**TRƯỜNG ĐẠI HỌC PHENIKAA**

**KHOA CÔNG NGHỆ THÔNG TIN**



# **Báo cáo bài tập lớn**

# **Môn học Lập trình hướng đối tượng**

Chủ đề nhỏ: Xây dựng chương trình đánh giá điểm rèn luyện cho sinh viên Đại học Phenikaa

# Giảng viên hướng dẫn: Trần Đăng Hoan

Thành viên:

* Nguyễn Minh Đức – 21010602
* Trần Mạnh Cường – 21011584
* Vũ Thành Đạt – 21010589
* Ngô Quang Dương – 21010592

*Hà Nội, Ngày 25 Tháng 4 Năm 2023*

# **MỤC LỤC**

[**MỤC LỤC** 2](#_Toc133187009)

[**CHƯƠNG I : PHÂN TÍCH YÊU CẦU** 4](#_Toc133187010)

[1. **Mục đích của dự án** 4](#_Toc133187011)

[**2.** **Yêu cầu về phần mềm** 4](#_Toc133187012)

[**CHƯƠNG II: PHÂN TÍCH THIẾT KẾ HỆ THỐNG** 5](#_Toc133187013)

[**1.** **Các tác nhân của hệ thống** 5](#_Toc133187014)

[**2.** **Sơ đồ UseCase sử dụng** 5](#_Toc133187015)

[**2.1.** **Usecase** 5](#_Toc133187016)

[**2.2.** **Đặc tả usecase** 5](#_Toc133187017)

[**2.2.1.** **Đăng nhập** 5](#_Toc133187018)

[**2.2.2.** **Quản lý giáo viên** 6](#_Toc133187019)

[**2.2.3.** **Quản lý sinh viên** 7](#_Toc133187020)

[**2.2.4.** **Quản lý điểm rèn luyện** 7](#_Toc133187021)

[**2.2.5.** **Thống kê** 8](#_Toc133187022)

[**3.** **Sơ đồ trạng thái ( State Diagram )** 9](#_Toc133187023)

[**3.1.** **Quản lý tài khoản giáo viên** 9](#_Toc133187024)

[**3.2.** **Quản lý sinh viên** 9](#_Toc133187025)

[**3.3.** **Quản lý điểm rèn luyện** 10](#_Toc133187026)

[**3.4.** **Thống kê** 11](#_Toc133187027)

[**4.** **Sơ đồ hoạt động ( Data Flow Diagram )** 11](#_Toc133187028)

[**4.1.** **Đăng nhập** 11](#_Toc133187029)

[**4.2.** **Quản lý tài khoản giáo viên** 12](#_Toc133187030)

[**4.3.** **Quản lý sinh viên** 12](#_Toc133187031)

[4.4. **Quản** **lý điểm rèn luyện** 13](#_Toc133187032)

[**4.5.** **Thống kê** 14](#_Toc133187033)

[**5.** **Cơ sở dữ liệu** 14](#_Toc133187034)

[**CHƯƠNG III : TÌM HIỂU CÔNG CỤ VÀ NGÔN NGỮ TRIỂN KHAI** 15](#_Toc133187035)

[**1.** **Java** 15](#_Toc133187036)

[**2.** **Design Pattern DAO** 15](#_Toc133187037)

[**3.** **JDBC** 16](#_Toc133187038)

[**CHƯƠNG IV : CÀI ĐẶT HỆ THỐNG** 18](#_Toc133187039)

# **Bảng phân công công việc**

|  |  |  |
| --- | --- | --- |
| Nguyễn Minh Đức | 21010602 | * Viết báo cáo, phân chia nhiệm vụ * Create Database * Đăng nhập ( 2 role: admin, giáo viên |
| Trần Mạnh Cường | 21011584 | * Viết báo cáo * Quản lý thông tin sinh viên : ( Thêm, sửa, xoá , tìm kiếm sinh viên ) |
| Vũ Thành Đạt | 21010589 | * Làm ppt * Quản lý giáo viên ( Thêm, sửa, xoá , tìm kiếm gíao viên ) |
| Ngô Quang Dương | 21010592 | * Làm ppt * Quản lý điểm sinh viên : (Thêm, sửa, xoá, tìm kiếm sinh viên, thống kê học lực, theo khoa, theo lóp ) |

# **CHƯƠNG I : PHÂN TÍCH YÊU CẦU**

1. **Mục đích của dự án**

Do số lượng sinh viên là rất lớn kéo theo lượng thông tin khổng lồ, việc quản lý sinh viên một cách tổng thể là không hề đơn giản. Hệ thống giúp giảng viên quản lý một cách dễ dàng về kết quả rèn luyện sinh viên, mang lại hiệu quả quản lý cao và tránh được các sai sót cũng như tiết kiệm được thời gian.

1. **Yêu cầu về phần mềm**

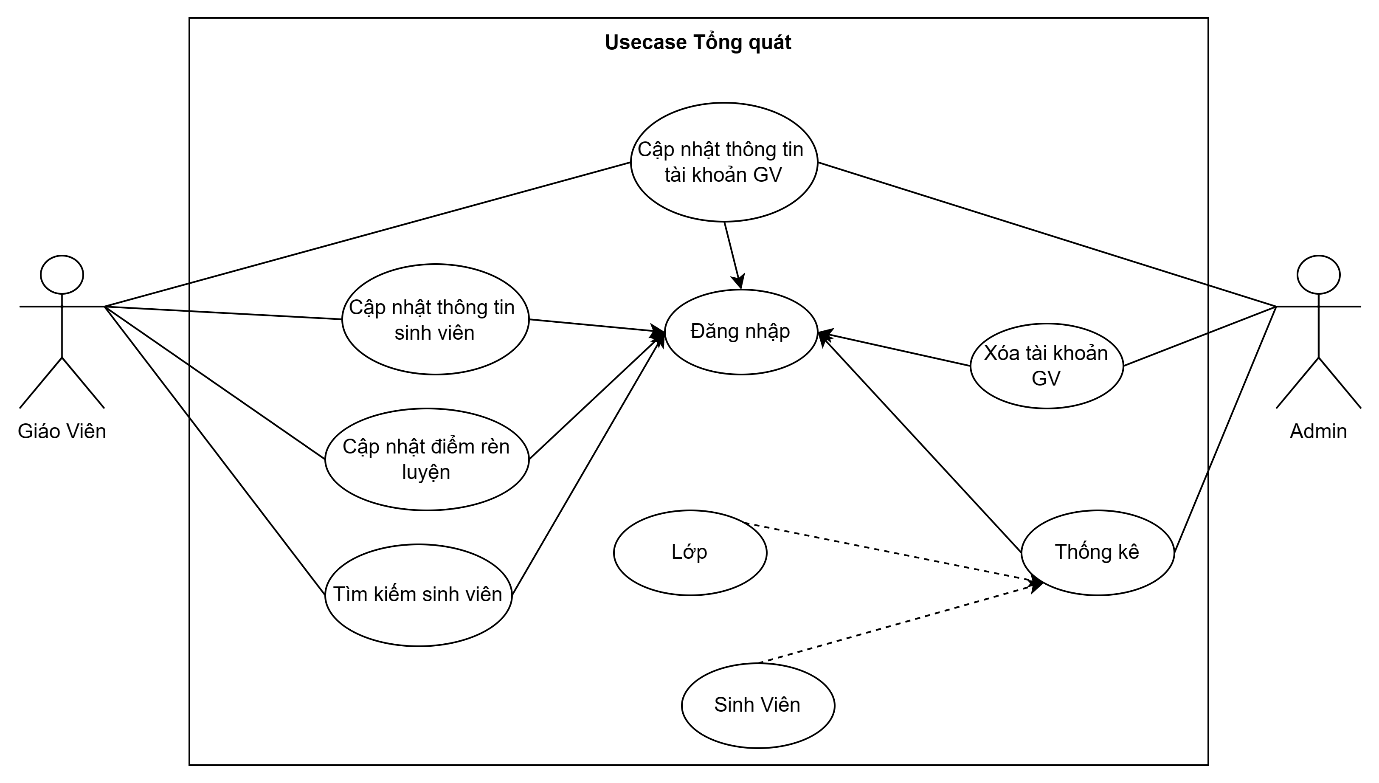
* Yêu cầu giao diện:
* Giao diện tinh gọn, khoa học phù hợp với từng đối tượng như cán bộ quản lý, giáo viên bộ môn
* Giao diện được sử dụng bằng tiếng Việt
* Yêu cầu về bảo mật:
* Tính an toàn về thông tin email, họ và tên, ngày tháng năm sinh, giới tính, khóa, lớp, chương trình đào tạo sinh viên
* Giáo viên đăng nhập bắt buộc phải có tài khoản mật khẩu do trường cấp và giáo viên chỉ có thể xem được thông tin lớp mình mà không thể xem được thông tin lớp của giáo viên khác. Sinh viên đào tạo và trưởng khoa có thể xem và sửa thông tin của tất cả các lớp.
* Yêu cầu về hệ thống:
* Khả năng phục hồi hệ thống khi xảy ra sự cố ngoài ý muốn như mất điện, lỗi hệ thống
* Các thao tác của người dùng đăng nhập vào hệ thống đều được ghi lại thông tin cá nhân, thời gian đăng nhập(xuất), các thao tác trên hệ thống để cán bộ quản lý dễ dàng quản lý.

