**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH**

**KHOA VIỄN THÔNG II**

**----------------------**

**BÁO CÁO MÔN HỌC**

**LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

NHÓM 5

**Đề tài: Quản Lý Khóa Học Trực Tuyến**

**Giảng viên: Nguyễn Văn Hiền**

**Sinh viên thực hiện:  
 Nguyễn Văn Phương N21DCVT078**

**Nguyễn Minh Quân N21DCVT80**

**Vũ Minh Quang N21DCVT079**

**Lê Hồng Anh N21DCVT001**

**Trần Hà Thanh Tuấn N21DCVT116**

**Năm học: *2024 - 2025***

**Mục lục**

[**Lời mở đầu 6**](#_Toc196092828)

[**I.** **Giới thiệu dự án** 7](#_Toc196092829)

[**1.** **Mục tiêu dự án** 7](#_Toc196092830)

[**2.** **Phần mềm và phương pháp triển khai** 7](#_Toc196092831)

[**3.** **Các class cần xây dựng** 7](#_Toc196092832)

[**3.1** **Package model** 7](#_Toc196092833)

[**3.2** **Package interface\_** 8](#_Toc196092834)

[**3.3** **Package dao** 9](#_Toc196092835)

[**3.4** **Package gui** 9](#_Toc196092836)

[**3.5** **Package hocVienGUI** 11](#_Toc196092837)

[**3.6** **Package giangVienGUI** 11](#_Toc196092838)

[**II.** **Mô hình dự án** 12](#_Toc196092839)

[**1.** **Sơ đồ lớp** 12](#_Toc196092840)

[**2.** **Sơ đồ Database** 13](#_Toc196092841)

[**3.** **Sơ đồ khối** 13](#_Toc196092842)

[**3.1** **LoginGUI** 13](#_Toc196092843)

[**3.2** **TaiKhoanGUI** 14](#_Toc196092844)

[**3.3** **AdminGUI** 14](#_Toc196092845)

[**3.4** **GiangVienGUI** 14](#_Toc196092846)

[3.5 HocVienGUI 21](#_Toc196092847)

[**III.** **Code dự án** 23](#_Toc196092848)

[**1.** **Package model** 23](#_Toc196092849)

[**1.1** **TaiKhoan** 23](#_Toc196092850)

[**1.2** **User** 23](#_Toc196092851)

[**1.3** **KhoaHoc** 24](#_Toc196092852)

[**1.4** **HocVien** 25](#_Toc196092853)

[**1.5** **GiangVien** 26](#_Toc196092854)

[**1.6** **DiemSo** 26](#_Toc196092855)

[**1.7** **DiemDanh** 27](#_Toc196092856)

[**2.** **Package interface\_** 29](#_Toc196092857)

[**2.1** **ITaiKhoanDAO** 29](#_Toc196092858)

[**2.2** **IKhoaHocDAO** 29](#_Toc196092859)

[**2.3** **IHocVienDAO** 29](#_Toc196092860)

[**2.4** **IGiangVienDAO** 30](#_Toc196092861)

[**2.5** **IDiemSoDAO** 31](#_Toc196092862)

[**2.6** **IDiemDanhDAO** 31](#_Toc196092863)

[**2.7** **IDangKyDAO** 31](#_Toc196092864)

[**3.** **Package dao** 32](#_Toc196092865)

[**3.1** **TaiKhoanDAO** 32](#_Toc196092866)

[**3.2** **KhoaHocDAO** 33](#_Toc196092867)

[**3.3** **HocVienDAO** 37](#_Toc196092868)

[**3.4** **GiangVienDAO** 41](#_Toc196092869)

[**3.5** **DiemSoDAO** 46](#_Toc196092870)

[**3.6** **DiemDanhDAO** 49](#_Toc196092871)

[**3.7** **DBConnection** 52](#_Toc196092872)

[**3.8** **DBCheckExist** 53](#_Toc196092873)

[**3.9** **IDangKyDAO** 54](#_Toc196092874)

[**4.** **Package gui** 56](#_Toc196092875)

[**4.1** **AdminGUI** 56](#_Toc196092876)

[**4.2** **GiangVienGUI** 58](#_Toc196092877)

[**4.3** **HocVienGUI** 69](#_Toc196092878)

[**4.4** **KhoaHocGUI** 75](#_Toc196092879)

[**4.5** **LoginGUI** 81](#_Toc196092880)

[**4.6** **SuaHocVienGUI** 83](#_Toc196092881)

[**4.7** **SuaKhoaHocGUI** 85](#_Toc196092882)

[**4.8** **TaiKhoanGUI** 87](#_Toc196092883)

[4.9 ThemKhoaHocGUI 92](#_Toc196092884)

[**5.** **Package hocVienGUI** 94](#_Toc196092885)

[**5.1 LoginHocVienGUI** 94](#_Toc196092886)

[**6.** **Package giangVienGUI** 101](#_Toc196092887)

[**6.1 LoginGiangVienGUI** 101](#_Toc196092888)

[**IV.** **Kết quả** 111](#_Toc196092889)

[**1.** **Đăng nhập bằng tài khoản Admin** 111](#_Toc196092890)

[**2.** **Đăng nhập bằng tài khoản Học viên** 131](#_Toc196092891)

[**3.** **Đăng nhập bằng tài khoản Giảng viên** 133](#_Toc196092892)

**DANH MỤC HÌNH ẢNH**

[Hình 1: Sơ đồ lớp 12](#_Toc196092893)

[Hình 2: Sơ đồ Database 13](#_Toc196092894)

[Hình 3: Sơ đồ khối LoginGUI 13](#_Toc196092895)

[Hình 4: Sơ đồ khối TaiKhoanGUI 14](#_Toc196092896)

[Hình 5: Sơ đồ khối AdminGUI 14](#_Toc196092897)

[Hình 6: Sơ đô khối GiangVienGUI 15](#_Toc196092898)

[Hình 7: Sơ đồ khối nút Danh sách khóa học giảng viên phụ trách 15](#_Toc196092899)

[Hình 8: Sơ đồ khối nút Danh sách học viên của giảng viên phụ trách 16](#_Toc196092900)

[Hình 9: Sơ đồ khối nút Xóa giảng viên 17](#_Toc196092901)

[Hình 10: Sơ đồ khối nút Tìm kiếm 17](#_Toc196092902)

[Hình 11: Sơ đồ khối nút Danh sách điểm danh 18](#_Toc196092903)

[Hình 12: Danh sách nút Điểm danh 19](#_Toc196092904)

[Hình 13: Danh sách nút Sửa điểm 20](#_Toc196092905)

[Hình 14: Sơ đồ khối nút Hiển thị danh sách 21](#_Toc196092906)

[Hình 15: Sơ đồ khối nút Xóa học viên 21](#_Toc196092907)

[Hình 16: Sơ đồ khối nút Tìm kiếm học viên 22](#_Toc196092908)

[Hình 17: Sơ đồ khối nút Chỉnh sửa thông tin học viên 22](#_Toc196092909)

[Hình 18: Đăng nhập bằng tài khoản admin 111](#_Toc196092910)

[Hình 19: Giao diện quản lý khóa học trực tuyến khi login bằng tài khoản admin 111](#_Toc196092911)

[Hình 20: Tạo các tài khoản học viên 112](#_Toc196092912)

[Hình 21: Tạo các tài khoản giảng viên 112](#_Toc196092913)

[Hình 22: Tạo các khóa học 113](#_Toc196092914)

[Hình 23: Nhập mã học viên cần đăng ký khóa học 113](#_Toc196092915)

[Hình 24: Nhập khóa học cần đăng ký 114](#_Toc196092916)

[Hình 25: Hiển thị danh sách khóa học 114](#_Toc196092917)

[Hình 26: Nhập mã khóa học cần tìm 115](#_Toc196092918)

[Hình 27: Kết quả tìm kiếm khóa học 115](#_Toc196092919)

[Hình 28: Chọn khóa học muốn xóa 116](file:///D:\Tài%20liệu%20học%20tập\Fourth%20year\Semester%202\Lập%20trình%20hướng%20đối%20tượng\project%20OOP.docx#_Toc196092920)

[Hình 29: Kết quả xóa khóa học 116](#_Toc196092921)

[Hình 30: Sửa khóa học 117](#_Toc196092922)

[Hình 31: Kết quả sửa khóa học 117](#_Toc196092923)

[Hình 32: Hiển thị danh sách học viên 118](#_Toc196092924)

[Hình 33: Nhập mã học viên cần tìm 118](#_Toc196092925)

[Hình 34: Kết quả tìm kiếm học viên 119](#_Toc196092926)

[Hình 35: Sửa học viên 119](#_Toc196092927)

[Hình 36: Kết quả sửa học viên 120](#_Toc196092928)

[Hình 37: Xóa học viên 120](#_Toc196092929)

[Hình 38: Kết quả xóa học viên 121](#_Toc196092930)

[Hình 39: Hiển thị danh sách giảng viên 121](#_Toc196092931)

[Hình 40: Nhâp mã giảng viên cần tìm kiếm 122](#_Toc196092932)

[Hình 41: Kết quả tìm kiếm giảng viên 122](#_Toc196092933)

[Hình 42: Nhập mã giảng viên 123](#_Toc196092934)

[Hình 43: Kết quả danh sách khóa học của giảng viên phụ trách 123](#_Toc196092935)

[Hình 44:Nhập mã giảng viên 124](#_Toc196092936)

[Hình 45: Nhập mã khóa học 124](#_Toc196092937)

[Hình 46: Kết quả danh sách học viên theo khóa học của giảng viên: 125](#_Toc196092938)

[Hình 47: Nhập mã giảng viên 125](#_Toc196092939)

[Hình 48: Nhập mã khóa học để mở danh sách sđiểm danh 126](#_Toc196092940)

[Hình 49: Kết quả danh sách điểm danh 126](#_Toc196092941)

[Hình 50: Nhập mã giảng viên 127](#_Toc196092942)

[Hình 51:Nhập mã khóa học của giảng viên 127](#_Toc196092943)

[Hình 52: Nhấn đúp váo sinh viên để điểm danh 128](#_Toc196092944)

[Hình 53: Nhập mã giảng viên 128](#_Toc196092945)

[Hình 54: Nhập mã khóa học 129](#_Toc196092946)

[Hình 55: Chọn học viên và nhập điểm 129](#_Toc196092947)

[Hình 56: Kết quả nhập điểm của học viên 130](#_Toc196092948)

[Hình 57: Chọn giảng viên cần xóa 130](#_Toc196092949)

[Hình 58: Kết quả xóa giảng viên 131](#_Toc196092950)

[Hình 59: Đăng nhập bằng tài khoản học viên 131](#_Toc196092951)

[Hình 60: Hiển thị thông tin cá nhân học viên 132](#_Toc196092952)

[Hình 61: Hiển thị điểm số của học viên 132](#_Toc196092953)

[Hình 62: Đăng ký khóa học cho học viên nếu còn khóa học 133](#_Toc196092954)

[Hình 63: Đăng nhập bằng tài khoản giảng viên 133](#_Toc196092955)

[Hình 64: Hiển thị thông tin cá nhân của giảng viên 134](#_Toc196092956)

[Hình 65: Hiển thị danh sách khóa học phụ trách của giảng viên 134](#_Toc196092957)

[Hình 66: Nhập mã khóa học cần hiển thị danh sách học viên 135](#_Toc196092958)

[Hình 67: Hiển thị kết quả danh sách học viên của khóa học 135](#_Toc196092959)

[Hình 68: Nhập mã khóa học cần tìm danh sách điểm danh 136](#_Toc196092960)

[Hình 69: Kết quả danh sách điểm danh của khóa học 136](#_Toc196092961)

[Hình 70: Nhập mã khóa học cần thực hiện điểm danh 137](#_Toc196092962)

[Hình 71: Nhấn đúp vào học viên để điểm danh 137](#_Toc196092963)

[Hình 72: Nhập khóa học cần điền điểm 138](#_Toc196092964)

[Hình 73: Nhấn đúp vào học viên và điển điểm cho học viên 138](#_Toc196092965)

# Lời mở đầu

Báo cáo này trình bày quá trình thiết kế và phát triển một hệ thống quản lý khóa học trực tuyến, tích hợp các chức năng như quản lý thông tin khóa học, đăng ký học viên, điểm danh, chấm điểm, tìm kiếm, chỉnh sửa, xóa dữ liệu, cũng như thống kê báo cáo. Hệ thống được xây dựng dựa trên các nguyên tắc lập trình hướng đối tượng bằng ngôn ngữ java, sử dụng kế thừa, interface, xử lý ngoại lệ, đọc/ghi file và giao diện người dùng (GUI) để đảm bảo tính trực quan và dễ sử dụng.

Mục tiêu của dự án là tạo ra một phần mềm đáp ứng tốt các yêu cầu thực tế, hỗ trợ hiệu quả cho công tác quản lý khóa học trực tuyến. Báo cáo sẽ trình bày chi tiết các bước thực hiện, từ phân tích yêu cầu, thiết kế hệ thống, đến triển khai và kiểm thử, nhằm mang lại cái nhìn toàn diện về quá trình phát triển phần mềm. Với sự kết hợp của công nghệ hiện đại và giao diện thân thiện, dự án hướng đến việc tối ưu hóa quy trình quản lý khóa học trực tuyến, đảm bảo tính chính xác, minh bạch và tiện lợi.

# **Giới thiệu dự án**

## **Mục tiêu dự án**

Hỗ trợ quản lý thông tin khóa học (mã, tên, giảng viên, số tín chỉ, học phí) và học viên (đăng ký, điểm danh, điểm số) một cách khoa học.

Cung cấp các chức năng tìm kiếm, chỉnh sửa, xóa thông tin khóa học và học viên, đáp ứng nhu cầu cập nhật dữ liệu nhanh chóng.

Thống kê danh sách học viên, giảng viên và khóa học, giúp người quản lý dễ dàng nắm bắt tình hình hoạt động.

Áp dụng các nguyên tắc lập trình hướng đối tượng (kế thừa, interface), xử lý ngoại lệ, tích hợp cơ sở dữ liệu (đọc/ghi DB) và xây dựng giao diện người dùng (GUI) trực quan.

* Mong muốn: Dự án chia làm 3 role gồm: Admin, Giảng viên, Học viên.
* Admin: có các quyền bao gồm: Tìm kiếm, sửa, xóa khóa học và học viên. Thống kê danh sách học viên, giảng viên khóa học.
* Giảng viên: có quyền tìm kiếm, xem thông tin khóa học và học viên đang dạy, điểm danh và sửa điểm số của các học sinh trong từng khóa học.
* Học sinh: chỉ xem được thông tin cá nhân, điểm số và đăng ký khóa học.

## **Phần mềm và phương pháp triển khai**

Ngôn ngữ lập trình: Sử dụng các ngôn ngữ lập trình Java

Cơ sở dữ liệu: Tích hợp hệ quản trị cơ sở dữ liệu oracle để lưu trữ và truy xuất thông tin.

Giao diện người dùng: Giao diện người dùng (GUI) đang được xây dựng trong Eclipse bằng công cụ WindowBuilder.

## **Các class cần xây dựng**

### **Package model**

|  |  |
| --- | --- |
| model | Mục đích |
| TaiKhoan | Dùng để lưu trữ và quản lý thông tin tài khoản người dùng, bao gồm tên đăng nhập, mật khẩu, vai trò và mã tài khoản. |
| User | Lớp User kế thừa từ lớp TaiKhoan , bổ sung thông tin số điện thoại và tên người dùng, dùng để quản lý chi tiết hơn về tài khoản người dùng cụ thể. |
| KhoaHoc | Dùng để lưu trữ và quản lý thông tin về khóa học, bao gồm mã khóa học, tên khóa học, mã giảng viên, số tín chỉ và học phí. |
| HocVien | Lớp HocVien kế thừa từ lớp User, bổ sung mã học viên, dùng để quản lý thông tin chi tiết về học viên trong hệ thống. |
| GiangVien | Lớp GiangVien kế thừa từ lớp User, bổ sung mã giảng viên và tiền lương, dùng để quản lý thông tin chi tiết về giảng viên trong hệ thống. |
| DiemSo | Dùng để lưu trữ và quản lý thông tin điểm số của học viên, bao gồm mã học viên, tên học viên, mã khóa học và điểm số. |
| DiemDanh | Dùng để lưu trữ và quản lý thông tin điểm danh của học viên, bao gồm mã khóa học, mã học viên, tên học viên, số điện thoại, ngày điểm danh và trạng thái có mặt. |

### **Package interface\_**

| interface\_ | Mục đích |
| --- | --- |
| ITaiKhoanDAO | Xử lý đăng nhập và tạo tài khoản (TaiKhoan). |
| IKhoaHocDAO | Quản lý khóa học: tạo, lấy danh sách, tìm kiếm, xóa, cập nhật, lấy khóa học theo tài khoản hoặc chưa đăng ký. |
| IHocVienDAO | Quản lý học viên: tạo, lấy danh sách, tìm kiếm, xóa, cập nhật, lấy thông tin học viên theo tài khoản. |
| IGiangVienDAO | Quản lý giảng viên: tạo, xóa, tìm kiếm, lấy thông tin giảng viên, danh sách khóa học và học viên theo giảng viên. |
| IDiemSoDAO | Quản lý điểm số: thêm điểm, lấy danh sách điểm theo giảng viên hoặc tài khoản. |
| IDiemDanhDAO | Quản lý điểm danh: ghi nhận điểm danh, lấy danh sách điểm danh theo khóa học và giảng viên. |
| IDangKyDAO | Quản lý đăng ký khóa học: kiểm tra và thực hiện đăng ký. |

### **Package dao**

|  |  |
| --- | --- |
| dao | Mục đích |
| DBConnection | Cung cấp phương thức để thiết lập và trả về kết nối đến cơ sở dữ liệu Oracle sử dụng JDBC. |
| DBCheckExist | Cung cấp phương thức kiểm tra sự tồn tại của một giá trị trong cột cụ thể của bảng trong cơ sở dữ liệu. |
| TaiKhoanDAO | Lớp TaiKhoanDAO triển khai giao diện ITaiKhoanDAO. Thực hiện các truy vấn với bảng TAIKHOAN trong Database |
| KhoaHocDAO | Lớp KhoaHocDAO triển khai giao diện IKhoaHocDAO. Thực hiện các truy vấn với bảng KHOAHOC trong Database |
| HocVienDAO | Lớp HocVienDAO triển khai giao diện IHocVienDAO Thực hiện các truy vấn với bảng HOCVIEN trong Database |
| GiangVienDAO | Lớp GiangVienDAO triển khai giao diện IGiangVienDAO. Thực hiện các truy vấn với bảng GIANGVIEN trong Database |
| DiemSoDAO | Lớp DiemSoDAO triển khai giao diện IDiemSoDAO. Thực hiện các truy vấn với bảng DIEMSO trong Database |
| DiemDanhDAO | Lớp DiemDanhDAO triển khai giao diện IDiemDanhDAO. Thực hiện các truy vấn với bảng TAIKHOAN trong Database |
| DangKyDAO | Lớp DangKyDAO triển khai giao diện IDangKyDAO. Thực hiện các truy vấn với bảng DANGKY trong Database |

### **Package gui**

|  |  |
| --- | --- |
| gui | Mục đích |
| AdminGUI | Lớp AdminGUI tạo giao diện người dùng cho quản trị viên, hiển thị các nút để chuyển hướng đến các giao diện quản lý khóa học, tài khoản, học viên, giảng viên, và đăng xuất để quay lại màn hình đăng nhập. |
| GiangVienGUI | Lớp GiangVienGUI tạo giao diện người dùng cho giảng viên, hiển thị bảng dữ liệu và các nút chức năng: xem danh sách giảng viên, danh sách khóa học của giảng viên, danh sách học viên theo khóa học, điểm danh, sửa điểm học viên, tìm kiếm giảng viên theo mã, xóa giảng viên, và quay lại giao diện quản trị viên. |
| HocVienGUI | Lớp HocVienGUI tạo giao diện người dùng để quản lý học viên, hiển thị bảng danh sách học viên và các nút chức năng: hiển thị danh sách học viên, tìm kiếm học viên theo mã, xóa học viên, sửa thông tin học viên (mở giao diện SuaHocVienGUI), và quay lại giao diện quản trị viên. |
| KhoaHocGUI | Lớp KhoaHocGUI tạo giao diện người dùng để quản lý khóa học, hiển thị bảng danh sách khóa học và các nút chức năng: hiển thị danh sách khóa học, tìm kiếm khóa học theo mã, xóa khóa học, thêm khóa học (mở giao diện ThemKhoaHocGUI), sửa thông tin khóa học (mở giao diện SuaKhoaHocGUI), đăng ký khóa học cho học viên, và quay lại giao diện quản trị viên. |
| LoginGUI | Lớp LoginGUI tạo giao diện đăng nhập cho hệ thống, với nút "Đăng nhập". Khi người dùng nhấn nút, hệ thống kiểm tra thông tin đăng nhập qua TaiKhoanDAO, sau đó chuyển hướng đến giao diện tương ứng (AdminGUI, LoginGiangVienGUI, hoặc LoginHocVienGUI) dựa trên vai trò của tài khoản (admin, giảng viên, hoặc học viên), hoặc hiển thị thông báo lỗi nếu đăng nhập thất bại. |
| SuaHocVienGUI | Lớp SuaHocVienGUI tạo giao diện chỉnh sửa thông tin học viên, hiển thị các trường mã học viên (không chỉnh sửa được), tên học viên, số điện thoại, và hai nút "Cập nhật" và "Hủy". Khi nhấn "Cập nhật", thông tin mới được lưu vào cơ sở dữ liệu qua HocVienDAO. |
| SuaKhoaHocGUI | Lớp SuaKhoaHocGUI tạo giao diện chỉnh sửa thông tin khóa học, hiển thị các trường mã khóa học (không chỉnh sửa được), tên khóa học, mã giảng viên, số tín chỉ, học phí, và hai nút "Cập nhật" và "Hủy". Khi nhấn "Cập nhật", thông tin mới được lưu vào cơ sở dữ liệu qua KhoaHocDAO. |
| TaiKhoanGUI | Lớp TaiKhoanGUI tạo giao diện để quản trị viên thêm tài khoản mới (học viên hoặc giảng viên). Sau khi nhập, nút "Thêm tài khoản" sẽ tạo tài khoản qua TaiKhoanDAO, sau đó tạo học viên hoặc giảng viên tương ứng qua HocVienDAO hoặc GiangVienDAO. |
| ThemKhoaHocGUI | Lớp ThemKhoaHocGUI tạo giao diện để thêm khóa họmới và hai nút "Thêm khóa học" và "Hủy". Khi nhấn "Thêm khóa học", thông tin được lưu vào cơ sở dữ liệu qua KhoaHocDAO. |

### **Package hocVienGUI**

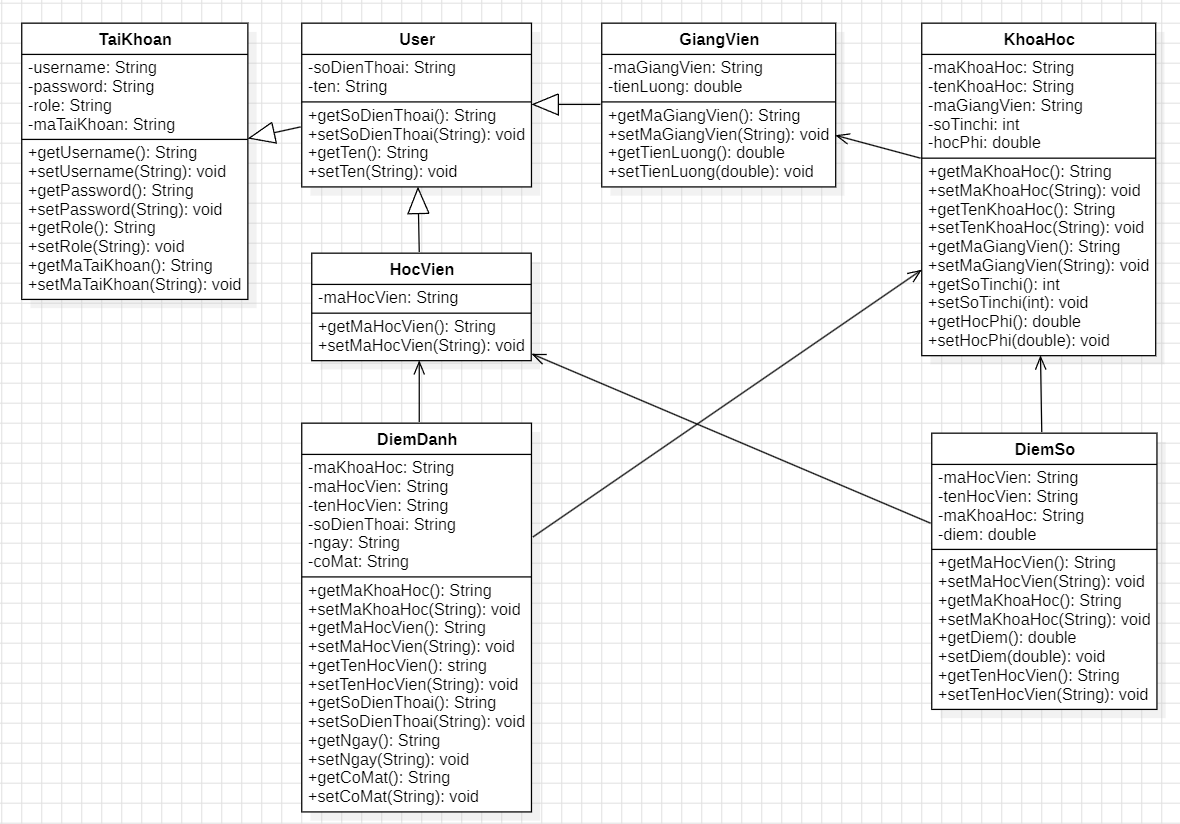
|  |  |
| --- | --- |
| hocVienGUI | Mục đích |
| LoginHocVienGUI | Lớp LoginHocVienGUI tạo giao diện cho học viên sau khi đăng nhập, hiển thị bảng dữ liệu và các nút: xem thông tin cá nhân, xem điểm số, đăng ký khóa học mới, và đăng xuất. Giao diện lấy dữ liệu thông qua các DAO như HocVienDAO, DiemSoDAO, KhoaHocDAO, và DangKyDAO để quản lý thông tin học viên và khóa học. |

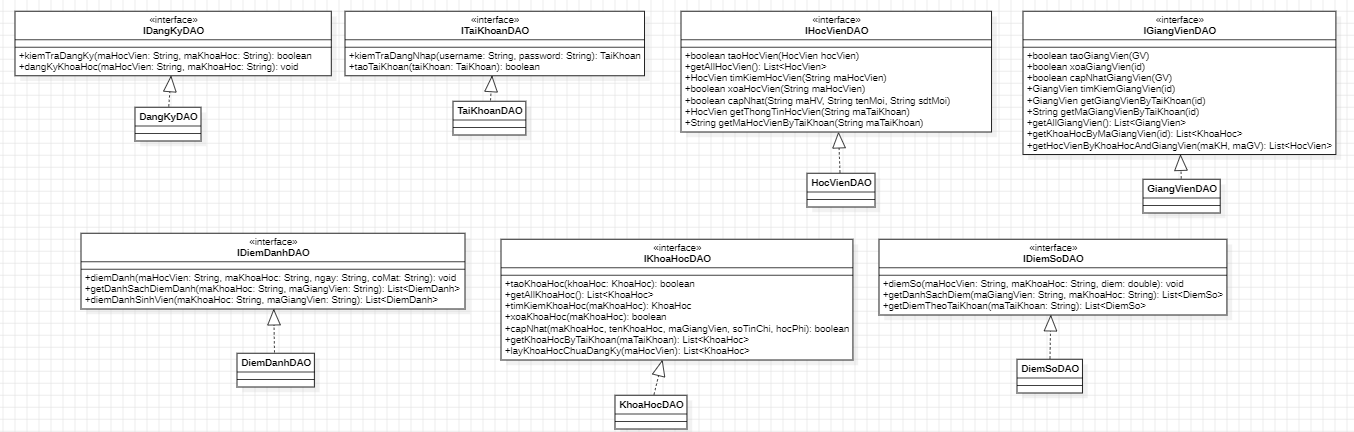
### **Package giangVienGUI**

|  |  |
| --- | --- |
| giangVienGUI | Mục đích |
| LoginGiangVienGUI | Lớp LoginGiangVienGUI tạo giao diện cho giảng viên sau khi đăng nhập, hiển thị bảng dữ liệu và các nút: xem thông tin cá nhân, danh sách khóa học phụ trách, danh sách học viên theo khóa học, điểm danh, sửa điểm học viên, và đăng xuất. Sử dụng các DAO như GiangVienDAO, KhoaHocDAO, DiemDanhDAO, và DiemSoDAO để quản lý dữ liệu. |

# **Mô hình dự án**

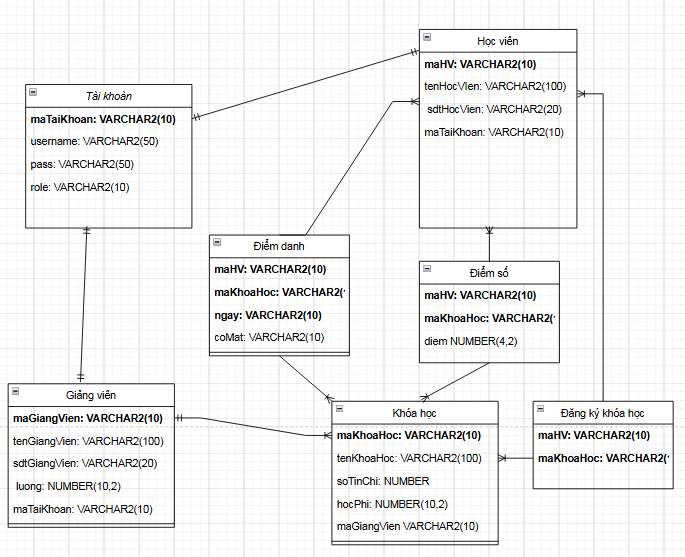
## **Sơ đồ lớp**





Hình 1: Sơ đồ lớp

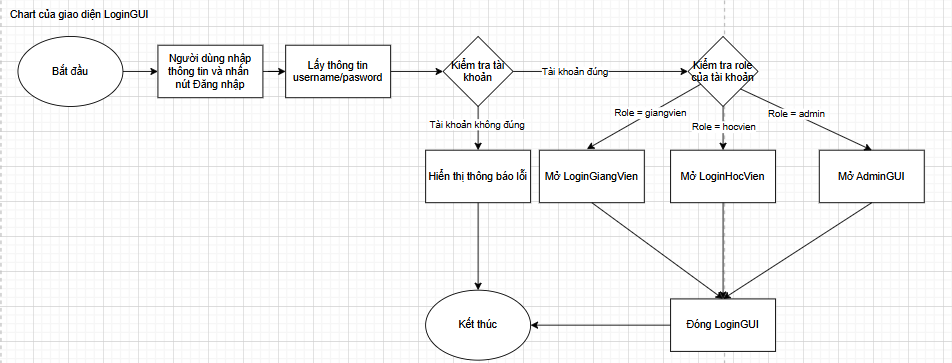
## **Sơ đồ Database**



Hình 2: Sơ đồ Database

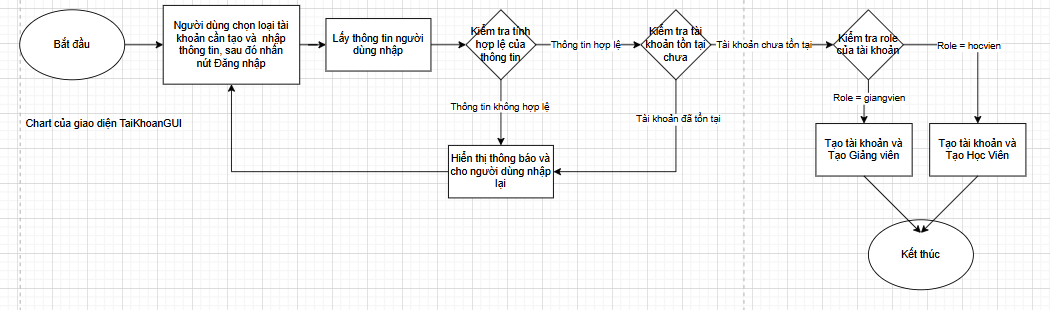
## **Sơ đồ khối**

### **LoginGUI**



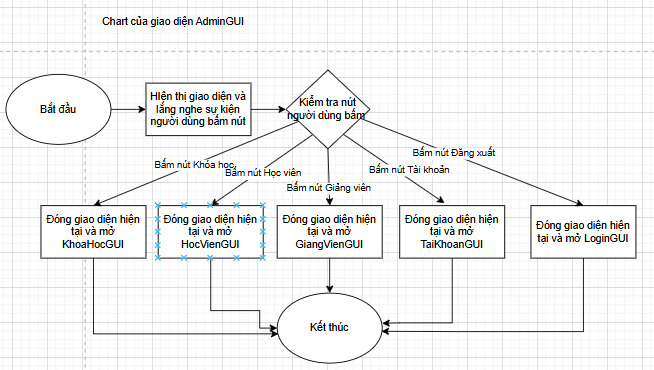
Hình 3: Sơ đồ khối LoginGUI

### **TaiKhoanGUI**



Hình 4: Sơ đồ khối TaiKhoanGUI

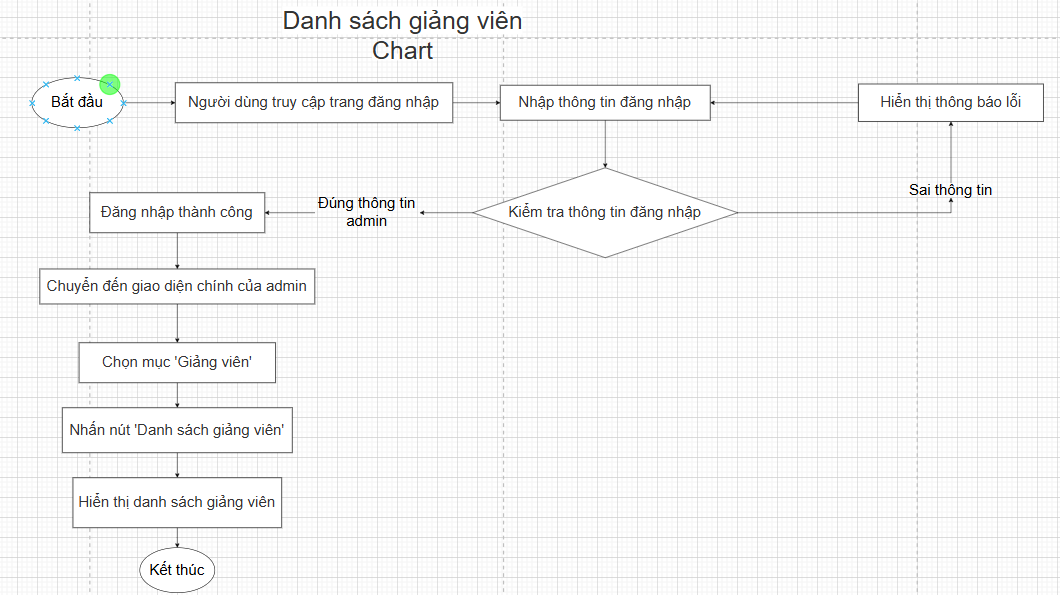
### **AdminGUI**



Hình 5: Sơ đồ khối AdminGUI

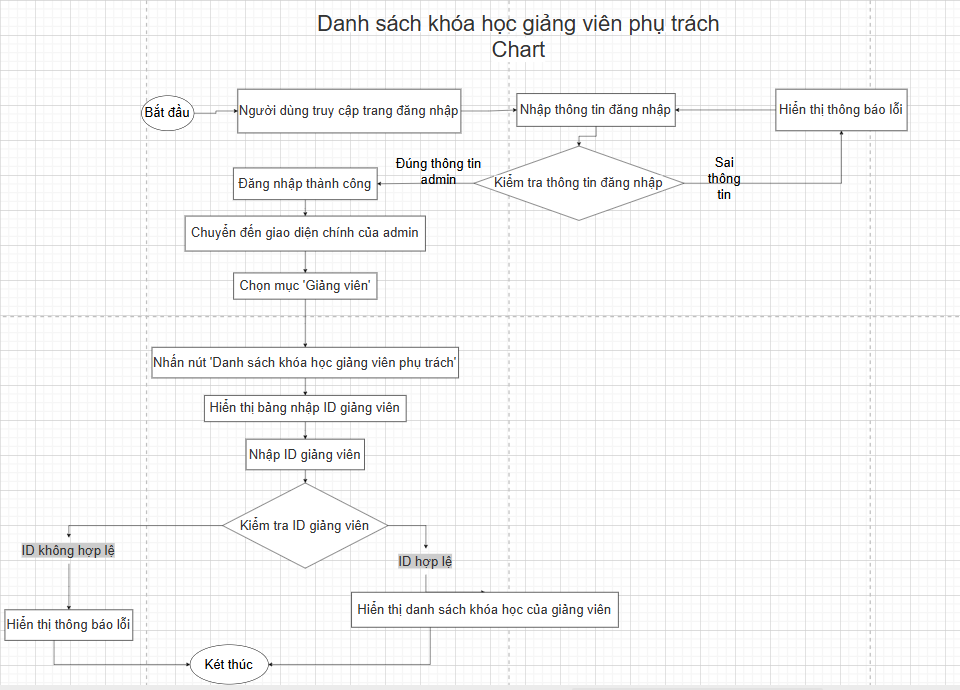
### **GiangVienGUI**

* Danh sách giảng viên



Hình 6: Sơ đô khối GiangVienGUI

* Danh sách khóa học giảng viên phụ trách



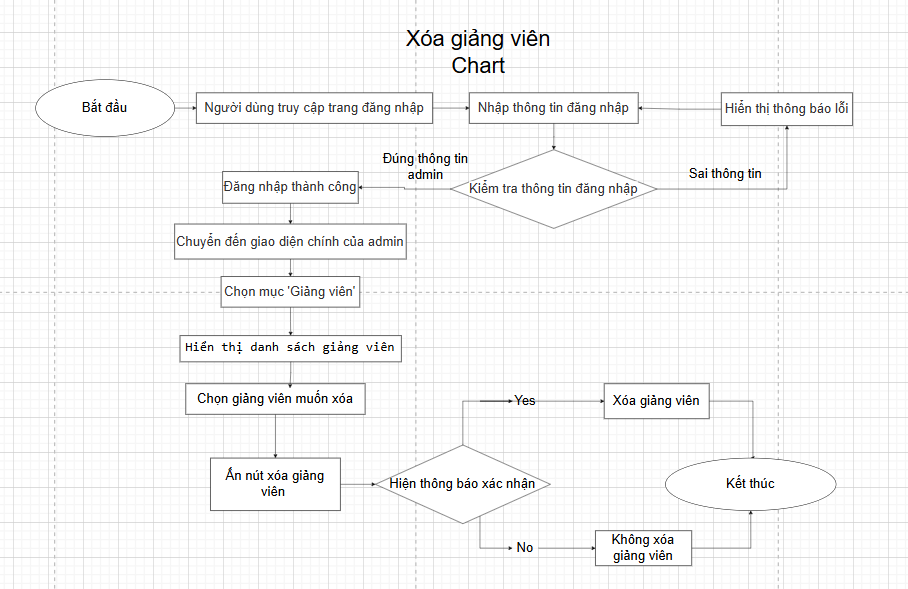
Hình 7: Sơ đồ khối nút Danh sách khóa học giảng viên phụ trách

