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

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN HỆ THỐNG THÔNG TIN**

**-----🙞🙞🙞🙞🙞-----**

A logo with blue letters and a ball

Description automatically generated

ĐỀ TÀI CUỐI KỲ MÔN HỆ QUẢN TRỊ CSDL

TÊN ĐỀ TÀI:

QUẢN LÝ HỆ THỐNG ĐĂNG KÝ MÔN HỌC

GVHD: TS. Nguyễn Thành Sơn

Lớp HP: DBMS330284\_23\_1\_05

Nhóm thực hiện: Nhóm 14

Học kỳ: 1

Năm học: 2023 – 2024

Thành phố Hồ Chí Minh, tháng 11, năm 2023

**DANH SÁCH SINH VIÊN NHÓM THỰC HIỆN**

**HỌC KÌ 1 NĂM HỌC 2023 - 2024**

**Nhóm 14**

***Đề tài: Quản lý hệ thống đăng ký môn học***

|  |  |
| --- | --- |
| 21133081 | Nguyễn Hoàn Thảo |
| 21133088 | Tăng Huỳnh Minh Tiến |
| 21133090 | Nguyễn Thành Trung |
| 21133113 | Nguyễn Anh Tuấn |

**NHẬN XÉT CỦA GV**Tp. Hồ Chí Minh, ngày 20, tháng 11, năm 2023

Giảng viên chấm điểm

**MỤC LỤC**

[**CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG 1**](#_Toc151414551)

[**1. Đặt tả bài toán 1**](#_Toc151414552)

[1.1. Mô tả bài toán 1](#_Toc151414553)

[1.3. Mô tả chức năng của bài toán 2](#_Toc151414554)

[1.2. Nghiệp vụ bài toán 3](#_Toc151414555)

[1.4. Tính học phí 4](#_Toc151414556)

[**CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 6**](#_Toc151414557)

[**1. Thiết kế cơ sở dữ liệu mức quan niệm 6**](#_Toc151414558)

[**2. Thiết kế cơ sở dữ liệu mức logic 8**](#_Toc151414559)

[**3. Các ràng buộc cần có 9**](#_Toc151414560)

[**4. Cài đặt các CSDL và các ràng buộc 10**](#_Toc151414561)

[**5. Các view 14**](#_Toc151414562)

[**6. Các trigger 15**](#_Toc151414563)

[**CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG 25**](#_Toc151414564)

[**1. Kết nối cơ sở dữ liệu 25**](#_Toc151414565)

[**2. Chức năng của người quản lý 26**](#_Toc151414566)

[2.1. Kiểm tra tài khoản mật khẩu của người quản lý: 26](#_Toc151414567)

[2.2. Tạo tài khoản 27](#_Toc151414568)

[2.3. Đổi mật khẩu 28](#_Toc151414569)

[2.4. Quản lý giảng viên 28](#_Toc151414570)

[*2.4.1. Hiển thị danh sách thông tin giảng viên: 28*](#_Toc151414571)

[*2.4.2. Tìm kiếm thông tin một giảng viên: 29*](#_Toc151414572)

[*2.4.3. Xoá thông tin một giảng viên: 30*](#_Toc151414573)

[*2.4.4. Thêm một giảng viên: 32*](#_Toc151414574)

[*2.4.5. Cập nhật thông tin một giảng viên: 33*](#_Toc151414575)

[2.5. Quản lý học viên 36](#_Toc151414576)

[*2.5.1. Hiển thị danh sách thông tin học viên: 36*](#_Toc151414577)

[*2.5.2. Tìm kiếm thông tin một học viên: 36*](#_Toc151414578)

[*2.5.3. Xoá thông tin một học viên: 37*](#_Toc151414579)

[*2.5.4. Thêm một học viên: 39*](#_Toc151414580)

[*2.5.5. Cập nhật thông tin một học viên: 41*](#_Toc151414581)

[2.6. Quản lý lớp học 43](#_Toc151414582)

[*2.6.1. Hiển thị danh sách thông tin lớp học: 43*](#_Toc151414583)

[*2.6.2. Tìm kiếm thông tin một lớp học: 44*](#_Toc151414584)

[*2.6.3. Xoá thông tin một lớp học: 45*](#_Toc151414585)

[*2.6.4. Thêm một lớp học: 46*](#_Toc151414586)

[*2.6.5. Cập nhật thông tin một lớp học: 48*](#_Toc151414587)

[**3. Chức năng của học viên 50**](#_Toc151414588)

[3.1. Đăng nhập vào hệ thống: 50](#_Toc151414589)

[3.2. Xem các môn có thể đăng kí: 52](#_Toc151414590)

[3.3. Xem các môn đã đăng kí: 52](#_Toc151414591)

[3.4. Xem danh sách lớp của một môn: 53](#_Toc151414592)

[3.5. Học viên đăng ký lớp học: 53](#_Toc151414593)

[3.6. Học viên xóa đăng ký lớp học: 54](#_Toc151414594)

[3.7. Học viên chuyển lớp: 56](#_Toc151414595)

[3.8. Học viên xem danh sách học phí: 57](#_Toc151414596)

[3.9. Học viên xem tổng học phí: 58](#_Toc151414597)

[3.10. Tìm kiếm các lớp đang mở của một môn: 59](#_Toc151414598)

[**4. Chức năng của giảng viên 59**](#_Toc151414599)

[4.1. Đăng nhập vào hệ thống: 59](#_Toc151414600)

[4.2. Xem danh sách lớp dạy: 61](#_Toc151414601)

[4.3. Xem danh sách học viên của lớp đang dạy: 61](#_Toc151414602)

[4.4. Gửi để xuất mở lớp: 62](#_Toc151414603)

[**CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN 63**](#_Toc151414604)

[**CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG 67**](#_Toc151414605)

[**1. Form đăng nhập 67**](#_Toc151414606)

[**2. Form đổi mật khẩu 72**](#_Toc151414607)

[**3. Form dành cho người quản lý 72**](#_Toc151414608)

[3.1. Form mở đầu 72](#_Toc151414609)

[3.2. Form tạo tài khoản cho người quản lý, giảng viên hoặc học viên 73](#_Toc151414610)

[3.3. Form quản lý thông tin giảng viên 74](#_Toc151414611)

[3.4. Form thêm, cập nhật thông tin giảng viên 75](#_Toc151414612)

[3.5. Form quản lý thông tin học viên 76](#_Toc151414613)

[3.6. Form thêm, cập nhật thông tin học viên 77](#_Toc151414614)

[3.7. Form quản lý thông tin lớp học 77](#_Toc151414615)

[3.8. Form thêm, cập nhật thông tin lớp học 78](#_Toc151414616)

[**4. Form chức năng giảng viên 79**](#_Toc151414617)

[4.1. Form mở đầu dành cho giảng viên 79](#_Toc151414618)

[4.2. Form xem danh sách học viên trong một lớp 80](#_Toc151414619)

[4.3. Form đề xuất mở thêm lớp 81](#_Toc151414620)

[**5. Form dành cho học viên 82**](#_Toc151414621)

[5.1. Form đăng ký môn học 82](#_Toc151414622)

[5.2. Form xem danh sách lớp học 83](#_Toc151414623)

[5.3. Form chi tiết học phí 84](#_Toc151414624)

**LỜI CẢM ƠN**

Kính thưa Thầy Nguyễn Thành Sơn,

Chúng em muốn bày tỏ lòng biết ơn chân thành đến Thầy vì sự hướng dẫn và sự giúp đỡ quý báu mà Thầy đã dành cho chúng em trong suốt quá trình hoàn thiện đồ án môn "Hệ quản trị cơ sở dữ liệu".

Nhờ sự tận tâm và kiên nhẫn của Thầy, chúng em đã có thể hoàn thành đồ án một cách hiệu quả và đạt được kết quả ấn tượng. Chúng em rất trân trọng thời gian và kiến thức mà Thầy đã dành để chia sẻ với chúng em, giúp chúng em tiến bộ và phát triển.

Những kiến thức và kỹ năng mà Thầy đã truyền đạt cho chúng em trong lĩnh vực quản trị cơ sở dữ liệu không chỉ giúp chúng em hoàn thành đồ án mà còn sẽ có giá trị lớn trong tương lai khi chúng em bước vào thế giới công việc.

Chúng em sẽ luôn mang theo những kiến thức này và hướng dẫn của Thầy trong hành trình phát triển của chúng em và sẽ không quên ơn đúng nghĩa của từ "người thầy" của mình.

Một lần nữa, xin chân thành tri ân Thầy Nguyễn Thành Sơn vì tất cả những điều tốt lành mà Thầy đã mang lại cho chúng em trong suốt thời gian học tập.

Trân trọng,

Tăng Huỳnh Minh Tiến,

Nguyễn Hoàn Thảo,

Nguyễn Thành Trung,

Nguyễn Anh Tuấn.

**LỜI NÓI ĐẦU**

Xin chào thầy, cô và quý các bạn,

Đồ án quản lý hệ thống đăng ký môn học là kết quả của sự đồng lòng và nỗ lực không ngừng của chúng tôi sau thời gian dài học tập và thực hành môn Hệ quản trị cơ sở dữ liệu. Đây là một dự án đầy thách thức, hứa hẹn mang đến kiến thức và kinh nghiệm đáng quý cho chúng em.

Trong quá trình hoàn thành dự án, chúng em đã đối mặt với nhiều khó khăn và thách thức. Tuy nhiên, nhờ sự hướng dẫn tận tâm của giảng viên và sự nỗ lực không ngừng của chúng em, dự án đã được hoàn thành một cách xuất sắc.

Hệ thống quản lý hệ thống đăng ký môn học là một công cụ thực tiễn, giúp các trường học và đại học quản lý quá trình đăng ký môn học của học sinh và sinh viên một cách hiệu quả và tiện lợi. Chúng em đã áp dụng kiến thức về hệ quản trị cơ sở dữ liệu để xây dựng một hệ thống hoàn chỉnh, bao gồm các chức năng được phân quyền và các hoạt động kinh doanh.

Chúng em hy vọng rằng dự án của chúng em sẽ mang lại kiến thức và trải nghiệm thực tế hữu ích cho quý thầy cô và các bạn sinh viên. Đồng thời, chúng em mong muốn dự án này sẽ truyền cảm hứng và khơi gợi ý tưởng sáng tạo cho các bạn sinh viên khác trong việc áp dụng hệ quản trị cơ sở dữ liệu vào thực tế.

Chúng em xin chân thành cảm ơn sự quan tâm và đóng góp ý kiến từ quý thầy cô và các bạn sinh viên.

# CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG

## 1. Đặt tả bài toán

### 1.1. Mô tả bài toán

Hệ thống quản lý đăng ký môn học cho sinh viên được phát triển để tối ưu hóa quá trình đăng ký môn học, giúp sinh viên, giảng viên và nhân viên quản lý trường đại học tương tác dễ dàng với hệ thống. Đây là một phần quan trọng trong việc tổ chức và quản lý hoạt động giảng dạy và học tập trong môi trường đại học hoặc các tổ chức giáo dục tương tự. Chương trình được thiết kế cho ba đối tượng sử dụng là người quản lý, giảng viên và sinh viên với các quyền khác nhau. Đối với đối tượng sử dụng là sinh viên, họ sẽ có quyền truy cập vào các chức năng như xem thông tin môn học, đăng ký môn học, xem thông tin đăng ký. Đối với đối tượng sử dụng là giảng viên, họ sẽ có quyền truy cập vào các chức năng như xem danh sách lớp học, quản lý thông tin lớp học. Cuối cùng, đối với đối tượng sử dụng là người quản lý, họ sẽ có quyền quản lý thông tin môn học, quản lý tài khoản, xem báo cáo và thống kê. Đối với hệ thống quản lý đăng ký môn học cần lưu trữ các dữ liệu sau:

Trong hệ thống có lưu trữ dữ liệu của các kỳ học của sinh viên, mỗi kỳ học có khóa chính là mã học kỳ (ví dụ 2023-2024/1). Ngoài ra còn có các thông tin như ngày khai giảng, ngày bế giảng của mỗi kỳ. Mỗi học kỳ gồm nhiều lớp học phần khác nhau mỗi lớp học phần phân biệt với nhau bằng mã lớp, mỗi lớp học phần có thể được mở trong các kỳ học khác nhau. Ngoài ra mỗi lớp học phần còn chứa các thông tin như tên lớp, số học viên đăng ký lớp, mã môn học lớp tổ chức giảng dạy, mã giảng viên giảng dạy, thời gian.

Trong mỗi lớp học sẽ tổ chức giảng dạy một môn học. Một môn học có thể có nhiều lớp học phần tổ chức giảng dạy và môn học có khóa chính là mã môn học để có thể phân biệt giữa các môn học và các tính chất khác như tên môn học, số tín chỉ, điều kiện tiên quyết (ví dụ như điều kiện tiên quyết bằng 1 là không có môn tiên quyết, điều kiện tiên quyết bằng 2 là có 1 môn tiên quyết,…).

Sinh viên được đăng ký nhiều lớp học phần khác nhau và mỗi lớp có thể được đăng ký bởi nhiều sinh viên (trong số lượng cho phép). Mỗi đối tượng sinh viên chứa các thông tin như mã số sinh viên (MSSV) để phân biệt các sinh viên với nhau, họ và tên sinh viên, ngày sinh, quê quán, địa chỉ, mã khoa.

Mỗi giảng viên có thể dạy nhiều lớp học phần khác nhau và mỗi lớp học phần chỉ được dạy bởi một giảng viên, để phân biệt các giảng viên với nhau mỗi giảng viên có một mã giảng viên. Ngoài ra mỗi đối tượng giảng viên còn chứa các thông tin như họ và tên giảng viên, địa chỉ, số điện thoại, địa chỉ email, trình độ chuyên môn, mã khoa.

Mỗi giảng viên thuộc một khoa, mỗi khoa có thể có nhiều giảng viên, thông tin của mỗi khoa gồm có mã khoa, tên khoa, mã trưởng khoa.

Hệ thống còn quản lý các phòng học của nhà trường để phục vụ cho việc sắp xếp và phân bố thời gian, phòng học cho các lớp học phần trong thời khóa biểu. Mỗi lớp học phần sẽ được học trong 1 phòng và mỗi phòng có thể được nhiều lớp học phần dùng (miễn là trong thời gian khác nhau). Mỗi phòng học bao gồm thông tin về mã phòng, loại phòng. Thời gian học bao gồm ngày học trong tuần, số tiết học, thời gian bắt đầu và thời gian kết thúc.

### 1.3. Mô tả chức năng của bài toán

*Chức năng của người quản lý*

- Mở thêm lớp học phần : Cho phép quản trị viên thêm lớp học phần trong kì học hiện tại để đáp ứng nhu cầu học của sinh viên.

- Xóa lớp học phần: Cho phép quản trị viên xóa lớp học phần nếu như lớp học phần đó không đủ điều kiện để mở trong kì học.

- Tạo tài khoản cho sinh viên và giảng viên: Cho phép người quản lý tạo tài khoản trong hệ thống đăng kí môn học cho sinh viên và giảng viên với các quyền truy cập khác nhau.

*Chức năng dành cho sinh viên*

- Đăng kí môn học: Cho phép sinh viên có thể đăng kí môn học mình mong muốn trong học kì.

- Hủy đăng kí môn học: Cho phép sinh viên hủy đăng kí môn học đã được đăng kí trước đó.

- Chuyển nhóm: Cho phép sinh viên chuyển lớp học phần hiện tại sang lớp học phần khác miễn là 2 lớp đều có chung mã môn học.

