit);  deThi.setThoiGianThi(thoiGianThi);  deThi.setThoiGianMo(thoiGianMo);  deThi.setThoiGianDong(thoiGianDong);  DBUtils.*EditDeThi*(conn, deThi);    request.setAttribute("deThi", deThi); response.sendRedirect(request.getContextPath() + "/DanhSachDeThi");  } |

## 4.18. Mẫu bảng mô tả các phương thức trong lớp ChiTietDeThi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stt | Tên phương thức | Tên SV phụ trách | Mục đích trong chương trình | Code chương trình |
| 1 | doGet | Phạm Hoàng Phúc | Kết nối CSDL lấy thông tin đề thi sau đó chuyển đến trang chi-tiet-de-thi.jsp và hiển thị nội dung đề thi lên | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String maDe = (String) request.getParameter("maDe");  DeThi deThi = **null**;  List<String> listTenLop =**null**;  List<CauHoi> listCauHoi =**null**;  deThi = DBUtils.*ShowDetailDeThi*(conn, maDe);  listTenLop = DBUtils.*ListLop\_De*(conn, maDe);  listCauHoi = DBUtils.*ListCauHoi\_De*(conn, maDe); |

## 4.19. Mẫu bảng mô tả các phương thức trong lớp DangXuatQLDT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stt | Tên phương thức | Tên SV phụ trách | Mục đích trong chương trình | Code chương trình |
| 1 | doGet | Phạm Hoàng Phúc | Xóa session userQLDT và chuyển đến trang Login | HttpSession session = request.getSession();  session.removeAttribute("userQLDT"); response.sendRedirect(request.getContextPath() + "/Login"); |

## 4.20. Mẫu bảng mô tả các phương thức trong lớp XoaDeThi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stt | Tên phương thức | Tên SV phụ trách | Mục đích trong chương trình | Code chương trình |
| 1 | doGet | Phạm Hoàng Phúc | Xóa một đề thi khỏi CSDL | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String maDe = (String) request.getParameter("maDe");  DBUtils.*DeleteDeThi*(conn, maDe); |

## 4.21. Mẫu bảng mô tả các phương thức trong lớp NguoiQuanLiDeThi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stt | Tên phương thức | Tên SV phụ trách | Mục đích trong chương trình | Code chương trình |
| 1 | doGet | Phạm Hoàng Phúc | Hiển thị trang nguoi-quan-li-de-thi.jsp | RequestDispatcher dispatcher = request.getServletContext() .getRequestDispatcher("/QuanLiDeThi/nguoi-quan-li-de-thi.jsp");  dispatcher.forward(request, response); |