* Danh sách học viên của giảng viên phụ trách



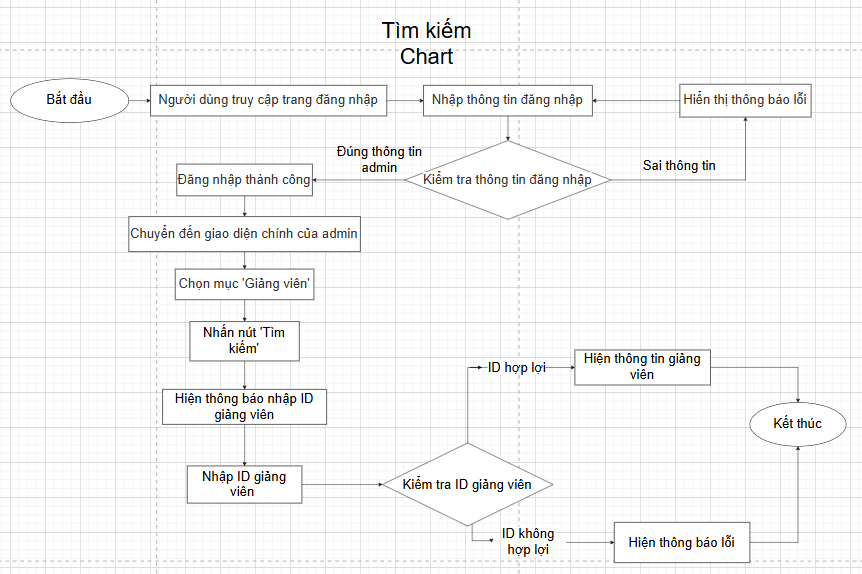
Hình 8: Sơ đồ khối nút Danh sách học viên của giảng viên phụ trách

* Xóa giảng viên



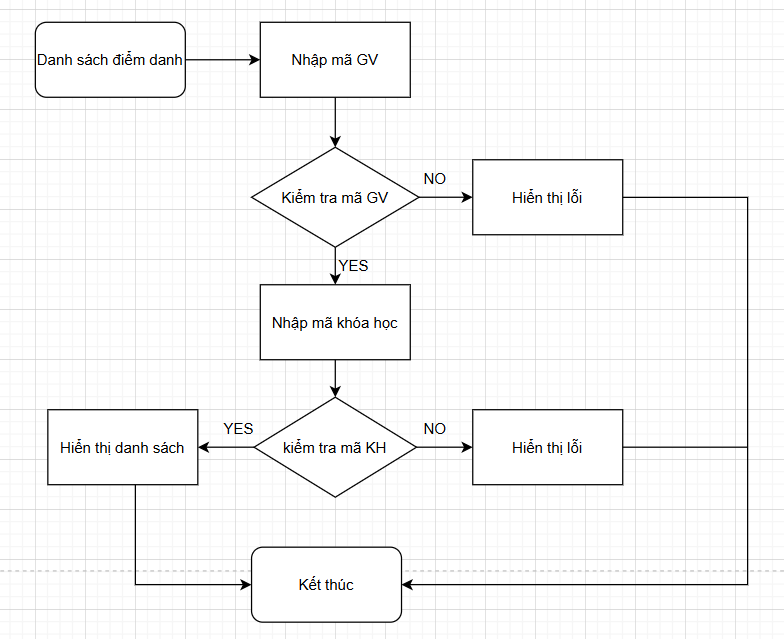
Hình 9: Sơ đồ khối nút Xóa giảng viên

* Tìm kiếm giảng viên



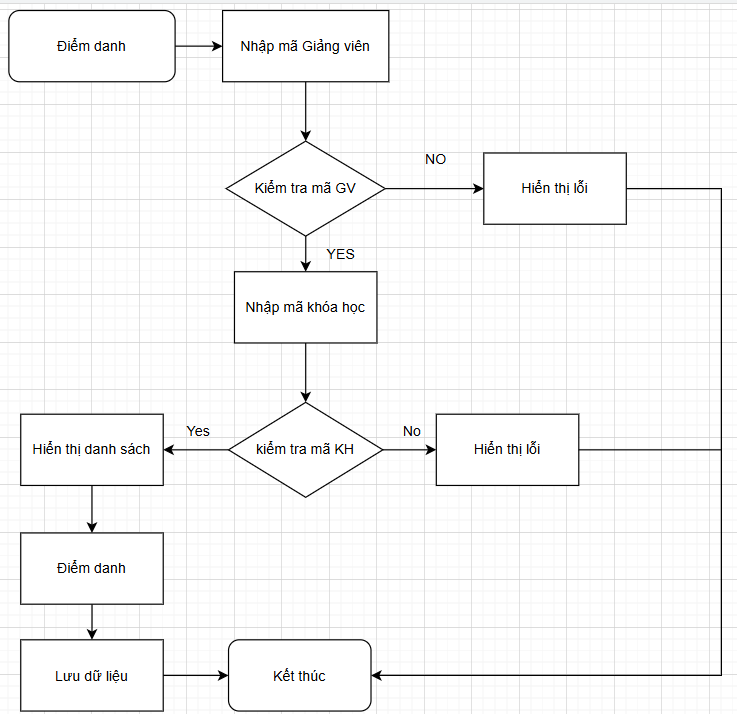
Hình 10: Sơ đồ khối nút Tìm kiếm

* Danh sách điểm danh



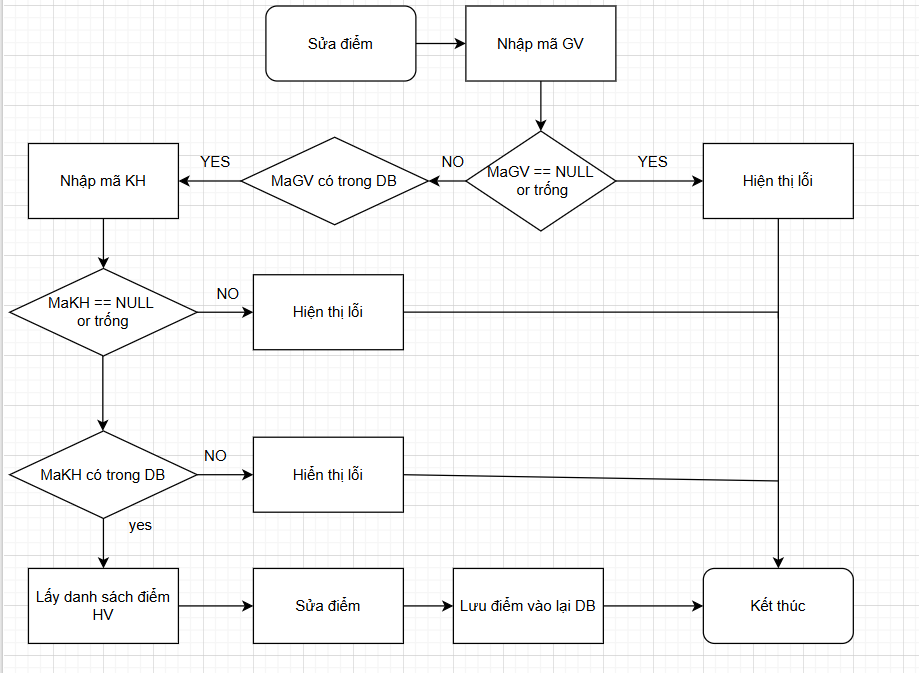
Hình 11: Sơ đồ khối nút Danh sách điểm danh

* Điểm danh



Hình 12: Danh sách nút Điểm danh

* Sửa điểm

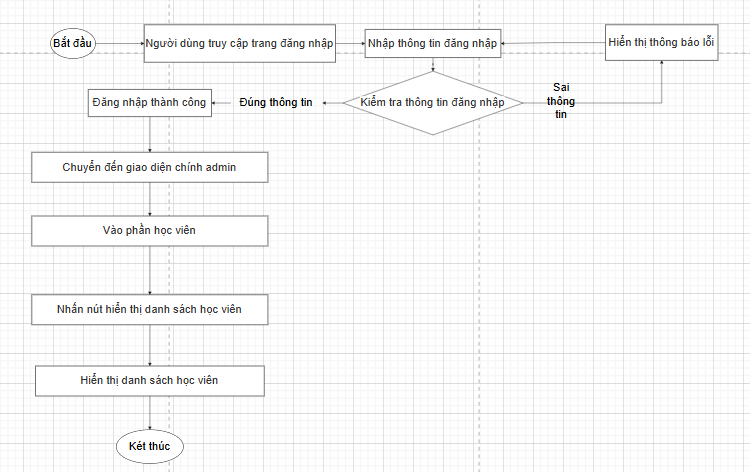


Hình 13: Danh sách nút Sửa điểm

### 

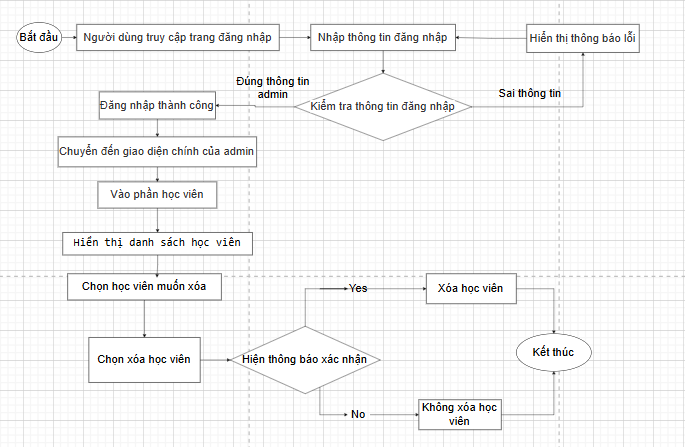
### 3.5 HocVienGUI

* Hiển thị danh sách



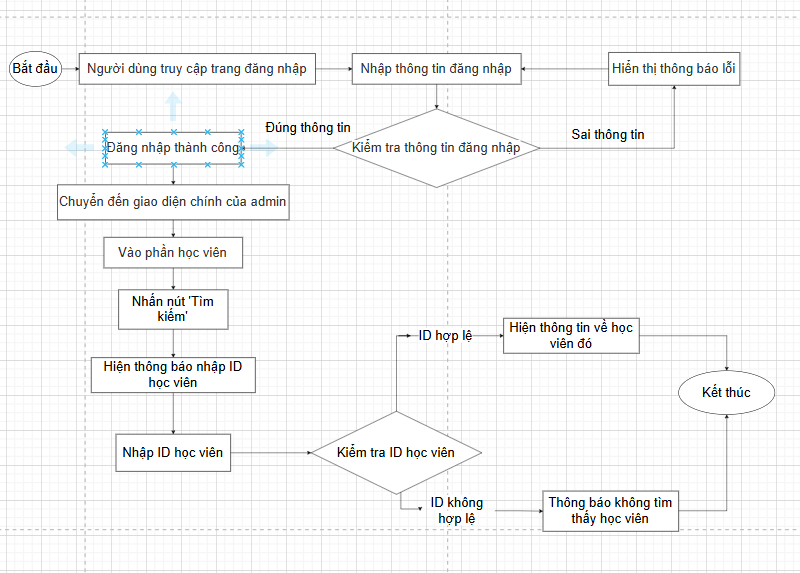
Hình 14: Sơ đồ khối nút Hiển thị danh sách

* Xóa học viên



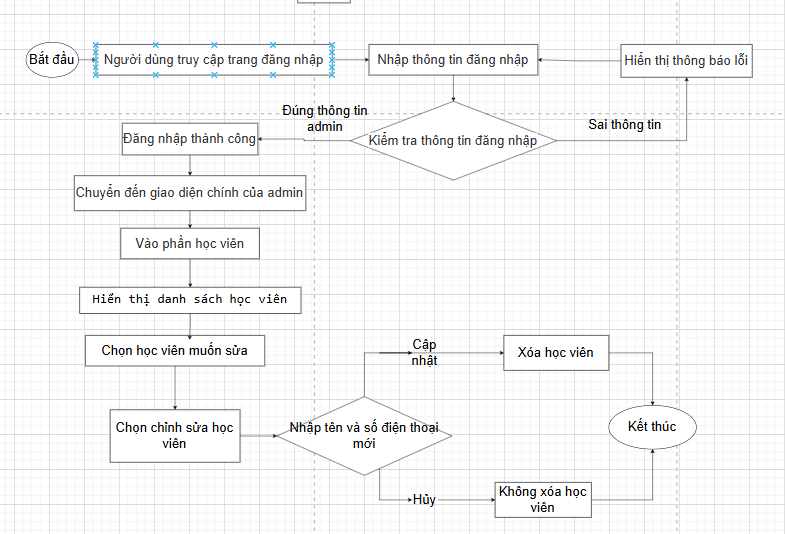
Hình 15: Sơ đồ khối nút Xóa học viên

* Tìm học viên theo mã



Hình 16: Sơ đồ khối nút Tìm kiếm học viên

* Chỉnh sửa thông tin học viên



Hình 17: Sơ đồ khối nút Chỉnh sửa thông tin học viên

# **Code dự án**

## **Package model**

### **TaiKhoan**

|  |
| --- |
| package model;  public class TaiKhoan {  private String username;  private String password;  private String role;  private String maTaiKhoan;    public TaiKhoan() {}    public TaiKhoan(String id) {this.maTaiKhoan = maTaiKhoan;}  public TaiKhoan(String username, String password, String role, String maTaiKhoan) {  this.username = username;  this.password = password;  this.role = role;  this.maTaiKhoan = maTaiKhoan;  }  public String getUsername() { return username; }  public void setUsername(String username) { this.username = username; }  public String getPassword() { return password; }  public void setPassword(String password) { this.password = password; }  public String getRole() { return role; }  public void setRole(String role) { this.role = role; }    public String getMaTaiKhoan() { return maTaiKhoan; }  public void setId(String id) { this.maTaiKhoan = maTaiKhoan; }  } |

### **User**

|  |
| --- |
| package model;  public class User extends TaiKhoan {  private String soDienThoai;  private String ten;  public User() {}  public User(String username, String password, String role, String maTaiKhoan, String soDienThoai, String ten) {  super(username, password, role, maTaiKhoan);  this.soDienThoai = soDienThoai;  this.ten = ten;    }  public String getSoDienThoai() {  return soDienThoai;  }  public void setSoDienThoai(String soDienThoai) {  this.soDienThoai = soDienThoai;  }  public String getTen() {  return ten;  }  public void setTen(String ten) {  this.ten = ten;  }  } |

### **KhoaHoc**

|  |
| --- |
| package model;  public class KhoaHoc {  private String maKhoaHoc;  private String tenKhoaHoc;  private String maGiangVien;  private int soTinchi;  private double hocPhi;  public KhoaHoc() {}  public KhoaHoc(String maKhoaHoc, String tenKhoaHoc, String maGiangVien, int soTinchi, double hocPhi) {  super();  this.maKhoaHoc = maKhoaHoc;  this.tenKhoaHoc = tenKhoaHoc;  this.maGiangVien = maGiangVien;  this.soTinchi = soTinchi;  this.hocPhi = hocPhi;  }  public String getMaKhoaHoc() {  return maKhoaHoc;  }  public void setMaKhoaHoc(String maKhoaHoc) {  this.maKhoaHoc = maKhoaHoc;  }  public String getTenKhoaHoc() {  return tenKhoaHoc;  }  public void setTenKhoaHoc(String tenKhoaHoc) {  this.tenKhoaHoc = tenKhoaHoc;  }  public String getMaGiangVien() {  return maGiangVien;  }  public void setMaGiangVien(String maGiangVien) {  this.maGiangVien = maGiangVien;  }  public int getSoTinchi() {  return soTinchi;  }  public void setSoTinchi(int soTinchi) {  this.soTinchi = soTinchi;  }  public double getHocPhi() {  return hocPhi;  }  public void setHocPhi(double hocPhi) {  this.hocPhi = hocPhi;  }  } |

### **HocVien**

|  |
| --- |
| **package** model;  **public** **class** HocVien **extends** User{  **private** String maHocVien;  **public** HocVien() {}  **public** HocVien(String username, String password, String role, String maTaiKhoan,String maHocVien, String soDienThoai, String ten) {  **super**(username, password, role, maTaiKhoan, soDienThoai, ten);  **this**.maHocVien = maHocVien;  // **TODO** Auto-generated constructor stub  }  **public** String getMaHocVien() {  **return** maHocVien;  }  **public** **void** setMaHocVien(String maHocVien) {  **this**.maHocVien = maHocVien;  }  } |

### **GiangVien**

|  |
| --- |
| **package** model;  **public** **class** GiangVien **extends** User{  **private** String maGiangVien;  **private** **double** tienLuong;  **public** GiangVien() {}  **public** GiangVien(String username, String password, String role, String maTaiKhoan, String soDienThoai, String ten, String maGiangVien, **double** tienLuong) {  **super**(username, password, role, maTaiKhoan, soDienThoai, ten);  **this**.tienLuong = tienLuong;  **this**.maGiangVien= maGiangVien;  }  **public** **double** getTienLuong() {  **return** tienLuong;  }  **public** **void** setTienLuong(**double** tienLuong) {  **this**.tienLuong = tienLuong;  }  **public** String getMaGiangVien() {  **return** maGiangVien;  }  **public** **void** setMaGiangVien(String maGiangVien) {  **this**.maGiangVien = maGiangVien;  }  } |

### **DiemSo**

|  |
| --- |
| **package** model;  **public** **class** DiemSo {  **private** String maHocVien;  **private** String tenHocVien;  **private** String maKhoaHoc;  **private** **double** diem;  **public** DiemSo() {}  **public** DiemSo(String maHocVien,String tenHocVien, String maKhoaHoc, **double** diem) {  **this**.maHocVien = maHocVien;  **this**.tenHocVien = tenHocVien;  **this**.maKhoaHoc = maKhoaHoc;  **this**.diem = diem;  }  **public** String getMaHocVien() {  **return** maHocVien;  }  **public** **void** setMaHocVien(String maHocVien) {  **this**.maHocVien = maHocVien;  }  **public** String getMaKhoaHoc() {  **return** maKhoaHoc;  }  **public** **void** setMaKhoaHoc(String maKhoaHoc) {  **this**.maKhoaHoc = maKhoaHoc;  }  **public** **double** getDiem() {  **return** diem;  }  **public** **void** setDiem(**double** diem) {  **this**.diem = diem;  }  **public** String getTenHocVien() {  **return** tenHocVien;  }  **public** **void** setTenHocVien(String tenHocVien) {  **this**.tenHocVien = tenHocVien;  }  } |

### **DiemDanh**

|  |
| --- |
| package model;  public class DiemDanh {  private String maKhoaHoc;  private String maHocVien;  private String tenHocVien;  private String soDienThoai;  private String ngay;  private String coMat;  // Constructor không tham số  public DiemDanh() {}  // Constructor đầy đủ  public DiemDanh(String maKhoaHoc, String maHocVien, String tenHocVien, String soDienThoai, String ngay, String coMat) {  this.maKhoaHoc = maKhoaHoc;  this.maHocVien = maHocVien;  this.tenHocVien = tenHocVien;  this.soDienThoai = soDienThoai;  this.ngay = ngay;  this.coMat = coMat;  }  // Getters và Setters  public String getMaKhoaHoc() {  return maKhoaHoc;  }  public void setMaKhoaHoc(String maKhoaHoc) {  this.maKhoaHoc = maKhoaHoc;  }  public String getMaHocVien() {  return maHocVien;  }  public void setMaHocVien(String maHocVien) {  this.maHocVien = maHocVien;  }  public String getTenHocVien() {  return tenHocVien;  }  public void setTenHocVien(String tenHocVien) {  this.tenHocVien = tenHocVien;  }  public String getSoDienThoai() {  return soDienThoai;  }  public void setSoDienThoai(String soDienThoai) {  this.soDienThoai = soDienThoai;  }  public String getNgay() {  return ngay;  }  public void setNgay(String ngay) {  this.ngay = ngay;  }  public String getCoMat() {  return coMat;  }  public void setCoMat(String coMat) {  this.coMat = coMat;  }  } |

## **Package interface\_**

### **ITaiKhoanDAO**

|  |
| --- |
| **package** interface\_;  **import** model.TaiKhoan;  **public** **interface** ITaiKhoanDAO {  TaiKhoan kiemTraDangNhap(String username, String password);  **boolean** taoTaiKhoan(TaiKhoan taiKhoan);  } |

### 

### **IKhoaHocDAO**

|  |
| --- |
| package interface\_;  import java.util.List;  import model.KhoaHoc;  public interface IKhoaHocDAO {  boolean taoKhoaHoc(KhoaHoc khoaHoc);  List<KhoaHoc> getAllKhoaHoc();  KhoaHoc timKiemKhoaHoc(String maKhoaHoc);  boolean xoaKhoaHoc(String maKhoaHoc);  boolean capNhat(String maKhoaHoc, String tenKhoaHoc, String maGiangVien, int soTinChi, double hocPhi);  List<KhoaHoc> getKhoaHocByTaiKhoan(String maTaiKhoan);  List<KhoaHoc> layKhoaHocChuaDangKy(String maHocVien);    } |

### **IHocVienDAO**

|  |
| --- |
| package interface\_;  import java.util.List;  import model.KhoaHoc;  public interface IKhoaHocDAO {  boolean taoKhoaHoc(KhoaHoc khoaHoc);  List<KhoaHoc> getAllKhoaHoc();  KhoaHoc timKiemKhoaHoc(String maKhoaHoc);  boolean xoaKhoaHoc(String maKhoaHoc);  boolean capNhat(String maKhoaHoc, String tenKhoaHoc, String maGiangVien, int soTinChi, double hocPhi);  List<KhoaHoc> getKhoaHocByTaiKhoan(String maTaiKhoan);  List<KhoaHoc> layKhoaHocChuaDangKy(String maHocVien);    } |

### **IGiangVienDAO**

|  |
| --- |
| package interface\_;  import java.util.List;  import model.GiangVien;  import model.HocVien;  import model.KhoaHoc;  public interface IGiangVienDAO {  boolean taoGiangVien(GiangVien giangVien);  boolean xoaGiangVien(String maGiangVien);  GiangVien timKiemGiangVien(String maGiangVien);  GiangVien getGiangVienByTaiKhoan(String maTaiKhoan);  String getMaGiangVienByTaiKhoan(String maTaiKhoan);  List<GiangVien> getAllGiangVien();  List<KhoaHoc> getKhoaHocByMaGiangVien(String maGiangVien);  List<HocVien> getHocVienByKhoaHocAndGiangVien(String maKhoaHoc, String maGiangVien);  } |

### **IDiemSoDAO**

|  |
| --- |
| package interface\_;  import java.util.List;  import model.DiemSo;  public interface IDiemSoDAO {  public void diemSo(String maHocVien, String maKhoaHoc, double diem);  List<DiemSo> getDanhSachDiem(String maGiangVien, String maKhoaHoc);  List<DiemSo> getDiemTheoTaiKhoan(String maTaiKhoan);    } |

### **IDiemDanhDAO**

|  |
| --- |
| package interface\_;  import java.util.List;  import model.DiemDanh;  public interface IDiemDanhDAO {  public void diemDanh(String maHocVien, String maKhoaHoc, String ngay, String coMat);  List<DiemDanh> getDanhSachDiemDanh(String maKhoaHoc, String maGiangVien);  List<DiemDanh> diemDanhSinhVien(String maKhoaHoc, String maGiangVien);  } |

### **IDangKyDAO**

|  |
| --- |
| package interface\_;  public interface IDangKyDAO {  boolean kiemTraDangKy(String maHocVien, String maKhoaHoc);  public void dangKyKhoaHoc(String maHocVien, String maKhoaHoc);    } |

## **Package dao**

### **TaiKhoanDAO**

|  |
| --- |
| package dao;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import javax.swing.JOptionPane;  import interface\_.ITaiKhoanDAO;  import model.TaiKhoan;  public class TaiKhoanDAO implements ITaiKhoanDAO {  public TaiKhoan kiemTraDangNhap(String username, String password) {  String sql = "SELECT \* FROM TaiKhoan WHERE username = ? AND password = ?";  try (Connection conn = DBConnection.getConnection();  PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, username);  stmt.setString(2, password);  ResultSet rs = stmt.executeQuery();  if (rs.next()) {  TaiKhoan tk = new TaiKhoan();  tk.setUsername(rs.getString("username"));  tk.setPassword(rs.getString("password"));  tk.setRole(rs.getString("role"));  tk.setId(rs.getString("maTaiKhoan"));  return tk;  }  } catch (Exception e) {  e.printStackTrace();  }  return null;  }    public boolean taoTaiKhoan(TaiKhoan taiKhoan) {  if (DBCheckExist.isMaTonTai("TaiKhoan", "maTaiKhoan", taiKhoan.getMaTaiKhoan())) {  System.out.println("Mã tài khoản đã tồn tại.");  JOptionPane.showMessageDialog(null,"Mã tài khoản đã tồn tại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  return false;  }  String sql = "INSERT INTO TaiKhoan (username, password, role, maTaiKhoan) VALUES (?, ?, ?, ?)";    try (Connection conn = DBConnection.getConnection();  PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, taiKhoan.getUsername());  stmt.setString(2, taiKhoan.getPassword()); // Lưu ý: nếu dùng mật khẩu hash thì hash trước  stmt.setString(3, taiKhoan.getRole());  stmt.setString(4, taiKhoan.getMaTaiKhoan());  int rowsInserted = stmt.executeUpdate();  return rowsInserted > 0;  } catch (Exception e) {  e.printStackTrace();  }  return false;  }  } |

### **KhoaHocDAO**

|  |
| --- |
| package dao;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  import javax.swing.JOptionPane;  import interface\_.IKhoaHocDAO;  import model.KhoaHoc;  public class KhoaHocDAO implements IKhoaHocDAO {  public boolean taoKhoaHoc(KhoaHoc khoaHoc) {  if (DBCheckExist.isMaTonTai("KhoaHoc", "makhoaHoc", khoaHoc.getMaKhoaHoc())) {  System.out.println("Mã khóa học đã tồn tại.");  JOptionPane.showMessageDialog(null, "Mã khóa học đã tồn tại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  return false;  }  String sql = "INSERT INTO KhoaHoc (maKhoaHoc,tenKhoaHoc, maGiangVien, soTinCHi, hocPhi ) VALUES (?, ?, ?, ?, ?)";  try (Connection conn = DBConnection.getConnection(); PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, khoaHoc.getMaKhoaHoc().toUpperCase());  stmt.setString(2, khoaHoc.getTenKhoaHoc());  stmt.setString(3, khoaHoc.getMaGiangVien().toUpperCase());  stmt.setInt(4, khoaHoc.getSoTinchi());  stmt.setDouble(5, khoaHoc.getHocPhi());  int rowsInserted = stmt.executeUpdate();  return rowsInserted > 0;  } catch (Exception e) {  e.printStackTrace();  }  return false;  }  public List<KhoaHoc> getAllKhoaHoc() {  List<KhoaHoc> danhSach = new ArrayList<>();  String sql = "SELECT \* FROM KhoaHoc";  try (Connection conn = DBConnection.getConnection();  PreparedStatement stmt = conn.prepareStatement(sql);  ResultSet rs = stmt.executeQuery()) {  while (rs.next()) {  KhoaHoc kh = new KhoaHoc();  kh.setMaKhoaHoc(rs.getString("maKhoaHoc"));  kh.setTenKhoaHoc(rs.getString("tenKhoaHoc"));  kh.setMaGiangVien(rs.getString("maGiangVien"));  kh.setSoTinchi(rs.getInt("soTinChi"));  kh.setHocPhi(rs.getDouble("hocPhi"));  danhSach.add(kh);  }  } catch (Exception e) {  e.printStackTrace();  }  return danhSach;  }  public KhoaHoc timKiemKhoaHoc(String maKhoaHoc) {  String sql = "SELECT maKhoaHoc, tenKhoaHoc, maGiangVien, soTinCHi, hocPhi FROM KhoaHoc WHERE maKhoaHoc = ?";  try (Connection conn = DBConnection.getConnection(); PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, maKhoaHoc);  ResultSet rs = stmt.executeQuery();  if (rs.next()) {  KhoaHoc kh = new KhoaHoc();  kh.setMaKhoaHoc(rs.getString("maKhoaHoc"));  kh.setTenKhoaHoc(rs.getString("tenKhoaHoc"));  kh.setMaGiangVien(rs.getString("maGiangVien"));  kh.setSoTinchi(rs.getInt("soTinChi"));  kh.setHocPhi(rs.getDouble("hocPhi"));  return kh;  }  } catch (Exception e) {  e.printStackTrace();  }  return null;  }  public boolean xoaKhoaHoc(String maKhoaHoc) {  String sql = "DELETE FROM KhoaHoc WHERE maKhoaHoc = ?";  try (Connection conn = DBConnection.getConnection(); PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, maKhoaHoc);  int rowsAffected = stmt.executeUpdate();  return rowsAffected > 0;  } catch (Exception e) {  e.printStackTrace();  }  return false;  }  public boolean capNhat(String maKhoaHoc, String tenKhoaHoc, String maGiangVien, int soTinChi, double hocPhi) {  String sql = "UPDATE KHOAHOC SET tenKhoaHoc = ?, maGiangVien = ?, soTinChi = ?, hocPhi = ? WHERE maKhoaHoc = ?";  try (Connection conn = DBConnection.getConnection(); PreparedStatement stmt = conn.prepareStatement(sql);) {  stmt.setString(1, tenKhoaHoc);  stmt.setString(2, maGiangVien);  stmt.setInt(3, soTinChi);  stmt.setDouble(4, hocPhi);  stmt.setString(5, maKhoaHoc);  int rowsAffected = stmt.executeUpdate();  return rowsAffected > 0;  } catch (SQLException e) {  e.printStackTrace();  }  return false;  }  public List<KhoaHoc> getKhoaHocByTaiKhoan(String maTaiKhoan) {  List<KhoaHoc> list = new ArrayList<>();  String sql = "SELECT kh.\* FROM KhoaHoc kh "  + "JOIN GiangVien gv ON kh.maGiangVien = gv.maGiangVien "  + "WHERE gv.maTaiKhoan = ?";  try (Connection conn = DBConnection.getConnection();  PreparedStatement pst = conn.prepareStatement(sql)) {  pst.setString(1, maTaiKhoan);  ResultSet rs = pst.executeQuery();  while (rs.next()) {  KhoaHoc kh = new KhoaHoc();  kh.setMaKhoaHoc(rs.getString("maKhoaHoc"));  kh.setTenKhoaHoc(rs.getString("tenKhoaHoc"));  kh.setMaGiangVien(rs.getString("maGiangVien"));  kh.setSoTinchi(rs.getInt("soTinChi"));  kh.setHocPhi(rs.getDouble("hocPhi"));  list.add(kh);  }  } catch (Exception e) {  e.printStackTrace();  }  return list;  }  public List<KhoaHoc> layKhoaHocChuaDangKy(String maHocVien) {  List<KhoaHoc> danhSach = new ArrayList<>();  String sql = "SELECT \* FROM KhoaHoc WHERE MaKhoaHoc NOT IN (" +  "SELECT MaKhoaHoc FROM DangKy WHERE MaHocVien = ?)";  try (Connection con = DBConnection.getConnection();  PreparedStatement stmt = con.prepareStatement(sql)) {  stmt.setString(1, maHocVien);  ResultSet rs = stmt.executeQuery();  while (rs.next()) {  KhoaHoc kh = new KhoaHoc();  kh.setMaKhoaHoc(rs.getString("MaKhoaHoc"));  kh.setTenKhoaHoc(rs.getString("TenKhoaHoc"));  kh.setSoTinchi(rs.getInt("soTinChi"));  kh.setHocPhi(rs.getDouble("hocPhi"));    danhSach.add(kh);  }  } catch (Exception e) {  e.printStackTrace();  }  return danhSach;  }  } |

### **HocVienDAO**

|  |
| --- |
| package dao;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  import javax.swing.JOptionPane;  import interface\_.IHocVienDAO;  import model.HocVien;  import model.User;  public class HocVienDAO implements IHocVienDAO {  public boolean taoHocVien(HocVien hocVien) {  if (DBCheckExist.isMaTonTai("HocVien", "maHocVien", hocVien.getMaHocVien())) {  System.out.println("Mã tài khoản đã tồn tại.");  JOptionPane.showMessageDialog(null, "Mã học viên đã tồn tại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  return false;  }  String sql = "INSERT INTO HocVien (maHocVien, tenHocVien, sdtHocVien, maTaiKhoan) VALUES (?, ?, ?, ?)";  try (Connection conn = DBConnection.getConnection(); PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, hocVien.getMaHocVien());  stmt.setString(2, hocVien.getTen());  stmt.setString(3, hocVien.getSoDienThoai());  stmt.setString(4, hocVien.getMaTaiKhoan());  int rowsInserted = stmt.executeUpdate();  return rowsInserted > 0;  } catch (Exception e) {  e.printStackTrace();  }  return false;  }  public List<HocVien> getAllHocVien() {  List<HocVien> danhSach = new ArrayList<>();  String sql = "SELECT \* FROM HocVien";  try (Connection conn = DBConnection.getConnection();  PreparedStatement stmt = conn.prepareStatement(sql);  ResultSet rs = stmt.executeQuery()) {  while (rs.next()) {  HocVien hv = new HocVien();  hv.setMaHocVien(rs.getString("maHocVien"));  hv.setTen(rs.getString("tenHocVien"));  hv.setSoDienThoai(rs.getString("sdtHocVien"));  danhSach.add(hv);  }  } catch (Exception e) {  e.printStackTrace();  }  return danhSach;  }  public HocVien timKiemHocVien(String maHocVien) {  String sql = "SELECT maHocVien, tenHocVien, sdtHocVien FROM HocVien WHERE maHocVien = ?";  try (Connection conn = DBConnection.getConnection(); PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, maHocVien);  ResultSet rs = stmt.executeQuery();  if (rs.next()) {  HocVien hv = new HocVien();  hv.setMaHocVien(rs.getString("maHocVien"));  hv.setTen(rs.getString("tenHocVien"));  hv.setSoDienThoai(rs.getString("sdtHocVien"));  return hv;  }  } catch (Exception e) {  e.printStackTrace();  }  return null;  }  public boolean xoaHocVien(String maHocVien) {  String sql = "DELETE FROM HocVien WHERE maHocVien = ?";  try (Connection conn = DBConnection.getConnection(); PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, maHocVien);  int rowsAffected = stmt.executeUpdate();  return rowsAffected > 0;  } catch (Exception e) {  e.printStackTrace();  }  return false;  }  public boolean capNhat(String maHV, String tenMoi, String sdtMoi) {  String sql = "UPDATE HOCVIEN SET tenHocVien = ?, sdtHocVien = ? WHERE maHocVien = ?";  try (Connection conn = DBConnection.getConnection(); PreparedStatement stmt = conn.prepareStatement(sql);) {  stmt.setString(1, tenMoi);  stmt.setString(2, sdtMoi);  stmt.setString(3, maHV);  int rowsAffected = stmt.executeUpdate();  return rowsAffected > 0;  } catch (SQLException e) {  e.printStackTrace();  }  return false;  }  public HocVien getThongTinHocVien(String maTaiKhoan) {  String sql = "SELECT \* FROM HOCVIEN WHERE maTaiKhoan = ?";  try (Connection conn = DBConnection.getConnection(); PreparedStatement pst = conn.prepareStatement(sql)) {  pst.setString(1, maTaiKhoan);  ResultSet rs = pst.executeQuery();  if (rs.next()) {  HocVien hv = new HocVien();  hv.setMaHocVien(rs.getString("maHocVien"));  hv.setTen(rs.getString("tenHocVien"));  hv.setSoDienThoai(rs.getString("sdtHocVien"));  return hv;  }  } catch (Exception e) {  e.printStackTrace();  }  return null;  }  public String getMaHocVIenByTaiKhoan(String maTaiKhoan) {  String sql = "SELECT maHocVien FROM HocVien WHERE maTaiKhoan = ?";  try (Connection conn = DBConnection.getConnection(); PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, maTaiKhoan);  ResultSet rs = stmt.executeQuery();  if (rs.next()) {  return rs.getString("maHocVien");  }  } catch (Exception e) {  e.printStackTrace();  }  return null;  }  } |