# **CHƯƠNG II: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

1. **Các tác nhân của hệ thống**

* Admin : Là giáo viên quản lý hệ thống, có thêm quyền hạn là thao tác và sử dụng các chức năng đặc biệt có trong hệ thống như có thể xóa tài khoản Giáo Viên, thống kê.
* Giáo Viên : Là tất cả người sử dụng hệ thống đề phục vụ nghiệp vụ nhập điểm rèn luyện cho sinh viên.

1. **Sơ đồ UseCase sử dụng** 
   1. **Usecase**

****

*Hình 1. UseCase quản lý hệ thống*

* 1. **Đặc tả usecase** 
     1. **Đăng nhập**

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Chức năng | Đăng nhập |
| Tác nhân | Giáo Viên |
| Điều kiện trước | Chưa đăng nhập vào hệ thống |
| Điều kiện sau |  |
| Mô tả | Giáo Viên đăng nhập vào hệ thống để thao tác và sử dụng các chức năng |
| Kịch bản | * Hành động của tác nhân : Nhập tên tài khoản và mật khẩu. * Hành động của hệ thống : Kiểm tra thông tin đăng nhập ; Thông tin đăng nhập chính xác thì truy cập thành công vào hệ thống ; Thông tin đăng nhập không chính xác thì xuất hiện thông báo “Tên tài khoản hoặc mật khẩu không chính xác”. |

* + 1. **Quản lý giáo viên**

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Chức năng | Quản lý Giáo Viên |
| Tác nhân | Admin |
| Điều kiện trước | Đăng nhập thành công hệ thống |
| Điều kiện sau |  |
| Mô tả | Xem thông tin của Giáo Viên |
| Kịch bản | * Hành động của tác nhân : Chọn chức năng “Quản lý Giáo Viên”. * Hành động của hệ thống : Hiển thị danh sách thông tin Giáo Viên từ cơ sở dữ liệu. |

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Chức năng | Quản lý Giáo Viên |
| Tác nhân | Giáo viên |
| Điều kiện trước | Đăng nhập thành công hệ thống |
| Điều kiện sau | Cập nhật vào cơ sở dữ liệu Giáo Viên |
| Mô tả | Cập nhật thông tin của Giáo Viên |
| Kịch bản | * Hành động của tác nhân : Nhập thông tin cần chỉnh sửa ( hoặc thêm mới ). * Hành động của hệ thống : Kiểm tra thông tin có hợp lệ hay không và xuất hiện thông báo “Cập nhật thành công” nếu thông tin được cập nhật thành công hay thông báo “Cập nhật thất bại. Xin mời thử lại” nếu thông tin nhập vào không hợp lệ. |

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Chức năng | Quản lý Giáo Viên |
| Tác nhân | Admin |
| Điều kiện trước | Đăng nhập thành công hệ thống |
| Điều kiện sau | Xóa tài khoản giáo viên khỏi cơ sở dữ liệu |
| Mô tả | Xóa tài khoản giáo viên |
| Kịch bản | * Hành động của tác nhân : Chọn tài khoản giáo viên muốn xóa và thao tác chức năng Xóa * Hành động của hệ thống : Kiểm tra thông tin có tồn tại hay không và xuất hiện thông báo “Xóa thành công” nếu thông tin tồn tại hay thông báo “Xóa thất bại. Xin mời thử lại” nếu thông tin không tồn tại. |

* + 1. **Quản lý sinh viên**

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Chức năng | Quản lý Sinh viên |
| Tác nhân | Giáo viên |
| Điều kiện trước | Đăng nhập thành công hệ thống |
| Điều kiện sau |  |
| Mô tả | Xem thông tin của Sinh viên |
| Kịch bản | * Hành động của tác nhân : Chọn chức năng “Quản lý Sinh viên”. * Hành động của hệ thống : Hiển thị danh sách thông tin Sinh viên từ cơ sở dữ liệu. |

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Chức năng | Quản lý Sinh viên |
| Tác nhân | Giáo viên |
| Điều kiện trước | Đăng nhập thành công hệ thống |
| Điều kiện sau | Cập nhật vào cơ sở dữ liệu Sinh viên |
| Mô tả | Cập nhật thông tin của Sinh viên |
| Kịch bản | * Hành động của tác nhân : Nhập thông tin cần chỉnh sửa ( hoặc thêm mới ). * Hành động của hệ thống : Kiểm tra thông tin có hợp lệ hay không và xuất hiện thông báo “Cập nhật thành công” nếu thông tin được cập nhật thành công hay thông báo “Cập nhật thất bại. Xin mời thử lại” nếu thông tin nhập vào không hợp lệ. |

* + 1. **Quản lý điểm rèn luyện**

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Chức năng | Quản lý Điểm rèn luyện |
| Tác nhân | Giáo viên |
| Điều kiện trước | Đăng nhập thành công hệ thống |
| Điều kiện sau |  |
| Mô tả | Xem thông tin của điểm rèn luyện |
| Kịch bản | * Hành động của tác nhân : Chọn chức năng “Quản lý Điểm rèn luyện”. * Hành động của hệ thống : Hiển thị danh sách thông tin Điểm rèn luyện từ cơ sở dữ liệu. |

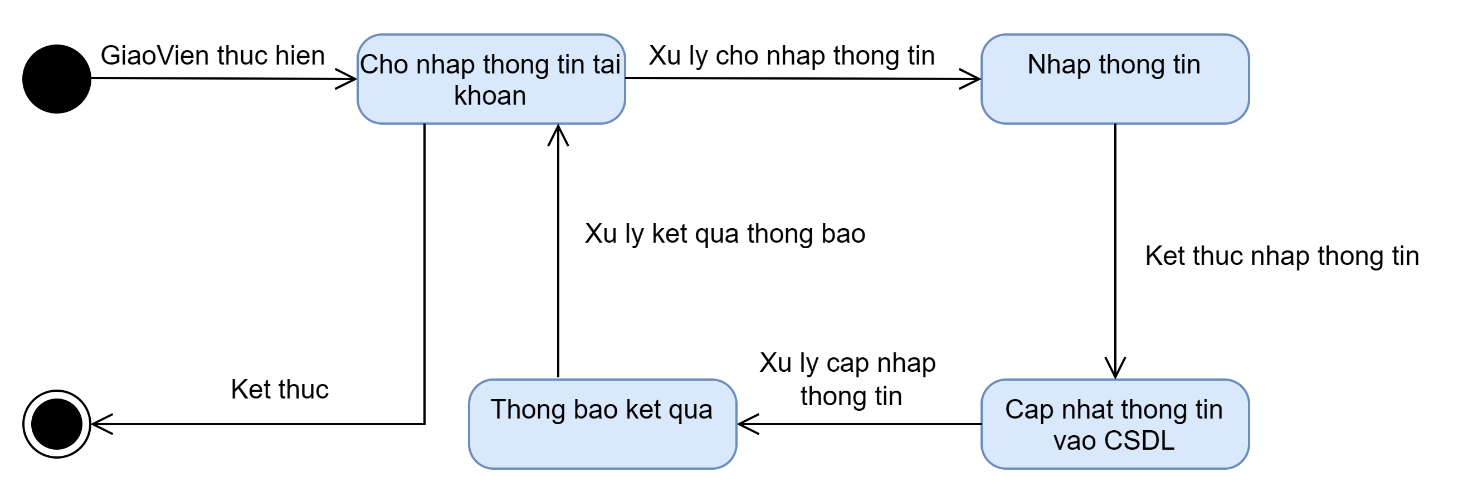
|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Chức năng | Quản lý Điểm rèn luyện |
| Tác nhân | Giáo viên |
| Điều kiện trước | Đăng nhập thành công hệ thống |
| Điều kiện sau | Cập nhật vào cơ sở dữ liệu Sinh viên |
| Mô tả | Cập nhật thông tin điểm rèn luyện của Sinh viên |
| Kịch bản | * Hành động của tác nhân : Nhập thông tin điểm rèn luyện cần chỉnh sửa ( hoặc thêm mới ). * Hành động của hệ thống : Kiểm tra thông tin có hợp lệ hay không và xuất hiện thông báo “Cập nhật thành công” nếu thông tin được cập nhật thành công hay thông báo “Cập nhật thất bại. Xin mời thử lại” nếu thông tin nhập vào không hợp lệ. |