- Xem thời khóa biểu: Cho phép sinh viên xem thời khóa biểu để biết thông tin lớp học, thời gian học, giảng viên

- Xuất phiếu đăng kí môn học: sau khi đăng kí xong có thể xuất phiếu đăng kí môn học để có thể làm minh chứng sau này.

- Xuất hóa đơn học phí: Sau khi đăng kí xong sinh viên có thể xuất hóa đơn học phí để có thể biết được tổng tiền phải trả.

*Chức năng của giảng viên*

- Xem danh sách học sinh: Cho phép giảng viên có thể xem danh sách sinh viên đã đăng kí trong lớp học phần mình giảng dạy.

- Đề nghị mở thêm lớp học phần: Cho phép giảng viên gửi yêu cầu mở thêm lớp cho người quản lý

### 1.2. Nghiệp vụ bài toán

Bài toán quản lý đăng ký môn học là quá trình quản lý và xử lý thông tin liên quan đến việc sinh viên đăng ký và thay đổi các môn học trong một hệ thống giáo dục. Bài toán quản lý đăng ký môn học thường được áp dụng trong các trường đại học, cao đẳng và các cơ sở giáo dục khác.

Dưới đây là một số nghiệp vụ chính trong bài toán quản lý đăng ký môn học:

*Hệ thống quản lý đăng ký môn học:*

- Hệ thống quản lý môn học phải lưu trữ và cung cấp thông tin chi tiết về các môn học trong tổ chức giáo dục. Điều này bao gồm tên môn học, mã môn học, mô tả môn học, số tín chỉ, giảng viên phụ trách, yêu cầu tiên quyết (nếu có), lịch học,phòng học và bất kỳ thông tin khác cần thiết.

- Hệ thống phải đảm bảo rằng sinh viên không bị trùng lịch giữa các môn học mà họ đăng ký. Nếu một sinh viên đã đăng ký một môn học mà có lịch trùng với môn học khác, hệ thống sẽ phải cảnh báo và yêu cầu sinh viên chọn một thời gian học khác.

- Xác định học phí: Hệ thống cần có các quy định và thông tin về học phí của từng môn học. Học phí có thể được xác định dựa trên nhiều yếu tố, bao gồm số tín chỉ của môn học, học phần môn học…

- Tính toán học phí: Khi sinh viên đăng ký môn học, hệ thống cần tính toán tổng học phí dựa trên thông tin về niên khóa của sinh viên tại trường, các môn học đã đăng ký và học phí tương ứng của từng môn. Tổng học phí này có thể được hiển thị cho sinh viên để xem trước hoặc thanh toán.

*Sinh viên:*

- Đăng ký môn học: Sinh viên có thể đăng ký các môn học mà họ muốn theo học thông qua hệ thống. Sinh viên truy cập vào trang đăng ký môn học của trường, sinh viên trong quá trình đăng ký môn học phải tuân thủ quy định, và thực hiện theo hướng dẫn có sẵn của nhà trường. Đầu tiên sinh viên chọn học phần mình muốn đăng ký, sau đó chọn lớp học phần mà mình muốn học, sinh viên phải xem xét tín chỉ tối đa mà sinh viên được phép đăng ký trong một kỳ học, lịch trình học có bị trùng hay không và kiểm tra xem sinh viên đã đáp ứng các yêu cầu tiên quyết của môn học hay chưa. Sau khi đã kiểm tra xong, sinh viên xác nhận đăng ký lớp học phần mà mình muốn đăng ký và cuối cùng là xuất phiếu đăng ký để có thể đối chiếu khi xảy ra sự cố hệ thống.

*Sau khi đăng ký:*

- Hủy đăng ký: Sinh viên có thể hủy đăng ký môn học trước khi kỳ học bắt đầu hoặc trong một khoảng thời gian nhất định sau khi kỳ học bắt đầu. Hệ thống cần hỗ trợ việc hủy đăng ký và cập nhật lại thông tin về lịch học và số lượng sinh viên trong mỗi môn học.

- Thống kê và báo cáo: Hệ thống cần cung cấp các chức năng thống kê và báo cáo về số lượng sinh viên đăng ký môn học, tình trạng lớp học, số lượng môn học còn trống, học phí cuối cùng sinh viên cần đóng và các thông tin liên quan khác. Thông tin này hữu ích cho việc quản lí và đánh giá quá trình đăng ký môn học.

### 1.4. Tính học phí

Việc tính tổng học phí của sinh viên sau khi đăng ký môn học bao gồm các yếu tố sau:

Tính học phí của từng môn học: đây là số tiền mà sinh viên phải thanh toán cho từng môn học mà mình đã đăng ký. Số tiền này được tính bằng cách nhân số tín chỉ của môn học cho đơn giá của một chỉ được quy định cho môn học đó. Đơn giá sẽ phụ thuộc vào mã môn học và niên học của sinh viên (dữ liệu về đơn giá được lấy từ bảng thông tin học phí được cung cấp từ hệ thống quản lý của nhà trường).

Tổng tiền môn học = số tín chỉ của môn học \* đơn giá của 1 chỉ.

Ví dụ:

Sinh viên có mã 21133xxx đăng ký môn học Hệ quản trị cơ sở dữ liệu có mã môn học là DBMS330284 với 3 tín chỉ thì mức học phí phải đóng cho môn học này được tính như sau:

Tổng tiền môn học (DBMS) = 3 \* **697.000** = 2.091.000.

Sinh viên có mã 23xxxxx đăng ký môn học Hệ quản trị cơ sở dữ liệu có mã môn học là DBMS330284CLC với 3 tín chỉ thì mức học phí phải đóng cho môn học này được tính như sau:

Tổng tiền môn học (DBMS) hệ CLC = 3 \* **1.270.000** = 3.810.000

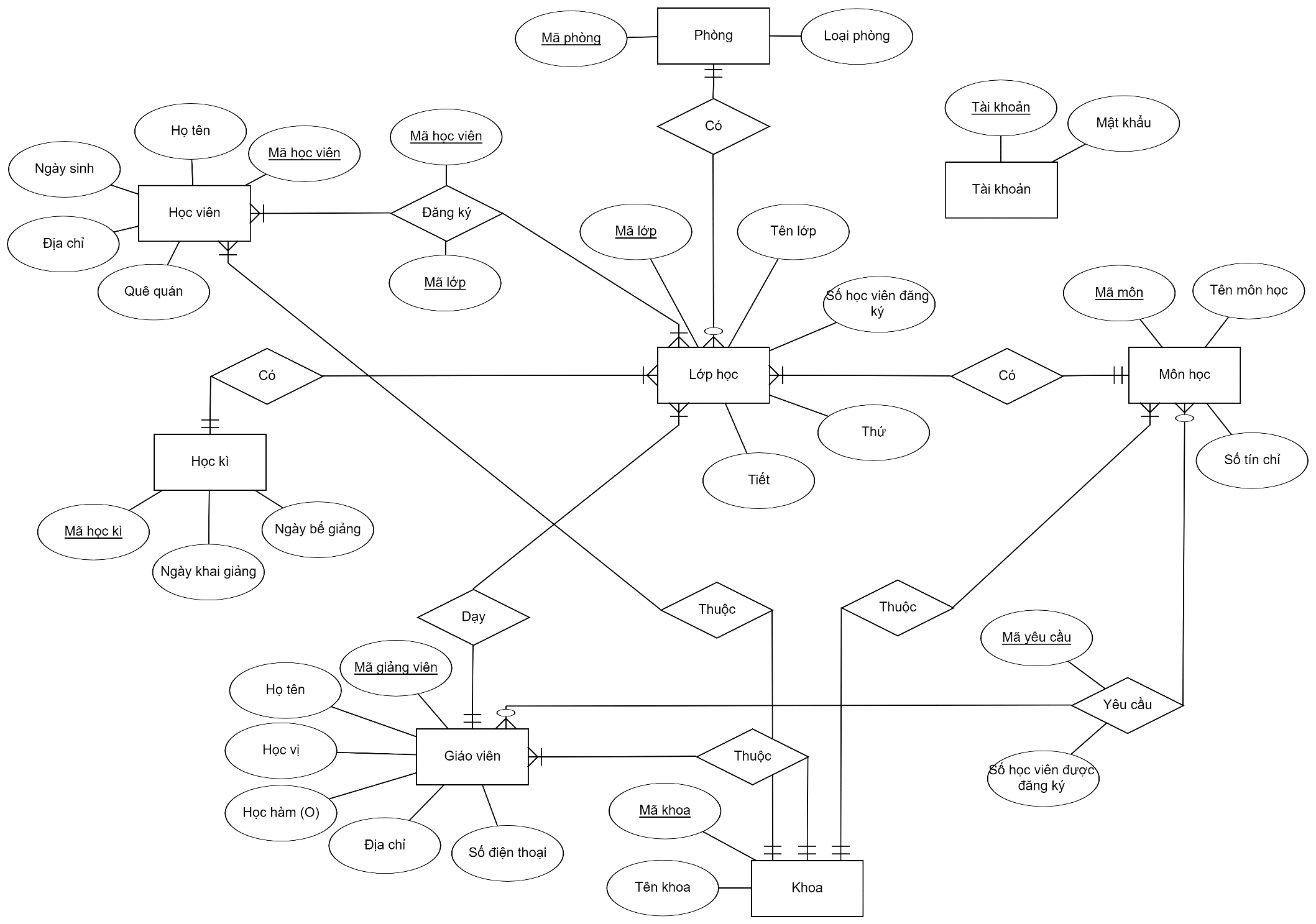
Tính tổng học phí của tất cả các môn học mà sinh viên đã đăng ký: đây là số tiền mà sinh viên phải thanh toán cho tất cả các học phần mà mình đã đăng ký. Số tiền này được tính bằng cách cộng tổng số tiền của tất cả các môn học mà sinh viên đã đăng ký.

Tổng học phí = Học phí môn học 1 + Học phí môn học 2 + …

# CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 1. Thiết kế cơ sở dữ liệu mức quan niệm

Từ những mô tả về dữ liệu cần có ở phần mô tả của bài toán ta hình thành được sơ đồ thực thể kết hợp (ERD):



## 2. Thiết kế cơ sở dữ liệu mức logic

Từ sơ đồ thực thể kết hợp (ERD), ta có các lược đồ quan hệ:

* HOCKI(MaHocKy, NgayKhaiGiang, NgayBeGiang)
* KHOA(MaKhoa, TenKhoa)
* GIANGVIEN(MaGiangVien, HoTen, HocVi, HocHam, DiaChi, Sdt, MaKhoa)
* MONHOC(MaMon, TenMonHoc, SoTinChi, MaKhoa)
* HOCVIEN(MaHocVien, HoTen, NgaySinh, DiaChi, QueQuan, MaKhoa)
* PHONGHOC(MaPhong, LoaiPhong)
* LOPHOC(MaLop, TenLop, MaGiangVien, MaMon, MaHocKy, SoHocVienDangKy, MaPhong, Thu, Tiet)
* DKMH(MaHocVien, MaLop)
* YEUCAU(MaGiangVien, MaMon, SoHocVienDuocDangKy)
* TAIKHOAN(TaiKhoan, MatKhau)

## 3. Các ràng buộc cần có

|  |  |  |
| --- | --- | --- |
| STT | BẢNG | RÀNG BUỘC |
| 1 | HOCKY | Khoá chính MaHocKy,  NOT NULL NgayKhaiGiang,  NOT NULL NgayBeGiang |
| 2 | KHOA | Khoá chính MaKhoa,  NOT NULL TenKhoa |
| 3 | GIANGVIEN | Khoá chính MaGiangVien,  NOT NULL HoTen,  NOT NULL HocVi,  NOT NULL Sdt,  Khoá ngoại MaKhoa đến bảng KHOA,  CHECK Sdt (= 10) |
| 4 | MONHOC | Khoá chính MaMon,  NOT NULL TenMonHoc,  NOT NULL SoTinChi,  Khoá ngoại MaKhoa đến bảng KHOA |
| 5 | HOCVIEN | Khoá chính MaHocVien,  NOT NULL HoTen,  NOT NULL NgaySinh,  NOT NULL DiaChi,  NOT NULL QueQuan,  Khoá ngoại MaKhoa đến bảng KHOA |
| 6 | PHONGHOC | Khoá chính MaPhong,  NOT NULL LoaiPhong |
| 7 | LOPHOC | Khoá chính MaLop,  NOT NULL TenLop,  NOT NULL SoHocVienDangKy,  NOT NULL Thu,  NOT NULL Tiet,  Khoá ngoại MaGiangVien đến bảng GIANGVIEN,  Khoá ngoại MaMon đến bảng MONHOC,  Khoá ngoại MaHocKy đến bảng HOCKY,  Khoá ngoại MaPhong đến bảng PHONGHOC,  CHECK SoHocVienDangKy (<= 75), |
| 8 | DKMH | Khóa chính (MaHocVien, MaLop),  Khoá ngoại MaHocVien đến bảng HOCVIEN,  Khoá ngoại MaLop đến bảng LOPHOC |
| 9 | YEUCAU | Khoá chính (MaMon, MaGiangVien),  NOT NULL SoHocVienDuocDangKy,  Khoá ngoại MaGiangVien đến bảng GIANGVIEN,  Khoá ngoại MaMon đến bảng MONHOC |
| 10 | TAIKHOAN | Khoá chính TaiKhoan |

## 4. Cài đặt các CSDL và các ràng buộc

***Bảng học kỳ***

|  |
| --- |
| CREATE TABLE HOCKY (  MaHocKy CHAR(9) NOT NULL, --HK1\_22-23, HK2\_22-23,..  NgayKhaiGiang DATE NOT NULL,  NgayBeGiang DATE NOT NULL,  CONSTRAINT PK\_HOCKY PRIMARY KEY (MaHocKy),  ); |

***Bảng khoa***

|  |
| --- |
| CREATE TABLE KHOA (  MaKhoa VARCHAR(5) NOT NULL, --FIT, FCFT, FCEAM, ITE,...  TenKhoa NVARCHAR(MAX) NOT NULL,  CONSTRAINT PK\_KHOA PRIMARY KEY (MaKhoa),  ); |

***Bảng giảng viên***

|  |
| --- |
| CREATE TABLE GIANGVIEN (  MaGiangVien CHAR(6) NOT NULL, --GV0001, GV0002,...  HoTen NVARCHAR(MAX) NOT NULL,  HocVi NVARCHAR(10) NOT NULL, --Thạc sĩ, Tiến sĩ, Cử nhân,...  HocHam NVARCHAR(12), --Phó Giáo sư, Giáo sư  DiaChi NVARCHAR(MAX),  Sdt CHAR(10) CHECK(len(Sdt) = 10) NOT NULL,  MaKhoa VARCHAR(5) NOT NULL,  CONSTRAINT PK\_GIANGVIEN PRIMARY KEY (MaGiangVien),  CONSTRAINT FK\_GIANGVIEN\_KHOA FOREIGN KEY (MaKhoa) REFERENCES KHOA(MaKhoa),  ); |

***Bảng môn học***

|  |
| --- |
| CREATE TABLE MONHOC (  MaMon CHAR(10) NOT NULL, --WEPR330479, DBMS330284,...  TenMonHoc NVARCHAR(30) NOT NULL,  SoTinChi INT NOT NULL, --(từ 1 đến 4)  MaKhoa VARCHAR(5) NOT NULL, --FIT, FCFT, FCEAM, ITE,...  CONSTRAINT PK\_MONHOC PRIMARY KEY (MaMon),  CONSTRAINT FK\_MONHOC\_KHOA FOREIGN KEY (MaKhoa) REFERENCES KHOA(MaKhoa),  ); |