## 4.22. Mẫu bảng mô tả các phương thức trong lớp DBUtils (quanlidethi.utils)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stt | Tên phương thức | Tên SV phụ trách | Mục đích trong chương trình | Code chương trình |
| 1 | EditDeThi | Phạm Hoàng Phúc | Update nội dung đề thi ở dưới CSDL | String sql = "UPDATE DeThi SET TenDe=?, SoLanSubmit=?, ThoiGianThi=?, ThoiGianMo=?, ThoiGianDong=? WHERE MaDe=?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, deThi.getTenDe());  pstm.setInt(2, deThi.getSoLanSubmit());  pstm.setInt(3, deThi.getThoiGianThi());  pstm.setString(4, deThi.getThoiGianMo());  pstm.setString(5, deThi.getThoiGianDong());  pstm.setString(6, deThi.getMaDe());  pstm.executeUpdate(); |
| 2 | ShowDetailDeThi | Phạm Hoàng Phúc | Lấy lên nội dung đề thi từ CSDL | String sql = "SELECT TenDe, SoCauHoi, ThoiGianThi, ThoiGianMo, ThoiGianDong, SoLanSubmit FROM DeThi WHERE MaDe=?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, maDe);  ResultSet rs = pstm.executeQuery();  **while**(rs.next()){  **int** soCauHoi = rs.getInt("SoCauHoi");  **int** thoiGianThi = rs.getInt("ThoiGianThi");  **int** soLanSubmit = rs.getInt("SoLanSubmit");  String thoiGianDong = rs.getString("ThoiGianDong").replace(" ", "T");  String thoiGianMo = rs.getString("ThoiGianMo").replace(" ", "T");  String tenDe = rs.getString("TenDe");  DeThi deThi = **new** DeThi();  deThi.setTenDe(tenDe);  deThi.setSoCauHoi(soCauHoi);  deThi.setThoiGianThi(thoiGianThi);  deThi.setThoiGianMo(thoiGianMo);  deThi.setThoiGianDong(thoiGianDong);  deThi.setSoLanSubmit(soLanSubmit);  **return** deThi;  }  **return** **null**; |
| 3 | ListDe\_Lop | Phạm Hoàng phúc | Lấy lên ds lớp của một đề thi từ CSDL | String sql = "SELECT TenLop FROM De\_Lop, Lop WHERE De\_Lop.MaLop = Lop.MaLop AND MaDe=?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, maDe);  ResultSet rs = pstm.executeQuery();  List<String> listTenMon = **new** ArrayList<String>();  **while**(rs.next()) {  String tenMon = rs.getString("TenLop");  listTenMon.add(tenMon);  }  **return** listTenMon; |
| 4 | ListCauHoi\_De | Phạm Hoàng Phúc | Lấy lên ds câu hỏi của một đề thi từ CSDL | String sql = "SELECT CauHoi.MaCauHoi, NoiDung, CauA, CauB, CauC, CauD, DapAn FROM CauHoi, De\_Cau WHERE CauHoi.MaCauHoi = De\_Cau.MaCauHoi AND MaDe=?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, maDe);  ResultSet rs = pstm.executeQuery();  List<CauHoi> listCauHoi = **new** ArrayList<CauHoi>();  **while**(rs.next()) {  String maCauHoi = rs.getString("MaCauHoi");  String noiDung = rs.getString("NoiDung");  String cauA = rs.getString("CauA");  String cauB = rs.getString("CauB");  String cauC = rs.getString("CauC");  String cauD = rs.getString("CauD");  String dapAn = rs.getString("DapAn");  CauHoi cauHoi = **new** CauHoi();  cauHoi.setMaCauHoi(maCauHoi);  cauHoi.setNoiDung(noiDung);  cauHoi.setCauA(cauA);  cauHoi.setCauB(cauB);  cauHoi.setCauC(cauC);  cauHoi.setCauD(cauD);  cauHoi.setDapAn(dapAn);  listCauHoi.add(cauHoi);  }  **return** listCauHoi; |
| 5 | ShowListDeThi | Phạm Hoàng Phúc | Lấy lên ds tất cả các đề thi từ CSDL | String sql = "SELECT MaDe, TenDe, MonHoc FROM DeThi";  PreparedStatement pstm = conn.prepareStatement(sql);  ResultSet rs = pstm.executeQuery();  List<DeThi> listDeThi = **new** ArrayList<DeThi>();  **while**(rs.next()) {  String maDe = rs.getString("MaDe");  String tenDe = rs.getString("TenDe");  String monHoc = rs.getString("MonHoc");  DeThi deThi = **new** DeThi();  deThi.setMaDe(maDe);  deThi.setTenDe(tenDe);  deThi.setMonHoc(monHoc);  listDeThi.add(deThi);  }  **return** listDeThi; |
| 6 | ShowListLopHoc | Phạm Hoàng Phúc | Lấy lên ds tất cả các lớp học từ CSDL | String sql = "SELECT Malop, TenLop, NgayLap FROM Lop";  PreparedStatement pstm = conn.prepareStatement(sql);  ResultSet rs = pstm.executeQuery();  List<Lop> listLopHoc = **new** ArrayList<Lop>();  **while**(rs.next()) {  String maLop = rs.getString("MaLop");  String tenLop = rs.getString("TenLop");  String ngayLap = rs.getString("NgayLap");  Lop lopHoc = **new** Lop(maLop, tenLop, ngayLap);  listLopHoc.add(lopHoc);  }  **return** listLopHoc; |
| 7 | ShowListMonHoc | Phạm Hoàng Phúc | Lấy lên ds môn học từ CSDL | String sql = "SELECT \* FROM MonHoc";  PreparedStatement pstm = conn.prepareStatement(sql);  ResultSet rs = pstm.executeQuery();  List<MonHoc> listMonHoc = **new** ArrayList<MonHoc>();  **while**(rs.next()) {  String maMon = rs.getString("MaMonHoc");  String tenMon = rs.getString("TenMonHoc");    MonHoc monHoc = **new** MonHoc(maMon, tenMon);  listMonHoc.add(monHoc);  }  **return** listMonHoc; |
| 8 | LuuDeThi | Phạm Hoàng Phúc | Thêm một đề thi vào CSDL | String sql = "INSERT INTO DeThi(MaDe, TenDe, MonHoc, SoCauHoi, ThoiGianThi, " + "ThoiGianMo, ThoiGianDong, SoLanSubmit) VALUES(?,?,?,?,?,?,?,?)";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, deThi.getMaDe());  pstm.setString(2, deThi.getTenDe());  pstm.setString(3, deThi.getMonHoc());  pstm.setInt(4, deThi.getSoCauHoi());  pstm.setInt(5, deThi.getThoiGianThi());  pstm.setString(6, deThi.getThoiGianMo());  pstm.setString(7, deThi.getThoiGianDong());  pstm.setInt(8, deThi.getSoLanSubmit());  pstm.executeUpdate(); |
| 9 | LuuCauHoi\_De | Phạm Hoàng Phúc | Thêm một câu hỏi vào bảng De\_Cau ở dưới CSDL | String sql = "INSERT INTO De\_Cau(MaDe, MaCauHoi) VALUES(?,?)";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, deThi.getMaDe());  pstm.setString(2, cauHoi.getMaCauHoi());  pstm.executeUpdate(); |
| 10 | LuuLop\_De | Phạm Hoàng Phúc | Thêm một lớp vào bảng De\_Lop ỏ dưới CSDL | String sql = "INSERT INTO De\_Lop(MaDe, MaLop) VALUES(?,?)";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, deThi.getMaDe());  pstm.setString(2, maLop);  pstm.executeUpdate(); |
| 11 | ListCauHoi | Phạm Hoàng Phúc | Lấy ngẫu nhiên ds câu hỏi theo môn học và mức độ | String sql = "SELECT MaCauHoi FROM CauHoi, MonHoc WHERE CauHoi.MaMonHoc = MonHoc.MaMonHoc " + "AND CauHoi.MaMonHoc=? AND MucDo=? ORDER BY RAND() LIMIT ?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, monHoc);  pstm.setString(2, mucDo);  pstm.setInt(3, soCau);  List<CauHoi> listCauHoi = **new** ArrayList<CauHoi>();  ResultSet rs = pstm.executeQuery();  **while**(rs.next()) {  String maCauHoi = rs.getString("MaCauHoi");  CauHoi cauHoi = **new** CauHoi(); cauHoi.setMaCauHoi(maCauHoi);  listCauHoi.add(cauHoi);  }  **return** listCauHoi; |
| 12 | DeleteDeThi | Phạm Hoàng Phúc | Xóa một đề thi khỏi CSDL | String sql = "DELETE FROM DeThi WHERE MaDe=?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, maDe);  pstm.executeUpdate(); |

## 4.23. Mẫu bảng mô tả các phương thức trong lớp DBUtils (public.utils)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stt | Tên phương thức | Tên SV phụ trách | Mục đích trong chương trình | Code chương trình |
| 1 | FindUserLogin | Phạm Hoàng Phúc | Lấy lên thông tin một user | String sql = "SELECT UserName, PassWord, Role FROM users WHERE UserName=?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, username);  ResultSet rs = pstm.executeQuery();  **while**(rs.next()) {  String username1 = rs.getString("UserName");  String password = rs.getString("PassWord");  String role = rs.getString("Role");  User user = **new** User(username1, password, role);  **return** user;  }  **return** **null**;  } |

## 4.24. Mẫu bảng mô tả các phương thức trong lớp DanhSachThanhVien

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách | Mục đích chính của phương thức trong chương trình | Code chương trình |
| 1 | doGet | Trần Văn Nam | Lấy danh sách thí sinh từ lớp queryThanhVien và đổ dữ liệu vào trang DanhSachThanhVien.jsp | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String errorString = **null**;  List<ThanhVien> list = **null**;  list = DBUtils.*queryThanhVien*(conn);  RequestDispatcher dispatcher = request.getServletContext() .getRequestDispatcher("/QuanLiThiSinh/DanhSachThanhVien.jsp");  dispatcher.forward(request, response); |