### **GiangVienDAO**

|  |
| --- |
| package dao;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  import javax.swing.JOptionPane;  import interface\_.IGiangVienDAO;  import model.GiangVien;  import model.HocVien;  import model.KhoaHoc;  import model.User;  public class GiangVienDAO implements IGiangVienDAO{  public boolean taoGiangVien(GiangVien giangVien) {  if (DBCheckExist.isMaTonTai("GiangVien", "maGiangVien", giangVien.getMaGiangVien())) {  JOptionPane.showMessageDialog(null, "Mã giảng viên đã tồn tại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  return false;  }  String sql = "INSERT INTO GiangVien (maGiangVien, tenGiangVien, sdtGiangVien, luong, maTaiKhoan) VALUES (?, ?, ?, ?, ?)";  try (Connection conn = DBConnection.getConnection(); PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, giangVien.getMaGiangVien());  stmt.setString(2, giangVien.getTen());  stmt.setString(3, giangVien.getSoDienThoai());  stmt.setDouble(4, giangVien.getTienLuong());  stmt.setString(5, giangVien.getMaTaiKhoan());  int rowsInserted = stmt.executeUpdate();  return rowsInserted > 0;  } catch (Exception e) {  e.printStackTrace();  }  return false;  }  public List<GiangVien> getAllGiangVien() {  List<GiangVien> danhSach = new ArrayList<>();  String sql = "SELECT \* FROM GiangVien";  try (Connection conn = DBConnection.getConnection();  PreparedStatement stmt = conn.prepareStatement(sql);  ResultSet rs = stmt.executeQuery()) {  while (rs.next()) {  GiangVien gv = new GiangVien();  gv.setMaGiangVien(rs.getString("maGiangVien"));  gv.setTen(rs.getString("tenGiangVien"));  gv.setSoDienThoai(rs.getString("sdtGiangVien"));  gv.setTienLuong(rs.getDouble("luong"));  danhSach.add(gv);  }  } catch (Exception e) {  e.printStackTrace();  }  return danhSach;  }  public List<KhoaHoc> getKhoaHocByMaGiangVien(String maGiangVien) {  List<KhoaHoc> danhSach = new ArrayList<>();  String sql = "SELECT \* FROM KhoaHoc WHERE maGiangVien = ?";  try (Connection conn = DBConnection.getConnection(); PreparedStatement pst = conn.prepareStatement(sql)) {  pst.setString(1, maGiangVien);  ResultSet rs = pst.executeQuery();  while (rs.next()) {  KhoaHoc kh = new KhoaHoc();  kh.setMaKhoaHoc(rs.getString("maKhoaHoc"));  kh.setTenKhoaHoc(rs.getString("tenKhoaHoc"));  kh.setMaGiangVien(rs.getString("maGiangVien"));  kh.setSoTinchi(rs.getInt("soTinChi"));  kh.setHocPhi(rs.getDouble("hocPhi"));  danhSach.add(kh);  }  } catch (Exception e) {  e.printStackTrace();  }  return danhSach;  }  public List<HocVien> getHocVienByKhoaHocAndGiangVien(String maKhoaHoc, String maGiangVien) {  List<HocVien> danhSach = new ArrayList<>();  String sql = """  SELECT hv.maHocVien, hv.tenHocVien, hv.sdtHocVien  FROM HOCVIEN hv  JOIN DANGKY dk ON hv.maHocVien = dk.maHocVien  JOIN KHOAHOC kh ON dk.maKhoaHoc = kh.maKhoaHoc  WHERE kh.maKhoaHoc = ? AND kh.maGiangVien = ?  """;  try (Connection conn = DBConnection.getConnection(); PreparedStatement pst = conn.prepareStatement(sql)) {  pst.setString(1, maKhoaHoc);  pst.setString(2, maGiangVien);  ResultSet rs = pst.executeQuery();  while (rs.next()) {  HocVien hv = new HocVien();  hv.setMaHocVien(rs.getString("maHocVien"));  hv.setTen(rs.getString("TenHocVien"));  hv.setSoDienThoai(rs.getString("SDTHocVien"));  danhSach.add(hv);  }  } catch (SQLException e) {  e.printStackTrace();  }  return danhSach;  }  public boolean xoaGiangVien(String maGiangVien) {  String sql = "DELETE FROM GiangVien WHERE maGiangVien = ?";  try (Connection conn = DBConnection.getConnection(); PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, maGiangVien);  int rowsAffected = stmt.executeUpdate();  return rowsAffected > 0;  } catch (Exception e) {  e.printStackTrace();  }  return false;  }  public GiangVien timKiemGiangVien(String maGiangVien) {  String sql = "SELECT \* FROM GIANGVIEN WHERE maGiangVien = ?";  try (Connection conn = DBConnection.getConnection(); PreparedStatement pst = conn.prepareStatement(sql)) {  pst.setString(1, maGiangVien);  ResultSet rs = pst.executeQuery();  if (rs.next()) {  GiangVien gv = new GiangVien();  gv.setMaGiangVien(rs.getString("maGiangVien"));  gv.setTen(rs.getString("tenGiangVien"));  gv.setSoDienThoai(rs.getString("sdtGiangVien"));  gv.setTienLuong(rs.getDouble("luong"));  return gv;  }  } catch (Exception e) {  e.printStackTrace();  }  return null;  }  public String getMaGiangVienByTaiKhoan(String maTaiKhoan) {  String sql = "SELECT maGiangVien FROM GiangVien WHERE maTaiKhoan = ?";  try (Connection conn = DBConnection.getConnection(); PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, maTaiKhoan);  ResultSet rs = stmt.executeQuery();  if (rs.next()) {  return rs.getString("maGiangVien");  }  } catch (Exception e) {  e.printStackTrace();  }  return null;  }  public GiangVien getGiangVienByTaiKhoan(String maTaiKhoan) {  String sql = "SELECT \* FROM GiangVien WHERE maTaiKhoan = ?";  try (Connection conn = DBConnection.getConnection(); PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, maTaiKhoan);  ResultSet rs = stmt.executeQuery();  if (rs.next()) {  GiangVien gv = new GiangVien();  gv.setMaGiangVien(rs.getString("maGiangVien"));  gv.setTen(rs.getString("tenGiangVien"));  gv.setSoDienThoai(rs.getString("sdtGiangVien"));  gv.setTienLuong(rs.getDouble("luong"));  return gv;  }  } catch (Exception e) {  e.printStackTrace();  }  return null;  }  } |

### **DiemSoDAO**

|  |
| --- |
| package dao;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  import interface\_.IDiemSoDAO;  import model.DiemSo;  public class DiemSoDAO implements IDiemSoDAO {    public void diemSo(String maHocVien, String maKhoaHoc, double diem) {  String checkQuery = "SELECT COUNT(\*) FROM DIEMSO WHERE maHocVien = ? AND maKhoaHoc = ?";  String insertQuery = "INSERT INTO DIEMSO(maHocVien, maKhoaHoc, diem) VALUES (?, ?, ?)";  String updateQuery = "UPDATE DIEMSO SET diem = ? WHERE maHocVien = ? AND maKhoaHoc = ?";  try (Connection conn = DBConnection.getConnection()) {    try (PreparedStatement checkStmt = conn.prepareStatement(checkQuery)) {  checkStmt.setString(1, maHocVien);  checkStmt.setString(2, maKhoaHoc);    ResultSet rs = checkStmt.executeQuery();  if (rs.next() && rs.getInt(1) > 0) {    try (PreparedStatement updateStmt = conn.prepareStatement(updateQuery)) {  updateStmt.setDouble(1, diem);  updateStmt.setString(2, maHocVien);  updateStmt.setString(3, maKhoaHoc);  updateStmt.executeUpdate();  }  } else {    try (PreparedStatement insertStmt = conn.prepareStatement(insertQuery)) {  insertStmt.setString(1, maHocVien);  insertStmt.setString(2, maKhoaHoc);  insertStmt.setDouble(3, diem);  insertStmt.executeUpdate();  }  }  }  } catch (SQLException e) {  e.printStackTrace();      }  }  public List<DiemSo> getDanhSachDiem(String maGiangVien, String maKhoaHoc) {  List<DiemSo> ds = new ArrayList<>();  String sql = "SELECT " + " hv.maHocVien, " + " hv.tenHocVien, " + " ds.diem " + "FROM "  + " KHOAHOC kh " + "JOIN " + " DANGKY dk ON kh.maKhoaHoc = dk.maKhoaHoc " + "JOIN "  + " HOCVIEN hv ON dk.maHocVien = hv.maHocVien " + "LEFT JOIN "  + " DIEMSO ds ON ds.maHocVien = hv.maHocVien AND ds.maKhoaHoc = kh.maKhoaHoc " + "WHERE "  + " kh.maKhoaHoc = ? AND kh.maGiangVien = ? " + "ORDER BY " + " hv.maHocVien";  try (Connection conn = DBConnection.getConnection(); PreparedStatement pst = conn.prepareStatement(sql)) {    pst.setString(1, maKhoaHoc );  pst.setString(2, maGiangVien);  try (ResultSet rs = pst.executeQuery()) {  while (rs.next()) {  DiemSo diem = new DiemSo();  diem.setMaHocVien(rs.getString("maHocVien"));  diem.setTenHocVien(rs.getString("tenHocVien"));  diem.setDiem(rs.getDouble("diem"));  ds.add(diem);  }  }  } catch (Exception e) {    e.printStackTrace();    }  return ds;  }  public List<DiemSo> getDiemTheoTaiKhoan(String maTaiKhoan) {  List<DiemSo> list = new ArrayList<>();  String sql = "SELECT dk.MaKhoaHoc, kh.TenKhoaHoc, ds.Diem "  + "FROM DangKy dk "  + "JOIN KhoaHoc kh ON dk.MaKhoaHoc = kh.MaKhoaHoc "  + "LEFT JOIN DiemSo ds ON dk.MaHocVien = ds.MaHocVien AND dk.MaKhoaHoc = ds.MaKhoaHoc "  + "WHERE dk.MaHocVien = ?";  try (Connection conn = DBConnection.getConnection();  PreparedStatement pst = conn.prepareStatement(sql)) {  pst.setString(1, maTaiKhoan);  ResultSet rs = pst.executeQuery();  while (rs.next()) {  DiemSo diem = new DiemSo();  diem.setMaKhoaHoc(rs.getString("maKhoaHoc"));  diem.setDiem(rs.getFloat("diem"));  list.add(diem);  }  } catch (Exception e) {  e.printStackTrace();  }  return list;  }  } |

### **DiemDanhDAO**

|  |
| --- |
| package dao;  import java.sql.Connection;  import java.sql.Date;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.text.SimpleDateFormat;  import java.util.ArrayList;  import java.util.List;  import interface\_.IDiemDanhDAO;  import model.DiemDanh;  import model.DiemSo;  public class DiemDanhDAO implements IDiemDanhDAO{  public void diemDanh(String maHocVien, String maKhoaHoc, String ngay, String coMat) {  String checkQuery = "SELECT COUNT(\*) FROM DIEMDANH WHERE maHocVien = ? AND maKhoaHoc = ? AND ngay = ?";  String insertQuery = "INSERT INTO DIEMDANH(maHocVien, maKhoaHoc, ngay, comat) VALUES (?, ?, ?, ?)";  String updateQuery = "UPDATE DIEMDANH SET comat = ? WHERE maHocVien = ? AND maKhoaHoc = ? AND ngay = ?";  try (Connection conn = DBConnection.getConnection()) {  // Kiểm tra xem học viên đã điểm danh ngày hôm đó chưa  try (PreparedStatement checkStmt = conn.prepareStatement(checkQuery)) {  checkStmt.setString(1, maHocVien);  checkStmt.setString(2, maKhoaHoc);  checkStmt.setString(3, ngay);  ResultSet rs = checkStmt.executeQuery();  if (rs.next() && rs.getInt(1) > 0) {    try (PreparedStatement updateStmt = conn.prepareStatement(updateQuery)) {  updateStmt.setString(1, coMat);  updateStmt.setString(2, maHocVien);  updateStmt.setString(3, maKhoaHoc);  updateStmt.setString(4, ngay);  updateStmt.executeUpdate();  }  } else {  try (PreparedStatement insertStmt = conn.prepareStatement(insertQuery)) {  insertStmt.setString(1, maHocVien);  insertStmt.setString(2, maKhoaHoc);  insertStmt.setString(3, ngay);  insertStmt.setString(4, coMat);  insertStmt.executeUpdate();  }  }  }  } catch (SQLException e) {  e.printStackTrace();  }  }  public List<DiemDanh> getDanhSachDiemDanh(String maKhoaHoc, String maGiangVien) {  List<DiemDanh> ds = new ArrayList<>();  String sql = "SELECT " +  " kh.maKhoaHoc, " +  " hv.maHocVien, " +  " hv.tenHocVien, " +  " hv.sdtHocVien, " +  " dd.ngay, " +  " dd.comat " +  "FROM " +  " KHOAHOC kh " +  "JOIN DANGKY dk ON kh.maKhoaHoc = dk.maKhoaHoc " +  "JOIN HOCVIEN hv ON dk.maHocVien = hv.maHocVien " +  "JOIN DIEMDANH dd ON dd.maHocVien = hv.maHocVien AND dd.maKhoaHoc = kh.maKhoaHoc " +  "WHERE kh.maKhoaHoc = ? AND kh.maGiangVien = ? " +  "ORDER BY dd.ngay, hv.maHocVien";  try (Connection conn = DBConnection.getConnection(); PreparedStatement pst = conn.prepareStatement(sql)) {    pst.setString(2, maGiangVien);  pst.setString(1, maKhoaHoc);  try (ResultSet rs = pst.executeQuery()) {  while (rs.next()) {  DiemDanh dd = new DiemDanh();  dd.setMaKhoaHoc(rs.getString("maKhoaHoc"));  dd.setMaHocVien(rs.getString("maHocVien"));  dd.setTenHocVien(rs.getString("tenHocVien"));  dd.setSoDienThoai(rs.getString("sdthocvien"));  dd.setNgay(rs.getString("ngay"));  dd.setCoMat(rs.getString("coMat"));  ds.add(dd);  }  }  } catch (Exception e) {  e.printStackTrace();  }  return ds;  }  public List<DiemDanh> diemDanhSinhVien(String maKhoaHoc, String maGiangVien) {  List<DiemDanh> ds = new ArrayList<>();  String sql = "SELECT "  + " kh.maKhoaHoc, "  + " hv.maHocVien, "  + " hv.tenHocVien, "  + " hv.sdtHocVien, "  + " NVL(dd.ngay, '') AS ngay, "  + " NVL(dd.comat, '') AS comat "  + "FROM "  + " KHOAHOC kh "  + "JOIN "  + " DANGKY dk ON kh.maKhoaHoc = dk.maKhoaHoc "  + "JOIN "  + " HOCVIEN hv ON dk.maHocVien = hv.maHocVien "  + "LEFT JOIN "  + " DIEMDANH dd ON dd.maHocVien = hv.maHocVien "  + " AND dd.maKhoaHoc = kh.maKhoaHoc "  + " AND dd.ngay = TO\_CHAR(SYSDATE, 'DD/MM/YYYY') "  + "WHERE "  + " kh.maKhoaHoc = ? "  + " AND kh.maGiangVien = ? "  + "ORDER BY "  + " dd.ngay ASC, hv.maHocVien";  try (Connection conn = DBConnection.getConnection(); PreparedStatement pst = conn.prepareStatement(sql)) {  pst.setString(2, maGiangVien);  pst.setString(1, maKhoaHoc);  try (ResultSet rs = pst.executeQuery()) {  while (rs.next()) {  DiemDanh dd = new DiemDanh();  dd.setMaKhoaHoc(rs.getString("maKhoaHoc"));  dd.setMaHocVien(rs.getString("maHocVien"));  dd.setTenHocVien(rs.getString("tenHocVien"));  dd.setSoDienThoai(rs.getString("sdthocvien"));  dd.setNgay(rs.getString("ngay"));  dd.setCoMat(rs.getString("coMat"));  ds.add(dd);  }  }  } catch (Exception e) {  e.printStackTrace();  }  return ds;  }  } |

### **DBConnection**

|  |
| --- |
| package dao;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.SQLException;  public class DBConnection {  private static Connection conn;  // Thông tin kết nối  private static final String URL = "jdbc:oracle:thin:@localhost:1521:orcl";  private static final String USER = "hr";  private static final String PASSWORD = "hr";  public static Connection getConnection() {  try {  if (conn == null || conn.isClosed()) {  Class.forName("oracle.jdbc.driver.OracleDriver");  conn = DriverManager.getConnection(URL, USER, PASSWORD);  System.out.println("Kết nối Oracle thành công!");  }  } catch (ClassNotFoundException e) {  System.err.println("Không tìm thấy driver Oracle JDBC.");  e.printStackTrace();  } catch (SQLException e) {  System.err.println("Không thể kết nối tới Oracle.");  e.printStackTrace();  }  return conn;  }  // Đóng kết nối (gọi ở nơi nào cần đóng lại)  public static void closeConnection() {  try {  if (conn != null && !conn.isClosed()) {  conn.close();  System.out.println("Đã đóng kết nối Oracle.");  }  } catch (SQLException e) {  e.printStackTrace();  }  }  } |

### **DBCheckExist**

|  |
| --- |
| package dao;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  public class DBCheckExist {  public static boolean isMaTonTai(String table, String column, String value) {  String sql = "SELECT 1 FROM " + table + " WHERE " + column + " = ?";  try (Connection conn = DBConnection.getConnection();  PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, value);  ResultSet rs = stmt.executeQuery();  return rs.next();  } catch (Exception e) {  e.printStackTrace();  }  return false;  }  } |

### **IDangKyDAO**

|  |
| --- |
| package dao;  import javax.swing.JOptionPane;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import interface\_.IDangKyDAO;  public class DangKyDAO implements IDangKyDAO{  // Hàm kiểm tra xem học viên đã đăng ký khóa học chưa  public boolean kiemTraDangKy(String maHocVien, String maKhoaHoc) {  String sql = "SELECT COUNT(\*) FROM DANGKY WHERE maHocVien = ? AND maKhoaHoc = ?";  try (Connection conn = DBConnection.getConnection(); PreparedStatement pst = conn.prepareStatement(sql)) {  pst.setString(1, maHocVien);  pst.setString(2, maKhoaHoc);  try (ResultSet rs = pst.executeQuery()) {  if (rs.next()) {  return rs.getInt(1) > 0; // Nếu trả về > 0, tức là học viên đã đăng ký  }  }  } catch (SQLException e) {  e.printStackTrace();  }  return false;  }  // Hàm đăng ký khóa học cho học viên và hiển thị dialog  public void dangKyKhoaHoc(String maHocVien, String maKhoaHoc) {  // Kiểm tra xem học viên đã đăng ký khóa học chưa  if (kiemTraDangKy(maHocVien, maKhoaHoc)) {  // Hiển thị hộp thoại thông báo học viên đã đăng ký  JOptionPane.showMessageDialog(null, "Học viên đã đăng ký khóa học này rồi!", "Thông báo",  JOptionPane.WARNING\_MESSAGE);  return; // Dừng lại nếu học viên đã đăng ký  }  String sql = "INSERT INTO DANGKY (maHocVien, maKhoaHoc) VALUES (?, ?)";  try (Connection conn = DBConnection.getConnection(); PreparedStatement pst = conn.prepareStatement(sql)) {  pst.setString(1, maHocVien);  pst.setString(2, maKhoaHoc);  int rowsAffected = pst.executeUpdate();  if (rowsAffected > 0) {  // Hiển thị hộp thoại thông báo thành công  JOptionPane.showMessageDialog(null, "Đăng ký khóa học thành công!", "Thông báo",  JOptionPane.INFORMATION\_MESSAGE);  } else {  // Hiển thị hộp thoại thông báo thất bại  JOptionPane.showMessageDialog(null, "Đăng ký thất bại! Vui lòng thử lại.", "Thông báo",  JOptionPane.ERROR\_MESSAGE);  }  } catch (SQLException e) {  e.printStackTrace();  // Hiển thị hộp thoại thông báo lỗi  JOptionPane.showMessageDialog(null, "Lỗi kết nối cơ sở dữ liệu!", "Lỗi", JOptionPane.ERROR\_MESSAGE);  }  }  } |

## **Package gui**

### **AdminGUI**

|  |
| --- |
| package gui;  import javax.swing.\*;  import java.awt.\*;  import java.awt.event.MouseAdapter;  import java.awt.event.MouseEvent;  import java.awt.event.ActionListener;  import java.awt.event.ActionEvent;  public class AdminGUI extends JFrame{  public AdminGUI() {  setTitle("ADMIN");  setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE); // Chỉ đóng frame này  setSize(800, 500);  setResizable(false);  getContentPane().setLayout(null);  JLabel titleLabel = new JLabel("Chào mừng đến với hệ thống quản lý khóa học trực tuyến", SwingConstants.CENTER);  titleLabel.setBounds(0, 20, 774, 30);  titleLabel.setFont(new Font("Arial", Font.BOLD, 18));  getContentPane().add(titleLabel);  JButton btnKhoaHoc = new JButton("Khóa học");  btnKhoaHoc.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  new KhoaHocGUI().setVisible(true);  setVisible(false);  }  });  btnKhoaHoc.setBounds(324, 115, 114, 40);  getContentPane().add(btnKhoaHoc);  JButton btnNewButton = new JButton("Tài khoản");  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  new TaiKhoanGUI().setVisible(true);  setVisible(false);  }  });  btnNewButton.setBounds(324, 165, 114, 40);  getContentPane().add(btnNewButton);  JButton btnHocVIen = new JButton("Học viên");  btnHocVIen.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  new HocVienGUI().setVisible(true);  setVisible(false);  }  });    btnHocVIen.setBounds(324, 215, 114, 40);  getContentPane().add(btnHocVIen);  JButton btnGiangVien = new JButton("Giảng viên");  btnGiangVien.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  new GiangVienGUI().setVisible(true);  setVisible(false);  }  });  btnGiangVien.setBounds(324, 265, 114, 40);  getContentPane().add(btnGiangVien);  JButton btnDangXuat = new JButton("Đăng xuất");  btnDangXuat.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  new LoginGUI().setVisible(true);  dispose();  }  });  btnDangXuat.setBounds(660, 411, 114, 40);  getContentPane().add(btnDangXuat);  setLocationRelativeTo(null);  setVisible(true);  }  public static void main(String[] args) {  SwingUtilities.invokeLater(() -> new AdminGUI());  }  } |

### **GiangVienGUI**

|  |
| --- |
| package gui;  import java.awt.BorderLayout;  import java.awt.Dimension;  import java.awt.EventQueue;  import javax.swing.BorderFactory;  import javax.swing.Box;  import javax.swing.BoxLayout;  import javax.swing.JButton;  import javax.swing.JFrame;  import javax.swing.JOptionPane;  import javax.swing.JPanel;  import javax.swing.JScrollPane;  import javax.swing.JTable;  import javax.swing.SwingUtilities;  import javax.swing.border.EmptyBorder;  import javax.swing.table.DefaultTableModel;  import dao.DBCheckExist;  import dao.DiemDanhDAO;  import dao.DiemSoDAO;  import dao.GiangVienDAO;  import dao.HocVienDAO;  import dao.KhoaHocDAO;  import model.DiemDanh;  import model.DiemSo;  import model.GiangVien;  import model.HocVien;  import model.KhoaHoc;  import java.awt.Component;  import java.awt.event.ActionListener;  import java.awt.event.MouseAdapter;  import java.awt.event.MouseEvent;  import java.awt.event.MouseListener;  import java.text.SimpleDateFormat;  import java.util.ArrayList;  import java.util.Date;  import java.util.List;  import java.awt.event.ActionEvent;  public class GiangVienGUI extends JFrame {  /\*\*  \* Create the frame.  \*  \*/  private String tableMode = "";  public GiangVienGUI() {  setTitle("Giảng viên");  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setSize(800, 500);  setMinimumSize(new Dimension(800, 500));    getContentPane().setLayout(new BorderLayout());    JTable table = new JTable(new DefaultTableModel());  JScrollPane scrollPane = new JScrollPane(table);  getContentPane().add(scrollPane, BorderLayout.CENTER);  table.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  if (e.getClickCount() == 2 && table.getSelectedRow() != -1) {  int selectedRow = table.getSelectedRow();  String maKhoaHoc = table.getValueAt(selectedRow, 0).toString();  String maHocVien = table.getValueAt(selectedRow, 1).toString();  String tenHocVien = table.getValueAt(selectedRow, 2).toString();  if (tableMode.equals("DIEM\_DANH")) {  SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");  String ngayHienTai = sdf.format(new Date());  table.setValueAt(ngayHienTai, selectedRow, 4);  int luaChon = JOptionPane.showConfirmDialog(  null,  "Học viên " + tenHocVien + " (" + maHocVien + ") có đi học không?",  "Xác nhận điểm danh",  JOptionPane.YES\_NO\_OPTION  );  if (luaChon != JOptionPane.CLOSED\_OPTION) {  boolean coMat = (luaChon == JOptionPane.YES\_OPTION);    table.setValueAt(coMat ? "Có" : "Vắng", selectedRow, 5);    DiemDanhDAO dao = new DiemDanhDAO();  dao.diemDanh(maHocVien, maKhoaHoc, ngayHienTai, coMat ? "Có" : "Vắng");  }  } else if (tableMode.equals("SUA\_DIEM")) {  String diemStr = JOptionPane.showInputDialog(  null,  "Nhập điểm cho học viên " + tenHocVien + " (" + maHocVien + "):",  "Nhập điểm",  JOptionPane.PLAIN\_MESSAGE  );  if (diemStr != null) {  try {  double diem = Double.parseDouble(diemStr);  table.setValueAt(diem, selectedRow, 3);    DiemSoDAO dao = new DiemSoDAO();  dao.diemSo(maHocVien, maKhoaHoc, diem);  } catch (NumberFormatException event) {  JOptionPane.showMessageDialog(null, "Vui lòng nhập số hợp lệ!", "Lỗi", JOptionPane.ERROR\_MESSAGE);  }  }  }  }  }  });  JPanel panel\_1 = new JPanel();  panel\_1.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));  getContentPane().add(panel\_1, BorderLayout.EAST);  panel\_1.setLayout(new BoxLayout(panel\_1, BoxLayout.Y\_AXIS));    JButton btnDSGV = new JButton("Danh sách giảng viên");  btnDSGV.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  GiangVienDAO dao = new GiangVienDAO();  List<GiangVien> dsGiangVien = dao.getAllGiangVien();  DefaultTableModel model = new DefaultTableModel();  table.setDefaultEditor(Object.class, null);  model.setColumnIdentifiers(new String[] { "Mã Giảng Viên", "Tên Giảng Viên", "SĐT", "Lương"});  table.setModel(model);  model.setRowCount(0);  for (GiangVien gv : dsGiangVien) {  model.addRow(new Object[] {  gv.getMaGiangVien(),  gv.getTen(),  gv.getSoDienThoai(),  gv.getTienLuong(),  });  }  }  });  btnDSGV.setMaximumSize(new Dimension(300, 30));  panel\_1.add(btnDSGV);    Component verticalStrut\_1 = Box.createVerticalStrut(10);  panel\_1.add(verticalStrut\_1);    JButton btnDSKHGVPT = new JButton("Danh sách khóa học giảng viên phụ trách");  btnDSKHGVPT.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  String maGiangVien = JOptionPane.showInputDialog(null, "Nhập ID giảng viên:");  if ( maGiangVien != null) {  maGiangVien = maGiangVien.toUpperCase();  if (maGiangVien.trim().isEmpty()) {  JOptionPane.showMessageDialog(null, "Bạn chưa nhập ID giảng viên");  }else if(DBCheckExist.isMaTonTai("GiangVien", "maGiangVien", maGiangVien)) {  GiangVienDAO dao = new GiangVienDAO();  List<KhoaHoc> dsKhoaHoc = dao.getKhoaHocByMaGiangVien(maGiangVien);  DefaultTableModel model = new DefaultTableModel();  table.setDefaultEditor(Object.class, null);  model.setColumnIdentifiers(  new String[] { "Mã Khóa Học", "Tên Khóa Học", "Mã Giảng Viên", "Số Tín Chỉ", "Học Phí" });  table.setModel(model);  model.setRowCount(0);  for (KhoaHoc kh : dsKhoaHoc) {  model.addRow(new Object[] { kh.getMaKhoaHoc(), kh.getTenKhoaHoc(), kh.getMaGiangVien(),  kh.getSoTinchi(), kh.getHocPhi(), });  }  }else {  JOptionPane.showMessageDialog(null,"Mã giảng viên không tồn tại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  }  }  }  });  btnDSKHGVPT.setMaximumSize(new Dimension(300, 30));  panel\_1.add(btnDSKHGVPT);      Component verticalStrut\_4\_1 = Box.createVerticalStrut(10);  panel\_1.add(verticalStrut\_4\_1);    JButton btnDSHV = new JButton("Danh sách học viên của giảng viên");  btnDSHV.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String maGiangVien = JOptionPane.showInputDialog(null, "Nhập mã giảng viên:");  if ( maGiangVien != null) {  maGiangVien = maGiangVien.toUpperCase();  if (maGiangVien.trim().isEmpty()) {  JOptionPane.showMessageDialog(null, "Bạn chưa nhập mã giảng viên");  }else if(DBCheckExist.isMaTonTai("GiangVien", "maGiangVien", maGiangVien)){  String maKhoaHoc = JOptionPane.showInputDialog(null, "Nhập mã khóa học:");  if ( maKhoaHoc != null) {  maKhoaHoc = maKhoaHoc.toUpperCase();  if (maKhoaHoc.trim().isEmpty()) {  JOptionPane.showMessageDialog(null, "Bạn chưa nhập mã khóa học");  }else if(DBCheckExist.isMaTonTai("KhoaHoc", "maKhoaHoc", maKhoaHoc)) {  GiangVienDAO dao = new GiangVienDAO();  List<HocVien> dsHocVien = dao.getHocVienByKhoaHocAndGiangVien(maKhoaHoc, maGiangVien );  DefaultTableModel model = new DefaultTableModel();  table.setDefaultEditor(Object.class, null);    model.setColumnIdentifiers(  new String[] { "Mã Khóa Học", "Mã Học Viên", "Tên Học Viên", "SĐT" });  table.setModel(model);  model.setRowCount(0);  for (HocVien hv : dsHocVien) {  model.addRow(new Object[] {maKhoaHoc, hv.getMaHocVien(), hv.getTen(), hv.getSoDienThoai() });  }  }else {  JOptionPane.showMessageDialog(null,"Mã khóa học không tồn tại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  }  }  }else {  JOptionPane.showMessageDialog(null,"Mã giảng viên không tồn tại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  }  }  }      });  btnDSHV.setMaximumSize(new Dimension(300, 30));  panel\_1.add(btnDSHV);    Component verticalStrut\_4 = Box.createVerticalStrut(10);  panel\_1.add(verticalStrut\_4);    JButton btnXoa = new JButton("Xóa giảng viên");  btnXoa.addActionListener(new ActionListener() {  DefaultTableModel model = (DefaultTableModel) table.getModel();    @Override  public void actionPerformed(ActionEvent e) {  int selectedRow = table.getSelectedRow();  if (selectedRow != -1) {  String maHV = table.getValueAt(selectedRow, 0).toString();  int confirm = JOptionPane.showConfirmDialog(  null,  "Bạn có chắc muốn xóa giảng viên này?\n",  "Xác nhận xóa",  JOptionPane.YES\_NO\_OPTION  );  if (confirm == JOptionPane.YES\_OPTION) {      GiangVienDAO dao = new GiangVienDAO();  boolean success= dao.xoaGiangVien(maHV);  if (success) {  DefaultTableModel model = (DefaultTableModel) table.getModel();  model.removeRow(selectedRow);  JOptionPane.showMessageDialog(null, "Xóa thành công!");    }  }  } else {  JOptionPane.showMessageDialog(null, "Vui lòng chọn một dòng để xóa.");    }    }  });  panel\_1.add(btnXoa);  btnXoa.setMaximumSize(new Dimension(300, 30));  panel\_1.add(Box.createVerticalStrut(10));      JButton btnTimKiem = new JButton("Tìm kiếm");  btnTimKiem.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String maGiangVien = JOptionPane.showInputDialog(null, "Nhập mã học viên:");  if ( maGiangVien != null) {  maGiangVien = maGiangVien.toUpperCase();  if (maGiangVien.trim().isEmpty()) {  JOptionPane.showMessageDialog(null, "Bạn chưa nhập mã học viên");  }else if(DBCheckExist.isMaTonTai("GiangVien", "maGiangVien", maGiangVien)){  GiangVienDAO dao = new GiangVienDAO();  GiangVien gv = dao.timKiemGiangVien(maGiangVien);  DefaultTableModel model = new DefaultTableModel();  table.setDefaultEditor(Object.class, null);  model.setColumnIdentifiers(new String[] { "Mã Giảng Viên", "Tên Giảng Viên", "SĐT", "Lương"});  table.setModel(model);  model.setRowCount(0);  if ( gv != null) {  model.addRow(new Object[] {  gv.getMaGiangVien(),  gv.getTen(),  gv.getSoDienThoai(),  gv.getTienLuong(),  });  }  }else {  JOptionPane.showMessageDialog(null,"Mã giảng viên không tồn tại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  }  }  }  });  btnTimKiem.setMaximumSize(new Dimension(300, 30));  panel\_1.add(btnTimKiem);    Component verticalStrut = Box.createVerticalStrut(10);  panel\_1.add(verticalStrut);    JButton btnDiemDanh = new JButton("Danh sách điểm danh");  btnDiemDanh.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {    String maGiangVien = JOptionPane.showInputDialog(null, "Nhập mã giảng viên:");  if ( maGiangVien != null) {  maGiangVien = maGiangVien.toUpperCase();  if (maGiangVien.trim().isEmpty()) {  JOptionPane.showMessageDialog(null, "Bạn chưa nhập mã giảng viên");  }else if(DBCheckExist.isMaTonTai("GiangVien", "maGiangVien", maGiangVien)){  String maKhoaHoc = JOptionPane.showInputDialog(null, "Nhập mã khóa học:");  if ( maKhoaHoc != null) {  maKhoaHoc = maKhoaHoc.toUpperCase();  if (maKhoaHoc.trim().isEmpty()) {  JOptionPane.showMessageDialog(null, "Bạn chưa nhập mã khóa học");  }else if(DBCheckExist.isMaTonTai("KhoaHoc", "maKhoaHoc", maKhoaHoc)) {    DiemDanhDAO dao = new DiemDanhDAO();  List<DiemDanh> dsHocVien = dao.getDanhSachDiemDanh(maKhoaHoc, maGiangVien );  DefaultTableModel model = new DefaultTableModel();  table.setModel(model);  table.setDefaultEditor(Object.class, null);  model.setColumnIdentifiers(  new String[] { "Mã Khóa Học", "Mã Học Viên", "Tên Học Viên", "SĐT" , "Ngày", "Điểm Danh" });    model.setRowCount(0);  for (DiemDanh hv : dsHocVien) {  model.addRow(new Object[] {maKhoaHoc, hv.getMaHocVien(), hv.getTenHocVien(), hv.getSoDienThoai(), hv.getNgay(), hv.getCoMat() });  }  }else {  JOptionPane.showMessageDialog(null,"Mã khóa học không tồn tại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  }      }  }else {  JOptionPane.showMessageDialog(null,"Mã giảng viên không tồn tại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  }  }  }    });  btnDiemDanh.setMaximumSize(new Dimension(300, 30));  panel\_1.add(btnDiemDanh);    Component verticalStrut\_2 = Box.createVerticalStrut(10);  panel\_1.add(verticalStrut\_2);    JButton btnSuaDiem = new JButton("Sửa điểm");  btnSuaDiem.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String maGiangVien = JOptionPane.showInputDialog(null, "Nhập mã giảng viên:");  if ( maGiangVien != null) {  maGiangVien = maGiangVien.toUpperCase();  if (maGiangVien.trim().isEmpty()) {  JOptionPane.showMessageDialog(null, "Bạn chưa nhập mã giảng viên");  }else if(DBCheckExist.isMaTonTai("GiangVien", "maGiangVien", maGiangVien)){  String maKhoaHoc = JOptionPane.showInputDialog(null, "Nhập mã khóa học:");  if ( maKhoaHoc != null) {  maKhoaHoc = maKhoaHoc.toUpperCase();  if (maKhoaHoc.trim().isEmpty()) {  JOptionPane.showMessageDialog(null, "Bạn chưa nhập mã khóa học");  }else if(DBCheckExist.isMaTonTai("KhoaHoc", "maKhoaHoc", maKhoaHoc)) {  tableMode = "SUA\_DIEM";  DiemSoDAO dao = new DiemSoDAO();  List<DiemSo> dsHocVien = dao.getDanhSachDiem(maGiangVien, maKhoaHoc );  DefaultTableModel model = new DefaultTableModel();  table.setDefaultEditor(Object.class, null);  model.setColumnIdentifiers(  new String[] { "Mã Khóa Học", "Mã Học Viên", "Tên Học Viên", "Điểm" });  table.setModel(model);  model.setRowCount(0);  for (DiemSo hv : dsHocVien) {  model.addRow(new Object[] {maKhoaHoc, hv.getMaHocVien(), hv.getTenHocVien(), hv.getDiem() });  }  }else {  JOptionPane.showMessageDialog(null,"Mã khóa học không tồn tại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  }      }  }else {  JOptionPane.showMessageDialog(null,"Mã giảng viên không tồn tại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  }  }  }  });    JButton btnDiemDanh\_1 = new JButton("Điểm danh");  btnDiemDanh\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String maGiangVien = JOptionPane.showInputDialog(null, "Nhập mã giảng viên:");  if ( maGiangVien != null) {  maGiangVien = maGiangVien.toUpperCase();  if (maGiangVien.trim().isEmpty()) {  JOptionPane.showMessageDialog(null, "Bạn chưa nhập mã giảng viên");  }else if(DBCheckExist.isMaTonTai("GiangVien", "maGiangVien", maGiangVien)){  String maKhoaHoc = JOptionPane.showInputDialog(null, "Nhập mã khóa học:");  if ( maKhoaHoc != null) {  maKhoaHoc = maKhoaHoc.toUpperCase();  if (maKhoaHoc.trim().isEmpty()) {  JOptionPane.showMessageDialog(null, "Bạn chưa nhập mã khóa học");  }else if(DBCheckExist.isMaTonTai("KhoaHoc", "maKhoaHoc", maKhoaHoc)) {  SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");  String ngayHienTai = sdf.format(new Date());  DefaultTableModel model = new DefaultTableModel();      tableMode = "DIEM\_DANH";  DiemDanhDAO dao = new DiemDanhDAO();  List<DiemDanh> dsHocVien = dao.diemDanhSinhVien(maKhoaHoc, maGiangVien );  table.setDefaultEditor(Object.class, null);  model.setColumnIdentifiers(  new String[] { "Mã Khóa Học", "Mã Học Viên", "Tên Học Viên", "SĐT" , "Ngày", "Điểm Danh" });  table.setModel(model);  model.setRowCount(0);    for (DiemDanh hv : dsHocVien) {  model.addRow(new Object[] {maKhoaHoc, hv.getMaHocVien(), hv.getTenHocVien(), hv.getSoDienThoai(), ngayHienTai, hv.getCoMat() });  }  }else {  JOptionPane.showMessageDialog(null,"Mã khóa học không tồn tại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  }      }  }else {  JOptionPane.showMessageDialog(null,"Mã giảng viên không tồn tại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  }  }  }  });  btnDiemDanh\_1.setMaximumSize(new Dimension(300, 30));  panel\_1.add(btnDiemDanh\_1);    Component verticalStrut\_2\_1 = Box.createVerticalStrut(10);  panel\_1.add(verticalStrut\_2\_1);  btnSuaDiem.setMaximumSize(new Dimension(300, 30));  panel\_1.add(btnSuaDiem);    Component verticalStrut\_3 = Box.createVerticalStrut(10);  panel\_1.add(verticalStrut\_3);    JButton btnBack = new JButton("Quay lại");  btnBack.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  new AdminGUI().setVisible(true);  dispose();  }  });  btnBack.setMaximumSize(new Dimension(300, 30));  panel\_1.add(btnBack);    setLocationRelativeTo(null);  setVisible(true);    }        public static void main(String[] args) {  SwingUtilities.invokeLater(() -> new GiangVienGUI());  Date a = new Date();  System.out.println(a.getDate());  }  } |