* + 1. **Thống kê**

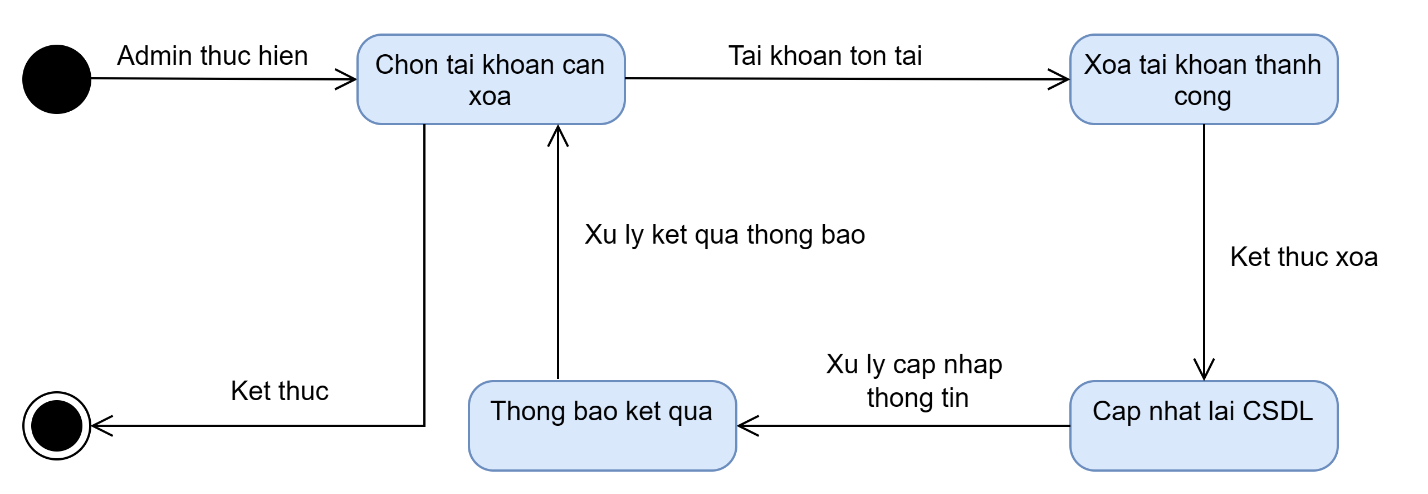
|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Chức năng | Thống kê dữ liệu theo sinh viên |
| Tác nhân | Admin |
| Điều kiện trước | Đăng nhập thành công hệ thống |
| Điều kiện sau |  |
| Mô tả | Xem các số liệu thống kê điểm rèn luyện theo sinh viên |
| Kịch bản | * Hành động của tác nhân : Thao tác chọn lớp muốn thống kê cần xem. * Hành động của hệ thống : Hiển thị thông tin số liệu đáp ứng kết quả thống kê. |

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Chức năng | Thống kê dữ liệu theo lớp |
| Tác nhân | Admin |
| Điều kiện trước | Đăng nhập thành công hệ thống |
| Điều kiện sau |  |
| Mô tả | Xem các số liệu thống kê điểm rèn luyện theo lớp |
| Kịch bản | * Hành động của tác nhân : Thao tác chọn khoa muốn thống kê cần xem. * Hành động của hệ thống : Hiển thị thông tin số liệu đáp ứng kết quả thống kê. |