## 4.25. Mẫu bảng mô tả các phương thức trong lớp DanhSachLop

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách | Mục đích chính của phương thức trong chương trình | Code chương trình |
| 1 | doGet | Ngô Tuấn Lĩnh | Lấy danh sách thí sinh từ lớp queryLop và đổ dữ liệu vào trang DanhSachLop.jsp | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String errorString = **null**;  List<Lop> list = **null**;  list = DBUtils.*queryLop*(conn); request.setAttribute("lopList", list); RequestDispatcher dispatcher = request.getServletContext() .getRequestDispatcher("/QuanLiThiSinh/DanhSachLop.jsp");  dispatcher.forward(request, response); |

## 4.26. Mẫu bảng mô tả các phương thức trong lớp DanhSachThanhVienLop

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết | Mục đích chính của phương thức | Code chương trình |
| 1 | doGet | Ngô Tuấn Lĩnh | Lấy danh sách thành viên theo mã lớp trong lớp TimThanhVienLop với biến MaLop được truyền vào. Đổ dữ liệu lên trang DanhSachThanhVienLop.jsp | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaLop = (String) request.getParameter("MaLop");  List<ThanhVien> tv = **null**;  String errorString = **null**;  tv = DBUtils.*TimThanhVienLop*(conn, MaLop);  **if** (tv == **null**) {  response.sendRedirect(request.getServletPath()  + "/DanhSachLop");  **return**;  }  request.setAttribute("thanhvienlop", tv);  request.setAttribute("MaLop", MaLop);  RequestDispatcher dispatcher = request.getServletContext() .getRequestDispatcher("/QuanLiThiSinh/DanhSachThanhVienLop.jsp");  dispatcher.forward(request, response); |

## 4.27. Mẫu bảng mô tả các phương thức trong lớp SuaLop

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách | Mục đích chính của phương thức trong chương trình | Code chương trình |
| 1 | doget | Ngô Tuấn Lĩnh | Lấy danh sách thành viên theo mã lớp trong PT TimThanhVienConLai, Lấy thông tin lớp PT từ TimLop với biến MaLop được truyền vào. Đổ dữ liệu lên trang SuaLop.jsp | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaLop = (String)request.getParameter("MaLop");  Lop lop = **null**;  List<ThanhVien> tv = **null**;  lop = DBUtils.*TimLop*(conn, MaLop);  tv = DBUtils.*TimThanhVienConlai*(conn, MaLop); request.setAttribute("errorString", errorString);  request.setAttribute("lop", lop);  request.setAttribute("thanhvien", tv);  RequestDispatcher dispatcher = request.getServletContext() .getRequestDispatcher("/QuanLiThiSinh/SuaLop.jsp");  dispatcher.forward(request, response); |
| 2 | dopost | Ngô Tuấn Lĩnh | Gọi PT SuaLop, ThemTVLop để sửa lớp và bổ sung thành viên | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaLop = (String) request.getParameter("MaLop");  String TenLop = (String) request.getParameter("TenLop");  String NgayLap = (String) request.getParameter("NgayLap");  String[] dsthanhvien = request.getParameterValues("DanhSach");  **if**(dsthanhvien!=**null**)  { **if**(dsthanhvien[0].equalsIgnoreCase("0"))  {  dsthanhvien=**null**;  }  }  Lop lop = **new** Lop(MaLop, TenLop, NgayLap);  String errorString = **null**;  DBUtils.*SuaLop*(conn,lop);  **if**(dsthanhvien!=**null**)  {  **for** (String tv : dsthanhvien) {  DBUtils.*ThemTVLop*(conn, MaLop, tv);  }  }  request.setAttribute("lop", lop); response.sendRedirect(request.getContextPath() + "/DanhSachLop"); |

## 4.28. Mẫu bảng mô tả các phương thức trong lớp SuaThanhVien

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách | Mục đích chính của phương thức | Code chương trình |
| 1 | Doget | Trần Văn Nam | Lấy thông tin thành viên theo mã thành viên trong PT TimThanhVien với biến MaThanhVien được truyền vào. Đổ dữ liệu lên trang SuaThanhVien.jsp | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaThanhVien = (String) request.getParameter("MaThanhVien");  ThanhVien tv = **null**;  String errorString = **null**;  tv = DBUtils.*TimThanhVien*(conn, MaThanhVien);  **if** (tv == **null**) { response.sendRedirect(request.getServletPath() + "/DanhSachThanhVien");  **return**;  } request.setAttribute("thanhvien", tv);  RequestDispatcher dispatcher = request.getServletContext() .getRequestDispatcher("/QuanLiThiSinh/SuaThanhVien.jsp");  dispatcher.forward(request, response); |
| 2 | dopost |  | Gọi PT SuaThanhVien để sửa thành viên | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaThanhVien = (String) request.getParameter("MaThanhVien");  String TenThanhVien = (String) request.getParameter("TenThanhVien");  String NgaySinh = (String) request.getParameter("NgaySinh");  String GioiTinh = (String) request.getParameter("GioiTinh");  String Email = (String) request.getParameter("Email");  String SoDienThoai = (String) request.getParameter("SoDienThoai");  String DiaChi = (String) request.getParameter("DiaChi");  String PassWord = (String) request.getParameter("PassWord");  String HinhAnh = (String) request.getParameter("HinhAnh");  ThanhVien tv = **new** ThanhVien(MaThanhVien, TenThanhVien, NgaySinh, GioiTinh, Email, SoDienThoai, DiaChi, PassWord, HinhAnh); DBUtils.*SuaThanhVien*(conn,tv);  request.setAttribute("thanhvien", tv);  response.sendRedirect(request.getContextPath() + "/DanhSachThanhVien"); |