### **HocVienGUI**

|  |
| --- |
| package gui;  import java.awt.EventQueue;  import java.awt.event.ActionListener;  import java.text.SimpleDateFormat;  import java.util.List;  import javax.swing.JButton;  import javax.swing.JFrame;  import javax.swing.JOptionPane;  import javax.swing.JPanel;  import javax.swing.JScrollPane;  import javax.swing.JTable;  import javax.swing.SwingUtilities;  import javax.swing.border.EmptyBorder;  import javax.swing.table.DefaultTableModel;  import dao.HocVienDAO;  import model.HocVien;  import java.awt.event.MouseAdapter;  import java.awt.event.MouseEvent;  import java.awt.event.ActionEvent;  public class HocVienGUI extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  private JTable table;  private JScrollPane scrollPane;  private JButton btn\_showDS;  private JButton btn\_TimKiem;  private JButton btn\_Xoa;  private JButton btn\_Them;  private JButton btn\_fixInfo;  private JButton btn\_Back;  public HocVienGUI() {  setTitle("Học viên");  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setResizable(false);  setSize(1000, 600);  contentPane = new JPanel();  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);  table = new JTable();  scrollPane = new JScrollPane(table);  scrollPane.setBounds(26, 10, 686, 536);  contentPane.add(scrollPane);  btn\_showDS = new JButton("Hiển thị danh sách");  btn\_showDS.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  HocVienDAO dao = new HocVienDAO();  List<HocVien> dsHocVien = dao.getAllHocVien();  DefaultTableModel model = new DefaultTableModel();  model.setColumnIdentifiers(new String[] { "Mã Học Viên", "Tên Học Viên", "SĐT" });  table.setModel(model);  model.setRowCount(0);  for (HocVien hv : dsHocVien) {  model.addRow(new Object[] { hv.getMaHocVien(), hv.getTen(), hv.getSoDienThoai(), });  }  }  });  btn\_showDS.setBounds(727, 20, 195, 40);  contentPane.add(btn\_showDS);  btn\_TimKiem = new JButton("Tìm kiếm học viên");  btn\_TimKiem.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String maHocVien = JOptionPane.showInputDialog(null, "Nhập mã học viên:");  if (maHocVien != null) {  maHocVien = maHocVien.toUpperCase();  if (maHocVien.trim().isEmpty()) {  JOptionPane.showMessageDialog(null, "Bạn chưa nhập mã học viên");  } else {  HocVienDAO dao = new HocVienDAO();  HocVien hv = dao.timKiemHocVien(maHocVien);  DefaultTableModel model = (DefaultTableModel) table.getModel();  table.setDefaultEditor(Object.class, null);  model.setColumnIdentifiers(new String[] { "Mã Học Viên", "Tên Học Viên", "SĐT" });  model.setRowCount(0);  if (hv != null) {  model.addRow(new Object[] { hv.getMaHocVien(), hv.getTen(), hv.getSoDienThoai() });  } else {  JOptionPane.showMessageDialog(null, "Không tìm thấy học viên!");  }  }  }  }  });  btn\_TimKiem.setBounds(727, 70, 195, 40);  contentPane.add(btn\_TimKiem);  btn\_Xoa = new JButton("Xóa học viên");  btn\_Xoa.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  int selectedRow = table.getSelectedRow();  if (selectedRow != -1) {  String maHV = table.getValueAt(selectedRow, 0).toString();  int confirm = JOptionPane.showConfirmDialog(null, "Bạn có chắc muốn xóa học viên " + maHV + "?",  "Xác nhận", JOptionPane.YES\_NO\_OPTION);  if (confirm == JOptionPane.YES\_OPTION) {  HocVienDAO dao = new HocVienDAO();  boolean success = dao.xoaHocVien(maHV);  if (success) {  DefaultTableModel model = (DefaultTableModel) table.getModel();  model.removeRow(selectedRow);  JOptionPane.showMessageDialog(null, "Xóa thành công!");    } else {  JOptionPane.showMessageDialog(null, "Xóa thất bại hoặc học viên không tồn tại!");  }  }  } else {  JOptionPane.showMessageDialog(null, "Vui lòng chọn học viên cần xóa!");  }  }  });  btn\_Xoa.setBounds(727, 120, 195, 40);  contentPane.add(btn\_Xoa);  btn\_fixInfo = new JButton("Sửa học viên");  btn\_fixInfo.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  int selectedRow = table.getSelectedRow();  if (selectedRow == -1) {  JOptionPane.showMessageDialog(null, "Vui lòng chọn học viên để chỉnh sửa.");  return;  }  String maHV = table.getValueAt(selectedRow, 0).toString();  String tenHV = table.getValueAt(selectedRow, 1).toString();  String sdtHV = table.getValueAt(selectedRow, 2).toString();  SuaHocVienGUI dialog = new SuaHocVienGUI(maHV, tenHV, sdtHV);  dialog.setVisible(true);  }  });  btn\_fixInfo.setBounds(727, 170, 195, 40);  contentPane.add(btn\_fixInfo);  btn\_Back = new JButton("Quay lại");  btn\_Back.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  new AdminGUI().setVisible(true);  setVisible(false);  }  });  btn\_Back.setBounds(727, 491, 195, 40);  contentPane.add(btn\_Back);  setLocationRelativeTo(null);  setVisible(true);  }  public void addShowListListener(ActionListener listener) {  btn\_showDS.addActionListener(listener);  }  public void addEditListener(ActionListener listener) {  btn\_fixInfo.addActionListener(listener);  }  public void displayHocVienList(Object[][] data, String[] columnNames) {  DefaultTableModel model = new DefaultTableModel(data, columnNames) {  @Override  public boolean isCellEditable(int row, int column) {  return false;  }  };  table.setModel(model);  // Tùy chỉnh hiển thị  table.getColumnModel().getColumn(0).setPreferredWidth(80);  table.getColumnModel().getColumn(1).setPreferredWidth(150);  table.getColumnModel().getColumn(2).setPreferredWidth(100);  table.getColumnModel().getColumn(3).setPreferredWidth(200);  table.setRowHeight(25);  }  public String getSelectedHocVien() {  int selectedRow = table.getSelectedRow();  if (selectedRow >= 0) {  return table.getValueAt(selectedRow, 0).toString();  }  return null;  }  public Object[] getSelectedHocVienDetails() {  int selectedRow = table.getSelectedRow();  if (selectedRow >= 0) {  return new Object[] { table.getValueAt(selectedRow, 0), table.getValueAt(selectedRow, 1),  table.getValueAt(selectedRow, 2), table.getValueAt(selectedRow, 3) };  }  return null;  }  // Xoa hoc vien  public void addDeleteListener(ActionListener listener) {  btn\_Xoa.addActionListener(listener); // Gán sự kiện cho nút Xóa  }  public void showMessage(String message, String title, int messageType) {  JOptionPane.showMessageDialog(this, message, title, messageType);  }  // Thêm phương thức mới để hiển thị hộp thoại xác nhận  public int showConfirmDialog(String message, String title) {  Object[] options = { "Xóa", "Hủy" };  return JOptionPane.showOptionDialog(this, message, title, JOptionPane.DEFAULT\_OPTION,  JOptionPane.WARNING\_MESSAGE, null, options, options[1] // Mặc định chọn "Hủy"  );  }  public void loadDataToTable(List<HocVien> dsHocVien) {  DefaultTableModel model = (DefaultTableModel) table.getModel();  table.setDefaultEditor(Object.class, null);  model.setRowCount(0);  for (HocVien hv : dsHocVien) {  model.addRow(new Object[] { hv.getMaHocVien(), hv.getTen(), hv.getSoDienThoai() });  }  }  // Nut xu ly them hoc vien  public void addInsertListener(ActionListener listener) {  btn\_Them.addActionListener(listener);  }  public void addSearchListener(ActionListener listener) {  btn\_TimKiem.addActionListener(listener);  }  public void addBackListener(ActionListener listener) {  btn\_Back.addActionListener(listener); // Gắn ActionListener cho btn\_Back  }  public static void main(String[] args) {  SwingUtilities.invokeLater(() -> new HocVienGUI());  }  } |

### **KhoaHocGUI**

|  |
| --- |
| package gui;  import java.awt.EventQueue;  import java.awt.event.ActionListener;  import java.text.SimpleDateFormat;  import java.util.List;  import javax.swing.JButton;  import javax.swing.JFrame;  import javax.swing.JOptionPane;  import javax.swing.JPanel;  import javax.swing.JScrollPane;  import javax.swing.JTable;  import javax.swing.SwingUtilities;  import javax.swing.border.EmptyBorder;  import javax.swing.table.DefaultTableModel;  import dao.DBCheckExist;  import dao.DangKyDAO;  import dao.GiangVienDAO;  import dao.HocVienDAO;  import dao.KhoaHocDAO;  import model.HocVien;  import model.KhoaHoc;  import java.awt.event.MouseAdapter;  import java.awt.event.MouseEvent;  import java.awt.event.ActionEvent;  public class KhoaHocGUI extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  private JTable table;  private JScrollPane scrollPane;  private JButton btn\_showDS;  private JButton btn\_TimKiem;  private JButton btn\_Xoa;  private JButton btn\_Them;  private JButton btn\_fixInfo;  private JButton btn\_Back;  public KhoaHocGUI() {  setTitle("Khóa học");  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setResizable(false);  setSize(1000, 600);  contentPane = new JPanel();  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);  table = new JTable();  scrollPane = new JScrollPane(table);  scrollPane.setBounds(26, 10, 686, 536);  contentPane.add(scrollPane);  btn\_showDS = new JButton("Hiển thị danh sách khóa học");  btn\_showDS.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  KhoaHocDAO dao = new KhoaHocDAO();  List<KhoaHoc> dsKhoaHoc = dao.getAllKhoaHoc();  DefaultTableModel model = new DefaultTableModel();  table.setDefaultEditor(Object.class, null);  model.setColumnIdentifiers(  new String[] { "Mã Khóa Học", "Tên Khóa Học", "Mã Giảng Viên", "Số Tín Chỉ", "Học Phí" });  table.setModel(model);  model.setRowCount(0);  for (KhoaHoc kh : dsKhoaHoc) {  model.addRow(new Object[] { kh.getMaKhoaHoc(), kh.getTenKhoaHoc(), kh.getMaGiangVien(),  kh.getSoTinchi(), kh.getHocPhi(), });  }  }  });  btn\_showDS.setBounds(727, 20, 195, 40);  contentPane.add(btn\_showDS);  btn\_TimKiem = new JButton("Tìm kiếm khóa học");  btn\_TimKiem.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String maKhoaHoc = JOptionPane.showInputDialog(null, "Nhập mã khóa học:");  if (maKhoaHoc != null) {  maKhoaHoc = maKhoaHoc.toUpperCase();  if (maKhoaHoc.trim().isEmpty()) {  JOptionPane.showMessageDialog(null, "Bạn chưa nhập mã khóa học");  } else {  KhoaHocDAO dao = new KhoaHocDAO();  KhoaHoc kh = dao.timKiemKhoaHoc(maKhoaHoc);  DefaultTableModel model = (DefaultTableModel) table.getModel();  table.setDefaultEditor(Object.class, null);  model.setColumnIdentifiers(new String[] { "Mã Khóa Học", "Tên Khóa Học", "Mã Giảng Viên",  "Số Tín Chỉ", "Học Phí" });  model.setRowCount(0);  if (kh != null) {  model.addRow(new Object[] { kh.getMaKhoaHoc(), kh.getTenKhoaHoc(), kh.getMaGiangVien(),  kh.getSoTinchi(), kh.getHocPhi(), });  } else {  JOptionPane.showMessageDialog(null, "Không tìm thấy khóa học!");  }  }  }  }  });  btn\_TimKiem.setBounds(727, 70, 195, 40);  contentPane.add(btn\_TimKiem);  btn\_Xoa = new JButton("Xóa khóa học");  btn\_Xoa.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  int selectedRow = table.getSelectedRow();  if (selectedRow != -1) {  String maKhoaHoc = table.getValueAt(selectedRow, 0).toString(); // cột 0 là mã học viên  int confirm = JOptionPane.showConfirmDialog(null,  "Bạn có chắc muốn xóa khóa học " + maKhoaHoc + "?", "Xác nhận", JOptionPane.YES\_NO\_OPTION);  if (confirm == JOptionPane.YES\_OPTION) {  KhoaHocDAO dao = new KhoaHocDAO();  boolean success = dao.xoaKhoaHoc(maKhoaHoc);  if (success) {  JOptionPane.showMessageDialog(null, "Xóa thành công. Vui lòng Reload lại để cập nhật!");    } else {  JOptionPane.showMessageDialog(null, "Xóa thất bại hoặc khóa học không tồn tại!");  }  }  } else {  JOptionPane.showMessageDialog(null, "Vui lòng chọn khóa học cần xóa!");  }  }  });  btn\_Xoa.setBounds(727, 120, 195, 40);  contentPane.add(btn\_Xoa);  btn\_Them = new JButton("Thêm khóa học");  btn\_Them.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  ThemKhoaHocGUI gui = new ThemKhoaHocGUI();  gui.setVisible(true);  }  });  btn\_Them.setBounds(727, 170, 195, 40);  contentPane.add(btn\_Them);  btn\_fixInfo = new JButton("Sửa khóa học");  btn\_fixInfo.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  int selectedRow = table.getSelectedRow();  if (selectedRow == -1) {  JOptionPane.showMessageDialog(null, "Vui lòng chọn Khóa học để chỉnh sửa");  return;  }  String maKhoaHoc = table.getValueAt(selectedRow, 0).toString();  String tenKhoaHoc = table.getValueAt(selectedRow, 1).toString();  String MaGiangVien = table.getValueAt(selectedRow, 2).toString();  String soTinChi = table.getValueAt(selectedRow, 3).toString();  String hocPhi = table.getValueAt(selectedRow, 4).toString();  SuaKhoaHocGUI dialog = new SuaKhoaHocGUI(maKhoaHoc, tenKhoaHoc, MaGiangVien, soTinChi, hocPhi);  dialog.setVisible(true);  }  });  btn\_fixInfo.setBounds(727, 220, 195, 40);  contentPane.add(btn\_fixInfo);  btn\_Back = new JButton("Quay lại");  btn\_Back.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  new AdminGUI().setVisible(true);  setVisible(false);  }  });  btn\_Back.setBounds(727, 491, 195, 40);  contentPane.add(btn\_Back);  JButton btn\_fixInfo\_1 = new JButton("Đăng ký cho học viên");  btn\_fixInfo\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  DangKyDAO dao = new DangKyDAO();  String maHocVien = JOptionPane.showInputDialog(null, "Nhập mã học viên:");  if (maHocVien != null) {  maHocVien = maHocVien.toUpperCase();  if (maHocVien.trim().isEmpty()) {  JOptionPane.showMessageDialog(null, "Bạn chưa nhập mã học viên");  } else if (DBCheckExist.isMaTonTai("HocVien", "maHocVien", maHocVien)) {  String maKhoaHoc = JOptionPane.showInputDialog(null, "Nhập mã khóa học:");  if (maKhoaHoc != null) {  maKhoaHoc = maKhoaHoc.toUpperCase();  if (maKhoaHoc.trim().isEmpty()) {  JOptionPane.showMessageDialog(null, "Bạn chưa nhập mã khóa học");  } else if (DBCheckExist.isMaTonTai("KhoaHoc", "maKhoaHoc", maKhoaHoc)) {  dao.dangKyKhoaHoc(maHocVien, maKhoaHoc);    } else {  JOptionPane.showMessageDialog(null, "Mã khóa học không tồn tại", "Lỗi",  JOptionPane.ERROR\_MESSAGE);  }  }  } else {  JOptionPane.showMessageDialog(null, "Mã học viên không tồn tại", "Lỗi",  JOptionPane.ERROR\_MESSAGE);  }  }    }  });  btn\_fixInfo\_1.setBounds(727, 270, 195, 40);  contentPane.add(btn\_fixInfo\_1);  setLocationRelativeTo(null);  setVisible(true);  }  public static void main(String[] args) {  SwingUtilities.invokeLater(() -> new KhoaHocGUI());  }  } |

### **LoginGUI**

|  |
| --- |
| package gui;  import java.awt.Font;  import javax.swing.\*;  import dao.TaiKhoanDAO;  import giangVienGUI.LoginGiangVienGUI;  import hocVienGUI.LoginHocVienGUI;  import model.TaiKhoan;  import java.awt.event.MouseAdapter;  import java.awt.event.MouseEvent;  import java.awt.event.ActionListener;  import java.awt.event.ActionEvent;  public class LoginGUI extends JFrame {  public LoginGUI() {  setTitle("Đăng nhập");    setSize(800, 500);  setDefaultCloseOperation( EXIT\_ON\_CLOSE);  getContentPane().setLayout(null);  // Label tiêu đề  JLabel titleLabel = new JLabel("Đăng nhập hệ thống", SwingConstants.CENTER);  titleLabel.setFont(new Font("Arial", Font.BOLD, 18));  titleLabel.setBounds(10, 54, 764, 25);  getContentPane().add(titleLabel);  // Label User  JLabel userLabel = new JLabel("Tài khoản:");  userLabel.setBounds(207, 121, 80, 25);  getContentPane().add(userLabel);  // TextField User  JTextField userText = new JTextField();  userText.setBounds(297, 121, 200, 25);  getContentPane().add(userText);  // Label Password  JLabel passLabel = new JLabel("Mật khẩu:");  passLabel.setBounds(207, 161, 80, 25);  getContentPane().add(passLabel);  // PasswordField  JPasswordField passText = new JPasswordField();  passText.setBounds(297, 161, 200, 25);  getContentPane().add(passText);  // Button Đăng nhập  JButton btnLogin = new JButton("Đăng nhập");  btnLogin.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String username = userText.getText().trim();  String password = new String(passText.getPassword()).trim();  TaiKhoanDAO dao = new TaiKhoanDAO();  TaiKhoan tk = dao.kiemTraDangNhap(username, password);  if (tk != null) {  // Điều hướng tới màn hình tương ứng  switch (tk.getRole()) {  case "admin":  // Mở giao diện admin  new AdminGUI().setVisible(true);  dispose();  break;  case "giangvien":  // Mở giao diện giảng viên  new LoginGiangVienGUI(tk.getMaTaiKhoan()).setVisible(true);  dispose();  break;  case "hocvien":  new LoginHocVienGUI(tk.getMaTaiKhoan()).setVisible(true);  dispose();  break;  }  } else {  JOptionPane.showMessageDialog(LoginGUI.this,  "Sai tên đăng nhập hoặc mật khẩu", "Lỗi", JOptionPane.ERROR\_MESSAGE);  }    }  });  btnLogin.setBounds(342, 196, 100, 30);  getContentPane().add(btnLogin);  // Hiển thị frame  setLocationRelativeTo(null); // Căn giữa màn hình    }  public static void main(String[] args) {  SwingUtilities.invokeLater(() -> new LoginGUI().setVisible(true));  }  } |

### **SuaHocVienGUI**

|  |
| --- |
| package gui;  import javax.swing.\*;  import dao.HocVienDAO;  import java.awt.\*;  public class SuaHocVienGUI extends JFrame {  private JTextField tfMaHV, tfTenHV, tfSdt;  private JButton btnXacNhan, btnHuy;  public SuaHocVienGUI(String maHV, String tenHV, String sdt) {  setTitle("Chỉnh sửa học viên");  setLayout(new GridLayout(4, 2, 5, 5));  setSize(400, 200);  setLocationRelativeTo(null);  setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE); // chỉ đóng cửa sổ này  add(new JLabel("Mã học viên:"));  tfMaHV = new JTextField(maHV != null ? maHV : "");  tfMaHV.setEditable(false);  add(tfMaHV);  add(new JLabel("Tên học viên:"));  tfTenHV = new JTextField(tenHV != null ? tenHV : "");  add(tfTenHV);  add(new JLabel("Số điện thoại:"));  tfSdt = new JTextField(sdt != null ? sdt : "");  add(tfSdt);  btnXacNhan = new JButton("Cập nhật");  btnHuy = new JButton("Hủy");  add(btnXacNhan);  add(btnHuy);  btnXacNhan.addActionListener(e -> {  String newTen = tfTenHV.getText().trim();  String newSdt = tfSdt.getText().trim();    HocVienDAO dao = new HocVienDAO();  boolean success = dao.capNhat(tfMaHV.getText(), newTen, newSdt);  if (success) {  JOptionPane.showMessageDialog(this, "Cập nhật thành công. Vui lòng Reload lại để cập nhật!");  dispose();  } else {  JOptionPane.showMessageDialog(this, "Cập nhật thất bại!");  }  dispose();  });  btnHuy.addActionListener(e -> dispose());  }  } |

### **SuaKhoaHocGUI**

|  |
| --- |
| package gui;  import javax.swing.\*;  import dao.HocVienDAO;  import dao.KhoaHocDAO;  import java.awt.\*;  public class SuaKhoaHocGUI extends JFrame {  private JTextField tftMaKhoaHoc, tftTenKhoaHoc, tftMaGiangVien, tftSoTinChi, tftHocPhi;  private JButton btnXacNhan, btnHuy;  public SuaKhoaHocGUI(String maKhoaHoc, String tenKhoaHoc, String maGiangVien, String soTinChi, String hocPhi) {  setTitle("Chỉnh sửa học viên");    getContentPane().setLayout(null);  setSize(320, 365);  setLocationRelativeTo(null);  setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE); // chỉ đóng cửa sổ này  JLabel label = new JLabel("Mã khóa học:");  label.setBounds(10, 10, 115, 30);  getContentPane().add(label);  tftMaKhoaHoc = new JTextField(maKhoaHoc != null ? maKhoaHoc : "");  tftMaKhoaHoc.setBounds(105, 10, 187, 30);  tftMaKhoaHoc.setEditable(false);  getContentPane().add(tftMaKhoaHoc);  JLabel label\_1 = new JLabel("Tên khóa học:");  label\_1.setBounds(10, 50, 115, 30);  getContentPane().add(label\_1);  tftTenKhoaHoc = new JTextField(tenKhoaHoc != null ? tenKhoaHoc : "");  tftTenKhoaHoc.setLocation(105, 49);  tftTenKhoaHoc.setSize(187, 30);  getContentPane().add(tftTenKhoaHoc);  JLabel label\_2 = new JLabel("Mã giảng viên:");  label\_2.setLocation(10, 90);  label\_2.setSize(115, 30);  getContentPane().add(label\_2);  tftMaGiangVien = new JTextField(maGiangVien != null ? maGiangVien : "");  tftMaGiangVien.setLocation(105, 90);  tftMaGiangVien.setSize(187, 30);  getContentPane().add(tftMaGiangVien);    JLabel label\_3 = new JLabel("Số tín chỉ:");  label\_3.setLocation(10, 130);  label\_3.setSize(115, 30);  getContentPane().add(label\_3);  tftSoTinChi = new JTextField(soTinChi != null ? soTinChi : "");  tftSoTinChi.setLocation(105, 130);  tftSoTinChi.setSize(187, 30);  getContentPane().add(tftSoTinChi);    JLabel label\_4 = new JLabel("Học phí:");  label\_4.setLocation(10, 170);  label\_4.setSize(115, 30);  getContentPane().add(label\_4);  tftHocPhi = new JTextField(hocPhi != null ? hocPhi : "");  tftHocPhi.setLocation(105, 170);  tftHocPhi.setSize(187, 30);  getContentPane().add(tftHocPhi);  btnXacNhan = new JButton("Cập nhật");  btnXacNhan.setLocation(37, 210);  btnXacNhan.setSize(150, 25);  btnHuy = new JButton("Hủy");  btnHuy.setLocation(197, 210);  btnHuy.setSize(71, 25);  getContentPane().add(btnXacNhan);  getContentPane().add(btnHuy);  btnXacNhan.addActionListener(e -> {  String newTenKhoaHoc = tftTenKhoaHoc.getText().trim();  String newMaGiangVien = tftMaGiangVien.getText().trim();  String newSoTinChi = tftSoTinChi.getText().trim();  String newHocPhi = tftHocPhi.getText().trim();    KhoaHocDAO dao = new KhoaHocDAO();  boolean success = dao.capNhat(tftMaKhoaHoc.getText().toUpperCase(), newTenKhoaHoc.toUpperCase(), newMaGiangVien.toUpperCase(), Integer.parseInt(newSoTinChi), Double.parseDouble(newHocPhi));  if (success) {  JOptionPane.showMessageDialog(this, "Cập nhật thành công. Vui lòng Reload lại để cập nhật!");  dispose();  } else {  JOptionPane.showMessageDialog(this, "Cập nhật thất bại!");  }  dispose();  });  btnHuy.addActionListener(e -> dispose());  }  } |

### **TaiKhoanGUI**

|  |
| --- |
| package gui;  import javax.swing.\*;  import dao.GiangVienDAO;  import dao.HocVienDAO;  import dao.TaiKhoanDAO;  import model.GiangVien;  import model.HocVien;  import model.TaiKhoan;  import model.User;  import java.awt.\*;  import java.awt.event.MouseAdapter;  import java.awt.event.MouseEvent;  import java.awt.event.ActionListener;  import java.awt.event.ActionEvent;  public class TaiKhoanGUI extends JFrame {  private static final long serialVersionUID = 1L;  private JTextField tfUsername, tfPassword, tfMaTaiKhoan, tfMaNguoiDung, tfName, tfPhone, tfSalary;  private JLabel lblSalary;  private JComboBox<String> comboBoxRole;  public TaiKhoanGUI() {  setTitle("Admin Panel");  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setSize(800, 500);  setMinimumSize(new Dimension(800, 500));  setLocationRelativeTo(null); // Căn giữa màn hình  setResizable(false);  // Panel chính dùng BoxLayout dọc để gói nội dung vào giữa  JPanel wrapperPanel = new JPanel();  wrapperPanel.setLayout(new BoxLayout(wrapperPanel, BoxLayout.Y\_AXIS));  wrapperPanel.setBorder(BorderFactory.createEmptyBorder(20, 50, 20, 50));  // Panel chứa các dòng nhập liệu, dùng GridBagLayout để cố định vị trí  JPanel formPanel = new JPanel(new GridBagLayout());  formPanel.setAlignmentX(Component.CENTER\_ALIGNMENT);  GridBagConstraints gbc = new GridBagConstraints();  gbc.insets = new Insets(8, 8, 8, 8);  gbc.anchor = GridBagConstraints.WEST;  int y = 0;  addRow(formPanel, gbc, y++, "Chọn vai trò:", comboBoxRole = new JComboBox<>(new String[]{"Học viên", "Giảng viên"}));  addRow(formPanel, gbc, y++, "Tên đăng nhập:", tfUsername = new JTextField(20));  addRow(formPanel, gbc, y++, "Mật khẩu:", tfPassword = new JTextField(20));  addRow(formPanel, gbc, y++, "Mã tài khoản:", tfMaTaiKhoan = new JTextField(20));  addRow(formPanel, gbc, y++, "Mã người dùng:", tfMaNguoiDung = new JTextField(20));  addRow(formPanel, gbc, y++, "Họ tên:", tfName = new JTextField(20));  addRow(formPanel, gbc, y++, "Số điện thoại:", tfPhone = new JTextField(20));  lblSalary = new JLabel("Lương:");  tfSalary = new JTextField(20);  addRow(formPanel, gbc, y++, lblSalary, tfSalary);  lblSalary.setVisible(false);  tfSalary.setVisible(false);  // Nút lưu  JButton btnSave = new JButton("Thêm tài khoản");  JPanel buttonPanel = new JPanel();  buttonPanel.setAlignmentX(Component.CENTER\_ALIGNMENT);  buttonPanel.add(btnSave);  wrapperPanel.add(formPanel);  wrapperPanel.add(Box.createRigidArea(new Dimension(0, 10)));  wrapperPanel.add(buttonPanel);    JButton btnNewButton = new JButton("Quay lại");  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  new AdminGUI().setVisible(true);  setVisible(false);  }  });  buttonPanel.add(btnNewButton);        getContentPane().add(wrapperPanel);  // Xử lý sự kiện chọn vai trò  comboBoxRole.addActionListener(e -> {  boolean isTeacher = comboBoxRole.getSelectedItem().equals("Giảng viên");  lblSalary.setVisible(isTeacher);  tfSalary.setVisible(isTeacher);  });  // Xử lý nút lưu  btnSave.addActionListener(e -> {  String role = (String) comboBoxRole.getSelectedItem();  String username = tfUsername.getText().trim();  String password = tfPassword.getText().trim();  String maTaiKhoan = tfMaTaiKhoan.getText().trim().toUpperCase();  String maNguoiDung = tfMaNguoiDung.getText().trim().toUpperCase();  String name = tfName.getText().trim();  String phone = tfPhone.getText().trim();  String salary = tfSalary.getText().trim();  if (username.isEmpty() || password.isEmpty() || maTaiKhoan.isEmpty() ||maNguoiDung.isEmpty() || name.isEmpty() || phone.isEmpty()) {  JOptionPane.showMessageDialog(this, "Vui lòng nhập đầy đủ thông tin!", "Thiếu thông tin", JOptionPane.WARNING\_MESSAGE);  return;  }  if (!phone.matches("\\d{10}")) {  JOptionPane.showMessageDialog(this, "Số điện thoại phải là 10 chữ số!", "Sai định dạng", JOptionPane.ERROR\_MESSAGE);  return;  }  if (role.equals("Giảng viên")) {  if (salary.isEmpty() || !salary.matches("\\d+")) {  JOptionPane.showMessageDialog(this, "Lương phải là số nguyên dương!", "Sai định dạng", JOptionPane.ERROR\_MESSAGE);  return;  }  }        if (role.equals("Giảng viên")){  role = "giangvien";  TaiKhoan tk = new TaiKhoan(username, password, role,maTaiKhoan);  Boolean check = taoTaiKhoan(tk);  if(check) {  GiangVien gv = new GiangVien(username, password, role, maTaiKhoan, phone, name, maNguoiDung, Double.parseDouble(salary));  taoGiangVien(gv);  }      }else {  role = "hocvien";  TaiKhoan tk = new TaiKhoan(username, password, role,maTaiKhoan);  Boolean check = taoTaiKhoan(tk);  if (check) {  HocVien hv = new HocVien(username, password, role, maTaiKhoan , maNguoiDung, phone, name);  taoHocVien(hv);  }    }      });  }  private void taoHocVien(HocVien hv) {  HocVienDAO dao = new HocVienDAO();    boolean success = dao.taoHocVien(hv);  if (success) {  System.out.println("Tạo học viên thành công!");  JOptionPane.showMessageDialog(this, "Tạo học viên thành công!", "Thành công", JOptionPane.INFORMATION\_MESSAGE);  } else {  System.out.println("Tạo học viên thất bại.");  }    }  private void taoGiangVien(GiangVien gv) {  GiangVienDAO dao = new GiangVienDAO();    boolean success = dao.taoGiangVien(gv);  if (success) {  System.out.println("Tạo giảng viên thành công!");  JOptionPane.showMessageDialog(this, "Tạo giảng viên thành công!", "Thành công", JOptionPane.INFORMATION\_MESSAGE);  } else {  System.out.println("Tạo giảng viên thất bại.");  }    }  private void addRow(JPanel panel, GridBagConstraints gbc, int y, String labelText, JComponent inputField) {  addRow(panel, gbc, y, new JLabel(labelText), inputField);  }  private void addRow(JPanel panel, GridBagConstraints gbc, int y, JLabel label, JComponent inputField) {  gbc.gridx = 0;  gbc.gridy = y;  gbc.fill = GridBagConstraints.NONE;  label.setPreferredSize(new Dimension(120, 25));  panel.add(label, gbc);  gbc.gridx = 1;  gbc.fill = GridBagConstraints.HORIZONTAL;  inputField.setPreferredSize(new Dimension(300, 25));  panel.add(inputField, gbc);  }  public boolean taoTaiKhoan(TaiKhoan tk) {  TaiKhoanDAO dao = new TaiKhoanDAO();    boolean success = dao.taoTaiKhoan(tk);  if (success) {  System.out.println("Tạo tài khoản thành công!");  JOptionPane.showMessageDialog(this, "Tạo tài khoản thành công!", "Thành công", JOptionPane.INFORMATION\_MESSAGE);  } else {  System.out.println("Tạo tài khoản thất bại.");  }  return success;  }  public static void main(String[] args) {  SwingUtilities.invokeLater(() -> new TaiKhoanGUI().setVisible(true));  }  } |