1. **Sơ đồ trạng thái ( State Diagram )**
   1. **Quản lý tài khoản giáo viên**

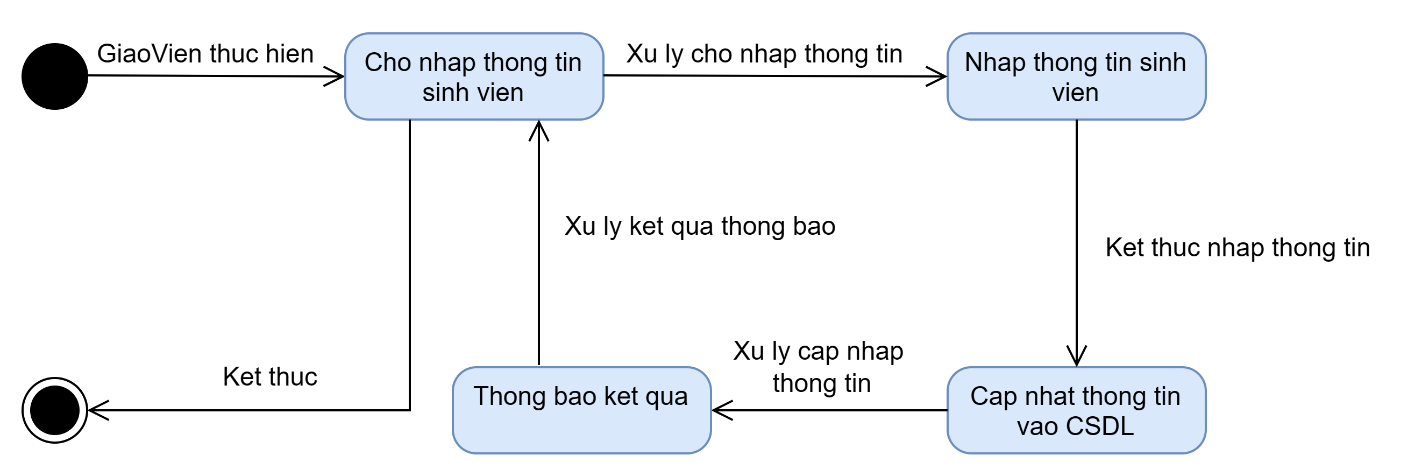
****

*Hình 2. State Diagram – Cập nhật thông tin tài khoản*

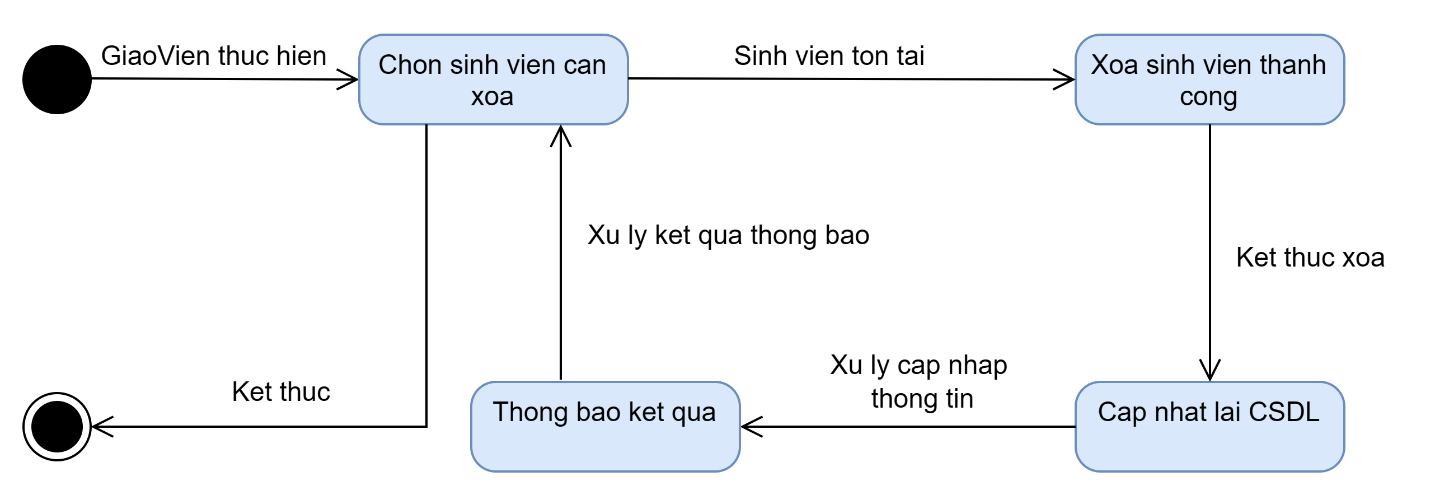
**

*Hình 3. State Diagram – Xóa tài khoản giáo viên*

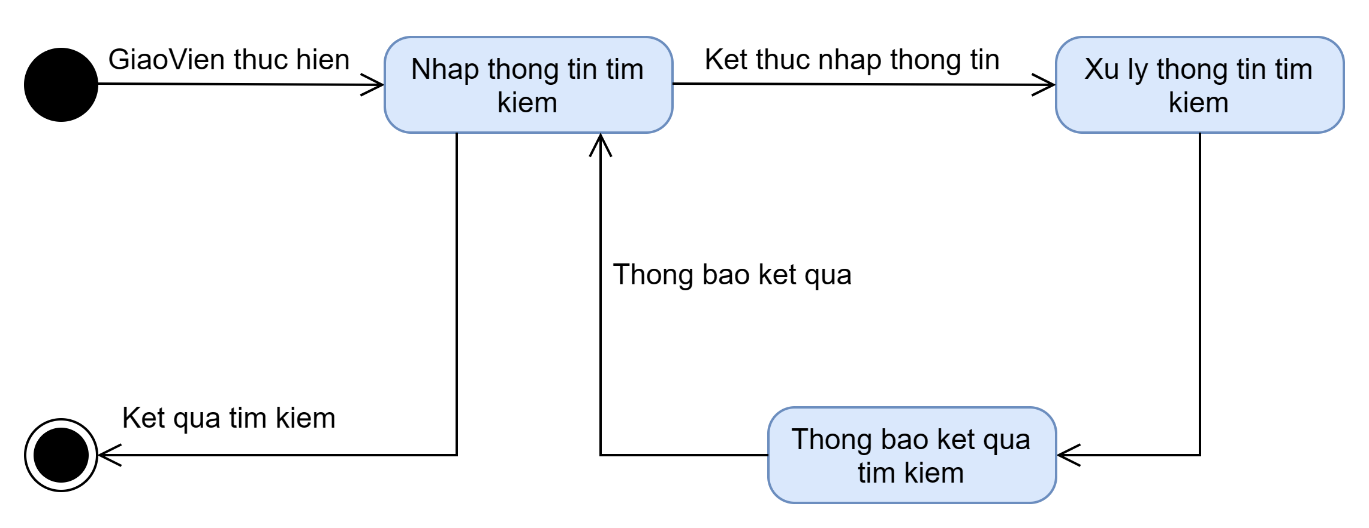
* 1. **Quản lý sinh viên**

**

*Hình 4. State Diagram – Cập nhật thông tin sinh viên*

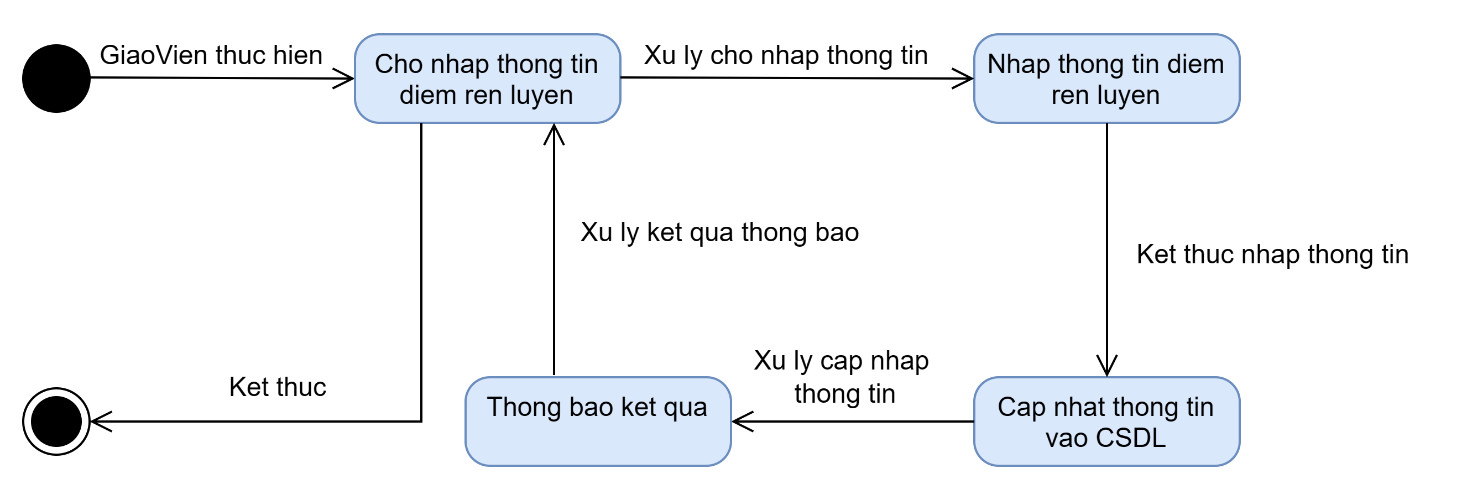
**

*Hình 5. State Diagram – Xóa sinh viên*

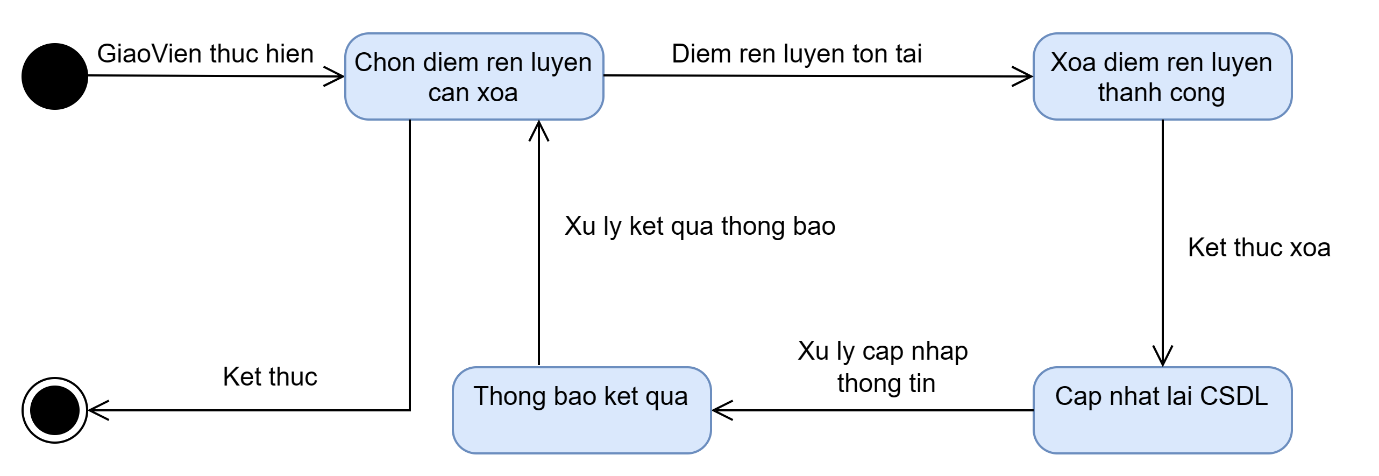
**

*Hình 6. State Diagram – Tìm kiếm sinh viên*

* 1. **Quản lý điểm rèn luyện**

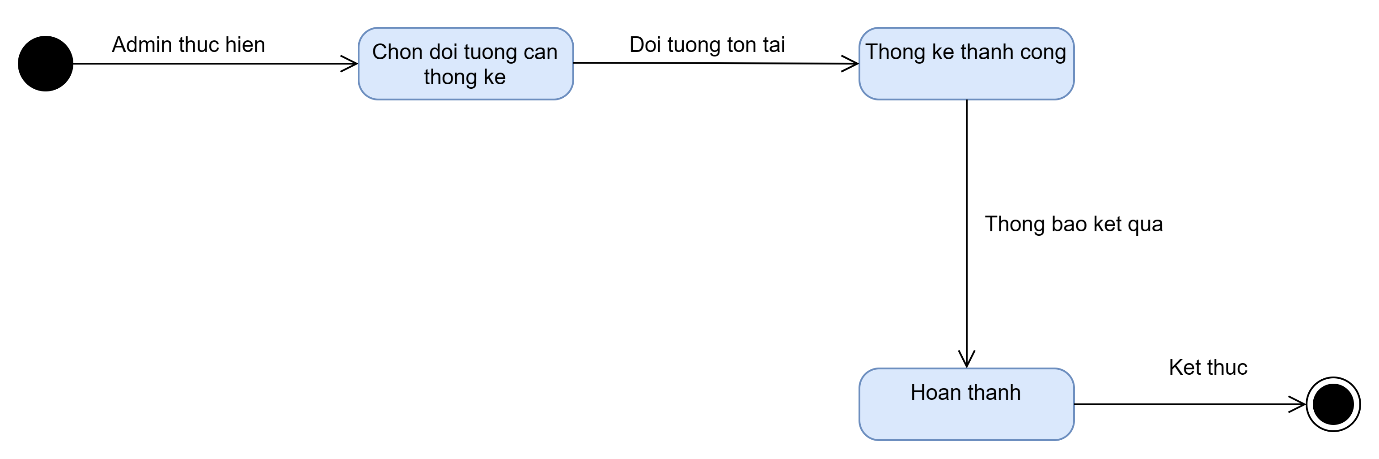
**

*Hình 7. State Diagram – Cập nhật điểm rèn luyện*

**

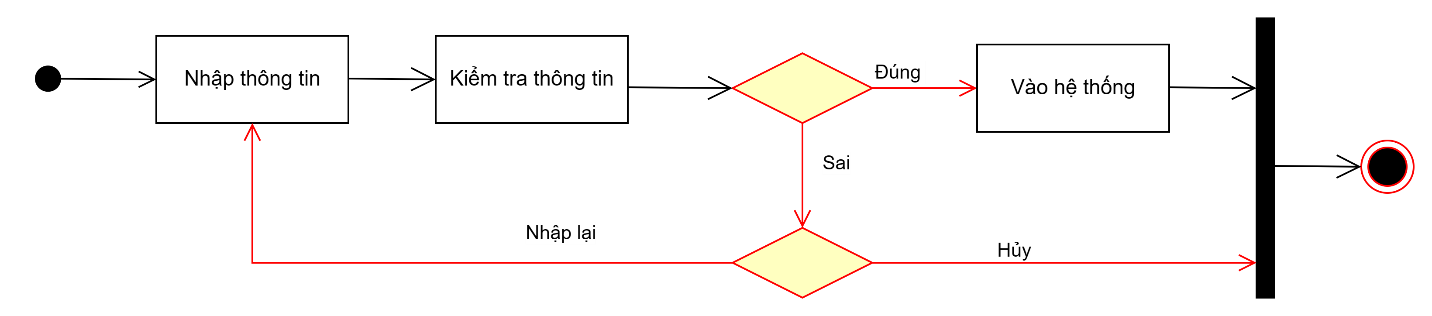
*Hình 8. State Diagram – Xóa điểm rèn luyện*