## 4.29. Mẫu bảng mô tả các phương thức trong lớp ThemLop

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết | Mục đích chính của phương thức trong chương trình | Code chương trình |
| 1 | doget | Ngô Tuấn Lĩnh | Gọi PT queryThanhVien và đổ dữ liệu vào trang ThemLop.jsp | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  List<ThanhVien> tv = **null**;  tv = DBUtils.*queryThanhVien*(conn);  **if** (tv == **null**) { response.sendRedirect(request.getServletPath() + "/DanhSachLop");  **return**;  }  request.setAttribute("thanhvien", tv);  RequestDispatcher dispatcher = request.getServletContext() .getRequestDispatcher("/QuanLiThiSinh/ThemLop.jsp");  dispatcher.forward(request, response); |
| 2 | dopost |  | Gọi PT ThemLop và themTVLop để tạo lớp và tạo danh sách thành viên cho lớp | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaLop = (String) request.getParameter("MaLop");  String TenLop = (String) request.getParameter("TenLop");  DateTimeFormatter dtf = DateTimeFormatter.*ofPattern*("dd/MM/yyyy");  LocalDate localDate = LocalDate.*now*();  String NgayLap = dtf.format(localDate);  String[] dsthanhvien = request.getParameterValues("DanhSach");  **if**(dsthanhvien!=**null**)  {  **if**(dsthanhvien[0].equalsIgnoreCase("0"))  {  dsthanhvien=**null**;  }  }  Lop lop = **new** Lop(MaLop, TenLop, NgayLap);  String regex = "\\w+";  DBUtils.*ThemLop*(conn, lop);  **if**(dsthanhvien!=**null**)  {  **for** (String tv : dsthanhvien) {  DBUtils.*ThemTVLop*(conn, MaLop, tv);  }  }  request.setAttribute("lop", lop);  response.sendRedirect(request.getContextPath() + "/DanhSachLop");} |

## 4.30. Mẫu bảng mô tả các phương thức trong lớp ThongTinTaiKhoan (quanlythisinh)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách | Mục đích chính của phương thức | Code chương trình |
|  | doget | Trần Văn Nam | Chuyển đến trang ThongTinTaiKhoan.jsp | RequestDispatcher rs = request.getServletContext().getRequestDispatcher("/QuanLiThiSinh/ThongTinTaiKhoan.jsp");  rs.forward(request, response); |

## 4.31. Mẫu bảng mô tả các phương thức trong lớp ThongTinThanhVien

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết | Mục đích chính của phương thức trong chương trình | Code chương trình |
| 1 | doget | Trần Văn Nam | Lấy thông tin thành viên theo mã thành viên trong PT TimThanhVien với biến MaThanhVien được truyền vào. Đổ dữ liệu lên trang ThongTinThanhVien.jsp | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaThanhVien = (String) request.getParameter("MaThanhVien");  ThanhVien tv = **null**;  tv = DBUtils.*TimThanhVien*(conn, MaThanhVien);  **if** (tv == **null**) { response.sendRedirect(request.getServletPath() + "/DanhSachThanhVien");  **return**;  }  request.setAttribute("thanhvien", tv);  RequestDispatcher dispatcher = request.getServletContext() .getRequestDispatcher("/QuanLiThiSinh/ThongTinThanhVien.jsp");  dispatcher.forward(request, response); |

## 4.32. Mẫu bảng mô tả các phương thức trong lớp XoaLop

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách | Mục đích chính của phương thức | Code chương trình |
| 1 | doget | Ngô Tuấn Lĩnh | Gọi PT XoaLop với biến MaLop được truyền vào | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaLop = (String) request.getParameter("MaLop");  DBUtils.*XoaLop*(conn, MaLop); response.sendRedirect(request.getContextPath() + "/DanhSachLop"); |

## 4.33. Mẫu bảng mô tả các phương thức trong lớp XoaThanhVien

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách | Mục đích chính của phương thức | Code chương trình |
|  | Doget | Trần Văn Nam | Gọi PT XoaThanhVien với biến MaThanhVien được truyền vào | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaThanhVien = (String) request.getParameter("MaThanhVien");  DBUtils.*XoaThanhVien*(conn, MaThanhVien); response.sendRedirect(request.getContextPath() + "/DanhSachThanhVien"); |

## 4.34. Mẫu bảng mô tả các phương thức trong lớp XoaThanhVienLop

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách | Mục đích chính của phương thức | Code chương trình |
| 1 | DoGet | Ngô Tuấn Lĩnh | Gọi PT XoaThanhVienLop với MaLop và MaThanhVien được truyền vào | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaThanhVien = (String) request.getParameter("MaThanhVien");  String MaLop = (String) request.getParameter("MaLop");  DBUtils.*XoaThanhVienLop*(conn, MaThanhVien, MaLop);  response.sendRedirect(request.getContextPath() + "/DanhSachLop"); |