### ThemKhoaHocGUI

|  |
| --- |
| package gui;  import javax.swing.\*;  import dao.HocVienDAO;  import dao.KhoaHocDAO;  import model.KhoaHoc;  import java.awt.\*;  public class ThemKhoaHocGUI extends JFrame {  private JTextField tftMaKhoaHoc, tftTenKhoaHoc, tftMaGiangVien, tftSoTinChi, tftHocPhi;  private JButton btnXacNhan, btnHuy;  public ThemKhoaHocGUI() {  setTitle("Thêm khóa học");  getContentPane().setLayout(null);  setSize(320, 365);  setLocationRelativeTo(null);  setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);  JLabel label = new JLabel("Mã khóa học:");  label.setBounds(10, 10, 115, 30);  getContentPane().add(label);  tftMaKhoaHoc = new JTextField();  tftMaKhoaHoc.setBounds(105, 10, 187, 30);  getContentPane().add(tftMaKhoaHoc);  JLabel label\_1 = new JLabel("Tên khóa học:");  label\_1.setBounds(10, 50, 115, 30);  getContentPane().add(label\_1);  tftTenKhoaHoc = new JTextField();  tftTenKhoaHoc.setLocation(105, 49);  tftTenKhoaHoc.setSize(187, 30);  getContentPane().add(tftTenKhoaHoc);  JLabel label\_2 = new JLabel("Mã giảng viên:");  label\_2.setLocation(10, 90);  label\_2.setSize(115, 30);  getContentPane().add(label\_2);  tftMaGiangVien = new JTextField();  tftMaGiangVien.setLocation(105, 90);  tftMaGiangVien.setSize(187, 30);  getContentPane().add(tftMaGiangVien);    JLabel label\_3 = new JLabel("Số tín chỉ:");  label\_3.setLocation(10, 130);  label\_3.setSize(115, 30);  getContentPane().add(label\_3);  tftSoTinChi = new JTextField();  tftSoTinChi.setLocation(105, 130);  tftSoTinChi.setSize(187, 30);  getContentPane().add(tftSoTinChi);    JLabel label\_4 = new JLabel("Học phí:");  label\_4.setLocation(10, 170);  label\_4.setSize(115, 30);  getContentPane().add(label\_4);  tftHocPhi = new JTextField();  tftHocPhi.setLocation(105, 170);  tftHocPhi.setSize(187, 30);  getContentPane().add(tftHocPhi);  btnXacNhan = new JButton("Thêm khóa học");  btnXacNhan.setLocation(37, 210);  btnXacNhan.setSize(150, 25);  btnHuy = new JButton("Hủy");  btnHuy.setLocation(197, 210);  btnHuy.setSize(71, 25);  getContentPane().add(btnXacNhan);  getContentPane().add(btnHuy);  btnXacNhan.addActionListener(e -> {  String newMaKhoaHoc = tftMaKhoaHoc.getText().trim();  String newTenKhoaHoc = tftTenKhoaHoc.getText().trim();  String newMaGiangVien = tftMaGiangVien.getText().trim();  String newSoTinChi = tftSoTinChi.getText().trim();  String newHocPhi = tftHocPhi.getText().trim();    KhoaHocDAO dao = new KhoaHocDAO();  KhoaHoc kh = new KhoaHoc(newMaKhoaHoc.toUpperCase(), newTenKhoaHoc, newMaGiangVien.toUpperCase(), Integer.parseInt(newSoTinChi), Double.parseDouble(newHocPhi));  boolean success = dao.taoKhoaHoc(kh);  if (success) {  JOptionPane.showMessageDialog(this, "Thêm khóa học thành công. Vui lòng Reload lại để cập nhật!");  dispose();  } else {  JOptionPane.showMessageDialog(this, "Thêm khóa học thất bại!");  }  dispose();  });  btnHuy.addActionListener(e -> dispose());  }  } |

## **Package hocVienGUI**

### **5.1 LoginHocVienGUI**

|  |
| --- |
| package hocVienGUI;  import java.awt.EventQueue;  import java.awt.event.ActionListener;  import java.text.SimpleDateFormat;  import java.util.Date;  import java.util.List;  import javax.swing.JButton;  import javax.swing.JFrame;  import javax.swing.JOptionPane;  import javax.swing.JPanel;  import javax.swing.JScrollPane;  import javax.swing.JTable;  import javax.swing.SwingUtilities;  import javax.swing.border.EmptyBorder;  import javax.swing.table.DefaultTableModel;  import dao.DangKyDAO;  import dao.DiemDanhDAO;  import dao.DiemSoDAO;  import dao.HocVienDAO;  import dao.KhoaHocDAO;  import gui.LoginGUI;  import model.DiemSo;  import model.HocVien;  import model.KhoaHoc;  import java.awt.event.MouseAdapter;  import java.awt.event.MouseEvent;  import java.awt.event.ActionEvent;  public class LoginHocVienGUI extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  private JTable table;  private JScrollPane scrollPane;  private JButton btn\_showDS;  private JButton btn\_TimKiem;  private JButton btn\_Them;  private JButton btn\_fixInfo;  private JButton btn\_Back;  /\*\*  \* /\*\* Create the frame.  \*/  public LoginHocVienGUI(String maTaiKhoan) {  HocVienDAO dao = new HocVienDAO();  HocVien hv = dao.getThongTinHocVien(maTaiKhoan);  String maHocVien = dao.getMaHocVIenByTaiKhoan(maTaiKhoan);  setTitle("Học viên " + hv.getTen());  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setResizable(false);  setSize(1000, 600);  contentPane = new JPanel();  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);  table = new JTable();  scrollPane = new JScrollPane(table);  scrollPane.setBounds(26, 10, 686, 536);  contentPane.add(scrollPane);    table.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  if (e.getClickCount() == 2 && table.getSelectedRow() != -1) {  int selectedRow = table.getSelectedRow();  String maKhoaHoc = table.getValueAt(selectedRow, 0).toString();  String tenKhoaHoc = table.getValueAt(selectedRow, 1).toString();    int luaChon = JOptionPane.showConfirmDialog(  null,  "Đăng ký khóa học: " + tenKhoaHoc ,  "Xác nhận đăng ký",  JOptionPane.YES\_NO\_OPTION  );    if (luaChon == JOptionPane.YES\_OPTION) {  DangKyDAO dangKyDAO = new DangKyDAO();  dangKyDAO.dangKyKhoaHoc(maHocVien, maKhoaHoc);  DefaultTableModel model = (DefaultTableModel) table.getModel();  model.removeRow(selectedRow);    }else {  JOptionPane.showMessageDialog(null,"Đăng ký không thành công", "Lỗi", JOptionPane.ERROR\_MESSAGE);  }    }  }  });  btn\_showDS = new JButton("Thông tin cá nhân");  btn\_showDS.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  HocVienDAO dao = new HocVienDAO();  HocVien hv = dao.getThongTinHocVien(maTaiKhoan);  DefaultTableModel model = new DefaultTableModel();  model.setColumnIdentifiers(new String[] { "Mã Học Viên", "Tên Học Viên", "SĐT" });  table.setModel(model);  model.setRowCount(0);  if (hv != null) {  model.addRow(new Object[] { hv.getMaHocVien(), hv.getTen(), hv.getSoDienThoai(), });  }  }  });  btn\_showDS.setBounds(727, 10, 195, 40);  contentPane.add(btn\_showDS);  btn\_TimKiem = new JButton("Điểm số");  btn\_TimKiem.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  DiemSoDAO dao = new DiemSoDAO();  List<DiemSo> danhSachDiem = dao.getDiemTheoTaiKhoan(maTaiKhoan);  DefaultTableModel model = (DefaultTableModel) table.getModel();  model.setColumnIdentifiers(new String[] { "Mã Khóa Học", "Điểm" });  model.setRowCount(0); // Xóa bảng cũ  for (DiemSo d : danhSachDiem) {  model.addRow(new Object[] { d.getMaKhoaHoc(), d.getDiem() });  }  }  });  btn\_TimKiem.setBounds(727, 70, 195, 40);  contentPane.add(btn\_TimKiem);  btn\_Back = new JButton("Đăng xuất");  btn\_Back.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  new LoginGUI().setVisible(true);  setVisible(false);  }  });  btn\_Back.setBounds(727, 491, 195, 40);  contentPane.add(btn\_Back);  JButton btn\_TimKiem\_1 = new JButton("Đăng ký khóa học mới");  btn\_TimKiem\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  DefaultTableModel model = (DefaultTableModel) table.getModel();  table.setDefaultEditor(Object.class, null);  HocVienDAO dao = new HocVienDAO();  String maHocVien = dao.getMaHocVIenByTaiKhoan(maTaiKhoan);  KhoaHocDAO khoaHocDAO = new KhoaHocDAO();  List<KhoaHoc> danhSachChuaDangKy = khoaHocDAO.layKhoaHocChuaDangKy(maHocVien);  if (danhSachChuaDangKy.isEmpty()) {  model.setRowCount(0);  JOptionPane.showMessageDialog(null, "Học viên không còn khóa học nào để đăng ký!", "Thông báo",  JOptionPane.INFORMATION\_MESSAGE);    } else {  // Gán vào bảng:    model.setRowCount(0);  model.setColumnIdentifiers(new String[] { "Mã Khóa Học", "Tên Khóa Học", "Số Tín Chỉ", "Học Phí" });  for (KhoaHoc kh : danhSachChuaDangKy) {  model.addRow(new Object[] { kh.getMaKhoaHoc(), kh.getTenKhoaHoc(), kh.getSoTinchi(),  kh.getHocPhi() });  }  }      }  });  btn\_TimKiem\_1.setBounds(727, 129, 195, 40);  contentPane.add(btn\_TimKiem\_1);  setLocationRelativeTo(null);  setVisible(true);  }  public void addShowListListener(ActionListener listener) {  btn\_showDS.addActionListener(listener);  }  public void addEditListener(ActionListener listener) {  btn\_fixInfo.addActionListener(listener);  }  public void displayHocVienList(Object[][] data, String[] columnNames) {  DefaultTableModel model = new DefaultTableModel(data, columnNames) {  @Override  public boolean isCellEditable(int row, int column) {  return false;  }  };  table.setModel(model);  // Tùy chỉnh hiển thị  table.getColumnModel().getColumn(0).setPreferredWidth(80);  table.getColumnModel().getColumn(1).setPreferredWidth(150);  table.getColumnModel().getColumn(2).setPreferredWidth(100);  table.getColumnModel().getColumn(3).setPreferredWidth(200);  table.setRowHeight(25);  }  public String getSelectedHocVien() {  int selectedRow = table.getSelectedRow();  if (selectedRow >= 0) {  return table.getValueAt(selectedRow, 0).toString();  }  return null;  }  public Object[] getSelectedHocVienDetails() {  int selectedRow = table.getSelectedRow();  if (selectedRow >= 0) {  return new Object[] { table.getValueAt(selectedRow, 0), table.getValueAt(selectedRow, 1),  table.getValueAt(selectedRow, 2), table.getValueAt(selectedRow, 3) };  }  return null;  }  public void showMessage(String message, String title, int messageType) {  JOptionPane.showMessageDialog(this, message, title, messageType);  }    public int showConfirmDialog(String message, String title) {  Object[] options = { "Xóa", "Hủy" };  return JOptionPane.showOptionDialog(this, message, title, JOptionPane.DEFAULT\_OPTION,  JOptionPane.WARNING\_MESSAGE, null, options, options[1] );  }  public void loadDataToTable(List<HocVien> dsHocVien) {  DefaultTableModel model = (DefaultTableModel) table.getModel();  table.setDefaultEditor(Object.class, null);  model.setRowCount(0);  for (HocVien hv : dsHocVien) {  model.addRow(new Object[] { hv.getMaHocVien(), hv.getTen(), hv.getSoDienThoai() });  }  }  // Nut xu ly them hoc vien  public void addInsertListener(ActionListener listener) {  btn\_Them.addActionListener(listener);  }  public void addSearchListener(ActionListener listener) {  btn\_TimKiem.addActionListener(listener);  }  public void addBackListener(ActionListener listener) {  btn\_Back.addActionListener(listener); // Gắn ActionListener cho btn\_Back  }  } |

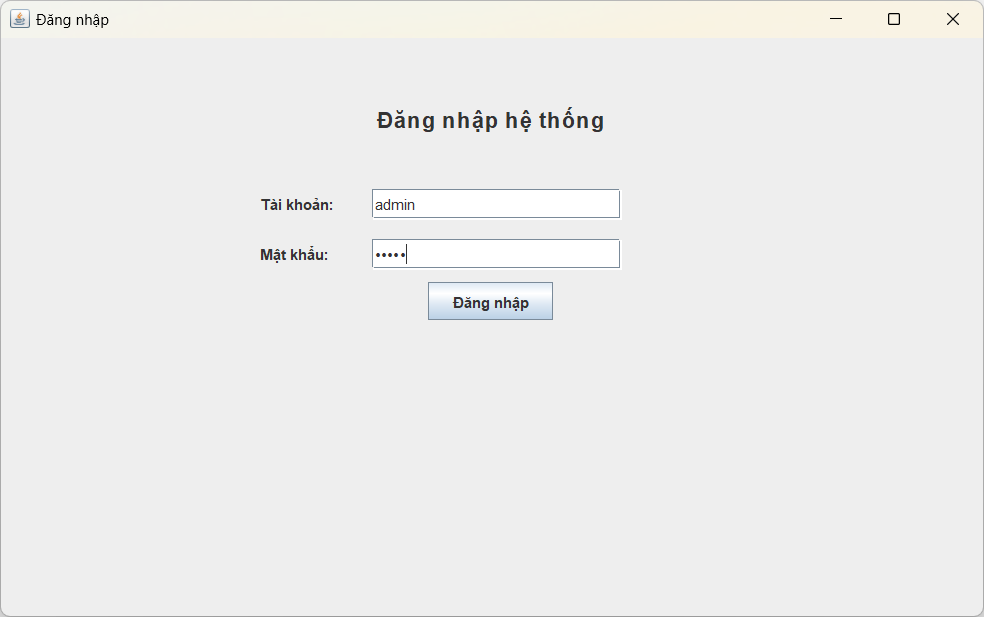
## **Package giangVienGUI**

### **6.1 LoginGiangVienGUI**

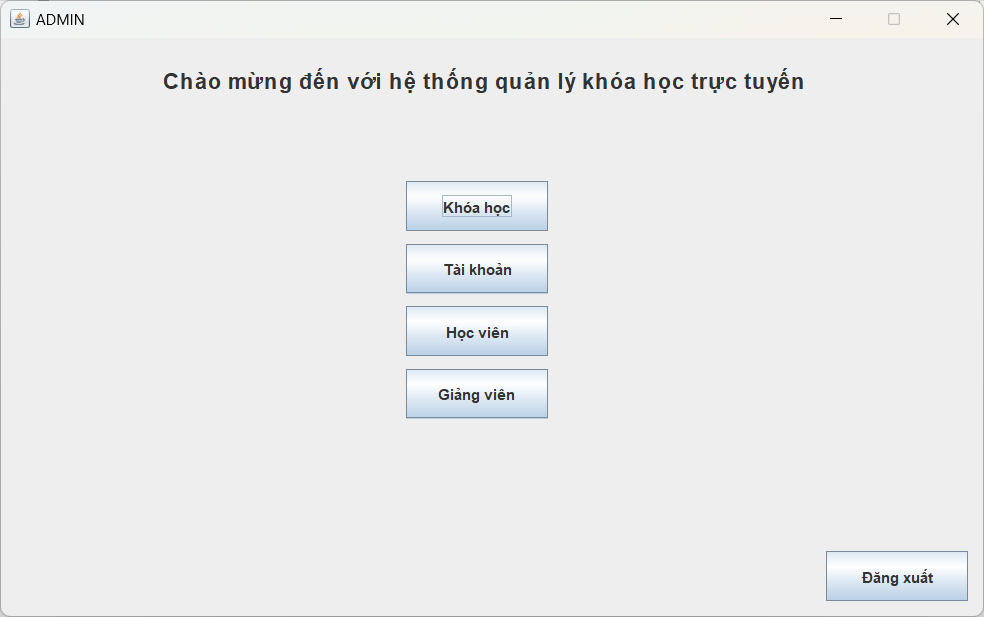
|  |
| --- |
| package giangVienGUI;  import java.awt.BorderLayout;  import java.awt.Dimension;  import java.awt.EventQueue;  import javax.swing.BorderFactory;  import javax.swing.Box;  import javax.swing.BoxLayout;  import javax.swing.JButton;  import javax.swing.JFrame;  import javax.swing.JOptionPane;  import javax.swing.JPanel;  import javax.swing.JScrollPane;  import javax.swing.JTable;  import javax.swing.SwingUtilities;  import javax.swing.border.EmptyBorder;  import javax.swing.table.DefaultTableModel;  import dao.DBCheckExist;  import dao.DiemDanhDAO;  import dao.DiemSoDAO;  import dao.GiangVienDAO;  import dao.HocVienDAO;  import dao.KhoaHocDAO;  import gui.LoginGUI;  import model.DiemDanh;  import model.DiemSo;  import model.GiangVien;  import model.HocVien;  import model.KhoaHoc;  import java.awt.Component;  import java.awt.event.ActionListener;  import java.awt.event.MouseAdapter;  import java.awt.event.MouseEvent;  import java.text.SimpleDateFormat;  import java.util.ArrayList;  import java.util.Date;  import java.util.List;  import java.awt.event.ActionEvent;  public class LoginGiangVienGUI extends JFrame {  /\*\*  \* Create the frame.  \*/  private String tableMode = "";  public LoginGiangVienGUI(String maTaiKhoan) {  GiangVienDAO dao = new GiangVienDAO();  GiangVien gv = dao.getGiangVienByTaiKhoan(maTaiKhoan);  setTitle("Giảng viên " + gv.getTen());  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setSize(800, 500);  setMinimumSize(new Dimension(800, 500));  getContentPane().setLayout(new BorderLayout());  JTable table = new JTable(new DefaultTableModel());  JScrollPane scrollPane = new JScrollPane(table);  getContentPane().add(scrollPane, BorderLayout.CENTER);  table.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  if (e.getClickCount() == 2 && table.getSelectedRow() != -1) {  int selectedRow = table.getSelectedRow();  String maKhoaHoc = table.getValueAt(selectedRow, 0).toString();  String maHocVien = table.getValueAt(selectedRow, 1).toString();  String tenHocVien = table.getValueAt(selectedRow, 2).toString();  if (tableMode.equals("DIEM\_DANH")) {  SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");  String ngayHienTai = sdf.format(new Date());  int luaChon = JOptionPane.showConfirmDialog(  null,  "Học viên " + tenHocVien + " (" + maHocVien + ") có đi học không?",  "Xác nhận điểm danh",  JOptionPane.YES\_NO\_OPTION  );  if (luaChon != JOptionPane.CLOSED\_OPTION) {  boolean coMat = (luaChon == JOptionPane.YES\_OPTION);  table.setValueAt(ngayHienTai, selectedRow, 4);  table.setValueAt(coMat ? "Có" : "Vắng", selectedRow, 5);    DiemDanhDAO dao = new DiemDanhDAO();  dao.diemDanh(maHocVien, maKhoaHoc, ngayHienTai, coMat ? "Có" : "Vắng");  }  } else if (tableMode.equals("SUA\_DIEM")) {  String diemStr = JOptionPane.showInputDialog(  null,  "Nhập điểm cho học viên " + tenHocVien + " (" + maHocVien + "):",  "Nhập điểm",  JOptionPane.PLAIN\_MESSAGE  );  if (diemStr != null) {  try {  double diem = Double.parseDouble(diemStr);  table.setValueAt(diem, selectedRow, 3);    DiemSoDAO dao = new DiemSoDAO();  dao.diemSo(maHocVien, maKhoaHoc, diem);  } catch (NumberFormatException event) {  JOptionPane.showMessageDialog(null, "Vui lòng nhập số hợp lệ!", "Lỗi", JOptionPane.ERROR\_MESSAGE);  }  }  }  }  }  });  JPanel panel\_1 = new JPanel();  panel\_1.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));  getContentPane().add(panel\_1, BorderLayout.EAST);  panel\_1.setLayout(new BoxLayout(panel\_1, BoxLayout.Y\_AXIS));  JButton btnDSGV = new JButton("Thông tin cá nhân");  btnDSGV.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  GiangVienDAO dao = new GiangVienDAO();  GiangVien gv = dao.getGiangVienByTaiKhoan(maTaiKhoan);  DefaultTableModel model = new DefaultTableModel();  table.setDefaultEditor(Object.class, null);  model.setColumnIdentifiers(new String[] { "Mã Giảng Viên", "Tên Giảng Viên", "SĐT", "Lương" });  table.setModel(model);  model.setRowCount(0);  if (gv != null) {  model.addRow(  new Object[] { gv.getMaGiangVien(), gv.getTen(), gv.getSoDienThoai(), gv.getTienLuong(), });  }  }  });  btnDSGV.setMaximumSize(new Dimension(300, 30));  panel\_1.add(btnDSGV);  Component verticalStrut\_1 = Box.createVerticalStrut(10);  panel\_1.add(verticalStrut\_1);  JButton btnDSKHGVPT = new JButton("Danh sách khóa học phụ trách");  btnDSKHGVPT.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  KhoaHocDAO dao = new KhoaHocDAO();  List<KhoaHoc> danhSachKhoaHoc = dao.getKhoaHocByTaiKhoan(maTaiKhoan);  DefaultTableModel model = new DefaultTableModel();  table.setDefaultEditor(Object.class, null);  model.setColumnIdentifiers(  new String[] { "Mã Khóa Học", "Tên Khóa Học", "Mã Giảng Viên", "Số Tín Chỉ", "Học Phí" });  table.setModel(model);  model.setRowCount(0);  for (KhoaHoc kh : danhSachKhoaHoc) {  model.addRow(new Object[] { kh.getMaKhoaHoc(), kh.getTenKhoaHoc(), kh.getMaGiangVien(),  kh.getSoTinchi(), kh.getHocPhi(), });  }  }  });  btnDSKHGVPT.setMaximumSize(new Dimension(300, 30));  panel\_1.add(btnDSKHGVPT);  Component verticalStrut\_4\_1 = Box.createVerticalStrut(10);  panel\_1.add(verticalStrut\_4\_1);  JButton btnDSHV = new JButton("Danh sách học viên của giảng viên");  btnDSHV.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String maKhoaHoc = JOptionPane.showInputDialog(null, "Nhập mã khóa học:");  if (maKhoaHoc != null) {  maKhoaHoc = maKhoaHoc.toUpperCase();  if (maKhoaHoc.trim().isEmpty()) {  JOptionPane.showMessageDialog(null, "Bạn chưa nhập mã khóa học");  } else if (DBCheckExist.isMaTonTai("KhoaHoc", "maKhoaHoc", maKhoaHoc)) {  GiangVienDAO dao = new GiangVienDAO();  String maGiangVien = dao.getMaGiangVienByTaiKhoan(maTaiKhoan);  List<HocVien> dsHocVien = dao.getHocVienByKhoaHocAndGiangVien(maKhoaHoc, maGiangVien);  DefaultTableModel model = new DefaultTableModel();  table.setDefaultEditor(Object.class, null);  model.setColumnIdentifiers(  new String[] { "Mã Khóa Học", "Mã Học Viên", "Tên Học Viên", "SĐT" });  table.setModel(model);  model.setRowCount(0);  for (HocVien hv : dsHocVien) {  model.addRow(  new Object[] { maKhoaHoc, hv.getMaHocVien(), hv.getTen(), hv.getSoDienThoai() });  }  } else {  JOptionPane.showMessageDialog(null, "Mã khóa học không tồn tại", "Lỗi",  JOptionPane.ERROR\_MESSAGE);  }  }  }  });  btnDSHV.setMaximumSize(new Dimension(300, 30));  panel\_1.add(btnDSHV);  Component verticalStrut\_4 = Box.createVerticalStrut(10);  panel\_1.add(verticalStrut\_4);  JButton btnDiemDanh = new JButton("Danh sách điểm danh");  btnDiemDanh.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String maKhoaHoc = JOptionPane.showInputDialog(null, "Nhập mã khóa học:");  if (maKhoaHoc != null) {  maKhoaHoc = maKhoaHoc.toUpperCase();  if (maKhoaHoc.trim().isEmpty()) {  JOptionPane.showMessageDialog(null, "Bạn chưa nhập mã khóa học");  } else if (DBCheckExist.isMaTonTai("KhoaHoc", "maKhoaHoc", maKhoaHoc)) {    GiangVienDAO giangVienDAO = new GiangVienDAO();  String maGiangVien = giangVienDAO.getMaGiangVienByTaiKhoan(maTaiKhoan);  DiemDanhDAO diemDanhDAO = new DiemDanhDAO();  List<DiemDanh> dsHocVien = diemDanhDAO.getDanhSachDiemDanh(maKhoaHoc, maGiangVien);  DefaultTableModel model = new DefaultTableModel();  table.setDefaultEditor(Object.class, null);  table.setModel(model);  model.setColumnIdentifiers(new String[] { "Mã Khóa Học", "Mã Học Viên", "Tên Học Viên", "SĐT",  "Ngày", "Điểm Danh" });    model.setRowCount(0);  for (DiemDanh hv : dsHocVien) {  model.addRow(new Object[] { maKhoaHoc, hv.getMaHocVien(), hv.getTenHocVien(),  hv.getSoDienThoai(), hv.getNgay(), hv.getCoMat() });  }  } else {  JOptionPane.showMessageDialog(null, "Mã khóa học không tồn tại", "Lỗi",  JOptionPane.ERROR\_MESSAGE);  }    }  }  });  btnDiemDanh.setMaximumSize(new Dimension(300, 30));  panel\_1.add(btnDiemDanh);  Component verticalStrut\_2 = Box.createVerticalStrut(10);  panel\_1.add(verticalStrut\_2);  JButton btnSuaDiem = new JButton("Sửa điểm");  btnSuaDiem.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String maKhoaHoc = JOptionPane.showInputDialog(null, "Nhập mã khóa học:");  if (maKhoaHoc != null) {  maKhoaHoc = maKhoaHoc.toUpperCase();  if (maKhoaHoc.trim().isEmpty()) {  JOptionPane.showMessageDialog(null, "Bạn chưa nhập mã khóa học");  } else if (DBCheckExist.isMaTonTai("KhoaHoc", "maKhoaHoc", maKhoaHoc)) {  tableMode = "SUA\_DIEM";  GiangVienDAO giangVienDAO = new GiangVienDAO();  String maGiangVien = giangVienDAO.getMaGiangVienByTaiKhoan(maTaiKhoan);  DiemSoDAO dao = new DiemSoDAO();  List<DiemSo> dsHocVien = dao.getDanhSachDiem(maGiangVien, maKhoaHoc);  DefaultTableModel model = new DefaultTableModel();  table.setDefaultEditor(Object.class, null);  model.setColumnIdentifiers(  new String[] { "Mã Khóa Học", "Mã Học Viên", "Tên Học Viên", "Điểm" });  table.setModel(model);  model.setRowCount(0);  for (DiemSo hv : dsHocVien) {  model.addRow(  new Object[] { maKhoaHoc, hv.getMaHocVien(), hv.getTenHocVien(), hv.getDiem() });  }  } else {  JOptionPane.showMessageDialog(null, "Mã khóa học không tồn tại", "Lỗi",  JOptionPane.ERROR\_MESSAGE);  }    }  }  });    JButton btnDiemDanh\_1 = new JButton("Điểm danh");  btnDiemDanh\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  GiangVienDAO giangVienDAO = new GiangVienDAO();  String maGiangVien = giangVienDAO.getMaGiangVienByTaiKhoan(maTaiKhoan);    String maKhoaHoc = JOptionPane.showInputDialog(null, "Nhập mã khóa học:");  if ( maKhoaHoc != null) {  maKhoaHoc = maKhoaHoc.toUpperCase();  if (maKhoaHoc.trim().isEmpty()) {  JOptionPane.showMessageDialog(null, "Bạn chưa nhập mã khóa học");  }else if(DBCheckExist.isMaTonTai("KhoaHoc", "maKhoaHoc", maKhoaHoc)) {  SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");  String ngayHienTai = sdf.format(new Date());  DefaultTableModel model = new DefaultTableModel();    tableMode = "DIEM\_DANH";  DiemDanhDAO khoaHocDAO = new DiemDanhDAO();  List<DiemDanh> dsHocVien = khoaHocDAO.diemDanhSinhVien(maKhoaHoc, maGiangVien );    table.setDefaultEditor(Object.class, null);  model.setColumnIdentifiers(  new String[] { "Mã Khóa Học", "Mã Học Viên", "Tên Học Viên", "SĐT" , "Ngày", "Điểm Danh" });  table.setModel(model);  model.setRowCount(0);    for (DiemDanh hv : dsHocVien) {  model.addRow(new Object[] {maKhoaHoc, hv.getMaHocVien(), hv.getTenHocVien(), hv.getSoDienThoai(), ngayHienTai, hv.getCoMat() });  }  }else {  JOptionPane.showMessageDialog(null,"Mã khóa học không tồn tại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  }      }          }  });  btnDiemDanh\_1.setMaximumSize(new Dimension(300, 30));  panel\_1.add(btnDiemDanh\_1);    Component verticalStrut\_2\_1 = Box.createVerticalStrut(10);  panel\_1.add(verticalStrut\_2\_1);  btnSuaDiem.setMaximumSize(new Dimension(300, 30));  panel\_1.add(btnSuaDiem);  Component verticalStrut\_3 = Box.createVerticalStrut(10);  panel\_1.add(verticalStrut\_3);  JButton btnBack = new JButton("Đăng xuất");  btnBack.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  new LoginGUI().setVisible(true);  dispose();  }  });  btnBack.setMaximumSize(new Dimension(300, 30));  panel\_1.add(btnBack);  setLocationRelativeTo(null);  setVisible(true);  }  public static void main(String[] args) {  SwingUtilities.invokeLater(() -> new LoginGiangVienGUI("TK002"));  }  } |

# **Kết quả**

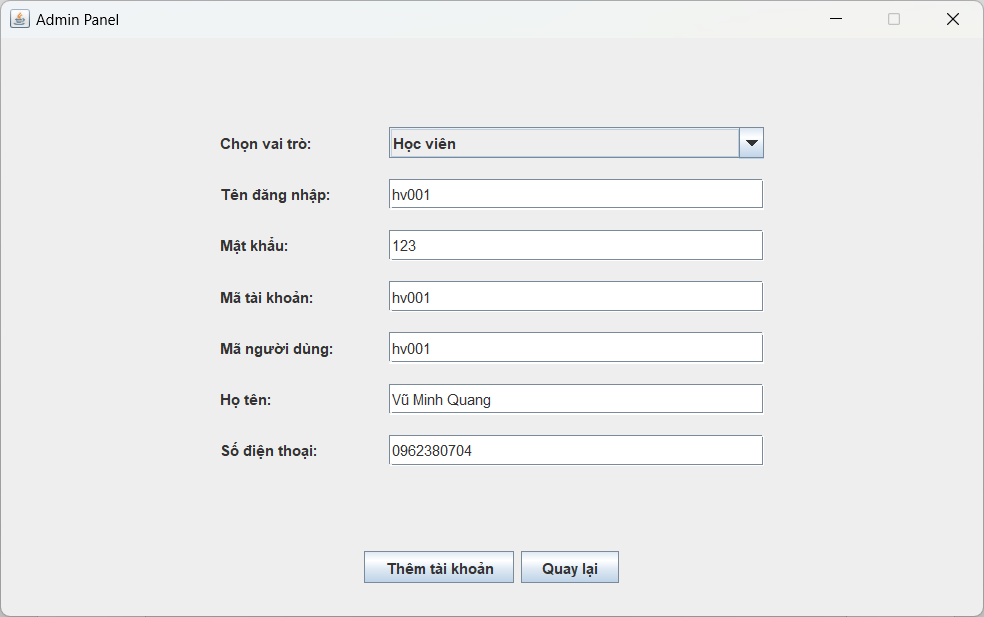
1. **Đăng nhập bằng tài khoản Admin**



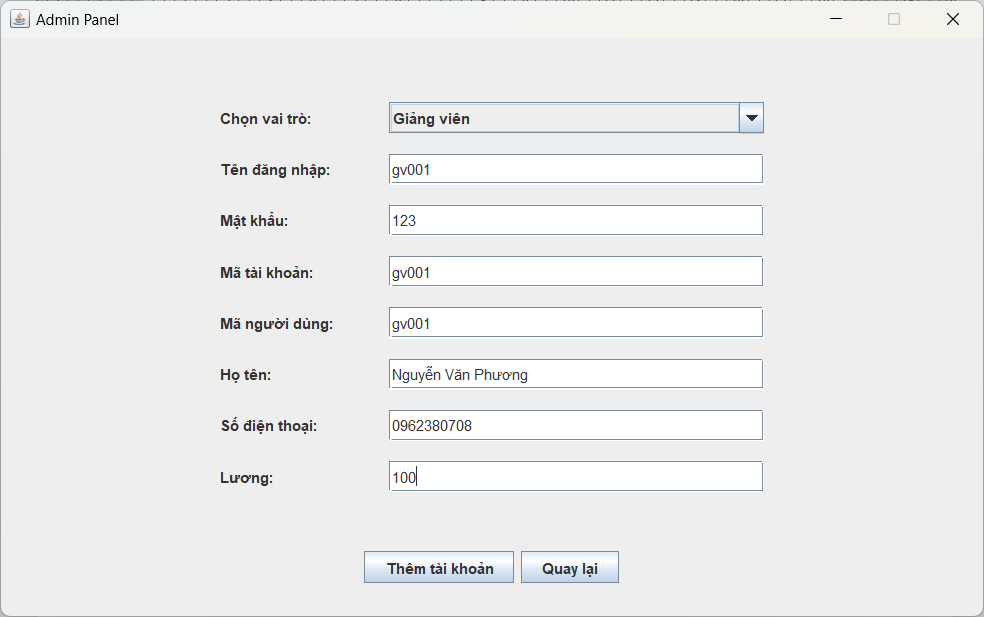
Hình 18: Đăng nhập bằng tài khoản admin



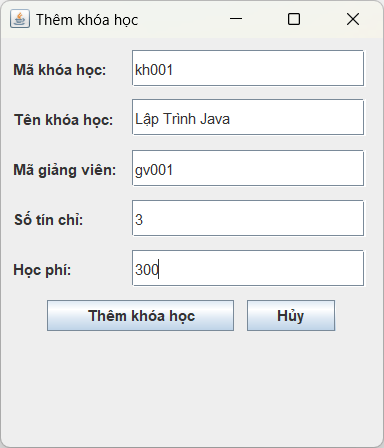
Hình 19: Giao diện quản lý khóa học trực tuyến khi login bằng tài khoản Admin