***Bảng học viên***

|  |
| --- |
| CREATE TABLE HOCVIEN (  MaHocVien CHAR(10) NOT NULL, --HV00000001, HV00000011,...  HoTen NVARCHAR(40) NOT NULL, --Lê Hoàng Hiếu Nghĩa Đệ Nhất Thương Tâm Nhân (tên dài nhất VN),...  NgaySinh DATE NOT NULL,  DiaChi NVARCHAR(MAX) NOT NULL,  QueQuan NVARCHAR(20) NOT NULL, --Bà Rịa - Vũng Tàu, Thừa Thiên - Huế,...  MaKhoa VARCHAR(5) NOT NULL, --FIT, FCFT, FCEAM, ITE,...  CONSTRAINT PK\_HOCVIEN PRIMARY KEY (MaHocVien),  CONSTRAINT FK\_HOCVIEN\_KHOA FOREIGN KEY (MaKhoa) REFERENCES KHOA(MaKhoa),  ); |

***Bảng phòng học***

|  |
| --- |
| CREATE TABLE PHONGHOC (  MaPhong VARCHAR(10) NOT NULL, --A5-203, A121, B108  LoaiPhong NVARCHAR(40) NOT NULL, --Phòng lý thuyết, Phòng máy tính, Phòng thực hành, Phòng thí nghiệm,...  CONSTRAINT PK\_PHONGHOC PRIMARY KEY (MaPhong),  ); |

***Bảng lớp học***

|  |
| --- |
| CREATE TABLE LOPHOC (  MaLop CHAR(18) NOT NULL, --DBMS330284\_23\_1\_05, WEPR330479\_23\_1\_11  TenLop NVARCHAR(MAX) NOT NULL,  MaGiangVien CHAR(6) NOT NULL, --GV0001, GV0002,...  MaMon CHAR(10) NOT NULL, --WEPR330479, DBMS330284,...  MaHocKy CHAR(9) NOT NULL, --HK1\_22-23, HK2\_22-23,..  SoHocVienDangKy INT NOT NULL CHECK (SoHocVienDangKy <= 75),  MaPhong VARCHAR(10) NOT NULL, --A5-203, A121, B108  Thu NVARCHAR(10) NOT NULL,  Tiet VARCHAR(10) NOT NULL,  CONSTRAINT PK\_LOPHOC PRIMARY KEY (MaLop),  CONSTRAINT FK\_LOPHOC\_GIANGVIEN FOREIGN KEY (MaGiangVien) REFERENCES GIANGVIEN(MaGiangVien),  CONSTRAINT FK\_LOPHOC\_MONHOC FOREIGN KEY (MaMon) REFERENCES MONHOC(MaMon),  CONSTRAINT FK\_LOPHOC\_HOCKY FOREIGN KEY (MaHocKy) REFERENCES HOCKY(MaHocKy),  CONSTRAINT FK\_LOPHOC\_PHONGHOC FOREIGN KEY (MaPhong) REFERENCES PHONGHOC(MaPhong),  ); |

***Bảng đăng ký môn học***

|  |
| --- |
| CREATE TABLE DKMH (  MaHocVien CHAR(10) NOT NULL,  MaLop CHAR(18) NOT NULL,  CONSTRAINT PK\_DKMH PRIMARY KEY (MaHocVien, MaLop),  CONSTRAINT FK\_DKHP\_LOPHOC FOREIGN KEY (MaLop) REFERENCES LOPHOC(MaLop),  CONSTRAINT FK\_DKHP\_HOCVIEN FOREIGN KEY (MaHocVien) REFERENCES HOCVIEN(MaHocVien),  ); |

***Bảng yêu cầu***

|  |
| --- |
| CREATE TABLE YEUCAU (  MaGiangVien CHAR(6) NOT NULL,  MaMon CHAR(10) NOT NULL,  SoHocVienDuocDangKy INT NOT NULL,  CONSTRAINT PK\_YEUCAU PRIMARY KEY (MaGiangVien, MaMon),  CONSTRAINT FK\_YEUCAU\_GIANGVIEN FOREIGN KEY (MaGiangVien) REFERENCES GIANGVIEN(MaGiangVien),  CONSTRAINT FK\_YEUCAU\_MONHOC FOREIGN KEY (MaMon) REFERENCES MONHOC(MaMon),  ); |

***Bảng tài khoản***

|  |
| --- |
| CREATE TABLE TAIKHOAN (  TaiKhoan NVARCHAR(10) NOT NULL,  MatKhau NVARCHAR(20) NOT NULL,  CONSTRAINT PK\_TAIKHOAN\_TAIKHOAN PRIMARY KEY (TaiKhoan)  ); |

A screenshot of a computer

Description automatically generated

## 5. Các view

1. **Xem danh sách các lớp học phần sinh viên có thể đăng kí**

|  |
| --- |
| CREATE VIEW v\_DSLopHoc AS  SELECT lh.MaLop, mh.TenMonHoc, mh.SoTinChi, lh.Tiet, lh.Thu, p.MaPhong Phong, gv.HoTen AS 'Giảng viên', mh.MaKhoa, mh.MaMon, lh.SoHocVienDangKy, lh.MaHocKy  FROM dbo.MONHOC mh  JOIN dbo.LOPHOC lh ON lh.MaMon = mh.MaMon  JOIN dbo.GIANGVIEN gv ON lh.MaGiangVien = gv.MaGiangVien  JOIN dbo.PHONGHOC p ON lh.MaPhong = p.MaPhong; |

1. **Xem danh sách các môn đã đăng kí**

|  |
| --- |
| CREATE VIEW v\_DSDaDangKi AS  SELECT lh.MaLop, mh.TenMonHoc, mh.SoTinChi, lh.Tiet, lh.Thu, p.MaPhong Phong, gv.HoTen AS GiangVien, dk.MaHocVien, lh.MaMon  FROM dbo.MONHOC mh  JOIN dbo.LOPHOC lh ON lh.MaMon = mh.MaMon  JOIN dbo.GIANGVIEN gv ON lh.MaGiangVien = gv.MaGiangVien  JOIN dbo.PHONGHOC p ON lh.MaPhong = p.MaPhong  JOIN dbo.DKMH dk ON dk.MaLop = lh.MaLop; |

1. **Xem danh sách giảng viên nào dạy lớp nào**

|  |
| --- |
| CREATE VIEW v\_DSLopDay AS  SELECT lh.MaGiangVien, mh.TenMonHoc, lh.MaLop, mh.MaMon, lh.Tiet, lh.Thu, lh.MaPhong Phong, lh.MaHocKy  FROM dbo.LOPHOC lh  JOIN dbo.MONHOC mh ON mh.MaMon = lh.MaMon; |

1. **Xem sinh viên nào đăng kí lớp nào**

|  |
| --- |
| CREATE VIEW v\_DSHV AS  SELECT lh.MaLop, hv.MaHocVien, hv.HoTen  FROM dbo.DKMH dk  JOIN dbo.LOPHOC lh ON dk.MaLop = lh.MaLop  JOIN dbo.HOCVIEN hv ON dk.MaHocVien = hv.MaHocVien; |

## 6. Các trigger

1. **Trigger ngăn chặn học viên đăng kí lớp học trùng lịch, môn, quá 35 tín chỉ hoặc lớp đã đủ số lượng sinh viên (mặc định là 75)**

|  |
| --- |
| CREATE TRIGGER Trigger\_TrungLichDangKyLop --Trigger ngăn chặn học viên đăng kí lớp học trùng lịch, môn, hoặc quá 35 tín chỉ  ON DKMH  INSTEAD OF INSERT  AS  BEGIN  DECLARE @I\_MaHocVien CHAR(10), @I\_MaLop CHAR(18) --Khai báo 2 biến chưa MaHocVien và Malop của học viên đăng kí  SELECT @I\_MaHocVien = i.MaHocVien, @I\_MaLop = i.MaLop  FROM inserted i  DECLARE @TimeTable1 TABLE ( --bảng lưu thời gian của các lớp mà học viên đó đã đăng ký  MaLop CHAR(18),  NamHoc INT,  SoHocKy INT,  Thu INT,  TietBD INT,  TietKT INT  );  DECLARE @TimeTable2 TABLE ( --bảng lưu thời gian lớp học mà học viên đó sắp đăng ký  MaLop CHAR(18),  NamHoc INT,  SoHocKy INT,  Thu INT,  TietBD INT,  TietKT INT  );  INSERT INTO @TimeTable1 (MaLop, NamHoc, SoHocKy, Thu, TietBD, TietKT)  SELECT MaLop,  CAST(SUBSTRING(MaLop, CHARINDEX('\_', MaLop) + 1, 2) AS INT) AS NamHoc,  CAST(SUBSTRING(MaLop, CHARINDEX('\_', MaLop) + 4, 1) AS INT) AS SoHocKy,  CAST(SUBSTRING(Thu, CHARINDEX(' ', Thu) + 1, LEN(Thu)) AS INT) AS Thu, --biến đổi thứ sang kiểu int để dễ xử lý  CAST(SUBSTRING(Tiet, CHARINDEX('-', Tiet) - 2, 2) AS INT) AS TietBD, --biến đổi tiết sang kiểu int trở thành tiết bắt đầu  CAST(SUBSTRING(Tiet, CHARINDEX('-', Tiet) + 1, LEN(Tiet)) AS INT) AS TietKT -- và tiết kết thúc để dễ xử lý  FROM (  SELECT MaLop, Thu, Tiet  FROM dbo.v\_DSDaDangKi ds  WHERE ds.MaHocVien = @I\_MaHocVien) A --A chứa thông tin thời gian học của Học viên    INSERT INTO @TimeTable2 (MaLop, NamHoc, SoHocKy, Thu, TietBD, TietKT)  SELECT MaLop,  CAST(SUBSTRING(MaLop, CHARINDEX('\_', MaLop) + 1, 2) AS INT) AS NamHoc,  CAST(SUBSTRING(MaLop, CHARINDEX('\_', MaLop) + 4, 1) AS INT) AS SoHocKy,  CAST(SUBSTRING(Thu, CHARINDEX(' ', Thu) + 1, LEN(Thu)) AS INT) AS Thu, --biến đổi thứ sang kiểu int để dễ xử lý  CAST(SUBSTRING(Tiet, CHARINDEX('-', Tiet) - 2, 2) AS INT) AS TietBD, --biến đổi tiết sang kiểu int trở thành tiết bắt đầu  CAST(SUBSTRING(Tiet, CHARINDEX('-', Tiet) + 1, LEN(Tiet)) AS INT) AS TietKT -- và tiết kết thúc để dễ xử lý  FROM (  SELECT MaLop, Thu, Tiet  FROM LOPHOC lh  WHERE lh.MaLop = @I\_MaLop ) A --A chứa thông tin thời gian học lớp sắp đăng kí    IF EXISTS (SELECT 1 FROM DKMH WHERE MaLop = @I\_MaLop and MaHocVien = @I\_MaHocVien)  BEGIN  RAISERROR (N'Đã đăng kí môn đó', 16, 1)  END  ELSE IF (  SELECT COUNT(ds.MaMon) --kiểm tra xem học viên đã đăng kí môn học đó ở lớp học khác chưa  FROM v\_DSDaDangKi ds  WHERE ds.MaHocVien = @I\_MaHocVien and ds.MaMon = (SELECT MaMon FROM LOPHOC WHERE MaLop = @I\_MaLop)  ) >= 1  BEGIN  RAISERROR (N'Đã đăng kí môn đó', 16, 1)  END  ELSE IF ( --kiểm tra xem số tín chỉ có vượt quá cho phép chưa  SELECT SUM(SoTinChi)  FROM v\_DSDaDangKi  WHERE MaHocVien = @I\_MaHocVien  ) > 35  BEGIN  RAISERROR (N'Số tín chỉ vượt quá giới hạn cho phép.', 16, 1)  END  ELSE IF EXISTS ( --kiểm tra xem có trùng lịch với môn nào hay không  SELECT 1  FROM (SELECT \* FROM @TimeTable1 ) AS tt1, (SELECT \* FROM @TimeTable2 ) AS tt2  WHERE (tt1.NamHoc = tt2.NamHoc AND tt1.SoHocKy = tt2.SoHocKy) AND  (tt1.Thu = tt2.Thu AND ((tt1.TietBD <= tt2.TietBD AND tt1.TietKT >= tt2.TietBD) OR (tt1.TietBD <= tt2.TietKT AND tt1.TietKT >= tt2.TietKT)))  )  BEGIN  RAISERROR(N'Trùng lịch', 16, 1)  END  ELSE IF ( --kiểm tra xem lớp đó đủ số lượng đăng kí chưa  SELECT SoHocVienDangKy  FROM LOPHOC  WHERE MaLop = @I\_MaLop ) = 75  )  BEGIN  RAISERROR(N'Đã đủ số lượng đăng kí', 16, 1)  END  ELSE  BEGIN  INSERT INTO DKMH (MaHocVien, MaLop)  VALUES(@I\_MaHocVien, @I\_MaLop)  END  END; |