* 1. **Thống kê**

****

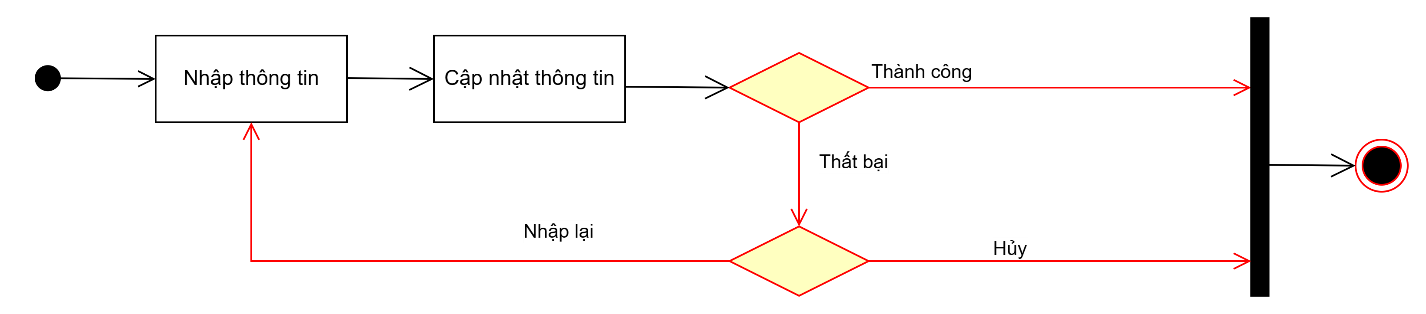
*Hình 9. State Diagram – Thống kê*

1. **Sơ đồ hoạt động ( Data Flow Diagram )**
   1. **Đăng nhập**

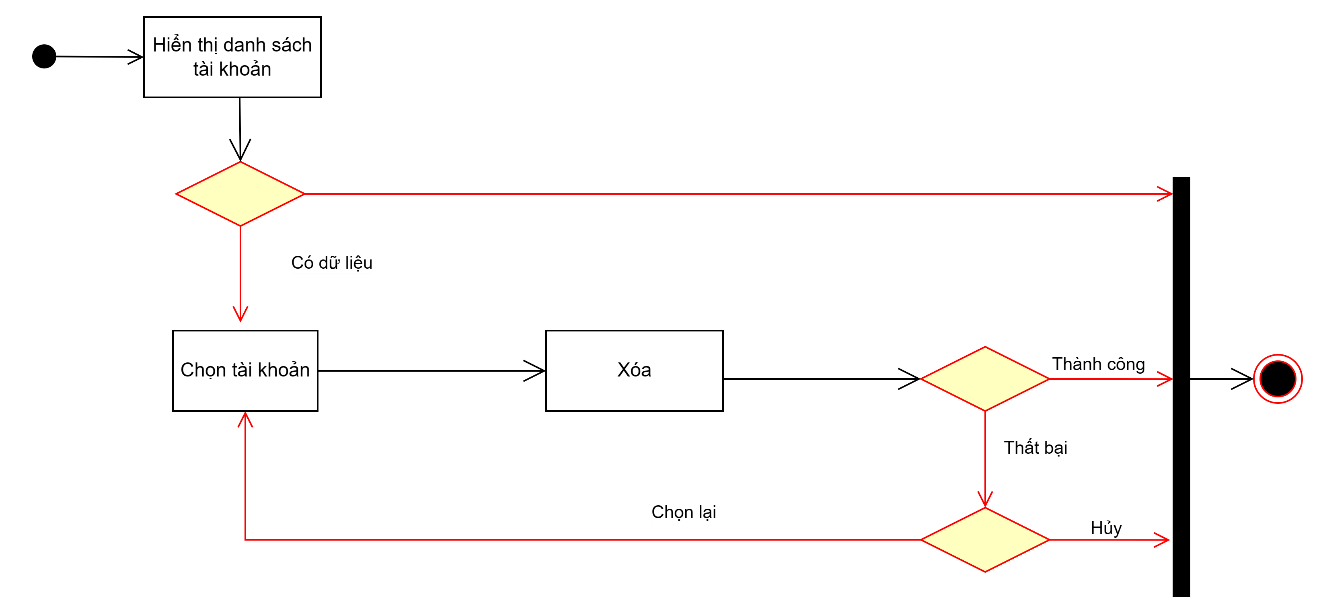
****

*Hình 10. DFD – Đăng nhập*

* 1. **Quản lý tài khoản giáo viên**

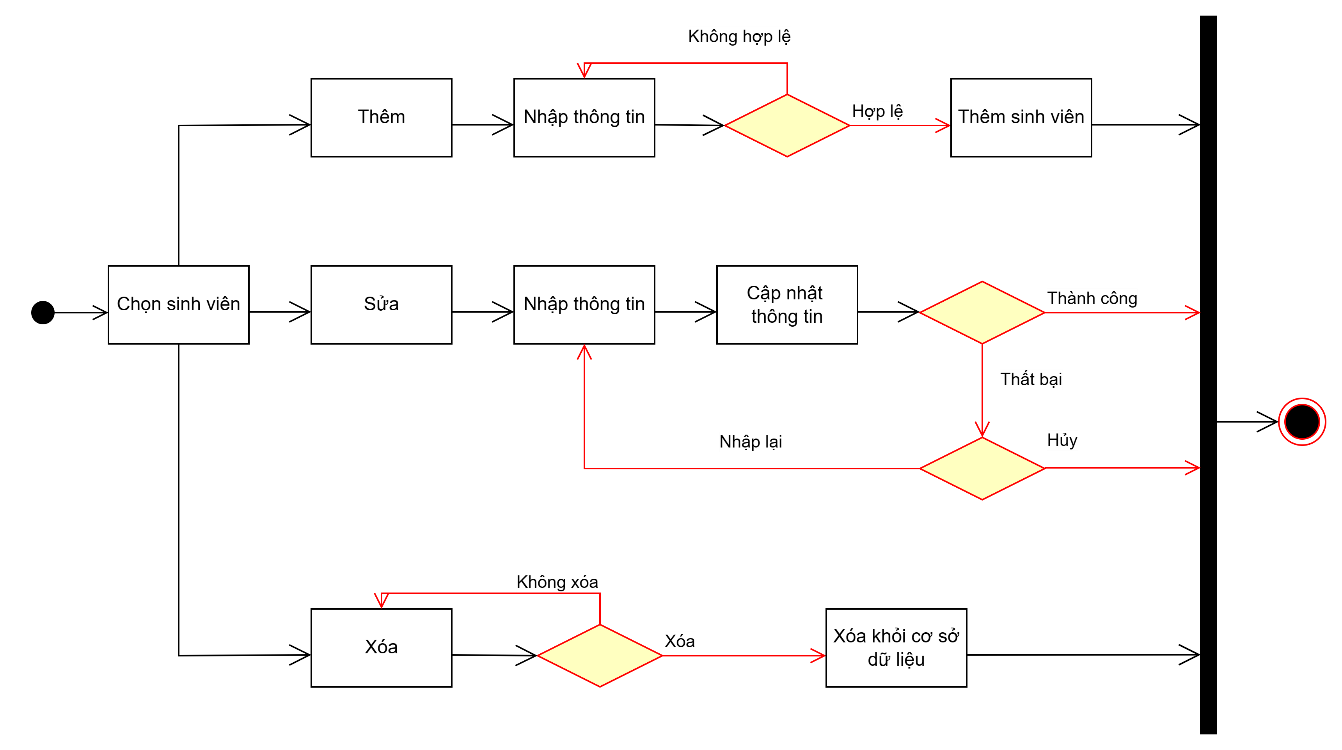
****

*Hình 11. DFD – Cập nhật thông tin tài khoản*

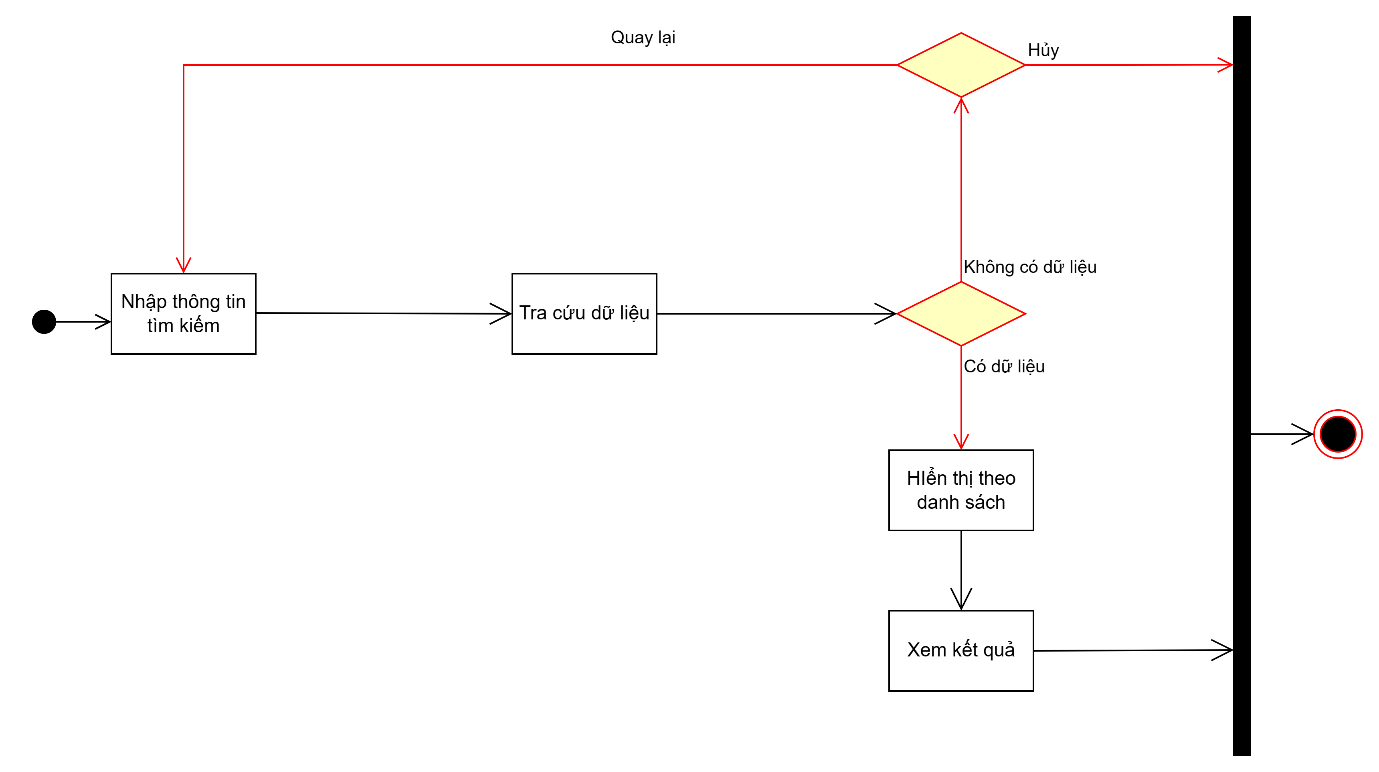


*Hình 12. DFD – Xóa tài khoản giáo viên*

* 1. **Quản lý sinh viên**

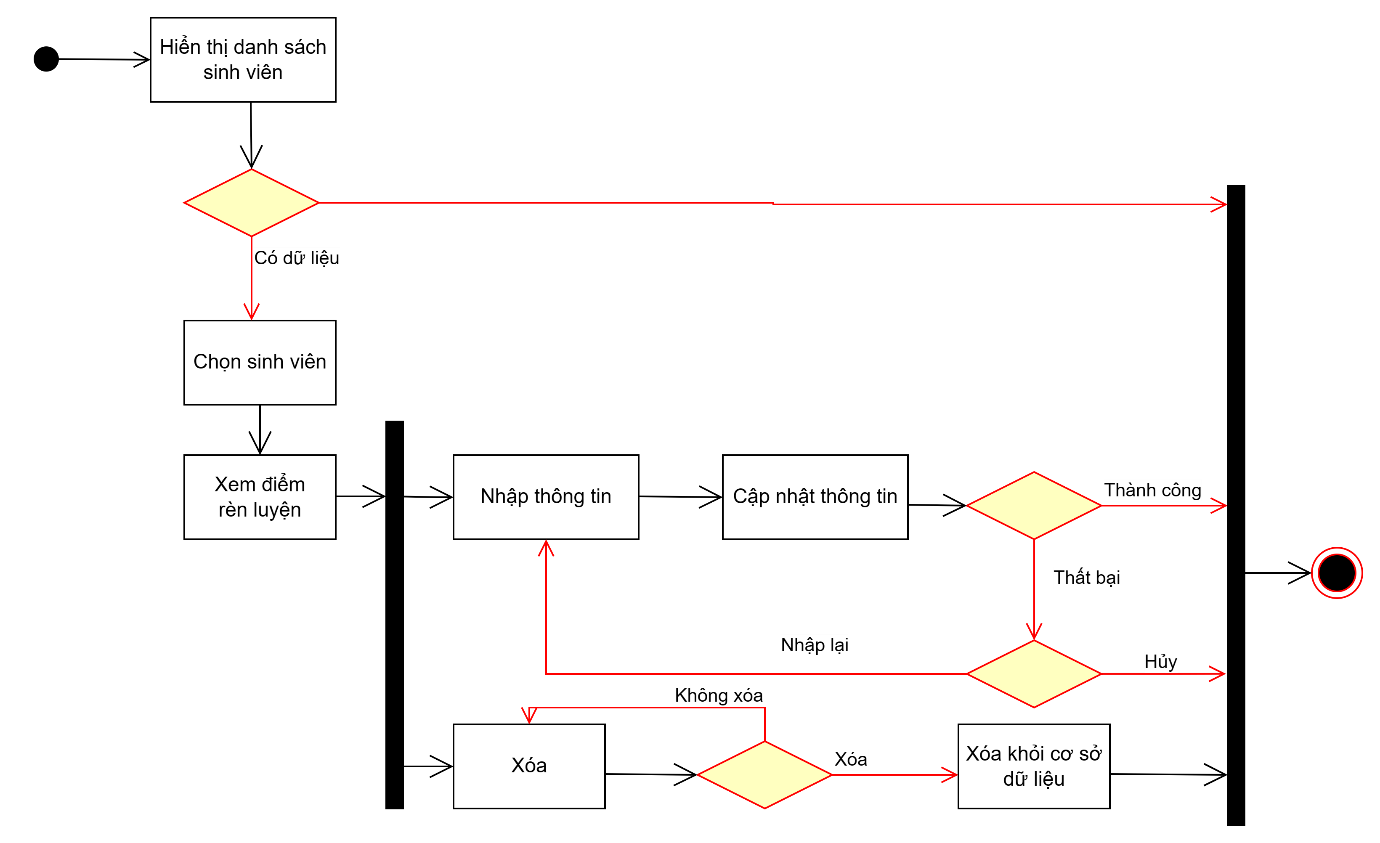


*Hình 13. DFD – Cập nhật thông tin sinh viên*

**

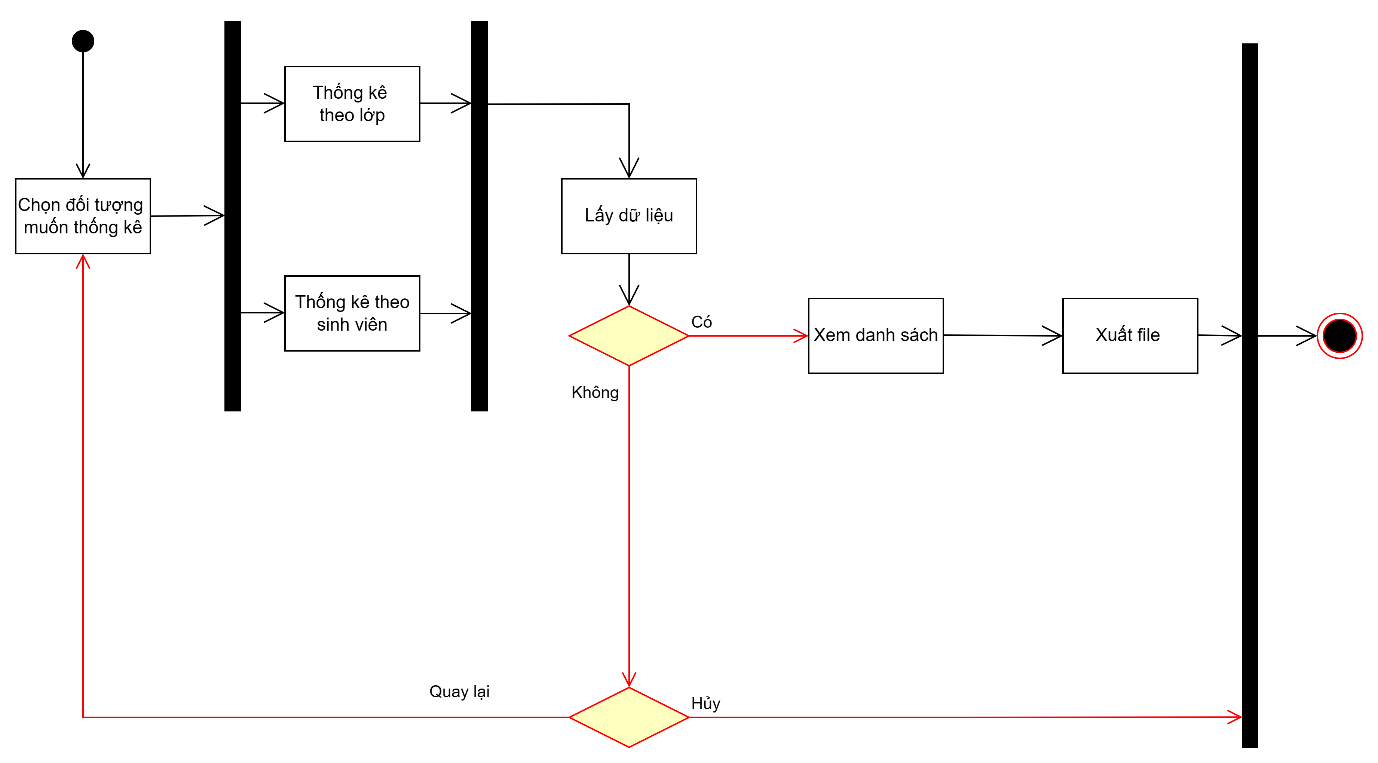
*Hình 14. DFD – Tìm kiếm sinh viên*

* 1. **Quản** **lý điểm rèn luyện**



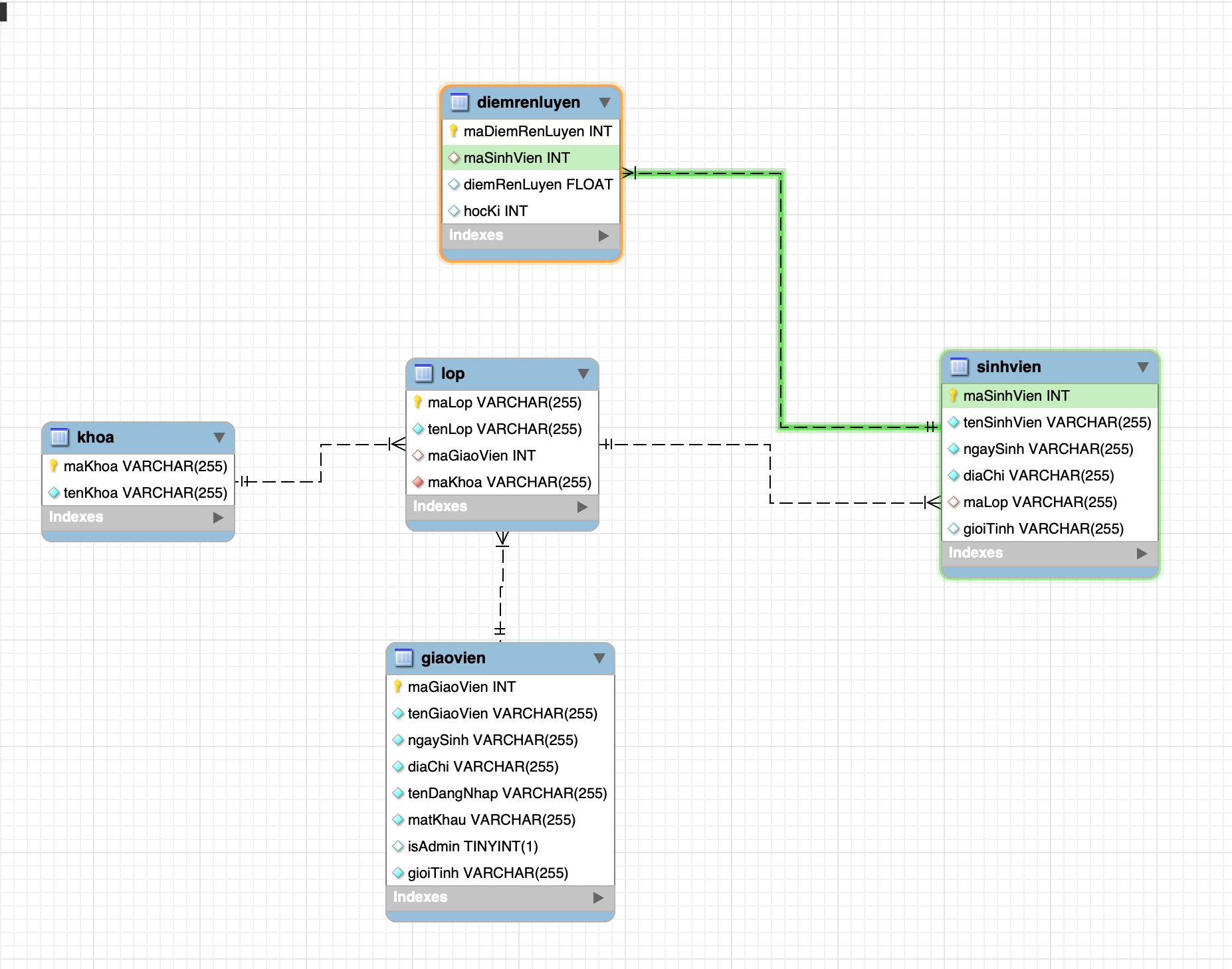
*Hình 15. DFD – Quản lý điểm rèn luyện*