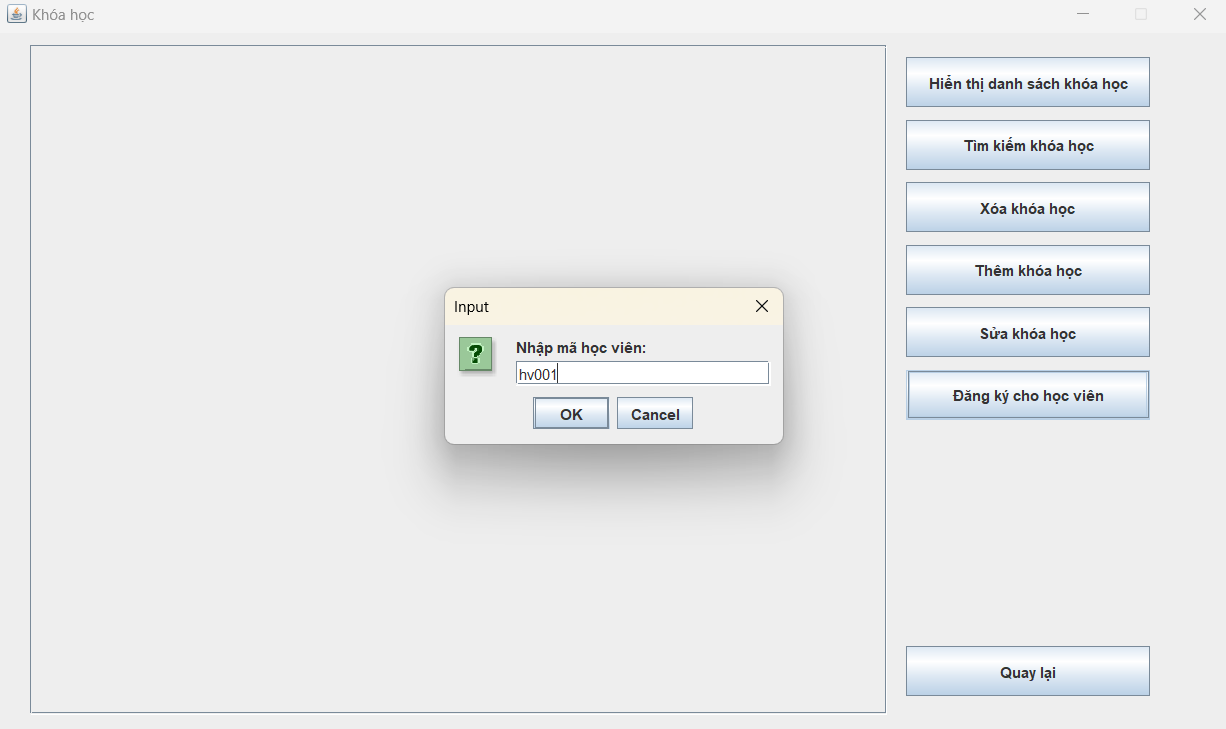
Hình 20: Tạo các tài khoản học viên



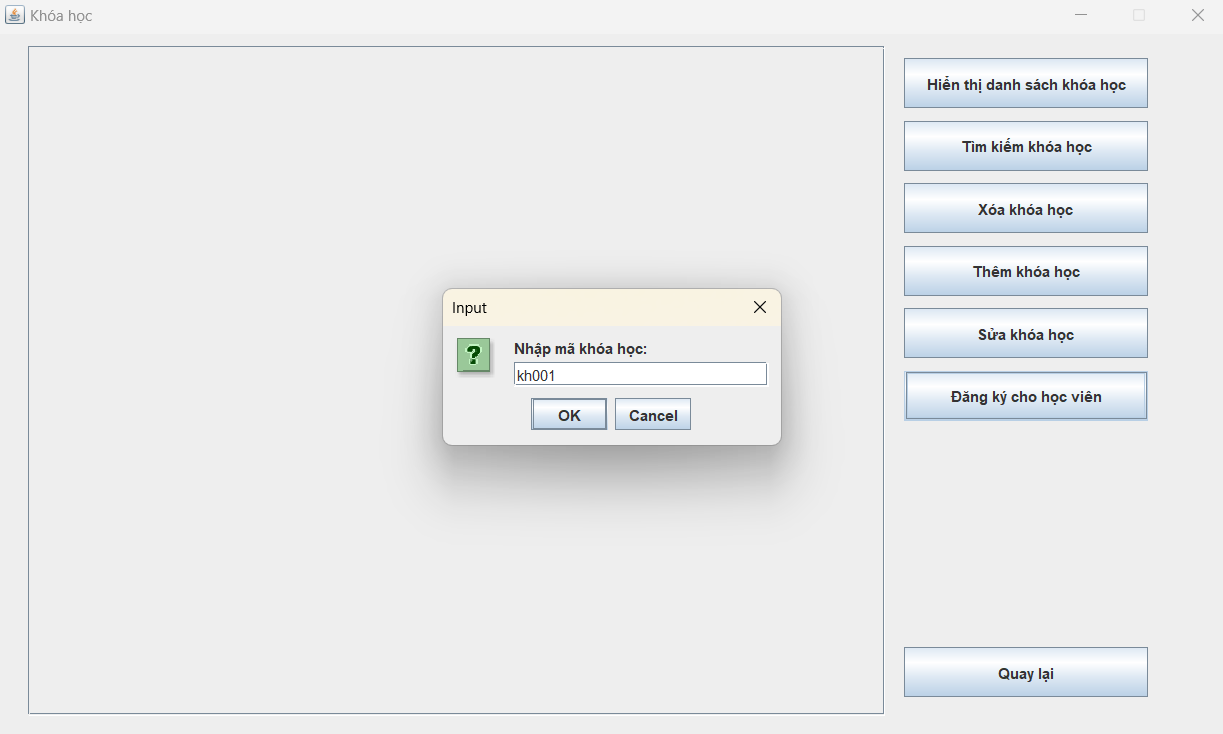
Hình 21: Tạo các tài khoản giảng viên



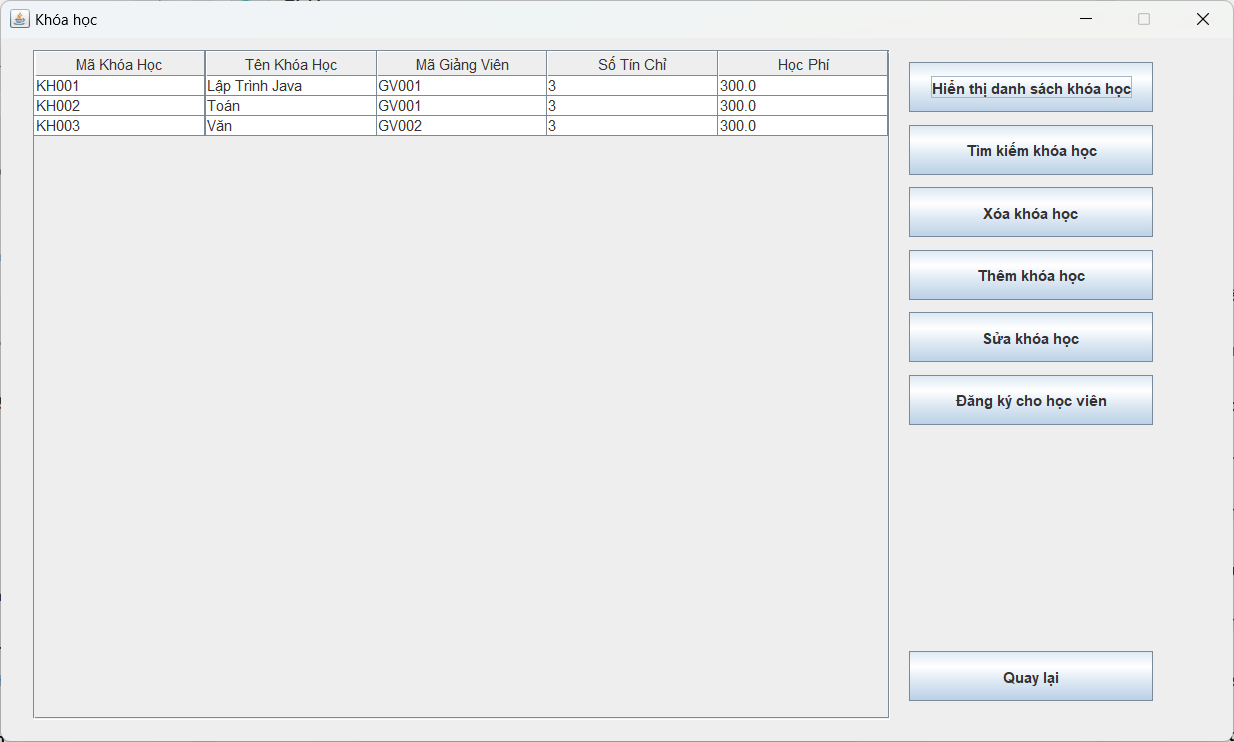
Hình 22: Tạo các khóa học



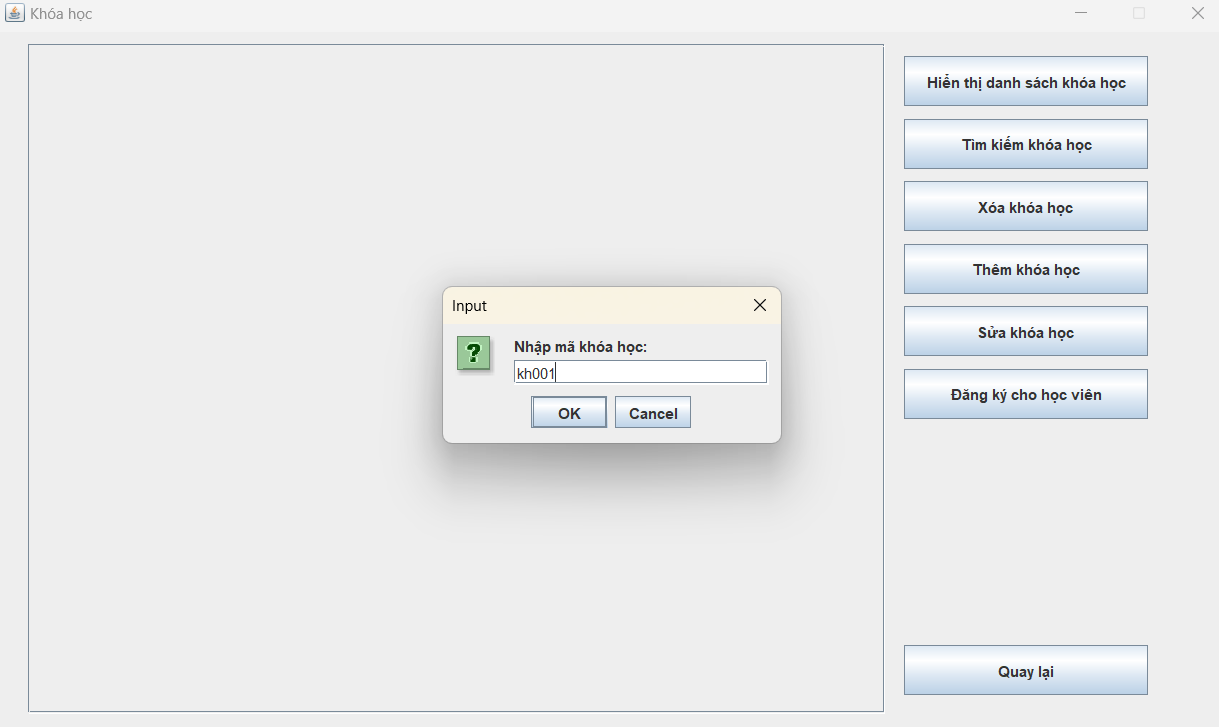
Hình 23: Nhập mã học viên cần đăng ký khóa học