1. **Trigger ngăn trùng lịch học khi học viên chuyển lớp**

|  |
| --- |
| CREATE TRIGGER Trigger\_TrungLichChuyenLop --Trigger ngăn trùng lịch học khi học viên chuyển lớp  ON DKMH  INSTEAD OF UPDATE  AS  BEGIN  DECLARE @I\_MaLop CHAR(18), @I\_MaHocVien CHAR(10)  SELECT @I\_MaLop = MaLop, @I\_MaHocVien = MaHocVien FROM inserted    DECLARE @D\_MaLop CHAR(18)  SELECT @D\_MaLop = MaLop FROM deleted  DECLARE @TimeTable1 TABLE ( --bảng lưu thời gian của các môn mà học viên đó đã đăng ký  MaLop CHAR(18),  NamHoc INT,  SoHocKy INT,  Thu INT,  TietBD INT,  TietKT INT  );  DECLARE @TimeTable2 TABLE ( --bảng lưu thời gian của các môn mà học viên đó đã đăng ký  MaLop CHAR(18),  NamHoc INT,  SoHocKy INT,  Thu INT,  TietBD INT,  TietKT INT  );  INSERT INTO @TimeTable1 (MaLop, NamHoc, SoHocKy, Thu, TietBD, TietKT)  SELECT MaLop,  CAST(SUBSTRING(MaLop, CHARINDEX('\_', MaLop) + 1, 2) AS INT) AS NamHoc,  CAST(SUBSTRING(MaLop, CHARINDEX('\_', MaLop) + 4, 1) AS INT) AS SoHocKy,  CAST(SUBSTRING(Thu, CHARINDEX(' ', Thu) + 1, LEN(Thu)) AS INT) AS Thu, --biến đổi thứ sang kiểu int để dễ xử lý  CAST(SUBSTRING(Tiet, CHARINDEX('-', Tiet) - 2, 2) AS INT) AS TietBD, --biến đổi tiết sang kiểu int trở thành tiết bắt đầu  CAST(SUBSTRING(Tiet, CHARINDEX('-', Tiet) + 1, LEN(Tiet)) AS INT) AS TietKT -- và tiết kết thúc để dễ xử lý  FROM (  SELECT MaLop, Thu, Tiet  FROM dbo.v\_DSDaDangKi ds  WHERE ds.MaHocVien = @I\_MaHocVien and ds.MaLop != @D\_MaLop) A --A chứa thông tin thời gian học của Học viên    INSERT INTO @TimeTable2 (MaLop, NamHoc, SoHocKy, Thu, TietBD, TietKT)  SELECT MaLop,  CAST(SUBSTRING(MaLop, CHARINDEX('\_', MaLop) + 1, 2) AS INT) AS NamHoc,  CAST(SUBSTRING(MaLop, CHARINDEX('\_', MaLop) + 4, 1) AS INT) AS SoHocKy,  CAST(SUBSTRING(Thu, CHARINDEX(' ', Thu) + 1, LEN(Thu)) AS INT) AS Thu, --biến đổi thứ sang kiểu int để dễ xử lý  CAST(SUBSTRING(Tiet, CHARINDEX('-', Tiet) - 2, 2) AS INT) AS TietBD, --biến đổi tiết sang kiểu int trở thành tiết bắt đầu  CAST(SUBSTRING(Tiet, CHARINDEX('-', Tiet) + 1, LEN(Tiet)) AS INT) AS TietKT -- và tiết kết thúc để dễ xử lý  FROM (  SELECT MaLop, Thu, Tiet  FROM LOPHOC lh  WHERE lh.MaLop = @I\_MaLop ) A --A chứa thông tin thời gian học lớp sắp đăng kí  IF (@D\_MaLop = @I\_MaLop)  BEGIN  RAISERROR (N'Đã đăng kí lớp này', 16, 1)  ROLLBACK;  END  ELSE IF EXISTS ( --kiểm tra xem có trùng lịch với môn nào hay không  SELECT 1  FROM (SELECT \* FROM @TimeTable1 ) AS tt1, (SELECT \* FROM @TimeTable2 ) AS tt2  WHERE (tt1.NamHoc = tt2.NamHoc AND tt1.SoHocKy = tt2.SoHocKy) AND  (tt1.Thu = tt2.Thu AND ((tt1.TietBD <= tt2.TietBD AND tt1.TietKT >= tt2.TietBD) OR (tt1.TietBD <= tt2.TietKT AND tt1.TietKT >= tt2.TietKT)))  )  BEGIN  RAISERROR(N'Trùng lịch', 16, 1)  END  ELSE IF ( --kiểm tra xem lướp đó đủ số lượng đăng kí chưa  (SELECT SoHocVienDangKy  FROM LOPHOC  WHERE MaLop = @I\_MaLop) = 75  )  BEGIN  RAISERROR(N'Đã đủ số lượng đăng kí', 16, 1)  ROLLBACK;  END  ELSE  BEGIN  UPDATE DKMH  SET MaLop = @I\_MaLop  WHERE MaHocVien = @I\_MaHocVien and MaLop = @D\_MaLop;  END  END; |

1. **Trigger hỗ trợ người quản lý xem xét mở lớp học mới**

|  |
| --- |
| CREATE TRIGGER Trigger\_MoLop --Trigger hỗ trợ người quản lý xem xét mở lớp học mới  ON LOPHOC  INSTEAD OF INSERT  AS  BEGIN  DECLARE @TimeTable1 TABLE ( --lưu thời gian các lớp học phần đang mở  MaLop CHAR(18),  MaGiangVien CHAR(10),  MaPhong CHAR(10),  NamHoc INT,  SoHocKy INT,  Thu INT,  TietBD INT,  TietKT INT);  INSERT INTO @TimeTable1 (MaLop, MaGiangVien, MaPhong, NamHoc, SoHocKy, Thu, TietBD, TietKT)  SELECT MaLop,  MaGiangVien,  MaPhong,  CAST(SUBSTRING(MaLop, CHARINDEX('\_', MaLop) + 1, 2) AS INT) AS NamHoc,  CAST(SUBSTRING(MaLop, CHARINDEX('\_', MaLop) + 4, 1) AS INT) AS SoHocKy,  CAST(SUBSTRING(Thu, CHARINDEX(' ', Thu) + 1, LEN(Thu)) AS INT) AS Thu,  CAST(SUBSTRING(Tiet, CHARINDEX('-', Tiet) - 2, 2) AS INT) AS TietBD,  CAST(SUBSTRING(Tiet, CHARINDEX('-', Tiet) + 1, LEN(Tiet)) AS INT) AS TietKT  FROM dbo.LOPHOC  DECLARE @TimeTable2 TABLE ( --lưu thời gian lớp học phần thêm vào (insert)  MaLop CHAR(18),  MaGiangVien CHAR(10),  MaPhong CHAR(10),  NamHoc INT,  SoHocKy INT,  Thu INT,  TietBD INT,  TietKT INT);  INSERT INTO @TimeTable2 (MaLop, MaGiangVien, MaPhong, NamHoc, SoHocKy, Thu, TietBD, TietKT)  SELECT MaLop,  MaGiangVien,  MaPhong,  CAST(SUBSTRING(MaLop, CHARINDEX('\_', MaLop) + 1, 2) AS INT) AS NamHoc,  CAST(SUBSTRING(MaLop, CHARINDEX('\_', MaLop) + 4, 1) AS INT) AS SoHocKy,  CAST(SUBSTRING(Thu, CHARINDEX(' ', Thu) + 1, LEN(Thu)) AS INT) AS Thu,  CAST(SUBSTRING(Tiet, CHARINDEX('-', Tiet) - 2, 2) AS INT) AS TietBD,  CAST(SUBSTRING(Tiet, CHARINDEX('-', Tiet) + 1, LEN(Tiet)) AS INT) AS TietKT  FROM inserted  DECLARE @I\_MaLop CHAR(18)  SELECT @I\_MaLop = i.MaLop  FROM inserted i  IF EXISTS (  SELECT 1  FROM LOPHOC lh,inserted i  WHERE lh.MaLop = i.MaLop  )  BEGIN  RAISERROR (N'Lớp học đã tồn tại trong bảng lớp học.', 16, 1)  END  ELSE IF EXISTS ( --chú ý  SELECT 1  FROM @TimeTable1 tt1 ,@TimeTable2 AS tt2  WHERE (tt1.NamHoc = tt2.NamHoc AND tt1.SoHocKy = tt2.SoHocKy) AND  (tt1.MaGiangVien = tt2.MaGiangVien AND tt1.Thu = tt2.Thu AND ((tt1.TietBD <= tt2.TietBD AND tt1.TietKT >= tt2.TietBD) OR (tt1.TietBD <= tt2.TietKT AND tt1.TietKT >= tt2.TietKT)))  )  BEGIN  RAISERROR (N'Giảng viên đã dạy lớp học khác trong thời gian diễn ra lớp học.', 16, 1)  END  ELSE IF EXISTS ( --chú ý  SELECT 1  FROM @TimeTable1 tt1 ,@TimeTable2 AS tt2  WHERE (tt1.NamHoc = tt2.NamHoc AND tt1.SoHocKy = tt2.SoHocKy) AND  (tt1.MaPhong = tt2.MaPhong AND tt1.Thu = tt2.Thu AND ((tt1.TietBD <= tt2.TietBD AND tt1.TietKT >= tt2.TietBD) OR (tt1.TietBD <= tt2.TietKT AND tt1.TietKT >= tt2.TietKT)))  )  BEGIN  RAISERROR (N'Phòng học đã phục vụ lớp học khác trong thời gian diễn ra lớp học.', 16, 1)  END  ELSE  BEGIN  INSERT INTO LOPHOC (MaLop, MaGiangVien, TenLop, SoHocVienDangKy, MaMon, MaHocKy, MaPhong, Tiet, Thu)  SELECT MaLop, MaGiangVien,TenLop, SoHocVienDangKy, MaMon, MaHocKy, MaPhong, Tiet,Thu  FROM inserted i;  PRINT N'Đã mở'  END  END; |

1. **Trigger hỗ trợ người quản lý xóa lớp không đủ điều kiện (có ít hơn 10 học viên đăng ký)**

|  |
| --- |
| CREATE TRIGGER Trigger\_XoaLopItHocVien --Trigger hỗ trợ người quản lý xóa lớp không đủ điều kiện  ON LOPHOC  INSTEAD OF DELETE  AS  BEGIN  DECLARE @D\_MaLop CHAR(18)  SELECT @D\_MaLop = d.MaLop  FROM deleted d  IF (  SELECT lh.SoHocVienDangKy  FROM dbo.LOPHOC lh  WHERE lh.MaLop = @D\_MaLop  ) < 10  BEGIN  DELETE dbo.LOPHOC  WHERE dbo.LOPHOC.MaLop = @D\_MaLop;  END  ELSE  BEGIN  RAISERROR (N'Lớp học không thể xoá', 16, 1)  END  END; |

1. **Trigger hỗ trợ người quản lý xóa giảng viên**

|  |
| --- |
| CREATE TRIGGER Trigger\_XoaGV --Trigger hỗ trợ người quản lý xóa Giang vien (Chỉ xóa khi giảng viên đó ko dạy lớp nào)  ON GIANGVIEN  INSTEAD OF DELETE  AS  BEGIN  DECLARE @D\_MaGV CHAR(18)  SELECT @D\_MaGV = d.MaGiangVien  FROM deleted d  IF NOT EXISTS (SELECT 1 FROM LOPHOC WHERE MaGiangVien = @D\_MaGV)  BEGIN  DELETE FROM GIANGVIEN WHERE MaGiangVien = @D\_MaGV  END  ELSE  BEGIN  RAISERROR (N'Giảng viên đang dạy 1 lớp', 16, 1)  END  END; |

1. **Trigger kiểm tra yêu cầu mở lớp của giảng viên**

|  |
| --- |
| CREATE TRIGGER Trigger\_CheckYeuCau -- Trigger kiểm tra yêu cầu của giáo viên  ON YEUCAU  INSTEAD OF INSERT  AS  BEGIN  DECLARE @I\_MaGV CHAR(18), @I\_MaMon CHAR(10)  SELECT @I\_MaGV = i.MaGiangVien, @I\_MaMon = i.MaMon  FROM inserted i  IF NOT EXISTS (SELECT 1 FROM YEUCAU WHERE MaGiangVien = @I\_MaGV and MaMon = @I\_MaMon)  BEGIN  INSERT INTO YEUCAU (MaGiangVien, MaMon, SoHocVienDuocDangKy)  SELECT i.MaGiangVien, i.MaMon, i.SoHocVienDuocDangKy  FROM inserted i  END  ELSE  BEGIN  RAISERROR (N'Giảng viên đã yêu cầu mở lớp cho môn này trước đó', 16, 1)  END  END; |

1. **Trigger hỗ trợ admin tạo tài khoản**

|  |
| --- |
| CREATE TRIGGER Trigger\_TaoTaiKhoan  ON TAIKHOAN  INSTEAD OF INSERT  AS  DECLARE @userName NVARCHAR(30), @passWord NVARCHAR(10)  SELECT @userName = nl.TaiKhoan, @passWord = nl.MatKhau  FROM inserted nl  IF EXISTS (SELECT 1 FROM TAIKHOAN WHERE TaiKhoan = @userName)  BEGIN  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi: Tài khoản đã tồn tại'  RAISERROR(@err, 16, 1)  END  ELSE  BEGIN  INSERT INTO TaiKhoan (TaiKhoan, MatKhau) VALUES (@userName, @passWord)  DECLARE @sqlString NVARCHAR(2000)  SET @sqlString = 'CREATE LOGIN [' + @userName + '] WITH PASSWORD=''' + @passWord + ''', DEFAULT\_DATABASE=[DANG\_KY\_MON\_HOC], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF'  EXEC (@sqlString)  SET @sqlString = 'CREATE USER ' + @userName +' FOR LOGIN '+ @userName  EXEC (@sqlString)  IF (LEFT(@userName, 2) = 'HV')  SET @sqlString = 'ALTER ROLE HV ADD MEMBER ' + @userName;  ELSE IF (LEFT(@userName, 2) = 'GV')  SET @sqlString = 'ALTER ROLE GV ADD MEMBER ' + @userName;  ELSE  SET @sqlString = 'ALTER SERVER ROLE sysadmin' + ' ADD MEMBER '+ @userName;  EXEC (@sqlString)  END; |

1. **Trigger hỗ trợ admin đổi mật khẩu**

|  |
| --- |
| CREATE TRIGGER Trigger\_DoiMatKhau  ON TAIKHOAN  INSTEAD OF UPDATE  AS  DECLARE @userName NVARCHAR(30), @N\_passWord NVARCHAR(20), @O\_passWord NVARCHAR(10)  SELECT @userName = nl.TaiKhoan, @N\_passWord = nl.MatKhau  FROM inserted nl  SELECT @O\_passWord = d.MatKhau  FROM deleted d  IF NOT EXISTS (SELECT 1 FROM TAIKHOAN WHERE TaiKhoan = @userName and MatKhau = @O\_passWord)  BEGIN  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi: Tên tài khoản hoặc mật khẩu không chính xác'  RAISERROR(@err, 16, 1)  END  ELSE  BEGIN  UPDATE TAIKHOAN  SET TaiKhoan = @userName,MatKhau = @N\_passWord  WHERE TaiKhoan = @userName and MatKhau = @O\_passWord  DECLARE @sqlString NVARCHAR(2000)  SET @sqlString = 'ALTER LOGIN [' + @userName + '] WITH PASSWORD = ''' + @N\_passWord+ ''''  EXEC (@sqlString)  END; |

# CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG

## 1. Kết nối cơ sở dữ liệu

|  |
| --- |
| public class DatabaseConnection  {  public SqlConnection connection;  private string connectionString;  public DatabaseConnection(string u, string p)  {  // Tạo chuỗi kết nối  connectionString = "Data Source=localhost;Initial Catalog=DANG\_KY\_MON\_HOC;User ID=" + u + ";Password=" + p;  connection = new SqlConnection(connectionString);  }  public DatabaseConnection()  {  // Tạo chuỗi kết nối  connectionString = "Data Source=localhost;Initial Catalog=DANG\_KY\_MON\_HOC;Integrated Security=True";  connection = new SqlConnection(connectionString);  }  public bool OpenConnection()  {  try  {  if (connection.State == System.Data.ConnectionState.Closed)  {  connection.Open();  }  return true;  }  catch (Exception ex)  {  Console.WriteLine("Lỗi kết nối: " + ex.Message);  return false;  }  }  public bool CloseConnection()  {  try  {  if (connection.State == System.Data.ConnectionState.Open)  {  connection.Close();  }  return true;  }  catch (Exception ex)  {  Console.WriteLine("Lỗi đóng kết nối: " + ex.Message);  return false;  }  }  } |

## 2. Chức năng của người quản lý

### 2.1. Kiểm tra tài khoản mật khẩu của người quản lý:

|  |
| --- |
| CREATE FUNCTION [dbo].[fu\_Check\_DangNhap\_ADMIN](@user NVARCHAR(50), @pass NVARCHAR(50))  RETURNS FLOAT  AS  BEGIN  DECLARE @result FLOAT;  IF EXISTS (SELECT (1) FROM TAIKHOAN WHERE TaiKhoan = @user  AND MatKhau = @pass  AND @user NOT IN (SELECT MaHocVien FROM HOCVIEN)  AND @user NOT IN (SELECT MaGiangVien FROM GIANGVIEN))  BEGIN  SET @result = 1  END  ELSE SET @result = 0  RETURN @result  END; |