## 4.35. Mẫu bảng mô tả các phương thức trong lớp DBUtils (quanlithisinh.utils)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stt | Tên phương thức | Tên SV phụ trách | Mục đích trong chương trình | Code chương trình |
| 1 | queryThanhVien | Trần Văn Nam | Lấy danh sách thành viên từ CSDL | String sql = "SELECT \* FROM ThanhVien";  PreparedStatement pstm = conn.prepareStatement(sql);  ResultSet rs = pstm.executeQuery();  List<ThanhVien> list = **new** ArrayList<ThanhVien>();  **while** (rs.next()) {  String MaThanhVien = rs.getString("MaThanhVien");  String TenThanhVien = rs.getString("TenThanhVien");  String NgaySinh = rs.getString("NgaySinh");  String GioiTinh =rs.getString("GioiTinh");  String Email =rs.getString("Email");  String SoDienThoai =rs.getString("SoDienThoai");  String DiaChi =rs.getString("DiaChi");  String PassWord =rs.getString("PassWord");  String HinhAnh =rs.getString("HinhAnh");  ThanhVien thanhvien = **new** ThanhVien(MaThanhVien, TenThanhVien, NgaySinh, GioiTinh, Email, SoDienThoai, DiaChi, PassWord, HinhAnh); thanhvien.setMaThanhVien(MaThanhVien); thanhvien.setTenThanhVien(TenThanhVien);  thanhvien.setNgaySinh(NgaySinh);  thanhvien.setGioiTinh(GioiTinh);  thanhvien.setEmail(Email); thanhvien.setSoDienThoai(SoDienThoai);  thanhvien.setDiaChi(DiaChi);  thanhvien.setPassWord(PassWord);  thanhvien.setHinhAnh(HinhAnh);  list.add(thanhvien);  }  **return** list; |
| 2 | queryLop | Ngô Tuấn Lĩnh | Lấy danh sách lớp học từ CSDL | String sql = "SELECT \* FROM Lop";  PreparedStatement pstm = conn.prepareStatement(sql);  ResultSet rs = pstm.executeQuery();  List<Lop> list = **new** ArrayList<Lop>();  **while** (rs.next()) {  String MaLop = rs.getString("MaLop");  String TenLop = rs.getString("TenLop");  String NgayLap = rs.getString("NgayLap");  Lop lop=**new** Lop();  lop.setMaLop(MaLop);  lop.setTenLop(TenLop);  lop.setNgayLap(NgayLap);  list.add(lop);  }  **return** list; |
| 3 | insertThanhVien | Trần Văn Nam | Thêm một thành viên mới xuống CSDL | String sql = "INSERT into ThanhVien ( MaThanhVien,TenThanhVien,NgaySinh,GioiTinh,Email,SoDienThoai,DiaChi,PassWord,HinhAnh) VALUES (?,?,?,?,?,?,?,?,?)";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, thanhvien.getMaThanhVien());  pstm.setString(2, thanhvien.getTenThanhVien());  pstm.setString(3, thanhvien.getNgaySinh());  pstm.setString(4, thanhvien.getGioiTinh());  pstm.setString(5, thanhvien.getEmail());  pstm.setString(6, thanhvien.getSoDienThoai(); pstm.setString(7, thanhvien.getDiaChi());  pstm.setString(8, thanhvien.getPassWord());  pstm.setString(9, thanhvien.getHinhAnh());  pstm.executeUpdate(); |
| 4 | SuaLop | Ngô Tuấn Lĩnh | Sửa thông tin lớp học theo mã lớp học dưới CSDL | String sql = "UPDATE Lop SET TenLop= ? WHERE MaLop = ?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(2, lop.getMaLop());  pstm.setString(1, lop.getTenLop()); pstm.executeUpdate(); |
| 5 | SuaThanhVien | Trần Văn Nam | Sửa thông tin thành viên theo mã thành viên dưới CSDL | String sql = "UPDATE ThanhVien SET TenThanhVien=?,NgaySinh=?,GioiTinh=?, Email=?,SoDienThoai=?,DiaChi=?,PassWord=?,HinhAnh=? WHERE MaThanhVien=?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, thanhvien.getTenThanhVien());  pstm.setString(2, thanhvien.getNgaySinh());  pstm.setString(3, thanhvien.getGioiTinh());  pstm.setString(4, thanhvien.getEmail());  pstm.setString(5, thanhvien.getSoDienThoai());  pstm.setString(6, thanhvien.getDiaChi());  pstm.setString(7, thanhvien.getPassWord());  pstm.setString(8, thanhvien.getHinhAnh());  pstm.setString(9, thanhvien.getMaThanhVien());  pstm.executeUpdate(); |
| 6 | ThemLop | Ngô Tuấn Lĩnh | Thêm một lớp mới xuống dưới CSDL | String sql = "INSERT into Lop ( MaLop,TenLop,NgayLap) VALUES (?,?,?)"; PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, lop.getMaLop());  pstm.setString(2, lop.getTenLop());  pstm.setString(3, lop.getNgayLap());  pstm.executeUpdate(); |
| 7 | ThemTVLop | Ngô Tuấn Lĩnh | Thêm mã lớp và mã thành viên vào bảng Lop-TV trong CSDL | String sql = "INSERT into Lop\_TV (MaThanhVien,MaLop) VALUES (?,?)";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(2, MaLop);  pstm.setString(1, MaThanhVien);  pstm.executeUpdate(); |
| 8 | TimLop | Ngô Tuấn Lĩnh | Lấy dữ liệu của một lớp học trong CSDL | String sql = "SELECT \* FROM Lop where MaLop=?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, MaLop);  ResultSet rs = pstm.executeQuery();  **while** (rs.next()) {  String TenLop = rs.getString("TenLop");  String NgayLap = rs.getString("NgayLap");  Lop lop = **new** Lop(MaLop,TenLop,NgayLap);  **return** lop;  }  **return** **null**; |
| 9 | TimThanhVien | Trần Văn Nam | Lấy dữ liệu của một thí sinh trong CSDL | String sql = "Select \* from ThanhVien where MathanhVien=?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, MaThanhVien);  ResultSet rs = pstm.executeQuery();  **while** (rs.next()) {  String TenThanhVien = rs.getString("TenThanhVien");  String NgaySinh = rs.getString("NgaySinh");  String GioiTinh =rs.getString("GioiTinh");  String Email =rs.getString("Email");  String SoDienThoai =rs.getString("SoDienThoai");  String DiaChi =rs.getString("DiaChi");  String PassWord =rs.getString("PassWord");  String HinhAnh =rs.getString("HinhAnh");  ThanhVien tv = **new** ThanhVien(MaThanhVien,TenThanhVien,NgaySinh,GioiTinh,Email,SoDienThoai,DiaChi,PassWord,HinhAnh);  **return** tv;  }  **return** **null**; |
| 10 | TimThanhVienLop | Ngô Tuấn Lĩnh | Lấy danh sách thành viên trong CSDL theo lớp học | String sql = "SELECT \* FROM Lop,Lop\_TV,ThanhVien " + "WHERE Lop\_TV.MaThanhVien=ThanhVien.MaThanhVien "  + "AND Lop\_TV.MaLop=Lop.MaLop AND Lop.MaLop=?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, MaLop);  ResultSet rs = pstm.executeQuery();  List<ThanhVien> list = **new** ArrayList<ThanhVien>();  **while** (rs.next()) {  String MaThanhVien = rs.getString("MaThanhVien");  String TenThanhVien = rs.getString("TenThanhVien");  String NgaySinh = rs.getString("NgaySinh");  String GioiTinh =rs.getString("GioiTinh");  String Email =rs.getString("Email");  String SoDienThoai =rs.getString("SoDienThoai");  String DiaChi =rs.getString("DiaChi");  String PassWord =rs.getString("PassWord");  String HinhAnh =rs.getString("HinhAnh");  ThanhVien thanhvien = **new** ThanhVien();  thanhvien.setMaThanhVien(MaThanhVien);  thanhvien.setTenThanhVien(TenThanhVien);  thanhvien.setNgaySinh(NgaySinh);  thanhvien.setGioiTinh(GioiTinh);  thanhvien.setEmail(Email);  thanhvien.setSoDienThoai(SoDienThoai);  thanhvien.setDiaChi(DiaChi);  thanhvien.setPassWord(PassWord);  thanhvien.setHinhAnh(HinhAnh);  list.add(thanhvien);  }  **return** list; |
| 11 | TimThanhVienConlai | Ngô Tuấn Lĩnh | Lấy danh sách thành viên không thuộc lớp trong CSDL | String sql = "SELECT \* FROM ThanhVien WHERE MaThanhVien NOT IN(SELECT MaThanhVien FROM Lop\_TV WHERE MaLop=?)";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, MaLop);  ResultSet rs = pstm.executeQuery();  List<ThanhVien> list = **new** ArrayList<ThanhVien>();  **while** (rs.next()) {  String MaThanhVien = rs.getString("MaThanhVien");  String TenThanhVien = rs.getString("TenThanhVien");  String NgaySinh = rs.getString("NgaySinh");  String GioiTinh =rs.getString("GioiTinh");  String Email =rs.getString("Email");  String SoDienThoai =rs.getString("SoDienThoai");  String DiaChi =rs.getString("DiaChi");  String PassWord =rs.getString("PassWord");  String HinhAnh =rs.getString("HinhAnh");  ThanhVien thanhvien = **new** ThanhVien();  thanhvien.setMaThanhVien(MaThanhVien);  thanhvien.setTenThanhVien(TenThanhVien);  thanhvien.setNgaySinh(NgaySinh);  thanhvien.setGioiTinh(GioiTinh);  thanhvien.setEmail(Email);  thanhvien.setSoDienThoai(SoDienThoai);  thanhvien.setDiaChi(DiaChi);  thanhvien.setPassWord(PassWord);  thanhvien.setHinhAnh(HinhAnh);  list.add(thanhvien);  }  **return** list; |
| 12 | XoaLop | Ngô Tuấn Lĩnh | Xóa một lớp học trong CSDL | String sql = "DELETE FROM Lop WHERE MaLop = ?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, MaLop);  pstm.executeUpdate(); |
| 13 | XoaThanhVien | Trần Văn Nam | Xóa một thành viên trong SDL | String sql = "DELETE FROM ThanhVien WHERE MaThanhVien = ?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, MaThanhVien);  pstm.executeUpdate(); |
| 14 | XoaThanhVienLop | Ngô Tuấn Lĩnh | Xóa một thành viên trong bảng Lop-TV trong CSDL | String sql = "DELETE FROM Lop\_TV WHERE MaThanhVien = ? AND MaLop = ?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, MaThanhVien);  pstm.setString(2, MaLop);  pstm.executeUpdate(); |