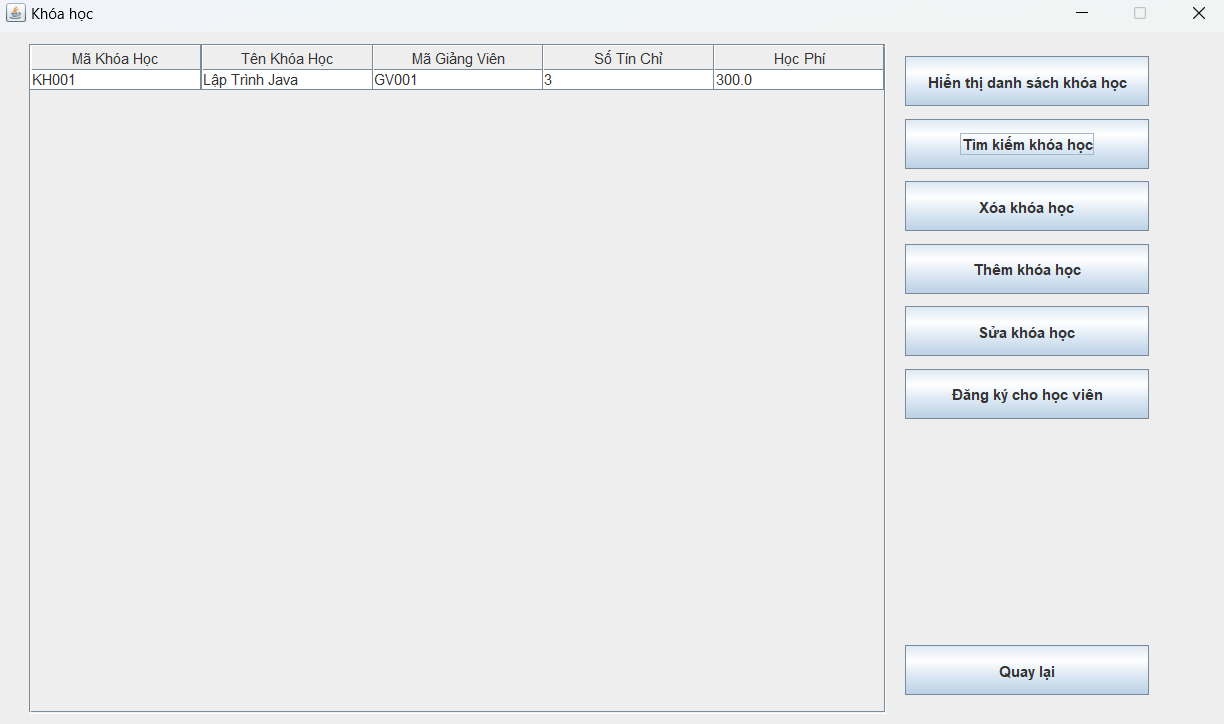
Hình 24: Nhập khóa học cần đăng ký



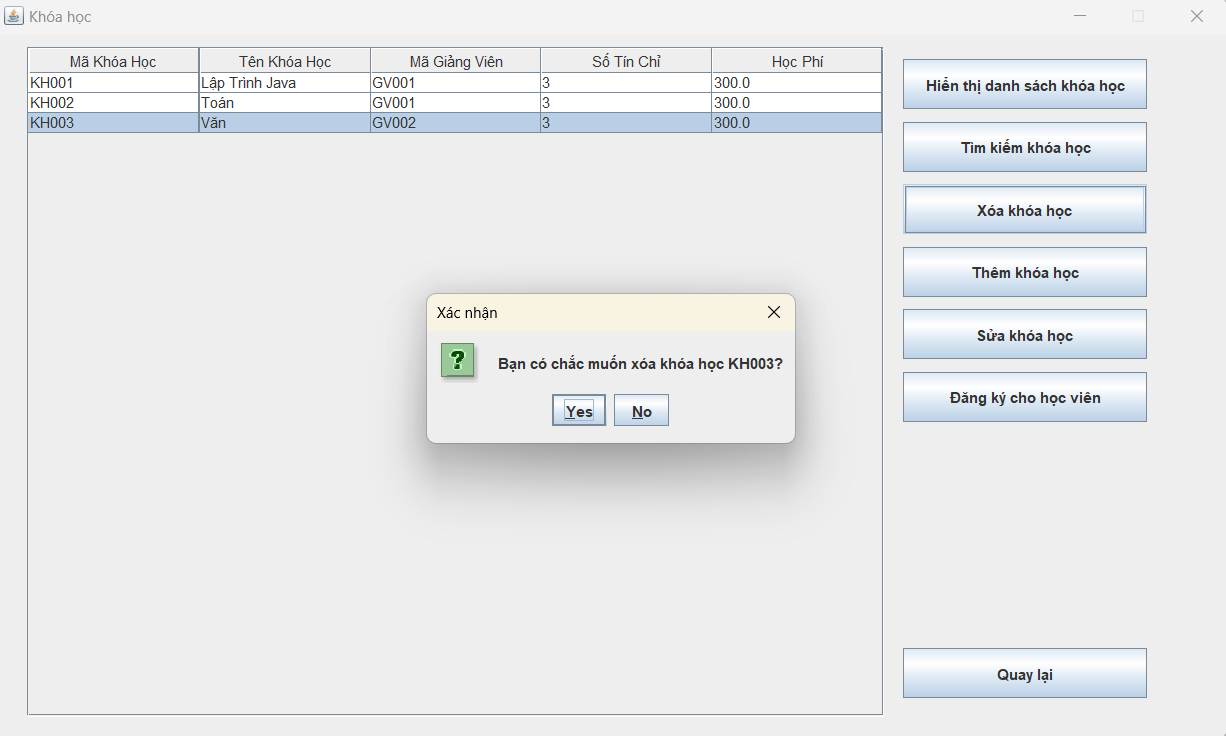
Hình 25: Hiển thị danh sách khóa học



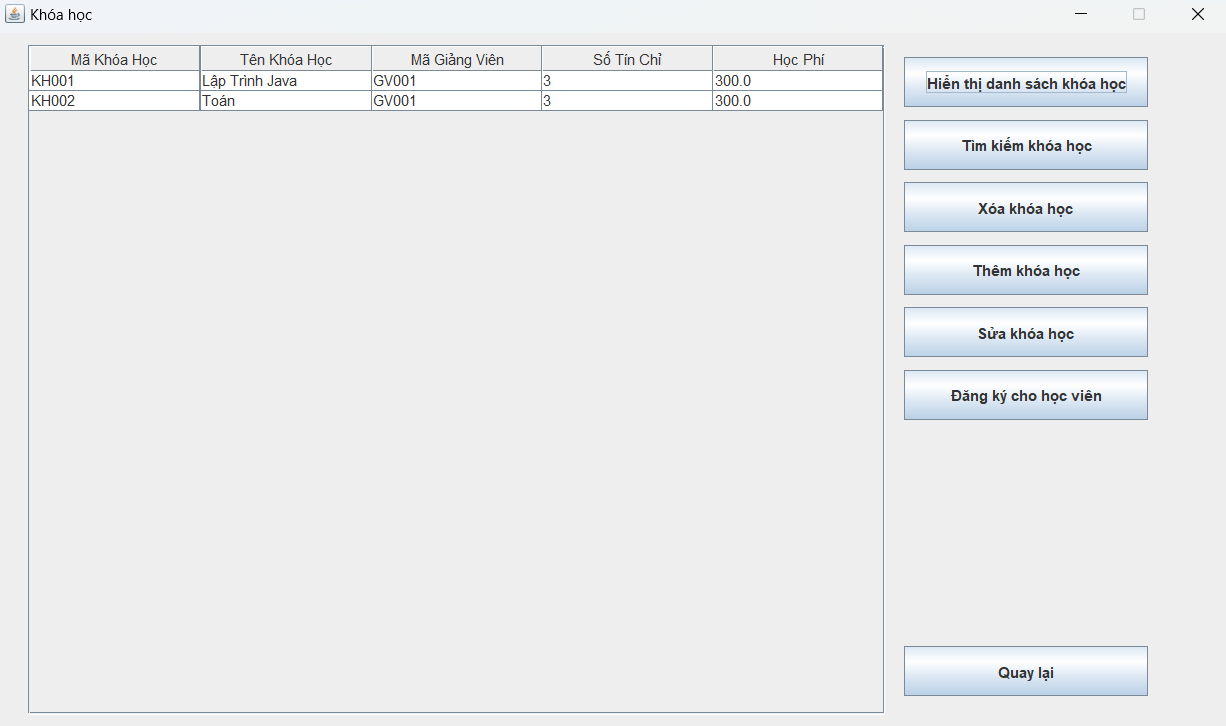
Hình 26: Nhập mã khóa học cần tìm



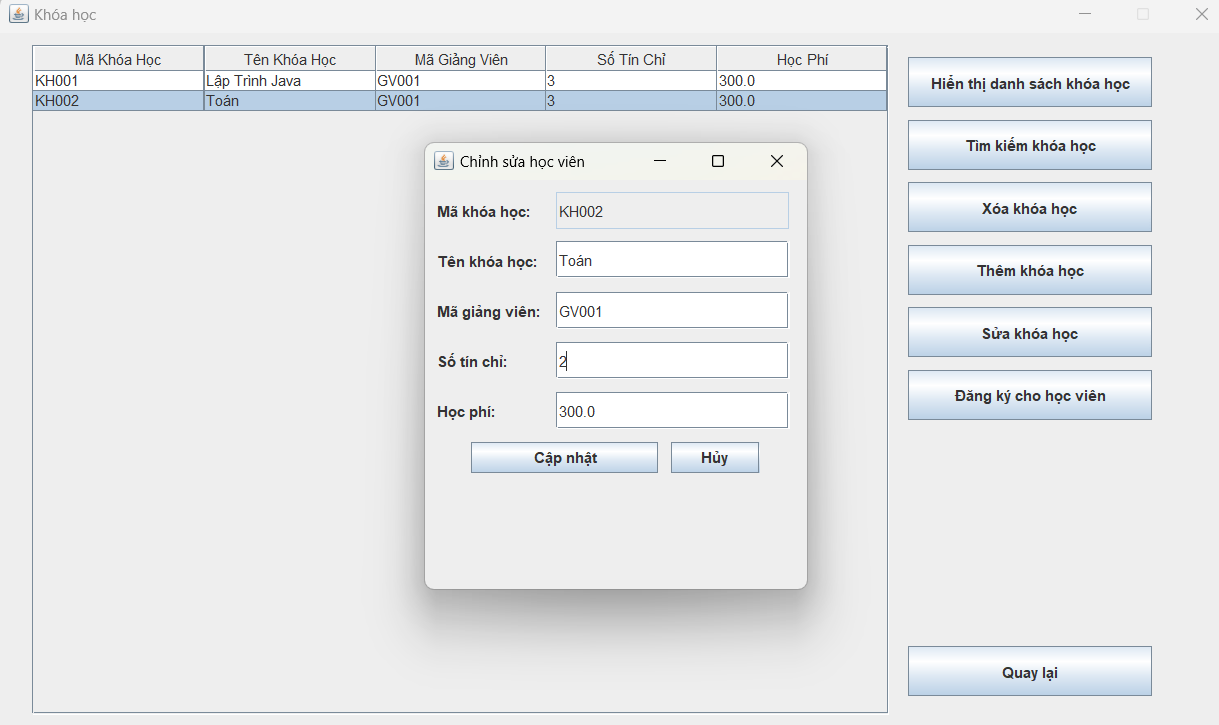
Hình 27: Kết quả tìm kiếm khóa học



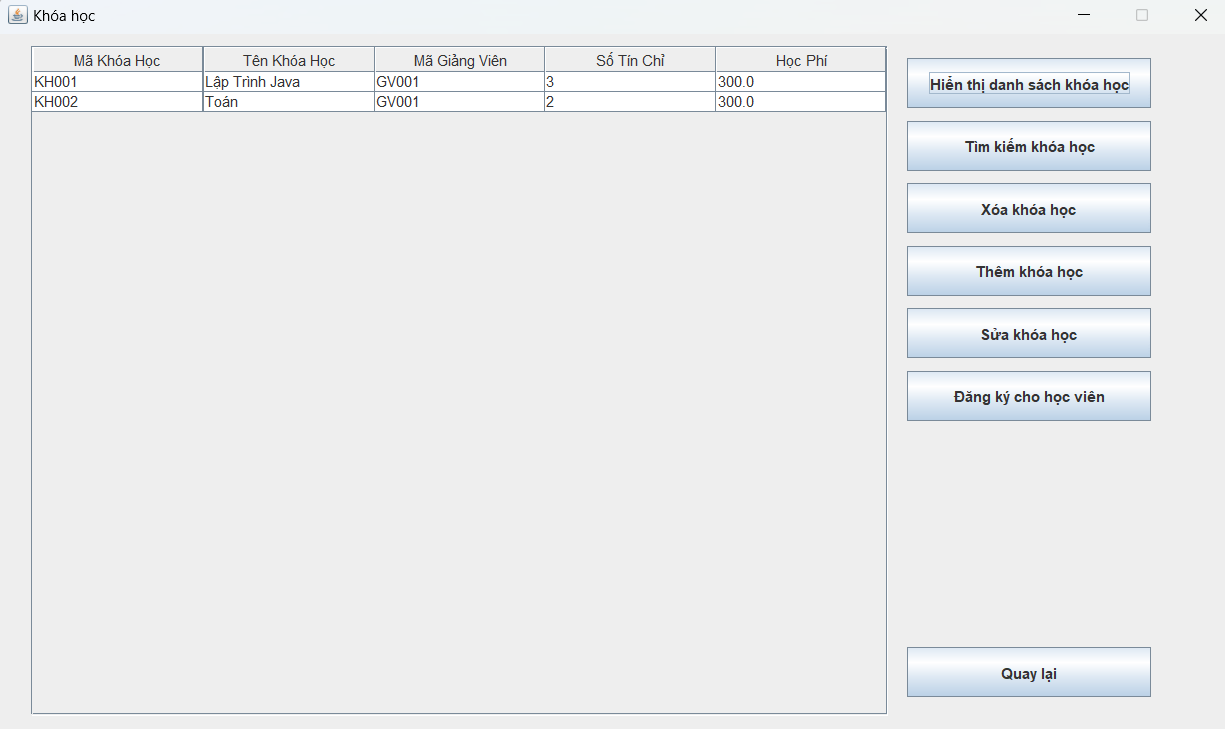
Hình 28: Chọn khóa học muốn xóa



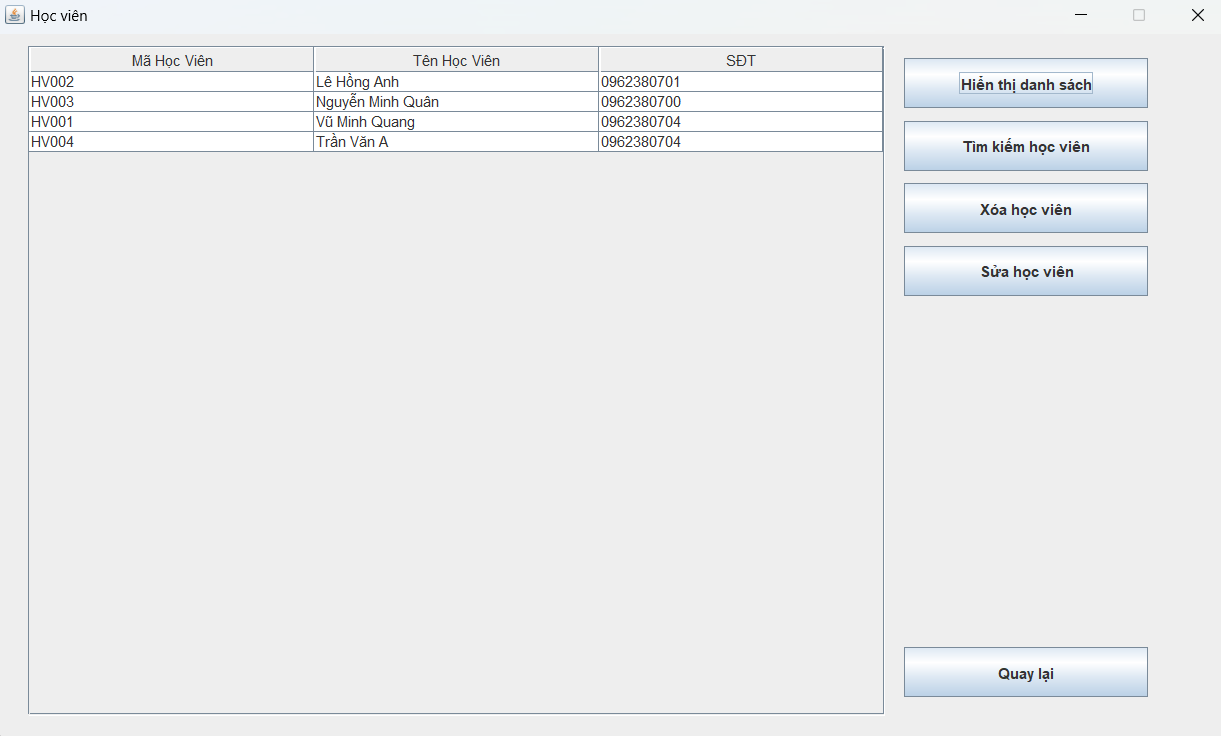
Hình 29: Kết quả xóa khóa học



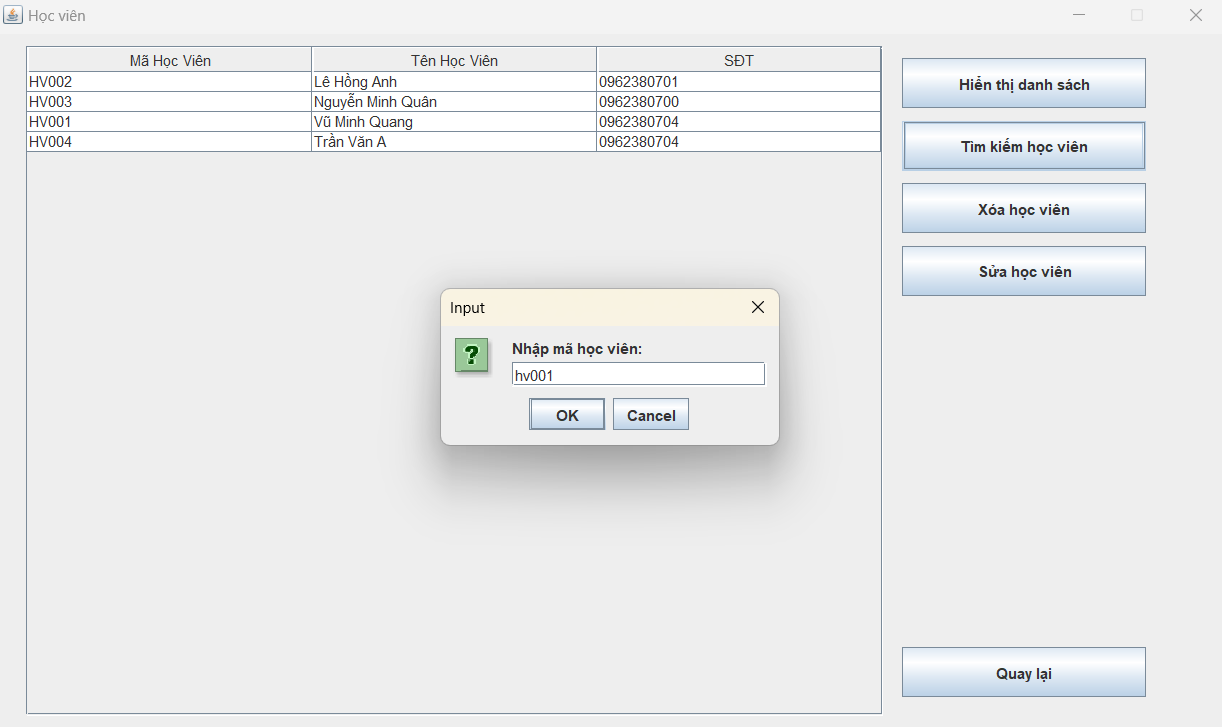
Hình 30: Sửa khóa học



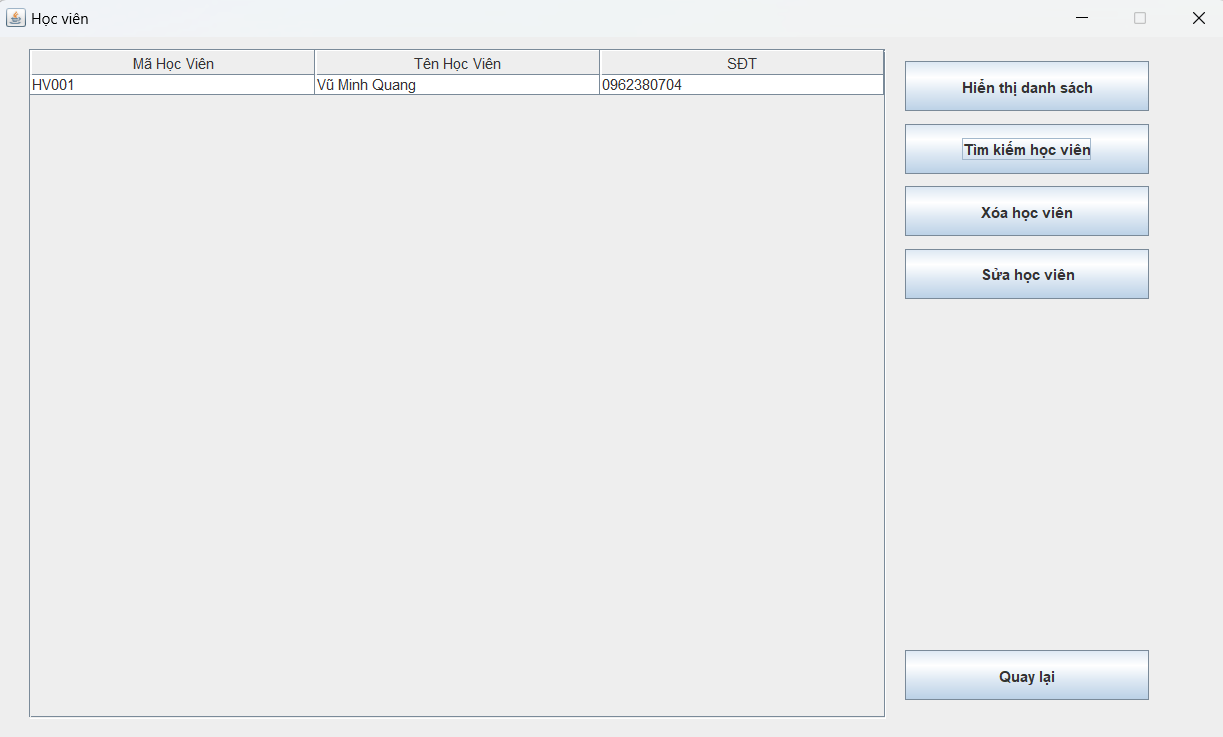
Hình 31: Kết quả sửa khóa học



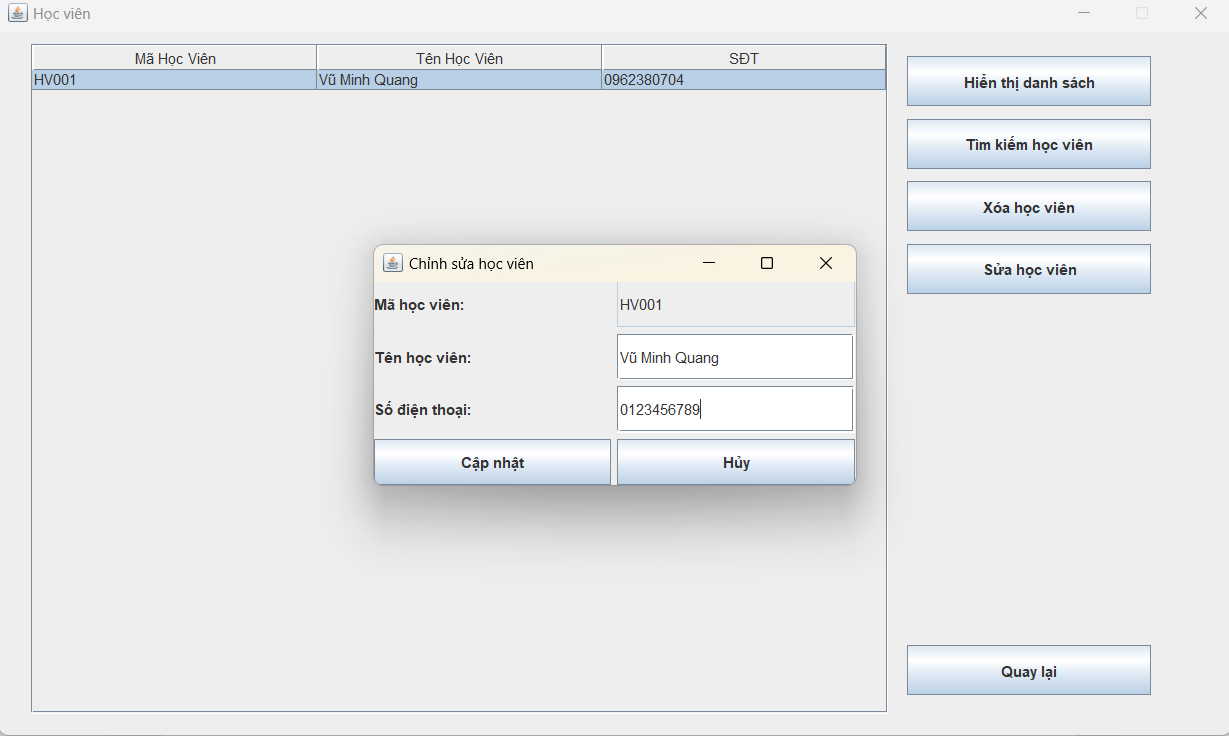
Hình 32: Hiển thị danh sách học viên



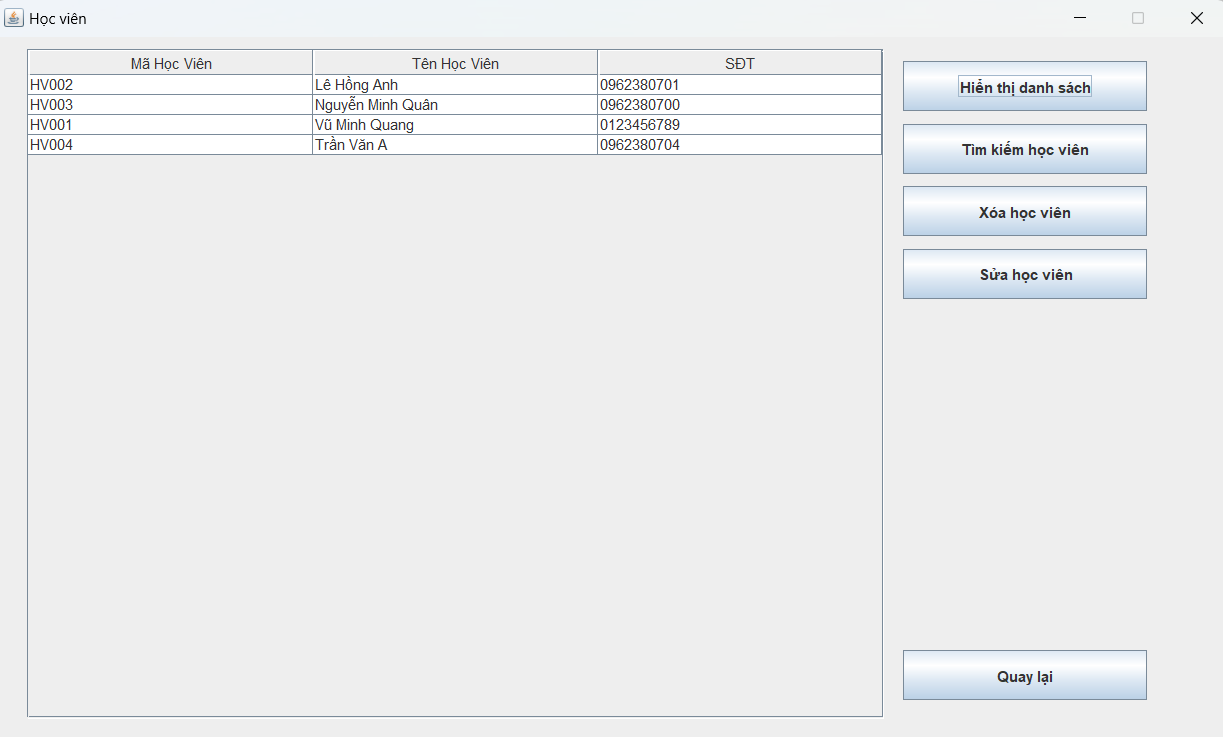
Hình 33: Nhập mã học viên cần tìm



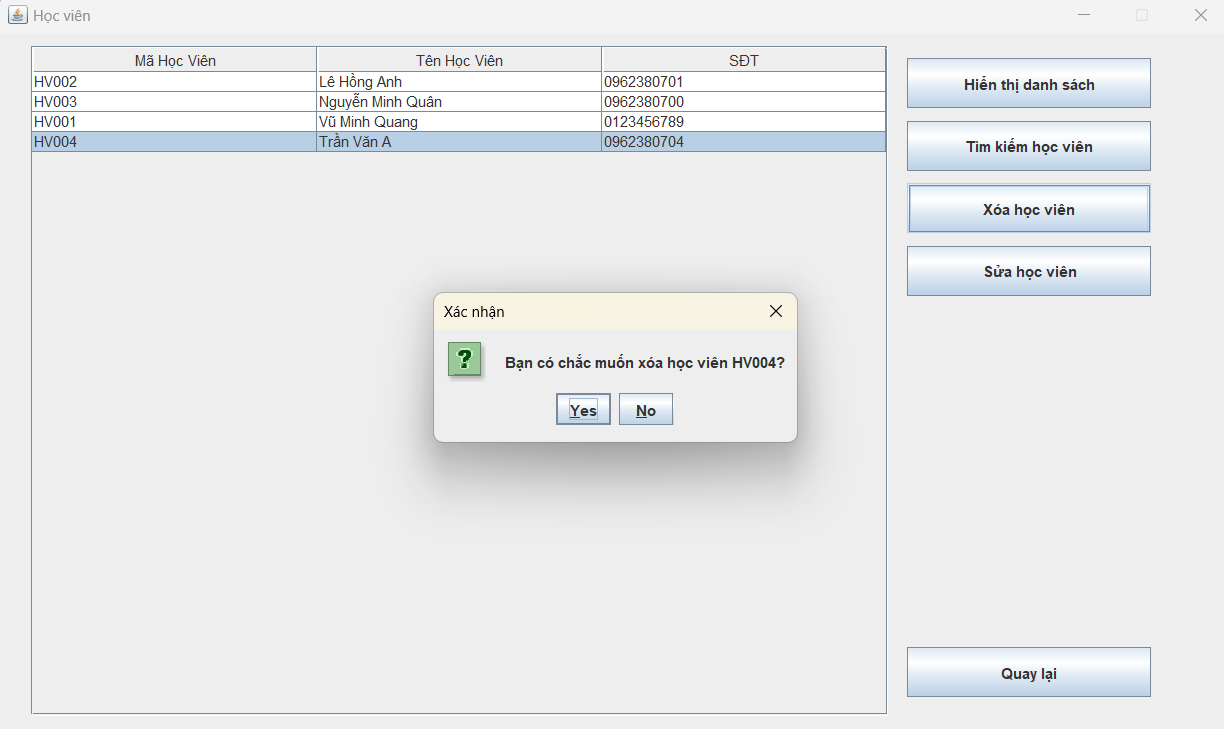
Hình 34: Kết quả tìm kiếm học viên



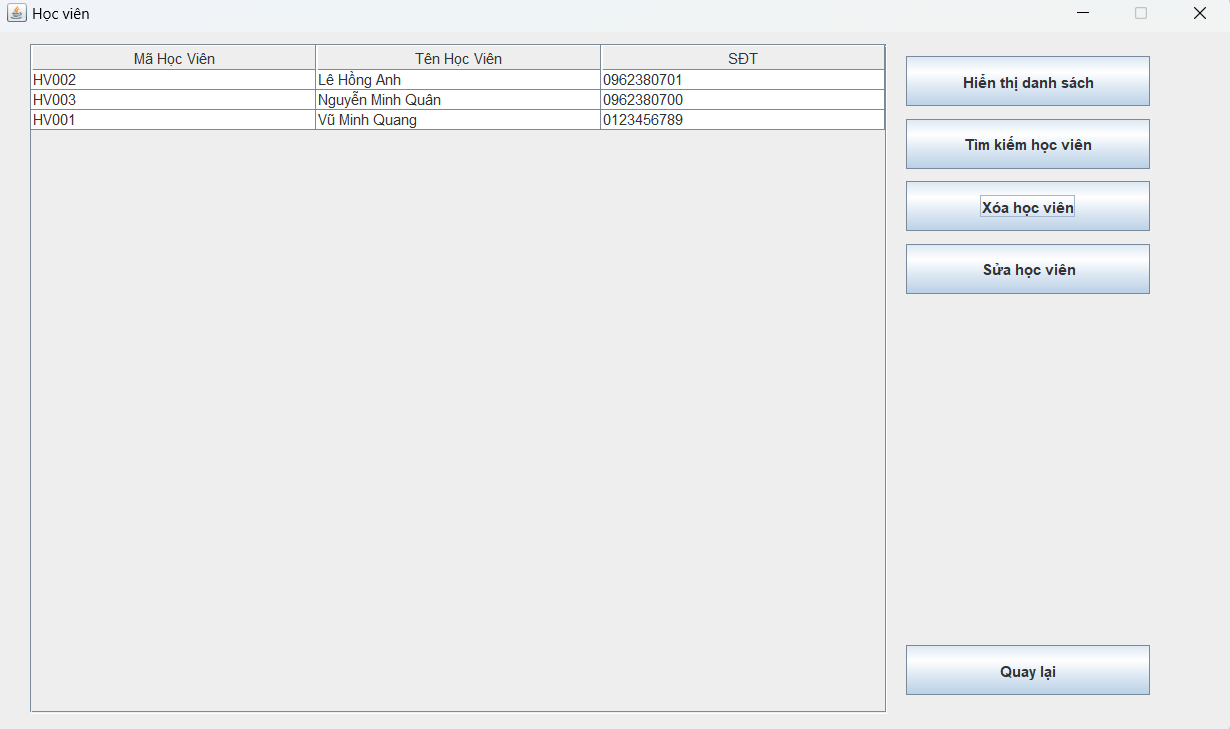
Hình 35: Sửa học viên



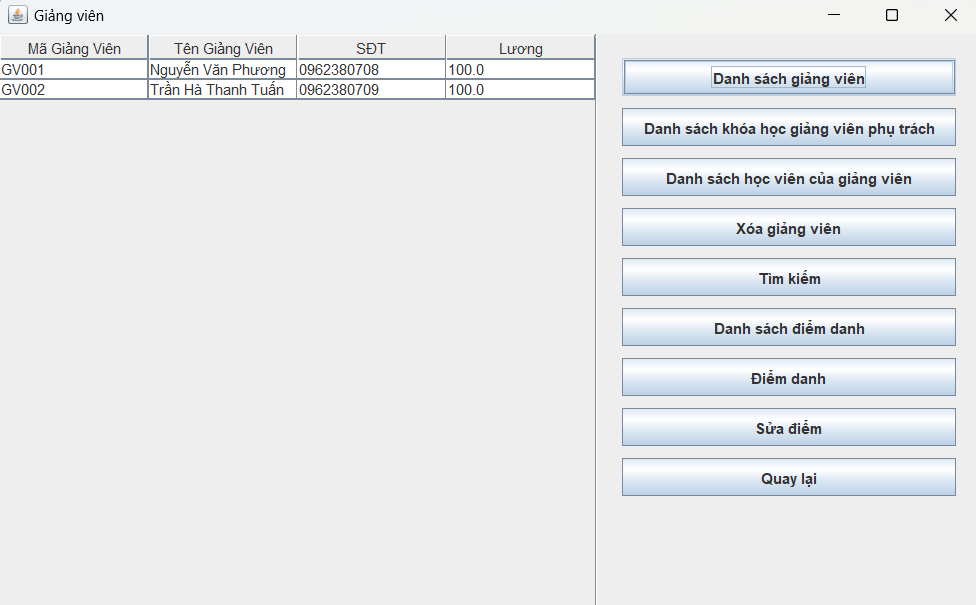
Hình 36: Kết quả sửa học viên



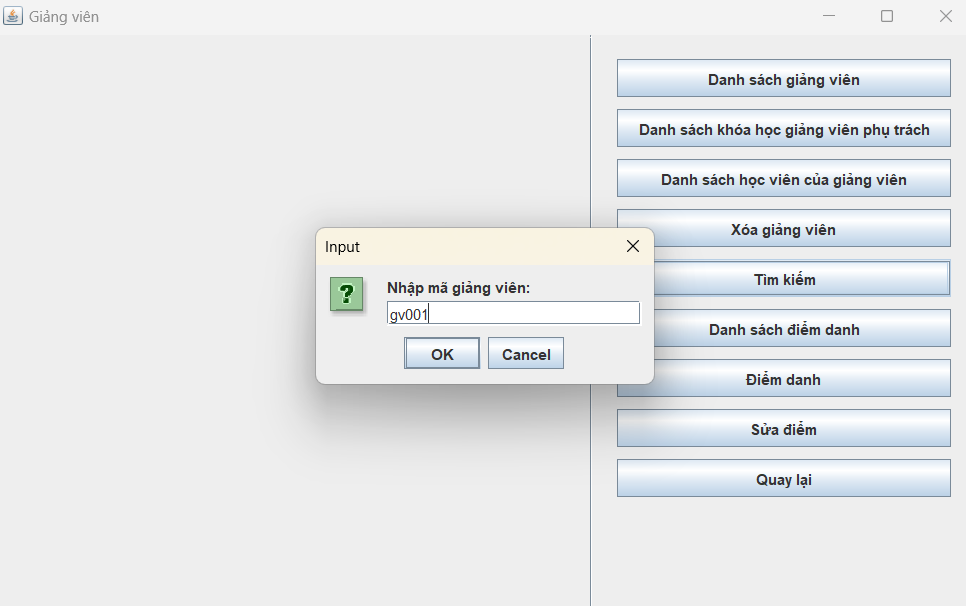
Hình 37: Xóa học viên



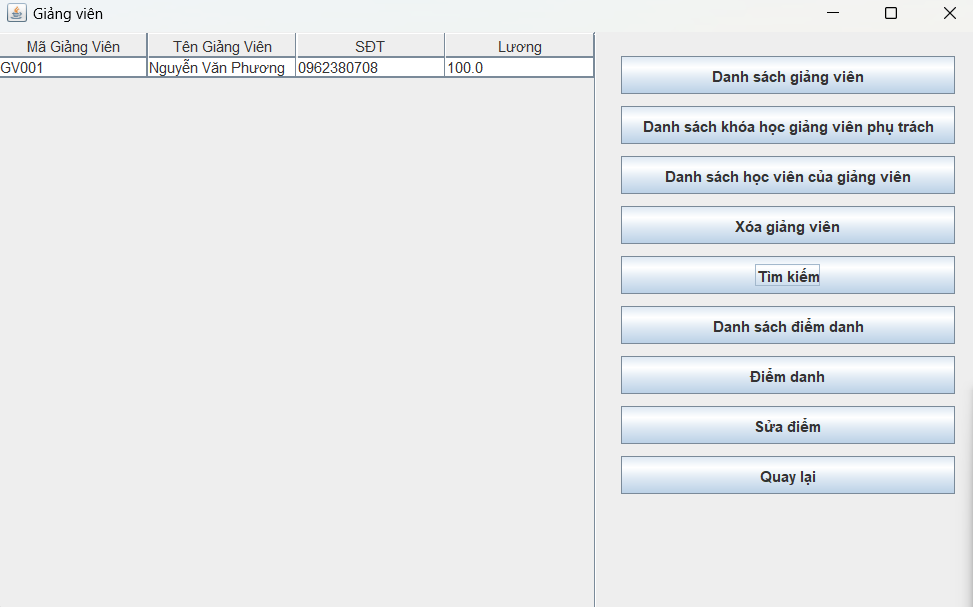
Hình 38: Kết quả xóa học viên



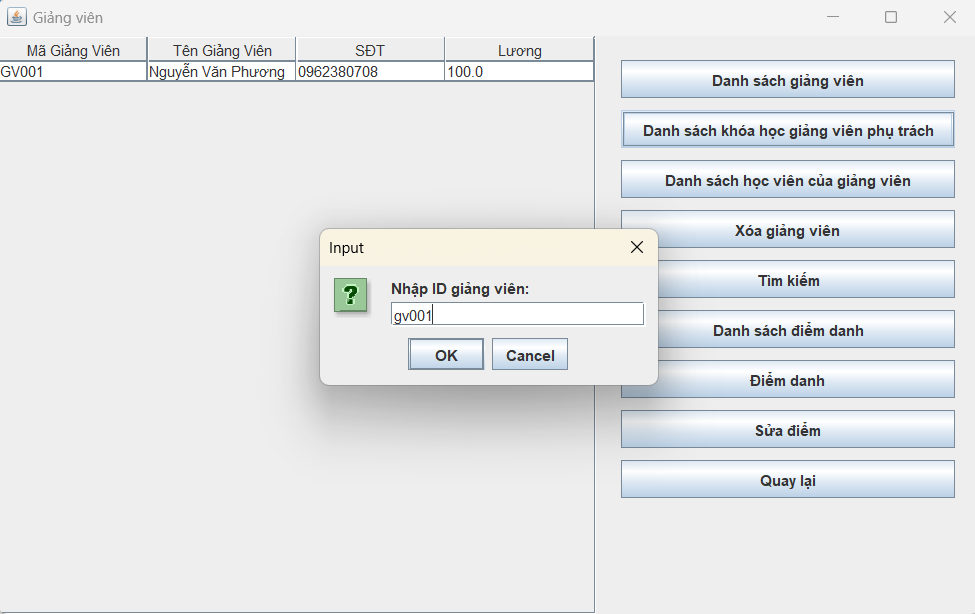
Hình 39: Hiển thị danh sách giảng viên



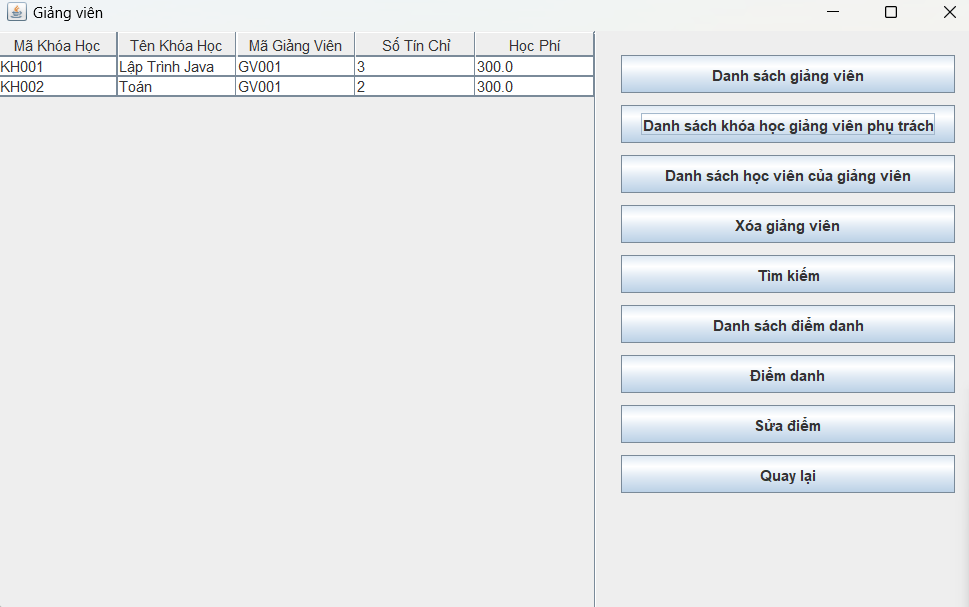
Hình 40: Nhâp mã giảng viên cần tìm kiếm



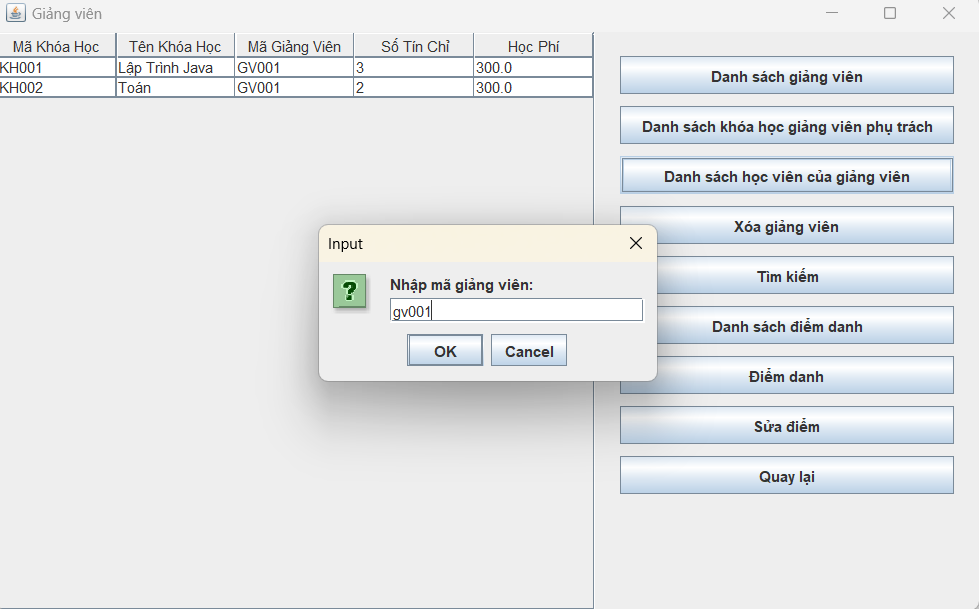
Hình 41: Kết quả tìm kiếm giảng viên



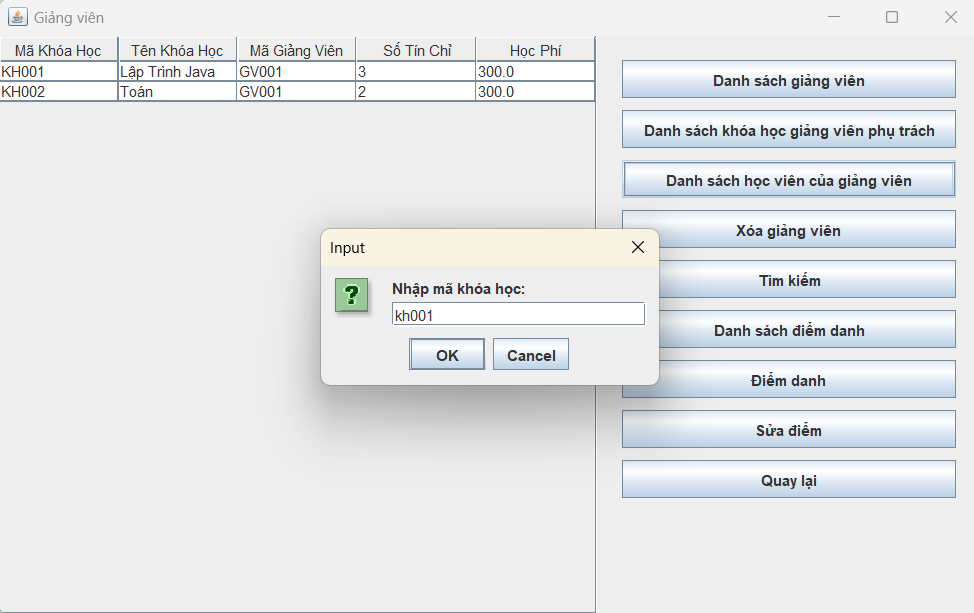
Hình 42: Nhập mã giảng viên



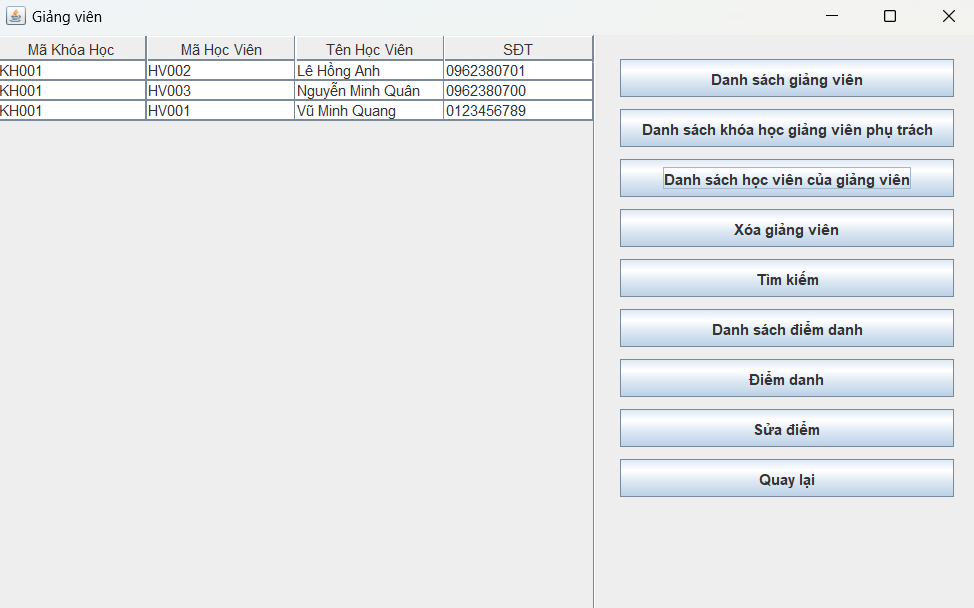
Hình 43: Kết quả danh sách khóa học của giảng viên phụ trách



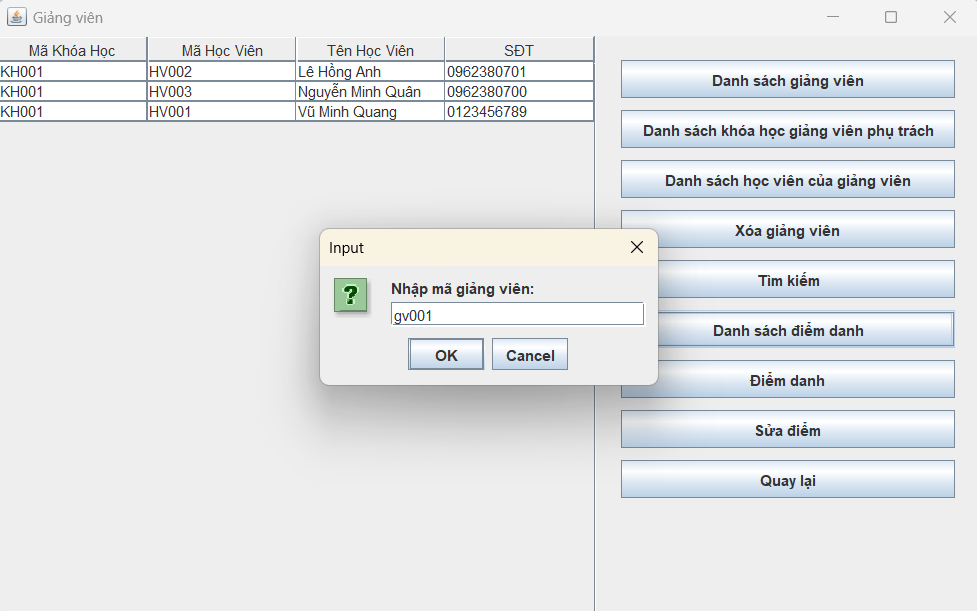
Hình 44:Nhập mã giảng viên



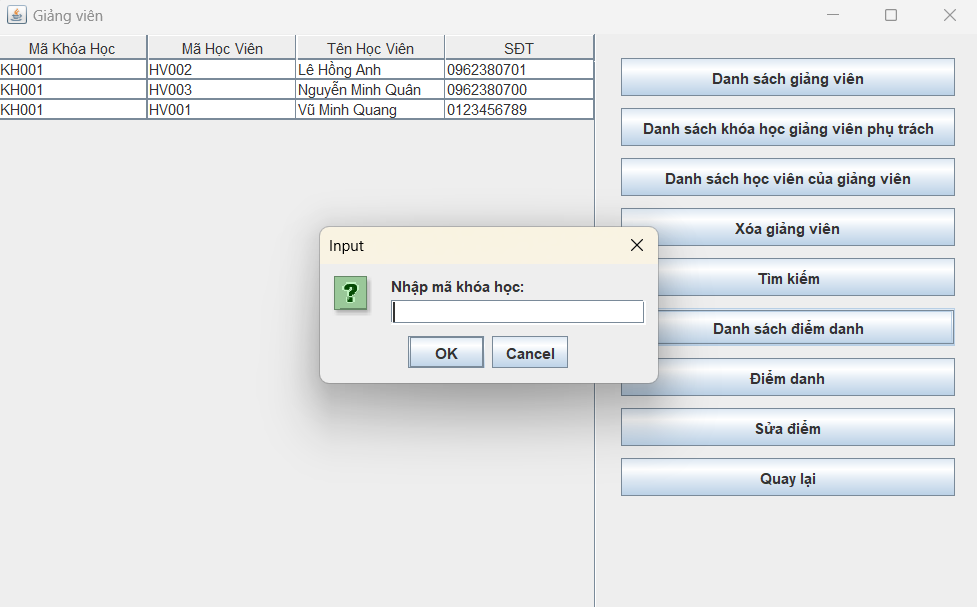
Hình 45: Nhập mã khóa học



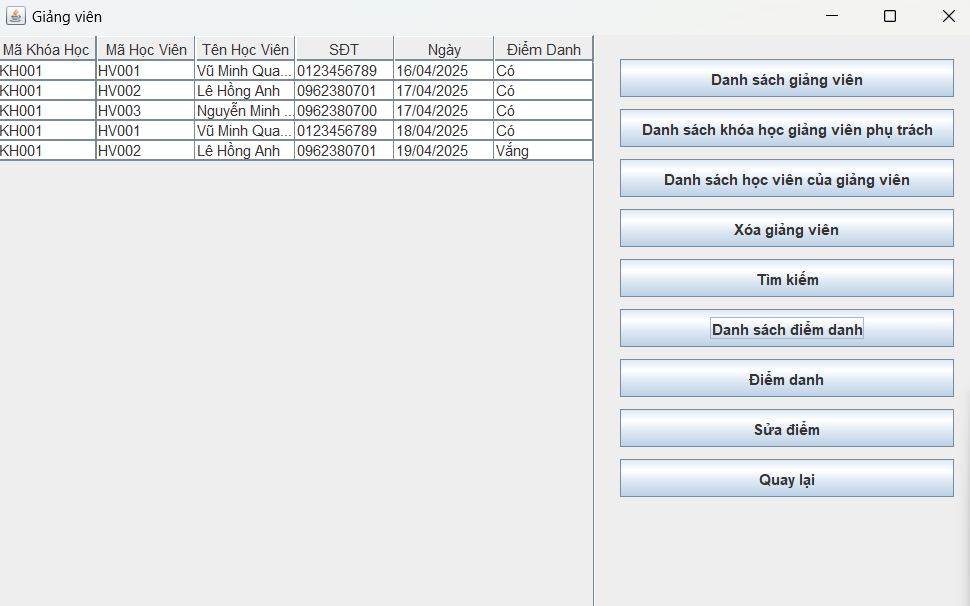
Hình 46: Kết quả danh sách học viên theo khóa học của giảng viên:



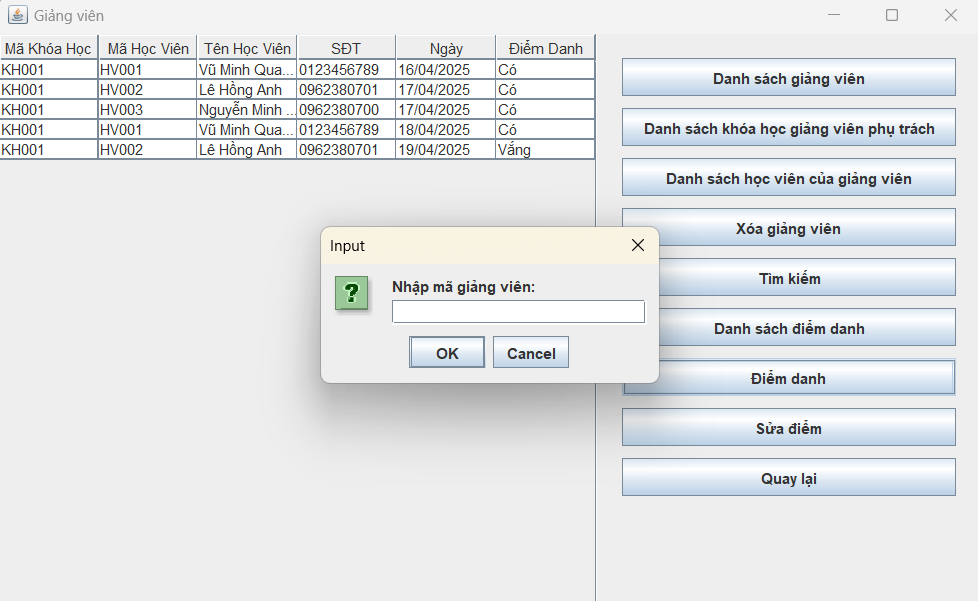
Hình 47: Nhập mã giảng viên



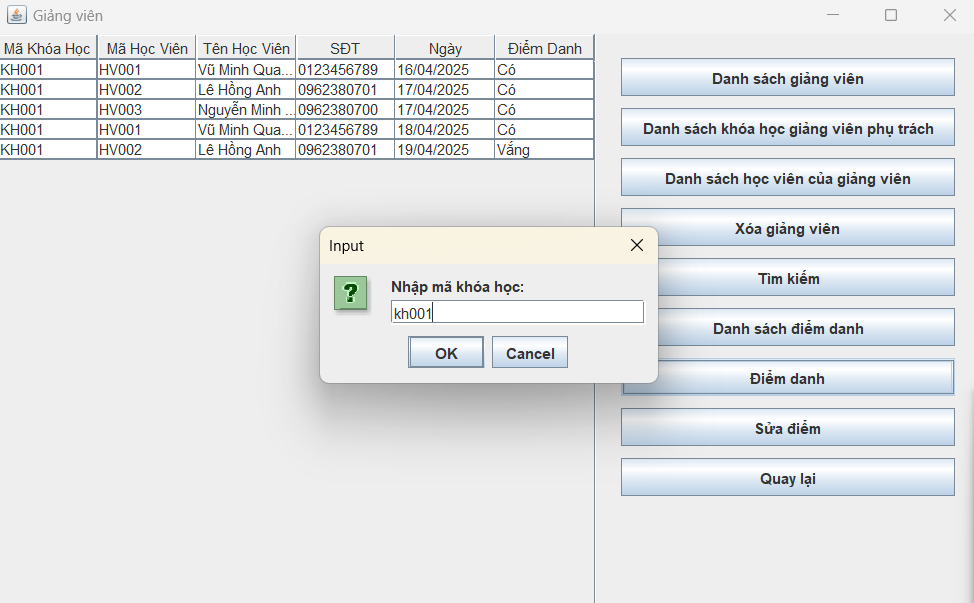
Hình 48: Nhập mã khóa học để mở danh sách sđiểm danh



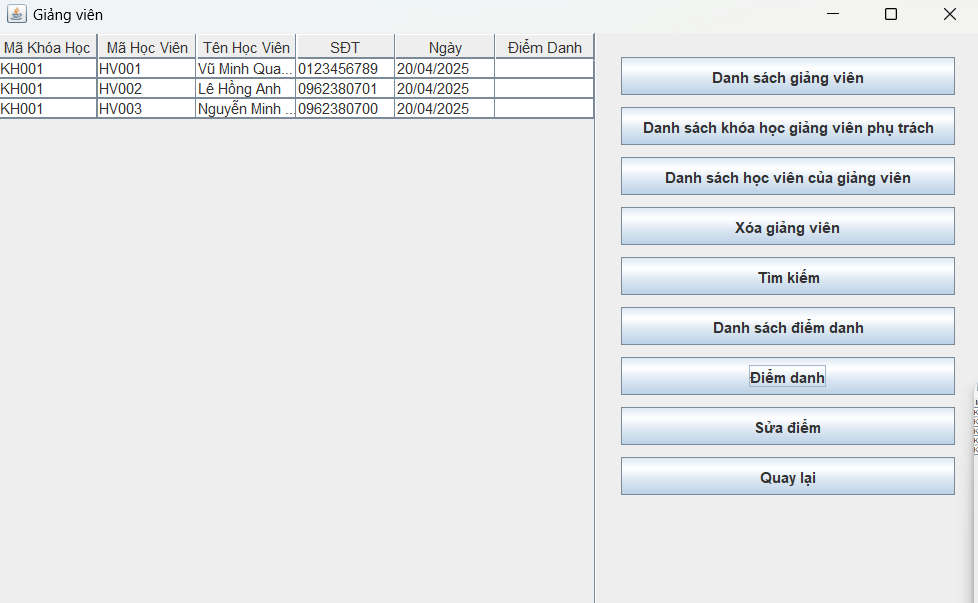
Hình 49: Kết quả danh sách điểm danh



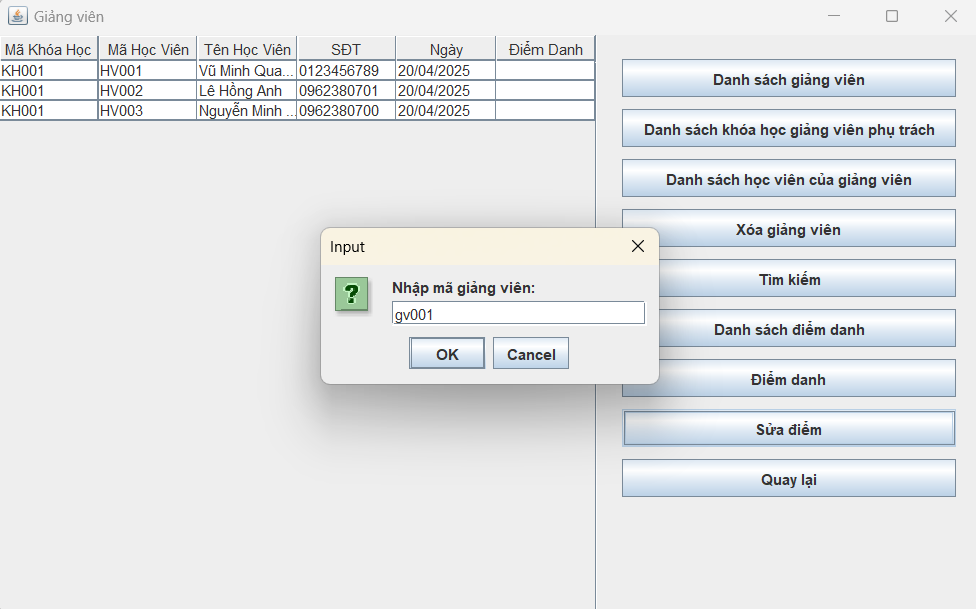
Hình 50: Nhập mã giảng viên



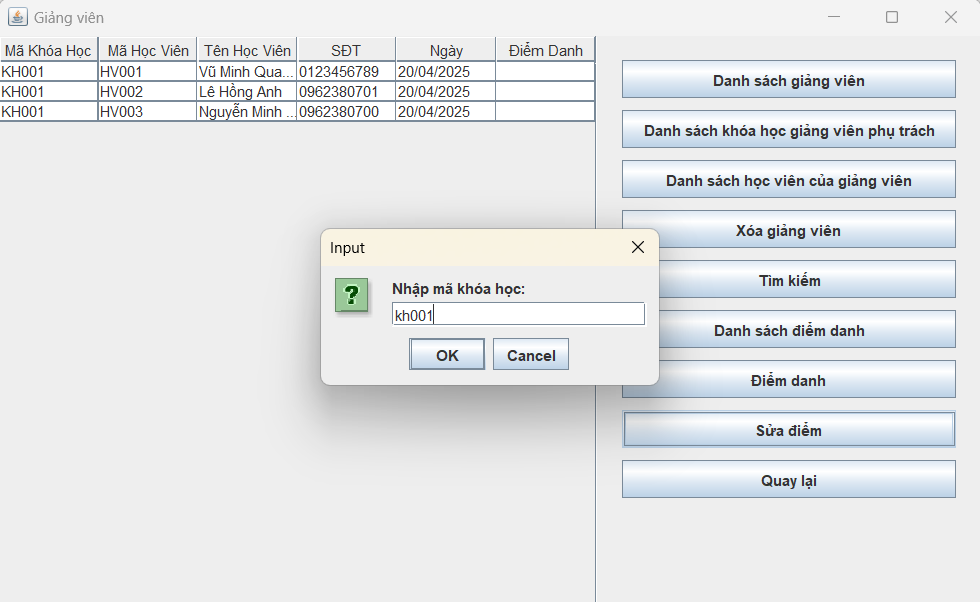
Hình 51:Nhập mã khóa học của giảng viên



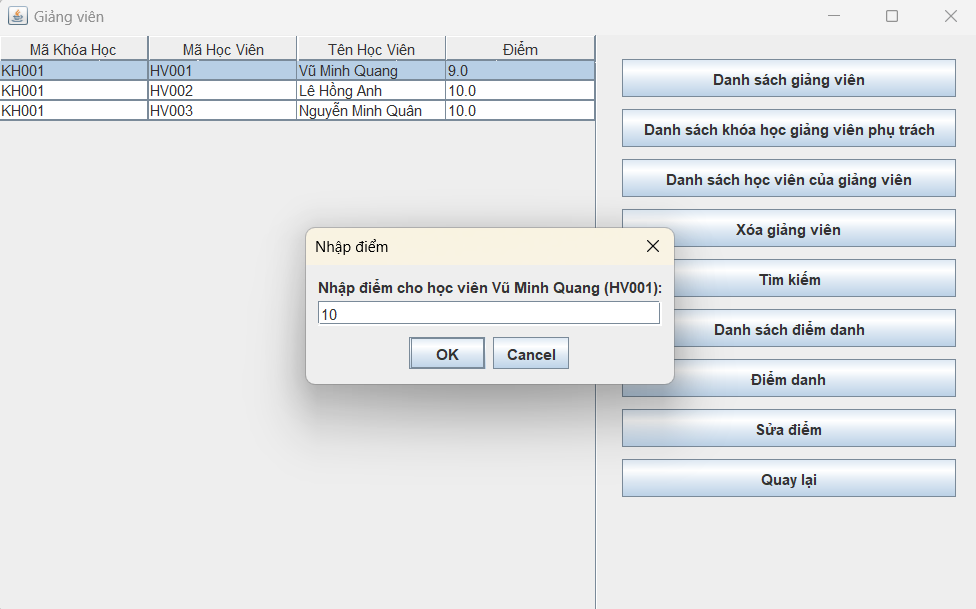
Hình 52: Nhấn đúp váo sinh viên để điểm danh