Thực thi trên C#:

|  |
| --- |
| public bool Admin\_DangNhap(DatabaseConnection db, TextBox TaiKhoan, TextBox MatKhau)  {  string username = TaiKhoan.Text;  string password = MatKhau.Text;  db.OpenConnection();  string query = "SELECT [dbo].[fu\_Check\_DangNhap\_ADMIN](@user, @pass)";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@user", username);  cmd.Parameters.AddWithValue("@pass", password);  double check = 0;  object result = cmd.ExecuteScalar(); // Sử dụng kiểu object để kiểm tra null  if (result != DBNull.Value && result != null)  {  check = (double)result;  }  db.CloseConnection();  if (check == 1)  {  FormLogIn fromAdmin = new FormLogIn();  DatabaseConnection db\_a = new DatabaseConnection(username, password);  Admin nguoiquanly = new Admin(db\_a);  nguoiquanly.ShowDialog();  nguoiquanly = null;  return true;  }  else  {  MessageBox.Show("Lỗi: Không tìm thấy tài khoản mật khẩu", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return false;  }  } |

### 2.2. Tạo tài khoản

|  |
| --- |
| CREATE PROCEDURE proc\_TaoTK(@TaiKhoan NVARCHAR(20), @MatKhau NVARCHAR(20))  AS  BEGIN  INSERT INTO TaiKhoan(TaiKhoan, MatKhau) VALUES (@TaiKhoan, @MatKhau)  END; |

Thực thi trên C#

|  |
| --- |
| public void TaoTaiKhoan(DatabaseConnection db, string u, string p)  {  try  {  db.OpenConnection();  string query = "proc\_TaoTK";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@TaiKhoan", u);  cmd.Parameters.AddWithValue("@MatKhau", p);  cmd.ExecuteNonQuery();  MessageBox.Show("Thông báo: Tạo tài khoản thành công.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.CloseConnection();  }  } |

### 2.3. Đổi mật khẩu

|  |
| --- |
| CREATE PROCEDURE proc\_DoiMK(@TaiKhoan NVARCHAR(20), @MatKhauCu NVARCHAR(20), @MatKhauMoi NVARCHAR(20))  AS  BEGIN  UPDATE TAIKHOAN  SET TaiKhoan = @TaiKhoan, MatKhau = @MatKhauMoi  WHERE TaiKhoan = @TaiKhoan AND MatKhau = @MatKhauCu  END; |

Thực thi trên C#

|  |
| --- |
| public void DoiMatKhau(DatabaseConnection db, string u, string O\_p, string N\_p)  {  try  {  db.OpenConnection();  string query = "proc\_DoiMK";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@TaiKhoan", u);  cmd.Parameters.AddWithValue("@MatKhauCu", O\_p);  cmd.Parameters.AddWithValue("@MatKhauMoi", N\_p);  cmd.ExecuteNonQuery();  MessageBox.Show("Thông báo: Đổi mật khẩu thành công.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.CloseConnection();  }  } |

### 2.4. Quản lý giảng viên

#### 2.4.1. Hiển thị danh sách thông tin giảng viên:

|  |
| --- |
| CREATE FUNCTION fu\_load\_GV( )  RETURNS TABLE  AS RETURN (  SELECT MaGiangVien, HoTen, HocVi, HocHam, Diachi, Sdt, MaKhoa, MatKhau  FROM GIANGVIEN LEFT JOIN TAIKHOAN ON TaiKhoan = MaGiangVien  ); |

Thực thi trên C#:

|  |
| --- |
| private void Admin\_GiangVien\_Load(object sender, EventArgs e)  {  DataTable dt = new DataTable();  dt = giangvienDao.load\_DSGiangVien(db);  dgvGiangVien.DataSource = dt;  Utils.AutosizeColumns(dgvGiangVien);  } |

|  |
| --- |
| public DataTable load\_DSGiangVien(DatabaseConnection db)  {  db.OpenConnection();  string query = "SELECT \* FROM fu\_load\_GV()";  SqlCommand cmd = new SqlCommand(query, db.connection);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt;  } |

#### 2.4.2. Tìm kiếm thông tin một giảng viên:

|  |
| --- |
| CREATE FUNCTION [dbo].[func\_getGiangVienByMaGiangVien] (@maGiangVien CHAR(6))  RETURNS TABLE  AS  RETURN  (  SELECT MaGiangVien, HoTen, HocVi, HocHam, Diachi, Sdt, MaKhoa, MatKhau  FROM GIANGVIEN LEFT JOIN TAIKHOAN ON TaiKhoan = MaGiangVien  WHERE MaGiangVien = @maGiangVien  ); |

Thực thi trên C#:

|  |
| --- |
| private void Search\_GiangVien\_Click(object sender, EventArgs e)  {  DataTable dt = new DataTable();  dt = giangvienDao.get\_GiangVien(db, txtMaGiangVien);  dgvGiangVien.DataSource = dt;  Utils.AutosizeColumns(dgvGiangVien);  } |

|  |
| --- |
| public DataTable get\_GiangVien(DatabaseConnection db, TextBox MaGiangVien)  {  string maGiangVien = MaGiangVien.Text;  DataTable dt = new DataTable();  try  {  db.OpenConnection();  string query = "SELECT MaGiangVien, HoTen, HocVi, HocHam, DiaChi, Sdt, MaKhoa, MatKhau\n" +  "FROM dbo.func\_getGiangVienByMaGiangVien (@maGiangVien)";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@maGiangVien", maGiangVien);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  adapter.Fill(dt);  }  catch (Exception ex)  {  // Xử lý ngoại lệ  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.CloseConnection();  }  return dt;  } |

#### 2.4.3. Xoá thông tin một giảng viên:

|  |
| --- |
| CREATE PROCEDURE DeleteGiangVien  @MaGiangVien CHAR(6)  AS  BEGIN  BEGIN TRANSACTION  DECLARE @username VARCHAR(15);  SELECT @username = MaGiangVien FROM GIANGVIEN WHERE MaGiangVien = @MaGiangVien  DECLARE @sql VARCHAR(100)  DECLARE @SessionID INT;  SELECT @SessionID = session\_id  FROM sys.dm\_exec\_sessions  WHERE login\_name = @username;  IF @SessionID IS NOT NULL  BEGIN  SET @sql = 'KILL ' + CONVERT(NVARCHAR(20), @SessionID)  EXEC(@sql)  END    BEGIN TRY  DELETE FROM GIANGVIEN  WHERE MaGiangVien = @MaGiangVien;  DELETE FROM TAIKHOAN  WHERE TaiKhoan = @MaGiangVien  SET @sql = 'DROP USER '+ @username  EXEC (@sql)  SET @sql = 'DROP LOGIN '+ @username  EXEC (@sql)  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi ' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH  END; |

Thực thi trên C#:

|  |
| --- |
| private void Delete\_GiangVien\_Click(object sender, EventArgs e)  {  giangvienDao.DeleteGiangVien(db, txtMaGiangVien);  } |

|  |
| --- |
| public void DeleteGiangVien(DatabaseConnection db, TextBox MaGiangVien)  {  string maGiangVien = MaGiangVien.Text;  try  {  db.OpenConnection();  string query = "DeleteGiangVien";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaGiangVien", maGiangVien);  DialogResult result = MessageBox.Show("Bạn có chắc chắn muốn xóa giảng viên này không?", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Information);  if (result == DialogResult.OK)  {  cmd.ExecuteNonQuery();  MessageBox.Show("Thành công: Xóa giảng viên thành công.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (Exception ex)  {  // Xử lý ngoại lệ  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.CloseConnection();  }  } |

#### 2.4.4. Thêm một giảng viên:

|  |
| --- |
| CREATE PROCEDURE [dbo].[InsertGiangVien]  @MaGiangVien CHAR(6),  @HoTen NVARCHAR(MAX),  @HocVi NVARCHAR(10),  @HocHam NVARCHAR(12),  @DiaChi NVARCHAR(MAX),  @Sdt CHAR(10),  @MaKhoa VARCHAR(5)  AS  BEGIN  IF EXISTS (SELECT 1 FROM GIANGVIEN WHERE MaGiangVien = @MaGiangVien)  BEGIN  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Giảng viên đã có trong cơ sở dữ liệu'  RAISERROR(@err, 16, 1)  END  ELSE  BEGIN  INSERT INTO GIANGVIEN (MaGiangVien, HoTen, HocVi, HocHam, DiaChi, Sdt, MaKhoa)  VALUES (@MaGiangVien, @HoTen, @HocVi, @HocHam, @DiaChi, @Sdt, @MaKhoa);  END  END; |

Thực thi trên C#:

|  |
| --- |
| private void btnThemGiangVien\_Click(object sender, EventArgs e)  {  GiangVienAdminDTO gv = new GiangVienAdminDTO(txtMaGiangVien.Text, txtTenGiangVien.Text, txtHocVi.Text, txtHocHam.Text, txtDiaChi.Text, txtSdt.Text, txtMaKhoa.Text, txtMatKhau.Text);  bool temp = giangvienDao.InsertGiangVien(db, gv);  if (temp)  {  MessageBox.Show("Thêm thành công!");  }  else  {  MessageBox.Show("Không thể thêm!");  }  } |

|  |
| --- |
| public bool InsertGiangVien(DatabaseConnection db, GiangVienAdminDTO gv)  {  try  {  db.OpenConnection();  string query = "InsertGiangVien";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaGiangVien", gv.MaGiangVien);  cmd.Parameters.AddWithValue("@HoTen", gv.HoTen);  cmd.Parameters.AddWithValue("@HocVi", gv.HocVi);  cmd.Parameters.AddWithValue("@HocHam", gv.HocHam);  cmd.Parameters.AddWithValue("@DiaChi", gv.DiaChi);  cmd.Parameters.AddWithValue("@Sdt", gv.Sdt);  cmd.Parameters.AddWithValue("@MaKhoa", gv.MaKhoa);  cmd.Parameters.AddWithValue("@MatKhau", gv.MatKhau);  int rowsAffected = cmd.ExecuteNonQuery();  if (rowsAffected > 0)  {  return true;  }  else  {  return false;  }  }  catch (Exception ex)  {  // Xử lý ngoại lệ  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return false;  }  finally  {  db.CloseConnection();  }  } |

#### 2.4.5. Cập nhật thông tin một giảng viên:

|  |
| --- |
| CREATE PROCEDURE [dbo].[UpdateGiangVien]  @MaGiangVien CHAR(6),  @HoTen NVARCHAR(MAX),  @HocVi NVARCHAR(10),  @HocHam NVARCHAR(12),  @DiaChi NVARCHAR(MAX),  @Sdt CHAR(10),  @MaKhoa VARCHAR(5),  @MatKhau VARCHAR(20)  AS  BEGIN  BEGIN TRY  BEGIN TRANSACTION;  IF EXISTS (SELECT 1 FROM GIANGVIEN WHERE MaGiangVien = @MaGiangVien)  BEGIN  UPDATE GIANGVIEN  SET  HoTen = @HoTen,  HocVi = @HocVi,  HocHam = @HocHam,  DiaChi = @DiaChi,  Sdt = @Sdt,  MaKhoa = @MaKhoa,  MatKhau = @MatKhau  WHERE MaGiangVien = @MaGiangVien;  END  ELSE  BEGIN  RAISERROR(N'Giảng viên với mã giảng viên không tồn tại.', 16, 1);  END  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  IF @@TRANCOUNT > 0  ROLLBACK TRANSACTION;  DECLARE @err NVARCHAR(MAX);  SELECT @err = N'Lỗi ' + ERROR\_MESSAGE();  RAISERROR(@err, 16, 1);  END CATCH  END; |

Thực thi trên C#:

|  |
| --- |
| private void btnCapNhatGV\_Click(object sender, EventArgs e)  {  GiangVienAdminDTO gv = new GiangVienAdminDTO(txtMaGiangVien.Text, txtTenGiangVien.Text, txtHocVi.Text, txtHocHam.Text, txtDiaChi.Text, txtSdt.Text, txtMaKhoa.Text, txtMatKhau.Text);  bool temp = giangvienDao.UpdateGiangVien(db, gv);  if (temp)  {  MessageBox.Show("Cập nhật thành công!");  }  else  {  MessageBox.Show("Không thể cập nhật!");  }  } |

|  |
| --- |
| public bool UpdateGiangVien(DatabaseConnection db, GiangVienAdminDTO gv)  {  try  {  db.OpenConnection();  string query = "UpdateGiangVien";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaGiangVien", gv.MaGiangVien);  cmd.Parameters.AddWithValue("@HoTen", gv.HoTen);  cmd.Parameters.AddWithValue("@HocVi", gv.HocVi);  cmd.Parameters.AddWithValue("@HocHam", gv.HocHam);  cmd.Parameters.AddWithValue("@DiaChi", gv.DiaChi);  cmd.Parameters.AddWithValue("@Sdt", gv.Sdt);  cmd.Parameters.AddWithValue("@MaKhoa", gv.MaKhoa);  cmd.Parameters.AddWithValue("@MatKhau", gv.MatKhau);  int rowsAffected = cmd.ExecuteNonQuery();  if (rowsAffected > 0)  {  return true;  }  else  {  return false;  }  }  catch (Exception ex)  {  // Xử lý ngoại lệ  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return false;  }  finally  {  db.CloseConnection();  }  } |

### 2.5. Quản lý học viên

#### 2.5.1. Hiển thị danh sách thông tin học viên:

|  |
| --- |
| CREATE FUNCTION fu\_load\_HocVien ( )  RETURNS TABLE  AS RETURN (  SELECT MaHocVien, HoTen, NgaySinh, DiaChi, QueQuan, MaKhoa, MatKhau  FROM HOCVIEN LEFT JOIN TAIKHOAN ON MaHocVien = TaiKhoan  ); |

Thực thi trên C#:

|  |
| --- |
| private void Admin\_HocVien\_Load(object sender, EventArgs e)  {  DataTable dt = new DataTable();  dt = hocvienDAO.load\_DSHocVien(db);  dgvHocVien.DataSource = dt;  Utils.AutosizeColumns(dgvGiangVien);  } |

|  |
| --- |
| public DataTable load\_DSHocVien(DatabaseConnection db)  {  db.OpenConnection();  string query = "SELECT \* FROM fu\_load\_HocVien()";  SqlCommand cmd = new SqlCommand(query, db.connection);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt;  } |

#### 2.5.2. Tìm kiếm thông tin một học viên:

|  |
| --- |
| CREATE FUNCTION [dbo].[func\_getHocVienByMaHocVien] (@maHocVien CHAR(10))  RETURNS TABLE  AS  RETURN  (  SELECT MaHocVien, HoTen, NgaySinh, DiaChi, QueQuan, MaKhoa, MatKhau  FROM HOCVIEN LEFT JOIN TAIKHOAN ON MaHocVien = TaiKhoan  WHERE MaHocVien = @maHocVien  ); |

Thực thi trên C#:

|  |
| --- |
| private void btnSearchHocVien\_Click(object sender, EventArgs e)  {  DataTable dt = new DataTable();  dt = hocvienDAO.get\_HocVien(db, txtMaHocVien);  dgvHocVien.DataSource = dt;  } |

|  |
| --- |
| public DataTable get\_HocVien(DatabaseConnection db, TextBox MaHocVien)  {  string maHocVien = MaHocVien.Text;  DataTable dt = new DataTable();  try  {  db.OpenConnection();  string query = "SELECT MaHocVien, HoTen, NgaySinh, DiaChi, QueQuan, MaKhoa, MatKhau\n" +  "FROM dbo.func\_getHocVienByMaHocVien(@maHocVien)";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@maHocVien", maHocVien);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  adapter.Fill(dt);  }  catch (Exception ex)  {  // Xử lý ngoại lệ  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.CloseConnection();  }  return dt;  } |