* 1. **Thống kê**

**

*Hình 16. DFD – Thống kê và xuất file*

1. **Cơ sở dữ liệu**



*Hình 17. Cơ sở dữ liệu*

# **CHƯƠNG III : TÌM HIỂU CÔNG CỤ VÀ NGÔN NGỮ TRIỂN KHAI**

1. **Java**



Java là một ngôn ngữ lập trình hướng đối tượng được phát triển bởi Sun Microsystems vào năm 1995. Ngôn ngữ này được thiết kế để chạy trên nền tảng đa nền tảng, cho phép các lập trình viên phát triển các ứng dụng đa nền tảng như ứng dụng desktop, ứng dụng web và ứng dụng di động.

Java có các đặc điểm sau:

* Hướng đối tượng: Java hỗ trợ hướng đối tượng với các tính năng như kế thừa, đa hình, trừu tượng hóa và đóng gói.
* Tính bảo mật: Java được thiết kế để có tính bảo mật cao, cho phép người dùng chạy ứng dụng Java trên mạng một cách an toàn.
* Garbage collector: Java có bộ thu gom rác tự động để giải phóng bộ nhớ khi các đối tượng không còn được sử dụng.
* Nền tảng đa nền tảng: Java được thiết kế để chạy trên nhiều nền tảng khác nhau như Windows, Mac OS và Linux.
* Cú pháp đơn giản: Java có cú pháp dễ đọc và dễ viết, giúp cho các lập trình viên dễ dàng học và sử dụng ngôn ngữ này.

Java có thể được sử dụng để phát triển các ứng dụng như ứng dụng desktop, ứng dụng web, ứng dụng di động, các hệ thống quản lý cơ sở dữ liệu và các ứng dụng máy tính khác.

1. **Design Pattern DAO**



DAO là viết tắt của *Data Access Object*, là một mẫu thiết kế (design pattern) cung cấp một giao diện trừu tượng để tương tác với cơ sở dữ liệu hoặc nguồn dữ liệu. Mục đích chính của mẫu thiết kế này là tách logic truy cập dữ liệu khỏi logic kinh doanh của ứng dụng.