## 4.36. Mẫu bảng mô tả các phương thức trong lớp ChiTietDeThiThiSinh

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách | Mục đích chính của phương thức | Code chương trình |
| 1 | DoGet | Lê Cao Liêm | Gọi và lấy lớp ChiTietDeThi trong PT queryChiTietDeThi với MaDe được truyền vào và đổ dữ liệu lên trang chitietdethi.jsp | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaThanhVien = (String) request.getParameter("MaThanhVien");  String MaMonHoc = (String) request.getParameter("MaMonHoc");  String MaDe = (String) request.getParameter("MaDe");  SoLanSubmitchophep = request.getParameter("SoLanSubmit");  **float** DiemTB =0;  **int** SoLanSubmit =0;  List<thisinh.beans.ChiTietDeThi> list = **null**;  list = DBUtils.*queryChiTietDeThi*(conn, MaDe);  DiemTB=DBUtils.*DiemTB*(conn, MaThanhVien, MaDe);  SoLanSubmit=DBUtils.*SoLanSubmit*(conn, MaThanhVien, MaDe);  **if** (list == **null**) { response.sendRedirect(request.getServletPath() + "/DanhSachDeThiThiSinh");  **return**;  }  String ThoiGianBD = **new** SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(Calendar.*getInstance*().getTime());  LocalDateTime datetime = LocalDateTime.*parse*(ThoiGianBD, DateTimeFormatter.*ofPattern*("yyyy-MM-dd HH:mm:ss"));  TGBD.add(datetime);  request.setAttribute("DiemTB",DiemTB);  request.setAttribute("ChiTietDeThiList", list);  request.setAttribute("MaThanhVien", MaThanhVien);  request.setAttribute("MaMonHoc", MaMonHoc);  request.setAttribute("MaDe", MaDe);  **if**(SoLanSubmit<Integer.*parseInt*(SoLanSubmitchophep))  {  request.setAttribute("TGBD", TGBD.get(0));  RequestDispatcher dispatcher = request.getServletContext() .getRequestDispatcher("/ThiSinh/chitietdethi.jsp");  dispatcher.forward(request, response);  }  **else**  {  TGBD.clear();  RequestDispatcher dispatcher = request.getServletContext() .getRequestDispatcher("/ThiSinh/ketquathi.jsp");  dispatcher.forward(request, response);} |
| 2 | DoPost | Lê Cao Liêm | Lấy đáp án của thí sinh, tính điểm cho thí sinh, gọi PT ThemDiem để lưu điểm cho thí sinh, gọi PT DiemTB để xuất điểm TB cho thí sinh. Chuyển đến trang ketquathi.jsp | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaDe = request.getParameter("MaDe");  String MaThanhVien = (String) request.getParameter("MaThanhVien");  String ThoiGianKT = **new** SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(Calendar.*getInstance*().getTime());  List<thisinh.beans.ChiTietDeThi> dc = **null**;  String errorString = **null**;  dc = DBUtils.*queryChiTietDeThi*(conn, MaDe);  List<String> da=**new** ArrayList<String>();  **for** (**int** i=0;i<dc.size();i++)  {  String dapanthisinh = (String)request.getParameter(String.*valueOf*(i+1));    **if**(dapanthisinh==**null**)  {  dapanthisinh="";  }  da.add(dapanthisinh);  }  **float** n=0;  **for** (**int** j=0;j<dc.size();j++)  { **if**(da.get(j).equals(dc.get(j).getDapAn().trim()))  n++;  }  **float** a = (n/dc.size())\*10;  a = (**float**)Math.*round*(a\*10)/10;  DiemThi dt = **new** DiemThi(MaDe,MaThanhVien,ThoiGianKT,a);  **int** SoLanSubmit=0;  **float** DiemTB=0;  SoLanSubmit=DBUtils.*SoLanSubmit*(conn, MaThanhVien, MaDe);  DiemTB=DBUtils.*DiemTB*(conn, MaThanhVien, MaDe);  **if**(SoLanSubmit<Integer.*parseInt*(SoLanSubmitchophep))  {  DBUtils.*ThemDiem*(conn, dt); SoLanSubmit=DBUtils.*SoLanSubmit*(conn, MaThanhVien, MaDe);  DiemTB=DBUtils.*DiemTB*(conn, MaThanhVien, MaDe);  }  request.setAttribute("DiemThi",a);  request.setAttribute("erorstring", errorString);  request.setAttribute("solanSubmit", SoLanSubmit);  request.setAttribute("DiemTB",DiemTB);  request.setAttribute("ChiTietDeThiList", dc);  request.setAttribute("MaThanhVien", MaThanhVien);  request.setAttribute("SoCauHoi", dc.size());  TGBD.clear();  RequestDispatcher dispatcher = request.getServletContext() .getRequestDispatcher("/ThiSinh/ketquathi.jsp");  dispatcher.forward(request, response); |