#### 2.5.3. Xoá thông tin một học viên:

|  |
| --- |
| CREATE PROCEDURE DeleteHocVien  @MaHocVien CHAR(10)  AS  BEGIN  BEGIN TRANSACTION  DECLARE @username VARCHAR(15);  SELECT @username = MaHocVien FROM HOCVIEN WHERE MaHocVien = @MaHocVien  DECLARE @sql VARCHAR(100)  DECLARE @SessionID INT;  SELECT @SessionID = session\_id  FROM sys.dm\_exec\_sessions  WHERE login\_name = @username;  IF @SessionID IS NOT NULL  BEGIN  SET @sql = 'KILL ' + CONVERT(NVARCHAR(20), @SessionID)  EXEC(@sql)  END    BEGIN TRY  UPDATE LOPHOC  SET LOPHOC.SoHocVienDangKy = LOPHOC.SoHocVienDangKy - 1  WHERE MaLop IN (SELECT MaLop FROM DKMH WHERE MaHocVien = @MaHocVien);  -- Xóa dòng dữ liệu có chứa @MaHocVien trong bảng DKMH  DELETE FROM DKMH  WHERE MaHocVien = @MaHocVien;  -- Xóa học viên từ bảng HOCVIEN  DELETE FROM HOCVIEN  WHERE MaHocVien = @MaHocVien;  ----  DELETE FROM TAIKHOAN  WHERE taikhoan = @MaHocVien  SET @sql = 'DROP USER ' + @username  EXEC (@sql)  SET @sql = 'DROP LOGIN ' + @username  EXEC (@sql)  COMMIT;  END TRY  BEGIN CATCH  -- Nếu có lỗi, hủy giao dịch và hiển thị thông báo lỗi  ROLLBACK;  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Không thể xóa, kiểm tra lại mã học viên'  RAISERROR(@err, 16, 1)  END CATCH;  END; |

Thực thi trên C#:

|  |
| --- |
| private void btnDeleteHocVien\_Click(object sender, EventArgs e)  {  hocvienDAO.DeleteHocVien(db, txtMaHocVien);  } |

|  |
| --- |
| public void DeleteHocVien(DatabaseConnection db, TextBox MaHocVien)  {  string maHocVien = MaHocVien.Text;  try  {  db.OpenConnection();  string query = "DeleteHocVien";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaHocVien", maHocVien);  DialogResult result = MessageBox.Show("Bạn có chắc chắn muốn xóa học viên này không?", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Information);  if (result == DialogResult.OK)  {  cmd.ExecuteNonQuery();  MessageBox.Show("Thành công: Xóa học viên thành công.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (Exception ex)  {  // Xử lý ngoại lệ  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.CloseConnection();  }  } |

#### 2.5.4. Thêm một học viên:

|  |
| --- |
| CREATE PROCEDURE [dbo].[InsertHocVien]  @MaHocVien CHAR(10),  @HoTen NVARCHAR(40),  @NgaySinh DATE,  @DiaChi NVARCHAR(MAX),  @QueQuan NVARCHAR(20),  @MaKhoa VARCHAR(5)  AS  BEGIN  IF EXISTS (SELECT 1 FROM HOCVIEN WHERE MaHocVien = @MaHocVien)  BEGIN  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Học viên đã tồn tại trong cơ sở dữ liệu'  RAISERROR(@err, 16, 1)  END  ELSE  BEGIN  INSERT INTO HOCVIEN (MaHocVien, HoTen, NgaySinh, DiaChi, QueQuan, MaKhoa)  VALUES (@MaHocVien, @HoTen, @NgaySinh, @DiaChi, @QueQuan, @MaKhoa);  END  END; |

Thực thi trên C#:

|  |
| --- |
| private void btnThemHocVien\_Click(object sender, EventArgs e)  {  HocVienAdminDTO hv = new HocVienAdminDTO(txtMaHocVien.Text, txtTenHocVien.Text, dtpNgaySinh.Value, txtDiaChi.Text, txtQueQuan.Text, txtMaKhoaHocVien.Text, txtMatKhauHocVien.Text);  bool temp = hocvienDAO.InsertHocVien(db, hv);  if (temp)  {  MessageBox.Show("Thêm thành công!");  }  else  {  MessageBox.Show("Không thể thêm!");  }  } |

|  |
| --- |
| public bool InsertHocVien(DatabaseConnection db, HocVienAdminDTO hv)  {  try  {  db.OpenConnection();  string query = "InsertHocVien";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaHocVien", hv.MaHocVien);  cmd.Parameters.AddWithValue("@HoTen", hv.HoTen);  cmd.Parameters.AddWithValue("@NgaySinh", hv.NgaySinh);  cmd.Parameters.AddWithValue("@DiaChi", hv.DiaChi);  cmd.Parameters.AddWithValue("@QueQuan", hv.QueQuan);  cmd.Parameters.AddWithValue("@MaKhoa", hv.MaKhoa);  cmd.Parameters.AddWithValue("@MatKhau", hv.MatKhau);  int rowsAffected = cmd.ExecuteNonQuery();  if (rowsAffected > 0)  {  return true;  }  else  {  return false;  }  }  catch (Exception ex)  {  // Xử lý ngoại lệ  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return false;  }  finally  {  db.CloseConnection();  }  } |

#### 2.5.5. Cập nhật thông tin một học viên:

|  |
| --- |
| CREATE PROCEDURE [dbo].[UpdateHocVien]  @MaHocVien CHAR(10),  @HoTen NVARCHAR(MAX),  @NgaySinh DATE,  @DiaChi NVARCHAR(MAX),  @QueQuan NVARCHAR(20),  @MaKhoa VARCHAR(5)  AS  BEGIN  BEGIN TRY  BEGIN TRANSACTION;  IF EXISTS (SELECT 1 FROM HOCVIEN WHERE MaHocVien = @MaHocVien)  BEGIN  UPDATE HOCVIEN  SET  HoTen = @HoTen,  NgaySinh = @NgaySinh,  DiaChi = @DiaChi,  QueQuan = @QueQuan,  MaKhoa = @MaKhoa  WHERE MaHocVien = @MaHocVien;  END  ELSE  BEGIN  RAISERROR(N'!', 16, 1);  END  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  IF @@TRANCOUNT > 0  ROLLBACK TRANSACTION;  DECLARE @err NVARCHAR(MAX);  SELECT @err = N'Học viên với mã học viên đã tồn tại' + ERROR\_MESSAGE();  RAISERROR(@err, 16, 1);  END CATCH  END; |

Thực thi trên C#:

|  |
| --- |
| private void btnCapNhatHV\_Click(object sender, EventArgs e)  {  HocVienAdminDTO hv = new HocVienAdminDTO(txtMaHocVien.Text, txtTenHocVien.Text,  dtpNgaySinh.Value, txtDiaChi.Text, txtQueQuan.Text, txtMaKhoaHocVien.Text, txtMatKhauHocVien.Text);  bool temp = hocvienDAO.UpdateHocVien(db, hv);  if (temp)  {  MessageBox.Show("Cập nhật thành công!");  }  else  {  MessageBox.Show("Không thể cập nhật!");  }  } |

|  |
| --- |
| public bool UpdateHocVien(DatabaseConnection db, HocVienAdminDTO hv)  {  try  {  db.OpenConnection();  string query = "UpdateHocVien";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaHocVien", hv.MaHocVien);  cmd.Parameters.AddWithValue("@HoTen", hv.HoTen);  cmd.Parameters.AddWithValue("@NgaySinh", hv.NgaySinh);  cmd.Parameters.AddWithValue("@DiaChi", hv.DiaChi);  cmd.Parameters.AddWithValue("@QueQuan", hv.QueQuan);  cmd.Parameters.AddWithValue("@MaKhoa", hv.MaKhoa);  cmd.Parameters.AddWithValue("@MatKhau", hv.MatKhau);  int rowsAffected = cmd.ExecuteNonQuery();  if (rowsAffected > 0)  {  return true;  }  else  {  return false;  }  }  catch (Exception ex)  {  // Xử lý ngoại lệ  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return false;  }  finally  {  db.CloseConnection();  }  } |

### 2.6. Quản lý lớp học

#### 2.6.1. Hiển thị danh sách thông tin lớp học:

|  |
| --- |
| CREATE FUNCTION [dbo].[DanhSachLopHoc] ( )  RETURNS TABLE  AS  RETURN  (  SELECT MaLop, TenLop, MaGiangVien, MaMon, MaHocKy, SoHocVienDangKy, MaPhong, Thu, Tiet  FROM LOPHOC  ); |

Thực thi trên C#:

|  |
| --- |
| private void QuanLyLopHoc\_Load(object sender, EventArgs e)  {  DataTable dt = new DataTable();  dt = lophocDAO.load\_lopHoc(db);  dgvLopHoc.DataSource = dt;  Utils.AutosizeColumns(dgvLopHoc);  } |

|  |
| --- |
| public DataTable load\_lopHoc(DatabaseConnection db)  {  db.OpenConnection();  string query = "SELECT \* FROM DanhSachLopHoc()";  SqlCommand cmd = new SqlCommand(query, db.connection);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt;  } |

#### 2.6.2. Tìm kiếm thông tin một lớp học:

|  |
| --- |
| CREATE FUNCTION [dbo].[LayThongTinLopHocPhan] (@MaLop CHAR(18))  RETURNS TABLE  AS  RETURN  (  SELECT MaLop, TenLop, MaGiangVien, MaMon, MaHocKy, SoHocVienDangKy, MaPhong, Thu, Tiet  FROM LOPHOC  WHERE MaLop = @MaLop  ); |

Thực thi trên C#:

|  |
| --- |
| private void btnSearchLopHoc\_Click(object sender, EventArgs e)  {  DataTable dt = new DataTable();  dt = lophocDAO.GetLopHoc(db, txtMaLopHoc);  dgvLopHoc.DataSource = dt;  Utils.AutosizeColumns(dgvLopHoc);  } |

|  |
| --- |
| public DataTable GetLopHoc(DatabaseConnection db, TextBox MaLop)  {  string maLop = MaLop.Text;  DataTable dt = new DataTable();  try  {  db.OpenConnection();  string query = "SELECT MaLop, TenLop, MaGiangVien, MaMon, MaHocKy, SoHocVienDangKy, MaPhong, Thu, Tiet\n" +  "FROM dbo.LOPHOC\n" +  "WHERE MaLop = @maLop";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@maLop", maLop);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  adapter.Fill(dt);  }  catch (Exception ex)  {  // Xử lý ngoại lệ  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.CloseConnection();  }  return dt;  } |

#### 2.6.3. Xoá thông tin một lớp học:

|  |
| --- |
| CREATE PROCEDURE [dbo].[XoaLopHoc]  @MaLop CHAR(18)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  DELETE FROM DKMH  WHERE MaLop = @MaLop  DELETE FROM LOPHOC  WHERE MaLop = @MaLop;  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi ' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH  END; |

Thực thi trên C#:

|  |
| --- |
| private void btnDeleteLopHP\_Click(object sender, EventArgs e)  {  lophocDAO.Delete\_LopHoc(db, txtMaLopHoc);  } |

|  |
| --- |
| public void DeleteLopHoc(DatabaseConnection db, TextBox MaLopHoc)  {  string maLopHoc = MaLopHoc.Text;  try  {  db.OpenConnection();  string query = "XoaLopHoc";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaLop", maLopHoc);  DialogResult result = MessageBox.Show("Bạn có chắc chắn muốn xóa lớp học này không?", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Information);  if (result == DialogResult.OK)  {  cmd.ExecuteNonQuery();  MessageBox.Show("Thành công: Xóa lớp học thành công.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (Exception ex)  {  // Xử lý ngoại lệ  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.CloseConnection();  }  } |

#### 2.6.4. Thêm một lớp học:

|  |
| --- |
| CREATE PROCEDURE [dbo].[ThemLopHoc]  @MaLop CHAR(18),  @TenLop NVARCHAR(MAX),  @MaGiangVien CHAR(6),  @MaMon CHAR(10),  @MaHocKy CHAR(9),  @SoHocVienDangKy INT,  @MaPhong VARCHAR(10),  @Thu NVARCHAR(10),  @Tiet VARCHAR(10)  AS  BEGIN  INSERT INTO LOPHOC (MaLop, TenLop, MaGiangVien, MaMon, MaHocKy, SoHocVienDangKy, MaPhong, Thu, Tiet)  VALUES (@MaLop, @TenLop, @MaGiangVien, @MaMon, @MaHocKy, @SoHocVienDangKy, @MaPhong, @Thu, @Tiet);  END; |

Thực thi trên C#:

|  |
| --- |
| private void btnThemLopHoc\_Click(object sender, EventArgs e)  {  LopHocAdminDTO lh = new LopHocAdminDTO(txtMaLop.Text, txtTenLop.Text, txtMaGiangVien.Text,  txtMaMon.Text, txtMaHocKy.Text, ((int)numericSoHV.Value), txtMaPhong.Text, txtThu.Text, txtTietHoc.Text);  bool temp = lophocDao.InsertLopHoc(db, lh);  if (temp)  {  MessageBox.Show("Thêm thành công!");  }  else  {  MessageBox.Show("Không thể thêm!");  }  } |

|  |
| --- |
| public bool InsertLopHoc(DatabaseConnection db, LopHocAdminDTO lopHoc)  {  try  {  db.OpenConnection();  string query = "ThemLopHoc";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaLop", lopHoc.MaLop);  cmd.Parameters.AddWithValue("@TenLop", lopHoc.TenLop);  cmd.Parameters.AddWithValue("@MaGiangVien", lopHoc.MaGiangVien);  cmd.Parameters.AddWithValue("@MaMon", lopHoc.MaMon);  cmd.Parameters.AddWithValue("@MaHocKy", lopHoc.MaHocKy);  cmd.Parameters.AddWithValue("@SoHocVienDangKy", lopHoc.SoHocVienDangKy);  cmd.Parameters.AddWithValue("@MaPhong", lopHoc.MaPhong);  cmd.Parameters.AddWithValue("@Thu", lopHoc.Thu);  cmd.Parameters.AddWithValue("@Tiet", lopHoc.Tiet);  int rowsAffected = cmd.ExecuteNonQuery();  if (rowsAffected > 0)  {  return true;  }  else  {  return false;  }  }  catch (Exception ex)  {  // Xử lý ngoại lệ  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return false;  }  finally  {  db.CloseConnection();  }  } |

#### 2.6.5. Cập nhật thông tin một lớp học:

|  |
| --- |
| CREATE PROCEDURE [dbo].[UpdateLopHoc]  @MaLop CHAR(18),  @TenLop NVARCHAR(50),  @MaGiangVien CHAR(10),  @MaMon CHAR(10),  @MaHocKy CHAR(10),  @SoHocVienDangKy INT,  @MaPhong CHAR(10),  @Thu NVARCHAR(10),  @Tiet NVARCHAR(50)  AS  BEGIN  BEGIN TRY  BEGIN TRANSACTION;  IF EXISTS (SELECT 1 FROM LOPHOC WHERE MaLop = @MaLop)  BEGIN  UPDATE LOPHOC  SET  TenLop = @TenLop,  MaGiangVien = @MaGiangVien,  MaMon = @MaMon,  MaHocKy = @MaHocKy,  SoHocVienDangKy = @SoHocVienDangKy,  MaPhong = @MaPhong,  Thu = @Thu,  Tiet = @Tiet  WHERE MaLop = @MaLop;  END  ELSE  BEGIN  RAISERROR(N'Lớp học với mã lớp = %s không tồn tại.', 16, 1, @MaLop);  END  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  IF @@TRANCOUNT > 0  ROLLBACK TRANSACTION;  DECLARE @err NVARCHAR(MAX);  SELECT @err = N'Lỗi ' + ERROR\_MESSAGE();  RAISERROR(@err, 16, 1);  END CATCH  END; |