Mục đích chính của mẫu thiết kế DAO là tách biệt logic truy cập dữ liệu ra khỏi các lớp xử lý kinh doanh trong ứng dụng. Theo cách này, nó cho phép chúng ta cập nhật cơ sở dữ liệu mà không làm ảnh hưởng đến các lớp khác trong ứng dụng.

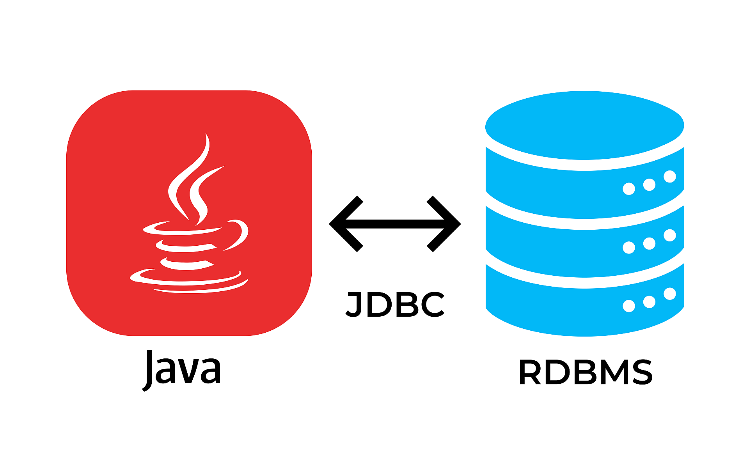
Một DAO thường được thiết kế để thực hiện các hoạt động cơ bản như thêm, sửa, xóa và truy vấn dữ liệu từ cơ sở dữ liệu. Thông thường, các DAO sẽ cung cấp một giao diện trừu tượng cho phép các lớp khác trong ứng dụng tương tác với cơ sở dữ liệu mà không cần biết chi tiết cài đặt của DAO.

Một DAO thường bao gồm các phương thức như create, read, update và delete (CRUD) để thực hiện các thao tác cơ bản với dữ liệu. DAO cũng có thể cung cấp các phương thức tìm kiếm và sắp xếp dữ liệu để giúp cho việc truy xuất dữ liệu trở nên dễ dàng hơn.

Ngoài ra, DAO cũng có thể được sử dụng để giảm thiểu sự phụ thuộc vào cơ sở dữ liệu bằng cách cung cấp một lớp trung gian giữa ứng dụng và cơ sở dữ liệu. Điều này có nghĩa là, nếu chúng ta muốn thay đổi cơ sở dữ liệu của mình, chúng ta chỉ cần thay đổi lớp DAO thay vì phải sửa đổi tất cả các lớp xử lý kinh doanh trong ứng dụng.

Tóm lại, mẫu thiết kế DAO cung cấp một cách tiếp cận tốt để tách biệt logic truy cập dữ liệu và logic kinh doanh trong ứng dụng, giúp cho ứng dụng trở nên dễ dàng bảo trì và mở rộng.

1. **JDBC**



JDBC (*Java Database Connectivity*) là một API trong Java được sử dụng để kết nối và tương tác với các cơ sở dữ liệu quan hệ (Relational Database) từ Java code. JDBC là một trong những công nghệ cơ bản cho phép các ứng dụng Java truy cập và thao tác với các hệ quản trị cơ sở dữ liệu (DBMS) như MySQL, Oracle, SQL Server, và PostgreSQL.

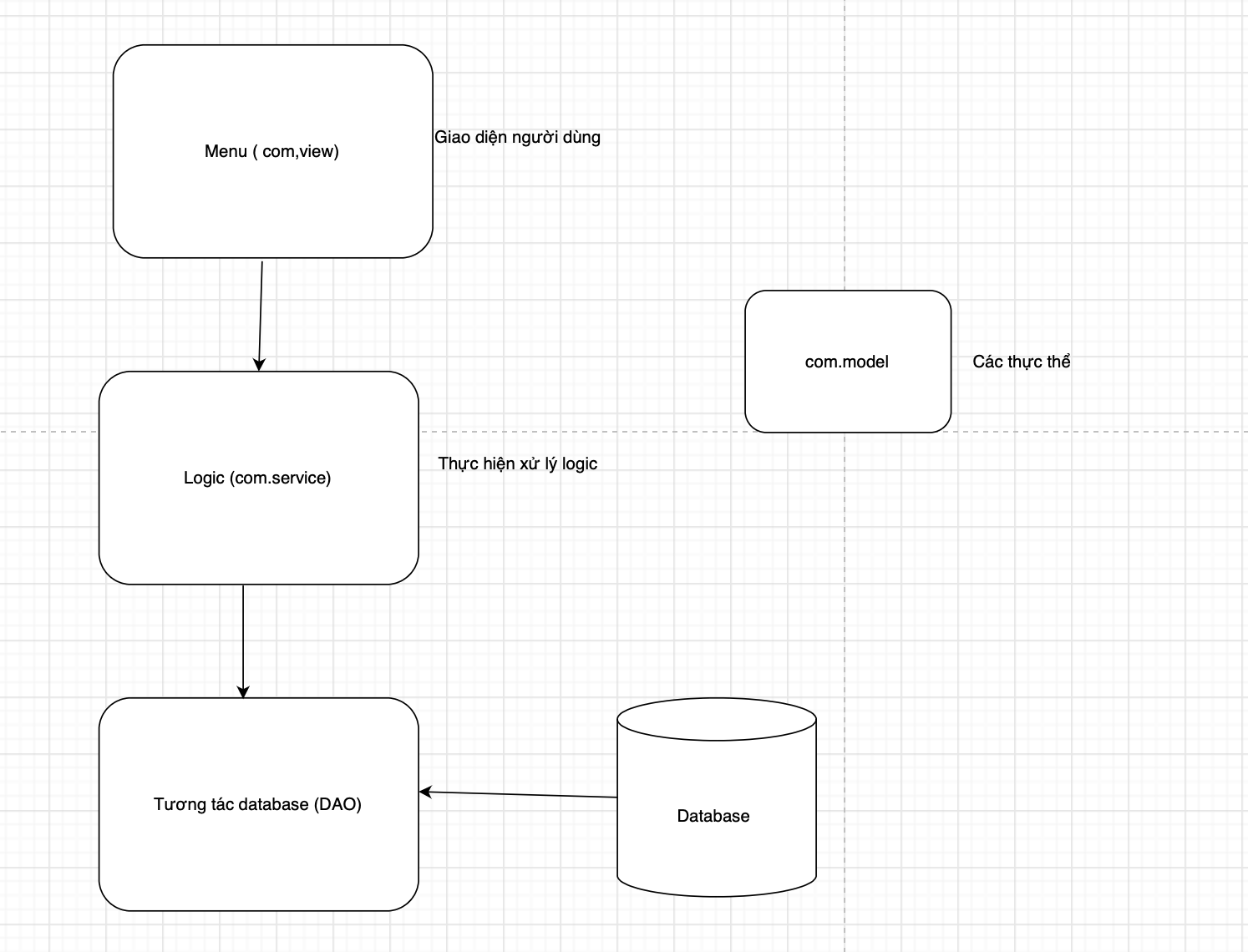
JDBC bao gồm một số giao diện và lớp để giúp tạo kết nối và thực hiện các thao tác cơ bản với cơ sở dữ liệu như tạo, đọc, cập nhật và xóa dữ liệu. Với JDBC, người lập trình Java có thể viết các ứng dụng tương tác với cơ sở dữ liệu một cách dễ dàng, linh hoạt và đáng tin cậy.

Các bước để sử dụng JDBC bao gồm:

* Tạo kết nối đến cơ sở dữ liệu bằng cách sử dụng một URL kết nối (connection string).
* Tạo đối tượng PreparedStatement để thực thi các truy vấn SQL và các thao tác trên cơ sở dữ liệu.
* Thực thi các truy vấn và các thao tác trên cơ sở dữ liệu bằng cách sử dụng các phương thức như executeQuery() hoặc executeUpdate() để truy xuất hoặc thay đổi dữ liệu trong cơ sở dữ liệu.
* Đóng kết nối đến cơ sở dữ liệu sau khi hoàn thành các thao tác.

JDBC cung cấp cho các nhà phát triển Java một cách tiếp cận mạnh mẽ để kết nối và tương tác với cơ sở dữ liệu từ Java code. Nó cho phép người lập trình có thể tương tác với cơ sở dữ liệu một cách nhanh chóng và hiệu quả, cung cấp cho người dùng kết quả chính xác và nhanh chóng.

# **CHƯƠNG IV: CÀI ĐẶT HỆ THỐNG.**

****