## 4.37. Mẫu bảng mô tả các phương thức trong lớp DangXuatTS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách | Mục đích chính của phương thức | Code chương trình |
| 1 | DoGet | Lê Cao Liêm | Xóa phiên hoạt động của thí sinh và đưa thí sinh trở lại trang login | HttpSession session = request.getSession();  session.removeAttribute("userTS");  response.sendRedirect(request.getContextPath() + "/Login"); |

## 4.38. Mẫu bảng mô tả các phương thức trong lớp DanhSachDeThiThiSinh

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách | Mục đích chính của phương thức | Code chương trình |
| 1 | DoGet | Lê Cao Liêm | Gọi và lấy lớp DeThi trong PT queryDeThi với MaMonHoc, MaLop, ThoiGianHienTai được truyền vào và đổ dữ liệu lên trang danhsachdethi.jsp | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaThanhVien = (String) request.getParameter("MaThanhVien");  String MaMonHoc = (String) request.getParameter("MaMonHoc");  String TenMonHoc = **null**;  String ThoiGianHienTai = **new** SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(Calendar.*getInstance*().getTime());  List<DeThi> list = **null**;  list = DBUtils.*queryDeThi*(conn, MaThanhVien, MaMonHoc,ThoiGianHienTai);  TenMonHoc =DBUtils.*findTenMonHoc*(conn,MaMonHoc);  **if** (list == **null**) {  response.sendRedirect(request.getServletPath() + "/DanhSachDeThiThiSinh");  **return**;  }  **if** (TenMonHoc == **null**) { response.sendRedirect(request.getServletPath() + "/DanhSachDeThiThiSinh");  **return**;  }  request.setAttribute("MaThanhVien", MaThanhVien);  request.setAttribute("TenMonHoc", TenMonHoc);  request.setAttribute("DeThiList", list);  RequestDispatcher dispatcher = request.getServletContext().getRequestDispatcher("/ThiSinh/danhsachdethi.jsp");  dispatcher.forward(request, response); |

## 4.39. Mẫu bảng mô tả các phương thức trong lớp DanhSachMonHoc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách | Mục đích chính của phương thức | Code chương trình |
| 1 | DoGet | Lê Cao Liêm | Gọi và lấy danh sách môn học trong PT queryMonHoc với và đổ dữ liệu lên trang danhsachmonhoc.jsp | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaThanhVien = (String) request.getParameter("MaThanhVien");  List<MonHoc> mh = **null**;  String errorString = **null**;  mh = DBUtils.*queryMonHoc*(conn);  **if** (mh == **null**) { response.sendRedirect(request.getServletPath() + "/login.jsp");  **return**;  }  request.setAttribute("MonHocList",mh);  request.setAttribute("MaThanhVien", MaThanhVien);  RequestDispatcher dispatcher = request.getServletContext() .getRequestDispatcher("/ThiSinh/danhsachmonhoc.jsp");  dispatcher.forward(request, response); |

## 4.40. Mẫu bảng mô tả các phương thức trong lớp ThongTinThiSinh

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách | Mục đích chính của phương thức | Code chương trình |
| 1 | DoGet | Lê Cao Liêm | Gọi và lấy thông tin thí sinh trong PT queryThanhVien với MaThanhVien được truyền vào và đổ dữ liệu lên trang taikhoan.jsp | Connection conn=**null**;  conn = ConnectionUtils.*getConnection*();  String MaThanhVien = (String) request.getParameter("MaThanhVien");  List<ThanhVien> list = **null**;  list = DBUtils.*queryThanhVien*(conn, MaThanhVien);  **if** (list == **null**) {  response.sendRedirect(request.getServletPath() + "/ThongTinThiSinh");  **return**;  }  request.setAttribute("MaThanhVien", MaThanhVien);  request.setAttribute("TaiKhoanList", list);  RequestDispatcher dispatcher = request.getServletContext().getRequestDispatcher("/ThiSinh/taikhoan.jsp");  dispatcher.forward(request, response); |