Thực thi trên C#:

|  |
| --- |
| private void btnUpdateLH\_Click(object sender, EventArgs e)  {  LopHocAdminDTO lh = new LopHocAdminDTO(txtMaLop.Text, txtTenLop.Text, txtMaGiangVien.Text, txtMaMon.Text, txtMaHocKy.Text, ((int)numericSoHV.Value), txtMaPhong.Text, txtThu.Text, txtTietHoc.Text);  bool temp = lophocDao.UpdateLopHoc(db, lh);  if (temp)  {  MessageBox.Show("Cập nhật thành công!");  }  else  {  MessageBox.Show("Không thể cập nhật!");  }  } |

|  |
| --- |
| public bool UpdateLopHoc(DatabaseConnection db, LopHocAdminDTO lopHoc)  {  try  {  db.OpenConnection();  string query = "UpdateLopHoc";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaLop", lopHoc.MaLop);  cmd.Parameters.AddWithValue("@TenLop", lopHoc.TenLop);  cmd.Parameters.AddWithValue("@MaGiangVien", lopHoc.MaGiangVien);  cmd.Parameters.AddWithValue("@MaMon", lopHoc.MaMon);  cmd.Parameters.AddWithValue("@MaHocKy", lopHoc.MaHocKy);  cmd.Parameters.AddWithValue("@SoHocVienDangKy", lopHoc.SoHocVienDangKy);  cmd.Parameters.AddWithValue("@MaPhong", lopHoc.MaPhong);  cmd.Parameters.AddWithValue("@Thu", lopHoc.Thu);  cmd.Parameters.AddWithValue("@Tiet", lopHoc.Tiet);  int rowsAffected = cmd.ExecuteNonQuery();  if (rowsAffected > 0)  {  return true;  }  else  {  return false;  }  }  catch (Exception ex)  {  // Xử lý ngoại lệ  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return false;  }  finally  {  db.CloseConnection();  }  } |

## 3. Chức năng của học viên

### 3.1. Đăng nhập vào hệ thống:

|  |
| --- |
| CREATE PROCEDURE proc\_HV\_DN  @TaiKhoan CHAR(10),  @MatKhau CHAR(18)  AS  BEGIN  IF EXISTS (SELECT 1 FROM TAIKHOAN, HOCVIEN WHERE TaiKhoan = @TaiKhoan and MatKhau = @MatKhau and MaHocVien = @TaiKhoan)  BEGIN  SELECT HoTen, MaKhoa  FROM HOCVIEN  WHERE MaHocVien = @TaiKHoan  END  ELSE  BEGIN  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Sai Tài khoản hoặc Mật khẩu'  RAISERROR(@err, 16, 1)  END  END; |

Thực thi trên C#:

|  |
| --- |
| public void HocVien\_DangNhap(DatabaseConnection db, TextBox U, TextBox P)  {  string username = U.Text;  string password = P.Text;  if (db.OpenConnection())  {  SqlCommand command = new SqlCommand("proc\_HV\_DN", db.connection);  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaHocVien", username);  command.Parameters.AddWithValue("@MatKhau", password);  SqlDataReader reader = command.ExecuteReader();  try  {  if (reader.Read())  {  string HoTen = reader["HoTen"].ToString();  string MaKhoa = reader["MaKHoa"].ToString();  HocVienDTO hv = new HocVienDTO(HoTen, username, MaKhoa);  FormHVDangKyLopHoc f\_sv = new FormHVDangKyLopHoc();  f\_sv.Update\_Form(hv);  f\_sv.ShowDialog();  reader.Close();  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.CloseConnection();  }  }  else  {  MessageBox.Show("Lỗi kết nối đến cơ sở dữ liệu.", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

### 3.2. Xem các môn có thể đăng kí:

|  |
| --- |
| CREATE FUNCTION fu\_load\_MonDK (@MaKhoa VARCHAR(5)) --Hàm load các môn học cho học viên đăng ký  RETURNS TABLE  AS RETURN (  SELECT MaMon, TenMonHoc, SoTinChi  FROM MonHoc  WHERE MaKhoa = @MaKhoa  ); |

Thực thi trên C#:

|  |
| --- |
| public DataTable load\_MonDK(DatabaseConnection db, HocVienDTO hv)  {  db.OpenConnection();  string query = "SELECT \* FROM fu\_load\_MonDK(@MaKhoa)";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@MaKhoa", hv.Makhoa);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt;  } |

### 3.3. Xem các môn đã đăng kí:

|  |
| --- |
| CREATE VIEW v\_DSDaDangKi AS  SELECT lh.MaLop, mh.TenMonHoc, mh.SoTinChi, lh.Tiet, lh.Thu, p.MaPhong Phong, gv.HoTen AS GiangVien, dk.MaHocVien, lh.MaMon  FROM dbo.MONHOC mh  JOIN dbo.LOPHOC lh ON lh.MaMon = mh.MaMon  JOIN dbo.GIANGVIEN gv ON lh.MaGiangVien = gv.MaGiangVien  JOIN dbo.PHONGHOC p ON lh.MaPhong = p.MaPhong  JOIN dbo.DKMH dk ON dk.MaLop = lh.MaLop; |

Thực thi trên C#:

|  |
| --- |
| public DataTable load\_MonDaDK(DatabaseConnection db, HocVienDTO hv)  {  db.OpenConnection();  string query = "SELECT \* FROM v\_DSDaDangKi WHERE MaHocVien = @MaHocVien";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@MaHocVien", hv.Mahv);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt;  } |

### 3.4. Xem danh sách lớp của một môn:

|  |
| --- |
| CREATE VIEW v\_DSLopHoc AS  SELECT lh.MaLop, mh.TenMonHoc, mh.SoTinChi, lh.Tiet, lh.Thu, p.MaPhong Phong, lh.MaHocKy, gv.HoTen AS 'Giảng viên', mh.MaKhoa, mh.MaMon, lh.SoHocVienDangKy  FROM dbo.MONHOC mh  JOIN dbo.LOPHOC lh ON lh.MaMon = mh.MaMon  JOIN dbo.GIANGVIEN gv ON lh.MaGiangVien = gv.MaGiangVien  JOIN dbo.PHONGHOC p ON lh.MaPhong = p.MaPhong; |

Thực thi trên C#:

|  |
| --- |
| public DataTable load\_DSLop(DatabaseConnection db, string MaMon, string MaHK)  {  db.OpenConnection();  string query = "SELECT \* FROM v\_DSLopHoc WHERE MaMon = @MaMon AND MaHocKy = @MaHocKy";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@MaMon", MaMon);  cmd.Parameters.AddWithValue("@MaHocKy", MaHK);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt;  } |

### 3.5. Học viên đăng ký lớp học:

|  |
| --- |
| CREATE PROCEDURE proc\_DKLopHoc --Học viên đăng ký lớp học  @MaHV CHAR(10),  @MaLH CHAR(18)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  INSERT INTO DKMH (MaHocVien, MaLop)  VALUES(@MaHV,@MaLH)  UPDATE LOPHOC  SET LOPHOC.SoHocVienDangKy = LOPHOC.SoHocVienDangKy + 1  WHERE MaLop = @MaLH  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi ' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH  END; |

Thực thi trên C#:

|  |
| --- |
| public void DK\_LopHoc(DatabaseConnection db, HocVienDTO hv, string LopHoc)  {  db.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_DKLopHoc", db.connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaLH", LopHoc);  cmd.Parameters.AddWithValue("@MaHV", hv.Mahv);  try  {  cmd.ExecuteNonQuery();  MessageBox.Show("Thành công: Đăng ký lớp học thành công.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.CloseConnection();  }  } |

### 3.6. Học viên xóa đăng ký lớp học:

|  |
| --- |
| CREATE PROCEDURE proc\_Xoa\_DKLopHoc --Học viên xoá đăng ký lớp học  @MaHV CHAR(10),  @MaLH CHAR(18)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  DELETE FROM DKMH WHERE MaHocVien = @MaHV AND MaLop = @MaLH  UPDATE LOPHOC  SET LOPHOC.SoHocVienDangKy = LOPHOC.SoHocVienDangKy - 1  WHERE MaLop = @MaLH  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi ' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH  END; |

Thực thi trên C#:

|  |
| --- |
| public void Xoa\_DK\_LopHoc(DatabaseConnection db, HocVienDTO hv, string LopHoc)  {  db.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_Xoa\_DKLopHoc", db.connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaLH", LopHoc);  cmd.Parameters.AddWithValue("@MaHV", hv.Mahv);  try  {  DialogResult result = MessageBox.Show("Bạn có chắc xóa môn này không", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Information);  if (result == DialogResult.OK)  {  cmd.ExecuteNonQuery();  MessageBox.Show("Thành công: Xóa lớp thành công.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.CloseConnection();  }    } |

### 3.7. Học viên chuyển lớp:

|  |
| --- |
| CREATE PROCEDURE proc\_ChuyenLopHoc --Học viên chuyển lớp học  @MaHV CHAR(10),  @MaLH CHAR(18), --lớp học hiện tại  @MaLHDK CHAR(18) --lớp học đăng ký mới  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  UPDATE DKMH  SET MaLop = @MaLHDK  WHERE MaHocVien = @MaHV and MaLop = @MaLH;  UPDATE LOPHOC  SET LOPHOC.SoHocVienDangKy = LOPHOC.SoHocVienDangKy - 1  WHERE MaLop = @MaLH  UPDATE LOPHOC  SET LOPHOC.SoHocVienDangKy = LOPHOC.SoHocVienDangKy + 1  WHERE MaLop = @MaLHDK  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi ' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH  END; |

Thực thi trên C#:

|  |
| --- |
| public bool Chuyen\_LopHoc(DatabaseConnection db, HocVienDTO hv, string LopHoc, string MaLopDK)  {  db.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_ChuyenLopHoc", db.connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaLH", LopHoc);  cmd.Parameters.AddWithValue("@MaHV", hv.Mahv);  cmd.Parameters.AddWithValue("@MaLHDK", MaLopDK);  try  {  cmd.ExecuteNonQuery();  MessageBox.Show("Thành công: Chuyển lớp học thành công.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return true;  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return false;  }  finally  {  db.CloseConnection();  }  } |

### 3.8. Học viên xem danh sách học phí:

|  |
| --- |
| CREATE FUNCTION fu\_load\_DSHocPhi (@MaHocVien CHAR(10), @MaKhoa VARCHAR(5)) --Hàm danh sách học phí  RETURNS TABLE  AS  RETURN  (  SELECT lh.MaLop, mh.TenMonHoc, lh.Tiet, lh.Thu, p.MaPhong Phong,  gv.HoTen AS GiangVien, dk.MaHocVien, lh.MaMon, mh.SoTinChi,  CASE  WHEN @MaKhoa = 'FHQ' THEN mh.SoTinChi \* 800000  ELSE mh.SoTinChi \* 600000  END AS HocPhi  FROM dbo.MONHOC mh  JOIN dbo.LOPHOC lh ON lh.MaMon = mh.MaMon  JOIN dbo.GIANGVIEN gv ON lh.MaGiangVien = gv.MaGiangVien  JOIN dbo.PHONGHOC p ON lh.MaPhong = p.MaPhong  JOIN dbo.DKMH dk ON dk.MaLop = lh.MaLop  WHERE dk.MaHocVien = @MaHocVien  ); |

Thực thi trên C#:

|  |
| --- |
| public DataTable load\_DSHP(DatabaseConnection db, HocVienDTO hv)  {  db.OpenConnection();  string query = "SELECT \* FROM fu\_load\_DSHocPhi(@MaHocVien, @MaKhoa)";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@MaHocVien", hv.Mahv);  cmd.Parameters.AddWithValue("@MaKhoa", hv.Makhoa);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt;  } |

### 3.9. Học viên xem tổng học phí:

|  |
| --- |
| CREATE FUNCTION [dbo].[fu\_TongHocPhi](@MaHocVien NVARCHAR(10), @MaKhoa VARCHAR(5))  RETURNS FLOAT  AS  BEGIN  DECLARE @tinchi FLOAT;  DECLARE @HocPhi FLOAT;  -- Sử dụng SELECT để gán giá trị cho biến result  SELECT @tinchi = SUM(SoTinChi) FROM v\_DSDaDangKi WHERE MaHocVien = @MaHocVien;  -- Nếu không có dữ liệu, gán giá trị mặc định cho biến result  IF (@tinchi IS NULL)  SET @tinchi = 0;  -- Trả về giá trị  IF ( @MaKhoa = 'FHQ')  BEGIN  SET @HocPhi = @tinchi \* 800000  END  ELSE  BEGIN  SET @HocPhi = @tinchi \* 600000  END  RETURN @HocPhi;  END; |

Thực thi trên C#:

|  |
| --- |
| public string TinhHocPhi(DatabaseConnection db, HocVienDTO hv)  {  db.OpenConnection();  string query = "SELECT [dbo].[fu\_TongHocPhi](@MaHocVien, @MaKhoa)";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@MaHocVien", hv.Mahv);  cmd.Parameters.AddWithValue("@MaKhoa", hv.Makhoa);  double hocPhi = 0;  object result = cmd.ExecuteScalar(); // Sử dụng kiểu object để kiểm tra null  if (result != DBNull.Value && result != null)  {  hocPhi = (double)result;  }  db.CloseConnection();  return hocPhi.ToString();  } |

### 3.10. Tìm kiếm các lớp đang mở của một môn:

|  |
| --- |
| CREATE FUNCTION fu\_load\_DSTimKiem (@string NVARCHAR(50), @MaHocKy CHAR(9)) --Hàm load các lớp học cho học viên đăng ký  RETURNS TABLE  AS RETURN (  SELECT \*  FROM v\_DSLopHoc  WHERE MaMon LIKE '%' + @string + '%' and MaHocKy = @MaHocKy  ); |

Thực thi trên C#:

|  |
| --- |
| public DataTable TimKiem(DatabaseConnection db, string s, string MaHK)  {  db.OpenConnection();  string query = "SELECT \* FROM fu\_load\_DSTimKiem (@string, @MaHocKy)";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@string", s);  cmd.Parameters.AddWithValue("@MaHocKy", MaHK);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt;  } |

## 4. Chức năng của giảng viên

### 4.1. Đăng nhập vào hệ thống:

|  |
| --- |
| CREATE PROCEDURE proc\_GV\_DN --Giảng viên đăng nhập  @TaiKhoan CHAR(10),  @MatKhau CHAR(18)  AS  BEGIN  IF EXISTS (SELECT 1 FROM TAIKHOAN, GIANGVIEN WHERE MaGiangVien = @TaiKhoan AND TaiKhoan = @TaiKhoan AND MatKhau = @MatKhau)  BEGIN  SELECT \*  FROM GIANGVIEN  WHERE MaGiangVien = @TaiKhoan  END  ELSE  BEGIN  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Sai Tài khoản hoặc Mật khẩu'  RAISERROR(@err, 16, 1)  END  END; |