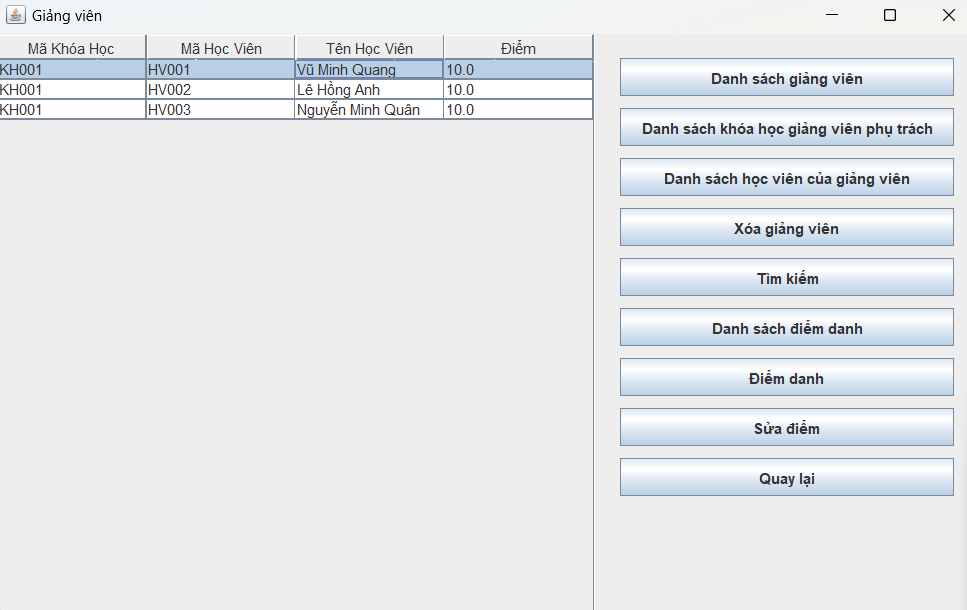
Hình 53: Nhập mã giảng viên



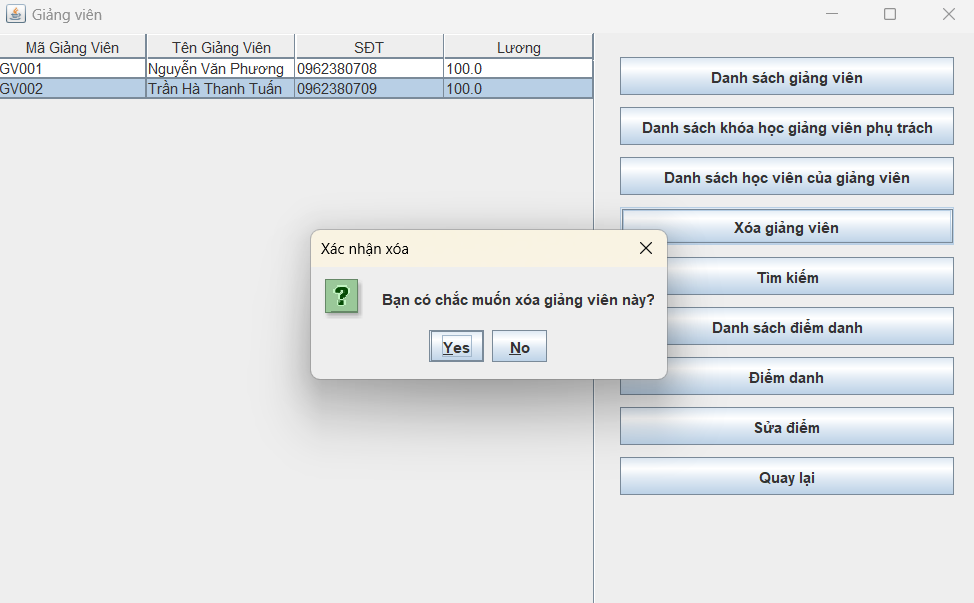
Hình 54: Nhập mã khóa học



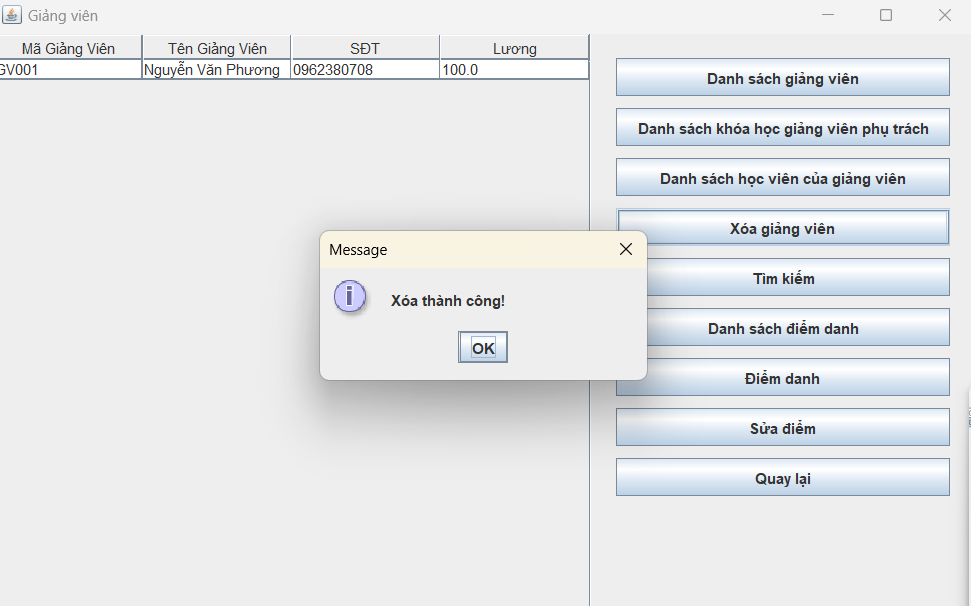
Hình 55: Chọn học viên và nhập điểm



Hình 56: Kết quả nhập điểm của học viên

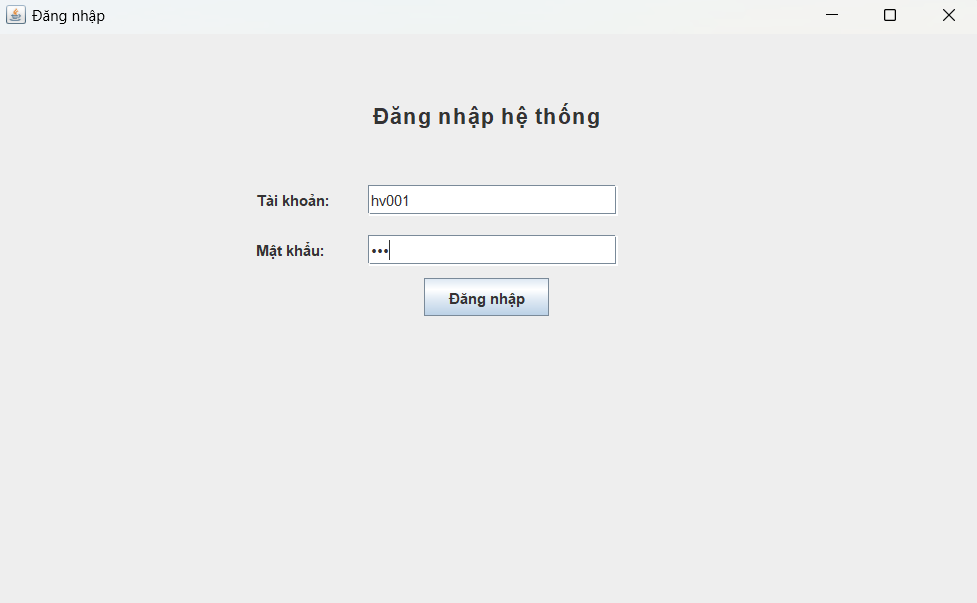


Hình 57: Chọn giảng viên cần xóa

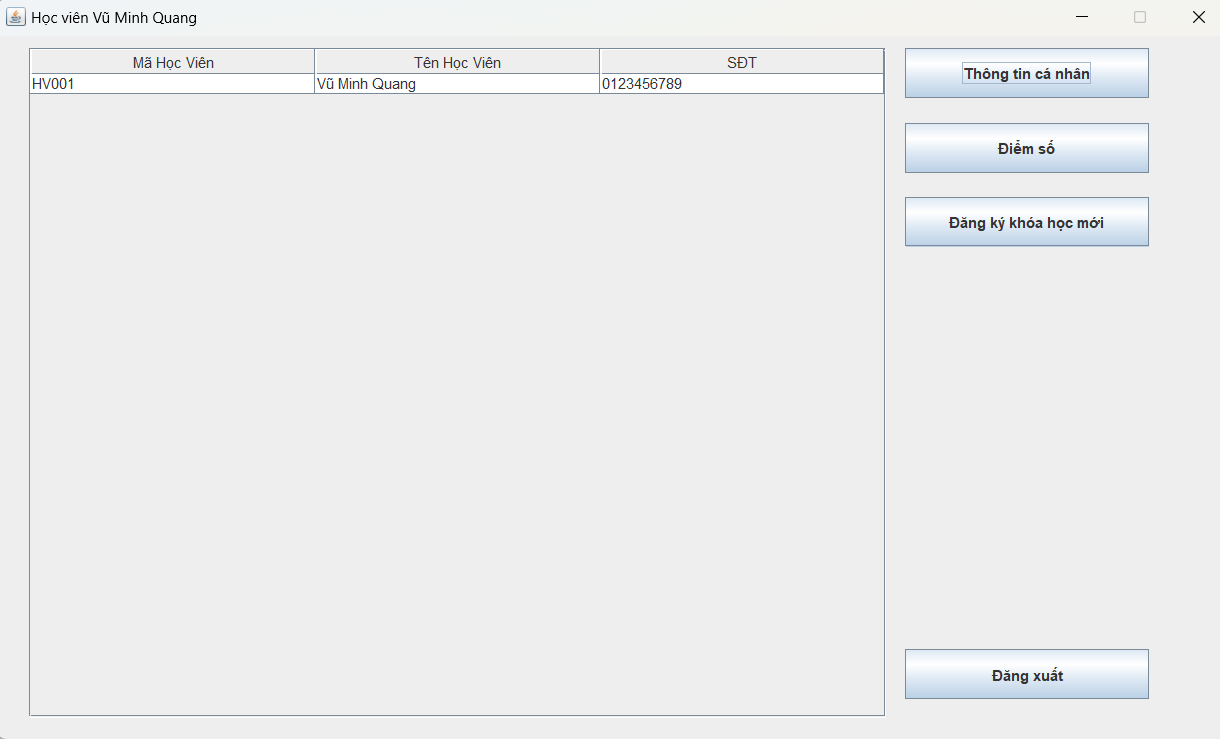


Hình 58: Kết quả xóa giảng viên

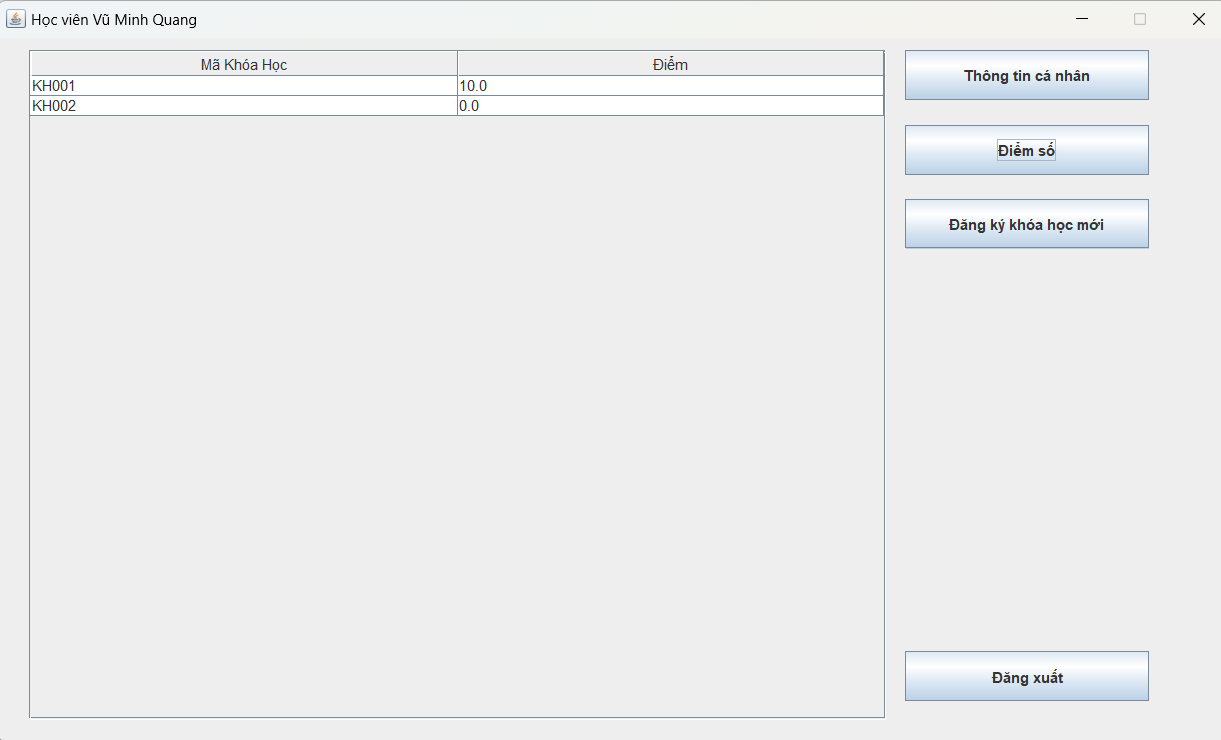
1. **Đăng nhập bằng tài khoản Học viên**



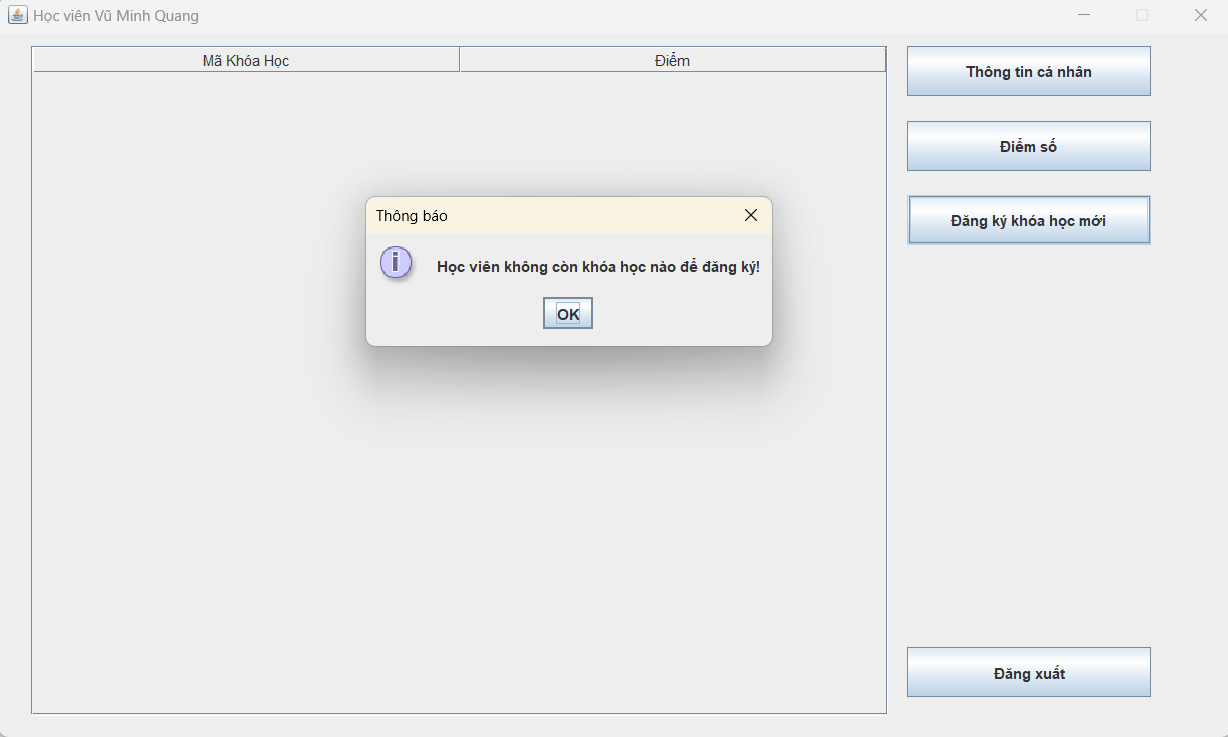
Hình 59: Đăng nhập bằng tài khoản học viên



Hình 60: Hiển thị thông tin cá nhân học viên

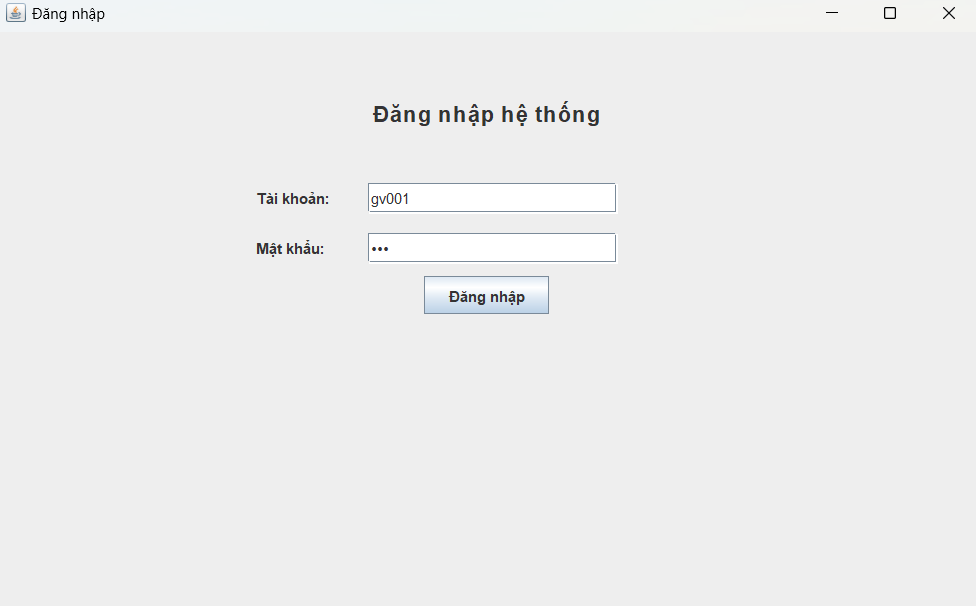


Hình 61: Hiển thị điểm số của học viên

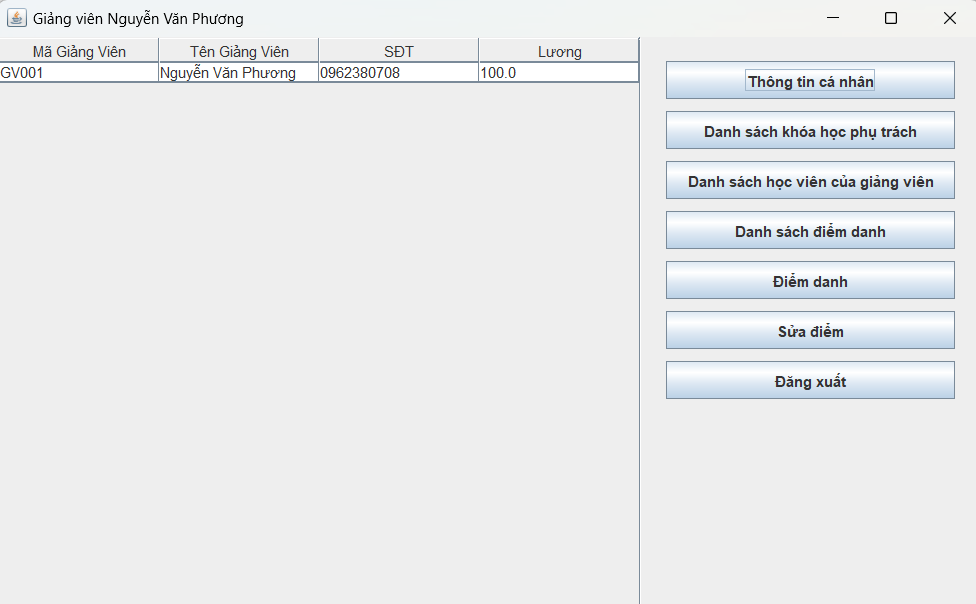


Hình 62: Đăng ký khóa học cho học viên nếu còn khóa học

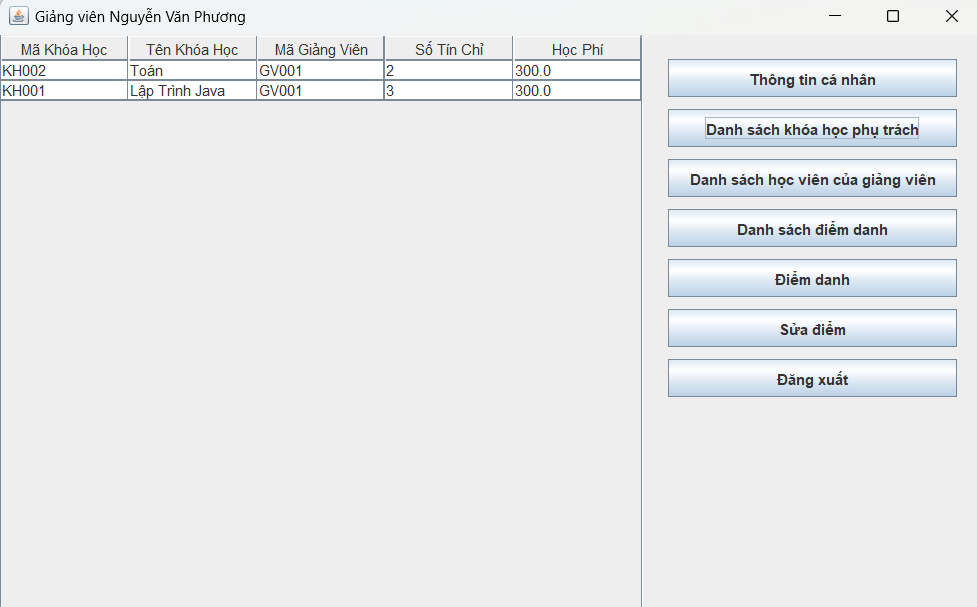
1. **Đăng nhập bằng tài khoản Giảng viên**



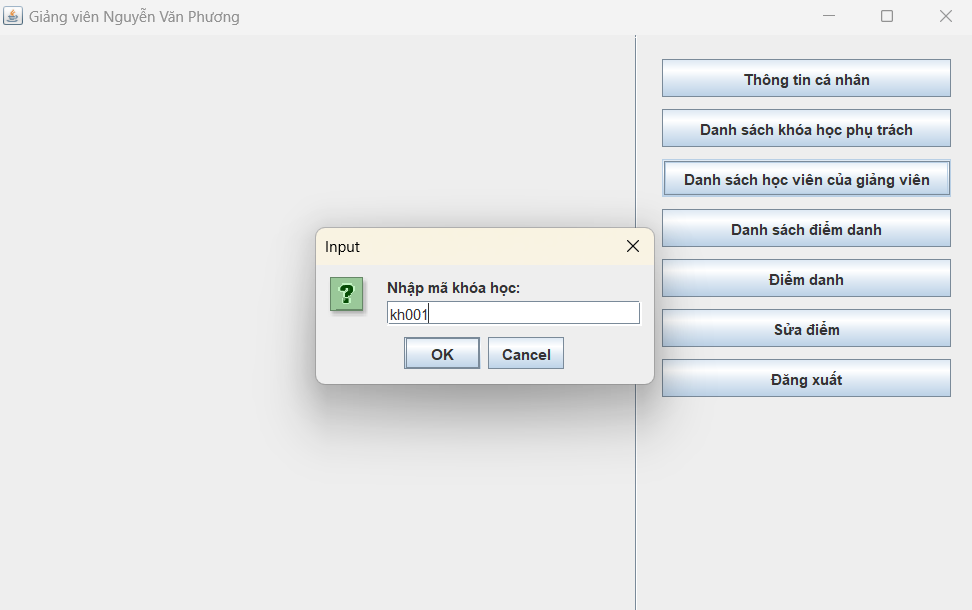
Hình 63: Đăng nhập bằng tài khoản giảng viên



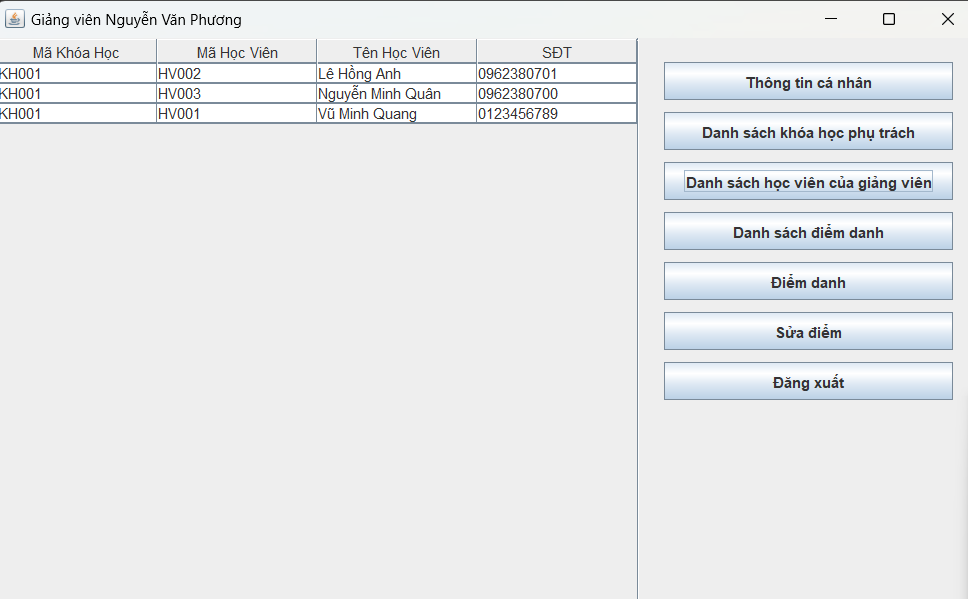
Hình 64: Hiển thị thông tin cá nhân của giảng viên



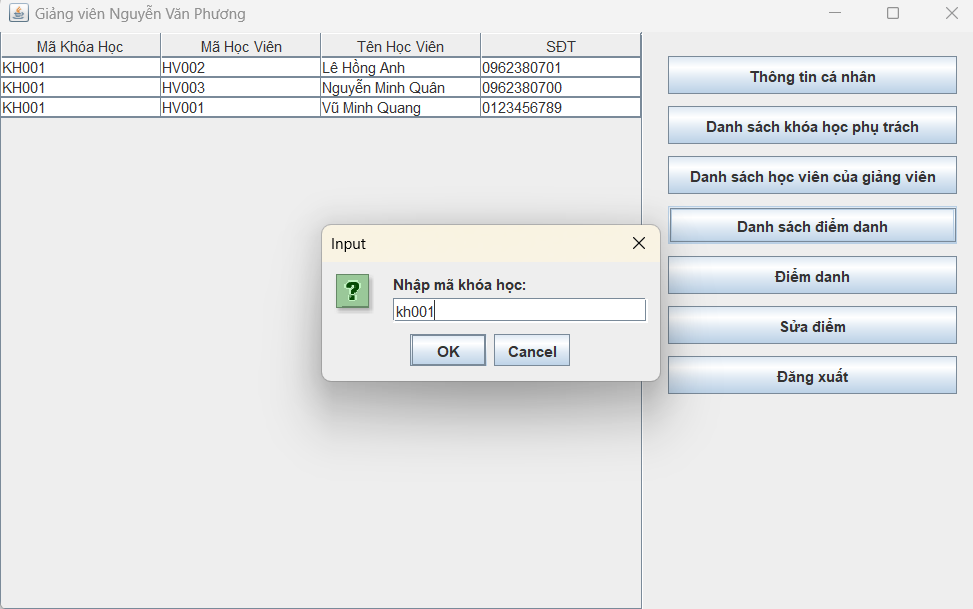
Hình 65: Hiển thị danh sách khóa học phụ trách của giảng viên



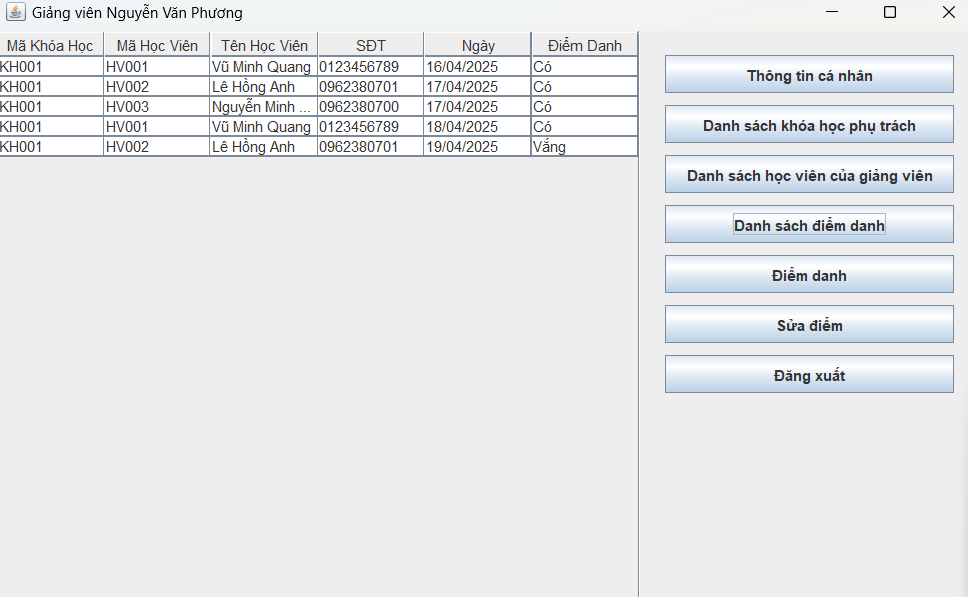
Hình 66: Nhập mã khóa học cần hiển thị danh sách học viên



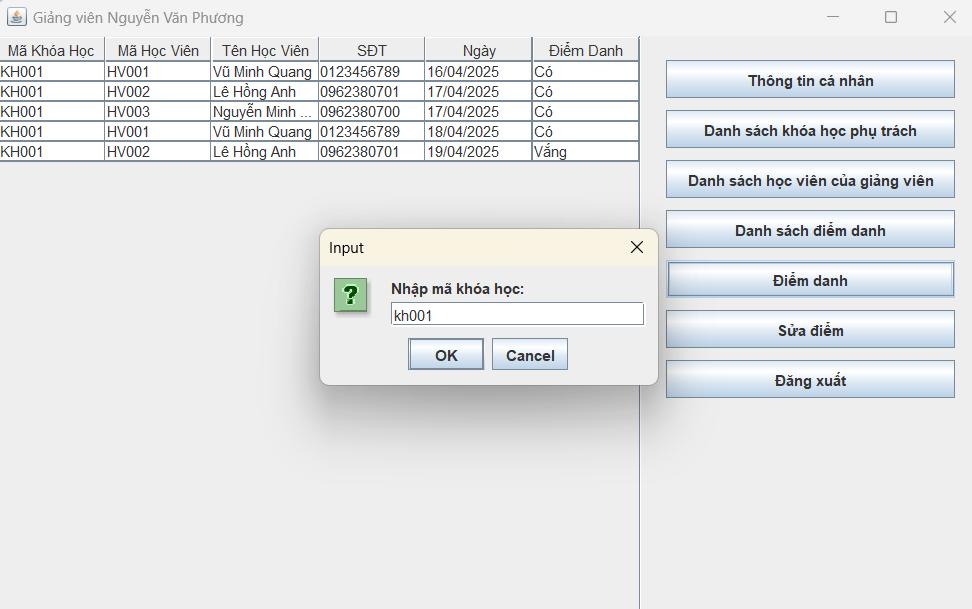
Hình 67: Hiển thị kết quả danh sách học viên của khóa học



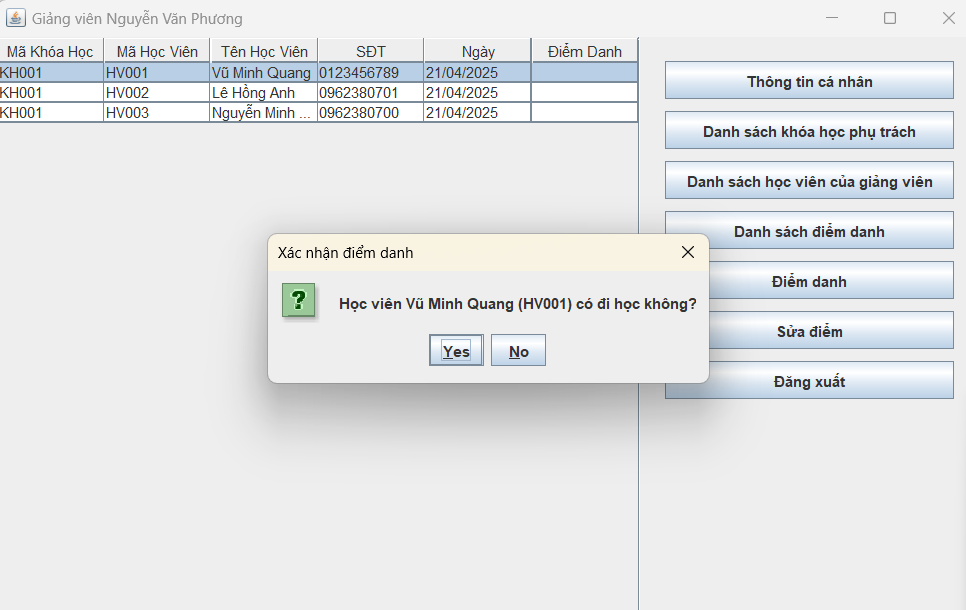
Hình 68: Nhập mã khóa học cần tìm danh sách điểm danh



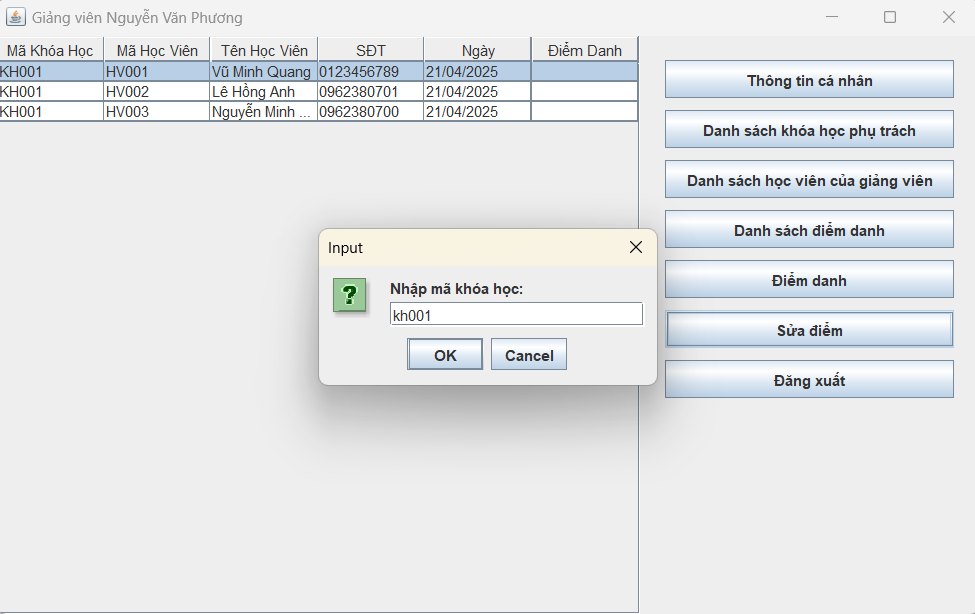
Hình 69: Kết quả danh sách điểm danh của khóa học



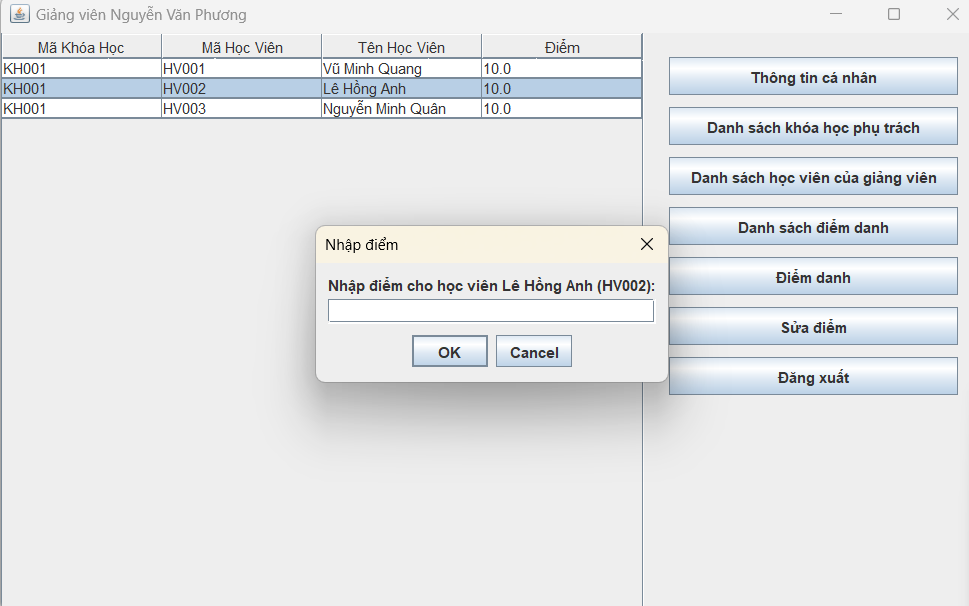
Hình 70: Nhập mã khóa học cần thực hiện điểm danh



Hình 71: Nhấn đúp vào học viên để điểm danh



Hình 72: Nhập khóa học cần điền điểm



Hình 73: Nhấn đúp vào học viên và điển điểm cho học viên