## 4.41. Mẫu bảng mô tả các phương thức trong lớp DBUtils (thisinh.utils)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stt | Tên phương thức | Tên SV phụ trách | Mục đích trong chương trình | Code chương trình |
| 1 | queryThanhVien | Lê Cao Liêm | Lấy thông tin thí sinh từ CSDL | String sql = "SELECT TenThanhVien, NgaySinh, GioiTinh, Email, SoDienThoai, DiaChi " + "FROM ThanhVien WHERE MaThanhVien = ?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, MaThanhVien);  ResultSet rs = pstm.executeQuery();  List<ThanhVien> list = **new** ArrayList<ThanhVien>();  **while** (rs.next()) {  String TenThanhVien = rs.getString("TenThanhVien");  String NgaySinh = rs.getString("NgaySinh");  String GioiTinh = rs.getString("GioiTinh");  String Email = rs.getString("Email");  String SoDienThoai = rs.getString("SoDienThoai");  String DiaChi = rs.getString("DiaChi");  ThanhVien thanhvien = **new** ThanhVien(**null**, TenThanhVien, NgaySinh, GioiTinh, Email, SoDienThoai, DiaChi, **null**, **null**); thanhvien.setTenThanhVien(TenThanhVien);  thanhvien.setNgaySinh(NgaySinh);  thanhvien.setGioiTinh(GioiTinh);  thanhvien.setEmail(Email);  thanhvien.setSoDienThoai(SoDienThoai);  thanhvien.setDiaChi(DiaChi);  list.add(thanhvien);  }  **return** list; |
| 2 | DiemTB | Lê Cao Liêm | Lấy điểm trung bình theo đề thi của thí sinh | String sql = "SELECT ROUND( DiemTB,2)AS DTB FROM DiemTrungBinh WHERE MaDe=? AND MaThanhVien=?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, MaDe);  pstm.setString(2, MaThanhVien);  ResultSet rs = pstm.executeQuery();  **while** (rs.next()) {  **float** DiemTB = rs.getFloat("DTB");  **return** DiemTB;  }  **return** 0; |
| 3 | queryChiTietDeThi | Lê Cao Liêm | Lấy thời gian làm bài, danh sách câu hỏi của một đề thi | String sql = "SELECT TenDe,ThoiGianThi, CauHoi.MaCauHoi,NoiDung,CauA,CauB,CauC,CauD,DapAn FROM De\_Cau,CauHoi,DeThi WHERE CauHoi.MaCauHoi=De\_Cau.MaCauHoi AND De\_Cau.MaDe=DeThi.MaDe AND De\_Cau.MaDe =?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, MaDe);  ResultSet rs = pstm.executeQuery();  List<ChiTietDeThi> list = **new** ArrayList<ChiTietDeThi>();  **while** (rs.next()) {  String TenDe = rs.getString("TenDe");  String ThoiGianThi = rs.getString("ThoiGianThi");  String MaCauHoi = rs.getString("MaCauHoi");  String NoiDung = rs.getString("NoiDung");  String CauA = rs.getString("CauA");  String CauB = rs.getString("CauB");  String CauC = rs.getString("CauC");  String CauD = rs.getString("CauD");  String DapAn = rs.getString("DapAn");  ChiTietDeThi chiTietDeThi = **new** ChiTietDeThi(TenDe, ThoiGianThi, MaCauHoi, NoiDung, CauA, CauB, CauC, CauD, DapAn);  list.add(chiTietDeThi);  }  **return** list; |
| 4 | queryDeThi | Lê Cao Liêm | Lấy danh sách các đề thi còn hạn của một thí sinh theo môn học mà thí sinh chọn | String sql = "SELECT DeThi.MaDe,TenDe,TenMonHoc,SoCauHoi,ThoiGianThi,SoLanSubmit,ThoiGianMo,ThoiGianDong " + "FROM De\_Lop,Lop\_TV,MonHoc,DeThi " + "WHERE DeThi.MaDe=De\_Lop.MaDe AND De\_Lop.MaLop=Lop\_TV.MaLop AND DeThi.MonHoc=MonHoc.MaMonHoc "+ "AND MaThanhVien=? AND DeThi.MonHoc=? AND ? >= ThoiGianMo AND ?<=ThoiGianDong";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, MaThanhVien);  pstm.setString(2, MaMonHoc);  pstm.setString(3, ThoiGianHienTai);  pstm.setString(4, ThoiGianHienTai);    ResultSet rs = pstm.executeQuery();  List<DeThi> list = **new** ArrayList<DeThi>();  **while** (rs.next()) {  String MaDe = rs.getString("MaDe");  String TenDe = rs.getString("TenDe");  String TenMonHoc = rs.getString("TenMonHoc");  **int** SoCauHoi = rs.getInt("SoCauHoi");  **int** ThoiGianThi = rs.getInt("ThoiGianThi");  **int** SoLanSubmit = rs.getInt("SoLanSubmit");  String ThoiGianMo = rs.getString("ThoiGianMo");  String ThoiGianDong = rs.getString("ThoiGianDong");  DeThi dethi = **new** DeThi();  dethi.setMaDe(MaDe);  dethi.setTenDe(TenDe);  dethi.setMonHoc(TenMonHoc);  dethi.setSoCauHoi(SoCauHoi);  dethi.setThoiGianThi(ThoiGianThi);  dethi.setSoLanSubmit(SoLanSubmit);  dethi.setThoiGianMo(ThoiGianMo);  dethi.setThoiGianDong(ThoiGianDong);  list.add(dethi);  }  **return** list; |
| 5 | queryMonHoc | Lê Cao Liêm | Lây danh sách môn học | String sql = "SELECT \* FROM MonHoc";  PreparedStatement pstm = conn.prepareStatement(sql);  ResultSet rs = pstm.executeQuery();  List<MonHoc> list = **new** ArrayList<MonHoc>();  **while** (rs.next()) {  String TenMonHoc = rs.getString("TenMonHoc");  String MaMonHoc = rs.getString("MaMonHoc");  MonHoc monhoc = **new** MonHoc(MaMonHoc, TenMonHoc);  monhoc.setTenMonHoc(TenMonHoc);  monhoc.setMaMonHoc(MaMonHoc);  list.add(monhoc);  }  **return** list; |
| 6 | SoLanSubmit | Lê Cao Liêm | Lấy số lần thí sinh đã làm đề thi đó | String sql = "SELECT COUNT(ThoiGianKetThuc) AS SoLanSubmit FROM Thi WHERE MaDe=? AND MaThanhVien=?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, MaDe);  pstm.setString(2, MaThanhVien);  ResultSet rs = pstm.executeQuery();  **while** (rs.next()) {  **int** SoLanSubmit = rs.getInt(1);  **return** SoLanSubmit;  }  **return** 0; |
| 7 | ThemDiem | Lê Cao Liêm | Thêm điểm của thí sinh vào CSDL sau khi thí sinh đã thi xong một lần thi | String sql = "INSERT into Thi (MaDe,MaThanhVien,ThoiGianKetThuc,Diem) VALUES (?,?,?,ROUND(?,2))";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, dt.getMaDe());  pstm.setString(2, dt.getMaThanhVien());  pstm.setString(3, dt.getThoiGianKT());  pstm.setFloat(4, dt.getDiem());  pstm.executeUpdate(); |

***​***