Thực thi trên C#:

|  |
| --- |
| public void GV\_DangNhap(DatabaseConnection db, TextBox U, TextBox P)  {  string username = U.Text;  string password = P.Text;  if (db.OpenConnection())  {  SqlCommand command = new SqlCommand("proc\_GV\_DN", db.connection);  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaGiangVien", username);  command.Parameters.AddWithValue("@MatKhau", password);  SqlDataReader reader = command.ExecuteReader();  try  {  if (reader.Read())  {  string HoTen = reader["HoTen"].ToString();  string MaKhoa = reader["MaKHoa"].ToString();  string SDT = reader["Sdt"].ToString();  string HocHam = reader["HocHam"].ToString();  string DiaChi = reader["DiaChi"].ToString();  GiangVienDTO gv = new GiangVienDTO(HoTen, username, MaKhoa, SDT, HocHam, DiaChi);  ChucNangGiangVien f\_GV = new ChucNangGiangVien();  f\_GV.Update\_Form(gv, "HK1\_22-23");  f\_GV.ShowDialog();  reader.Close();  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.CloseConnection();  }  }  else  {  MessageBox.Show("Lỗi kết nối đến cơ sở dữ liệu.", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

### 4.2. Xem danh sách lớp dạy:

|  |
| --- |
| CREATE VIEW v\_DSLopDay AS  SELECT lh.MaGiangVien, mh.TenMonHoc, lh.MaLop, mh.MaMon, lh.Tiet, lh.Thu, lh.MaPhong Phong,lh.MahocKy  FROM dbo.LOPHOC lh  JOIN dbo.MONHOC mh ON mh.MaMon = lh.MaMon;  GO |

Thực thi trên C#:

|  |
| --- |
| public DataTable load\_DSLopDay(DatabaseConnection db, GiangVienDTO gv, string MaHocKy)  {  db.OpenConnection();  string query = "SELECT \* FROM v\_DSLopDay WHERE MaGiangVien = @MaGiangVien AND MaHocKy = @MaHocKy";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@MaGiangVien", gv.Magiangvien);  cmd.Parameters.AddWithValue("@MaHocKy", MaHocKy);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt;  } |

### 4.3. Xem danh sách học viên của lớp đang dạy:

|  |
| --- |
| CREATE VIEW v\_DSHV AS  SELECT lh.MaLop, hv.MaHocVien, hv.HoTen  FROM dbo.DKMH dk  JOIN dbo.LOPHOC lh ON dk.MaLop = lh.MaLop  JOIN dbo.HOCVIEN hv ON dk.MaHocVien = hv.MaHocVien; |

Thực thi trên C#:

|  |
| --- |
| public DataTable load\_DSHocVien(DatabaseConnection db, string LopHoc)  {  db.OpenConnection();  string query = "SELECT \* FROM v\_DSHV WHERE MaLop = @MaLop";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@MaLop", LopHoc);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt;  } |

### 4.4. Gửi để xuất mở lớp:

|  |
| --- |
| CREATE PROCEDURE proc\_GuiYeuCau  @MaGiangVien CHAR(6),  @MaMon CHAR(10),  @SoHocVienDuocDangKy INT  AS  BEGIN  BEGIN TRY  INSERT INTO YEUCAU (MaGiangVien, MaMon, SoHocVienDuocDangKy)  VALUES (@MaGiangVien, @MaMon, @SoHocVienDuocDangKy)  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi ' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH  END; |

Thực thi trên C#:

|  |
| --- |
| public void GV\_GuiDeXuat(DatabaseConnection db, GiangVienDTO gv, string MaMon, int SoHV)  {  db.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_GuiYeuCau", db.connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaGiangVien", gv.Magiangvien);  cmd.Parameters.AddWithValue("@MaMon", MaMon);  cmd.Parameters.AddWithValue("@SoHocVienDuocDangKy", SoHV);  try  {  cmd.ExecuteNonQuery();    }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);    }  finally  {  db.CloseConnection();  }  } |

# CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN

Ứng với ba nhóm người dùng (người quản lý, giảng viên, học viên) là role riêng

+ Đối với role người quản lý (admin) ta sử dụng server role sysadmin:

* Trigger tạo tài khoản:

|  |
| --- |
| CREATE TRIGGER Trigger\_TaoTaiKhoan  ON TAIKHOAN  INSTEAD OF INSERT  AS  DECLARE @userName NVARCHAR(30), @passWord NVARCHAR(10)  SELECT @userName = nl.TaiKhoan, @passWord = nl.MatKhau  FROM inserted nl  IF EXISTS (SELECT 1 FROM TAIKHOAN WHERE TaiKhoan = @userName)  BEGIN  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi: Tài khoản đã tồn tại'  RAISERROR(@err, 16, 1)  END  ELSE  BEGIN  INSERT INTO TaiKhoan (TaiKhoan, MatKhau) VALUES (@userName, @passWord)  DECLARE @sqlString NVARCHAR(2000)  SET @sqlString = 'CREATE LOGIN [' + @userName + '] WITH PASSWORD=''' + @passWord + ''', DEFAULT\_DATABASE=[DANG\_KY\_MON\_HOC], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF'  EXEC (@sqlString)  SET @sqlString = 'CREATE USER ' + @userName +' FOR LOGIN '+ @userName  EXEC (@sqlString)  IF (LEFT(@userName, 2) = 'HV')  SET @sqlString = 'ALTER ROLE HV ADD MEMBER ' + @userName;  ELSE IF (LEFT(@userName, 2) = 'GV')  SET @sqlString = 'ALTER ROLE GV ADD MEMBER ' + @userName;  ELSE  SET @sqlString = 'ALTER SERVER ROLE sysadmin' + ' ADD MEMBER '+ @userName;  EXEC (@sqlString)  END; |

* Procedure xoá giảng viên:

|  |
| --- |
| CREATE PROCEDURE DeleteGiangVien  @MaGiangVien CHAR(6)  AS  BEGIN  BEGIN TRANSACTION  DECLARE @username VARCHAR(15);  SELECT @username = MaGiangVien FROM GIANGVIEN WHERE MaGiangVien = @MaGiangVien  DECLARE @sql VARCHAR(100)  DECLARE @SessionID INT;  SELECT @SessionID = session\_id  FROM sys.dm\_exec\_sessions  WHERE login\_name = @username;  IF @SessionID IS NOT NULL  BEGIN  SET @sql = 'KILL ' + CONVERT(NVARCHAR(20), @SessionID)  EXEC(@sql)  END    BEGIN TRY  DELETE FROM GIANGVIEN  WHERE MaGiangVien = @MaGiangVien;  DELETE FROM TAIKHOAN  WHERE TaiKhoan = @MaGiangVien  SET @sql = 'DROP USER '+ @username  EXEC (@sql)  SET @sql = 'DROP LOGIN '+ @username  EXEC (@sql)  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi ' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH  END; |

* Procedure xoá học viên:

|  |
| --- |
| CREATE PROCEDURE DeleteHocVien  @MaHocVien CHAR(10)  AS  BEGIN  BEGIN TRANSACTION  DECLARE @username VARCHAR(15);  SELECT @username = MaHocVien FROM HOCVIEN WHERE MaHocVien = @MaHocVien  DECLARE @sql VARCHAR(100)  DECLARE @SessionID INT;  SELECT @SessionID = session\_id  FROM sys.dm\_exec\_sessions  WHERE login\_name = @username;  IF @SessionID IS NOT NULL  BEGIN  SET @sql = 'KILL ' + CONVERT(NVARCHAR(20), @SessionID)  EXEC(@sql)  END    BEGIN TRY  UPDATE LOPHOC  SET LOPHOC.SoHocVienDangKy = LOPHOC.SoHocVienDangKy - 1  WHERE MaLop IN (SELECT MaLop FROM DKMH WHERE MaHocVien = @MaHocVien);  -- Xóa dòng dữ liệu có chứa @MaHocVien trong bảng DKMH  DELETE FROM DKMH  WHERE MaHocVien = @MaHocVien;  -- Xóa học viên từ bảng HOCVIEN  DELETE FROM HOCVIEN  WHERE MaHocVien = @MaHocVien;  ----  DELETE FROM TAIKHOAN  WHERE taikhoan = @MaHocVien  SET @sql = 'DROP USER ' + @username  EXEC (@sql)  SET @sql = 'DROP LOGIN ' + @username  EXEC (@sql)  COMMIT;  END TRY  BEGIN CATCH  -- Nếu có lỗi, hủy giao dịch và hiển thị thông báo lỗi  ROLLBACK;  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Không thể xóa, kiểm tra lại mã học viên'  RAISERROR(@err, 16, 1)  END CATCH;  END; |

+ Đối với role giảng viên:

|  |
| --- |
| CREATE ROLE GV  GRANT SELECT ON GIANGVIEN TO GV  GRANT SELECT ON HOCKY TO GV  GRANT SELECT ON HOCVIEN TO GV  GRANT SELECT ON KHOA TO GV  GRANT SELECT ON LOPHOC TO GV  GRANT SELECT ON LOPHOC TO GV  GRANT SELECT ON DKMH TO GV  GRANT SELECT ON PHONGHOC TO GV  GRANT SELECT, INSERT ON YEUCAU TO GV  GRANT EXECUTE TO GV  GRANT SELECT TO GV;  DENY EXECUTE ON DeleteGiangVien TO GV;  DENY EXECUTE ON InsertGiangVien TO GV;  DENY EXECUTE ON UpdateGiangVien TO GV;  DENY EXECUTE ON DeleteHocVien TO GV;  DENY EXECUTE ON InsertHocVien TO GV;  DENY EXECUTE ON UpdateHocVien TO GV;  DENY EXECUTE ON XoaLopHoc TO GV;  DENY EXECUTE ON ThemLopHoc TO GV;  DENY EXECUTE ON UpdateLopHoc TO GV;  DENY EXECUTE ON proc\_HV\_DN TO GV;  DENY EXECUTE ON proc\_GV\_DN TO GV;  DENY EXECUTE ON proc\_TaoTK TO GV;  DENY EXECUTE ON proc\_DKLopHoc TO GV;  DENY EXECUTE ON proc\_Xoa\_DKLopHoc TO GV;  DENY EXECUTE ON proc\_ChuyenLopHoc TO GV;  DENY EXECUTE ON proc\_TaoTK TO GV; |

+ Đối với role học viên:

|  |
| --- |
| CREATE ROLE HV  GRANT SELECT, INSERT, DELETE, UPDATE, REFERENCES ON DKMH TO HV  GRANT SELECT ON HOCKY TO HV  GRANT SELECT ON HOCVIEN TO HV  GRANT SELECT ON MONHOC TO HV  GRANT SELECT ON LOPHOC TO HV  GRANT SELECT ON KHOA TO HV  GRANT SELECT ON LOPHOC TO HV  GRANT EXECUTE TO HV  GRANT SELECT TO HV;  DENY EXECUTE ON DeleteGiangVien TO HV;  DENY EXECUTE ON InsertGiangVien TO HV;  DENY EXECUTE ON UpdateGiangVien TO HV;  DENY EXECUTE ON DeleteHocVien TO HV;  DENY EXECUTE ON InsertHocVien TO HV;  DENY EXECUTE ON UpdateHocVien TO HV;  DENY EXECUTE ON XoaLopHoc TO HV  DENY EXECUTE ON ThemLopHoc TO HV;  DENY EXECUTE ON UpdateLopHoc TO HV;  DENY EXECUTE ON proc\_HV\_DN TO HV;  DENY EXECUTE ON proc\_GV\_DN TO HV;  DENY EXECUTE ON proc\_GuiYeuCau TO HV;  DENY EXECUTE ON proc\_TaoTK TO HV; |

# CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG

**Môi trường cài đặt và công nghệ sử dụng**

- Hệ cơ sở dữ liệu Microsoft SQL Server Management Studio 19.2.

- Chương trình Winform được viết bằng Microsoft Visual Studio 2022 - trên nền tảng

ASP.NET Web Application (.NET Framework 4.7.2) C#.

Giao diện và chức năng của chương trình:

## Form đăng nhập

A computer screen with a graduation cap on it

Description automatically generated

Kiểm tra đăng nhập admin:

Code C#:

|  |
| --- |
| public bool Admin\_DangNhap(DatabaseConnection db, TextBox TaiKhoan, TextBox MatKhau)  {  string username = TaiKhoan.Text;  string password = MatKhau.Text;  db.OpenConnection();  string query = "SELECT [dbo].[fu\_Check\_DangNhap\_ADMIN](@user, @pass)";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@user", username);  cmd.Parameters.AddWithValue("@pass", password);  double check = 0;  object result = cmd.ExecuteScalar(); // Sử dụng kiểu object để kiểm tra null  if (result != DBNull.Value && result != null)  {  check = (double)result;  }  db.CloseConnection();  if (check == 1)  {  FormLogIn fromAdmin = new FormLogIn();  DatabaseConnection db\_a = new DatabaseConnection(username, password);  Admin nguoiquanly = new Admin(db\_a);  nguoiquanly.ShowDialog();  nguoiquanly = null;  return true;  }  else  {  MessageBox.Show("Lỗi: Không tìm thấy tài khoản mật khẩu", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return false;  }  } |

Code SQL:

|  |
| --- |
| CREATE FUNCTION [dbo].[fu\_Check\_DangNhap\_ADMIN](@user NVARCHAR(50), @pass NVARCHAR(50))  RETURNS FLOAT  AS  BEGIN  DECLARE @result FLOAT;  IF EXISTS (SELECT (1) FROM TAIKHOAN WHERE TaiKhoan = @user  AND MatKhau = @pass  AND @user NOT IN (SELECT MaHocVien FROM HOCVIEN)  AND @user NOT IN (SELECT MaGiangVien FROM GIANGVIEN))  BEGIN  SET @result = 1  END  ELSE SET @result = 0  RETURN @result  END; |

Thực thi trên C# đăng nhập của HOCVIEN

|  |
| --- |
| public bool HocVien\_DangNhap(DatabaseConnection db, TextBox U, TextBox P)  {  string username = U.Text;  string password = P.Text;  if (db.OpenConnection())  {  SqlCommand command = new SqlCommand("proc\_HV\_DN", db.connection);  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@TaiKhoan", username);  command.Parameters.AddWithValue("@MatKhau", password);  try  {  SqlDataReader reader = command.ExecuteReader();  if (reader.Read())  {  string HoTen = reader["HoTen"].ToString();  string MaKhoa = reader["MaKHoa"].ToString();  reader.Close();  HocVienDTO hv = new HocVienDTO(HoTen, username, MaKhoa);  DatabaseConnection db\_H = new DatabaseConnection(username, password);  FormHVDangKyLopHoc f\_sv = new FormHVDangKyLopHoc(db\_H);  f\_sv.Update\_Form(hv);  f\_sv.ShowDialog();  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.CloseConnection();  }  return true;  }  else  {  MessageBox.Show("Lỗi kết nối đến cơ sở dữ liệu.", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return false;  }  } |

Code trong sql

|  |
| --- |
| CREATE PROCEDURE proc\_HV\_DN --Học viên đăng nhập  @TaiKhoan CHAR(10),  @MatKhau CHAR(18)  AS  BEGIN  IF EXISTS (SELECT 1 FROM TAIKHOAN, HOCVIEN WHERE TaiKhoan = @TaiKhoan AND MatKhau = @MatKhau AND MaHocVien = @TaiKhoan )  BEGIN  SELECT HoTen,MaKhoa  FROM HOCVIEN  WHERE MaHocVien = @TaiKHoan  END  ELSE  BEGIN  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Sai Tài khoản hoặc Mật khẩu'  RAISERROR(@err, 16, 1)  END  END; |

Thực thi trên C# của GIANGVIEN

|  |
| --- |
| public bool GV\_DangNhap(DatabaseConnection db, TextBox U, TextBox P)  {  string username = U.Text;  string password = P.Text;  if (db.OpenConnection())  {  SqlCommand command = new SqlCommand("proc\_GV\_DN", db.connection);  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@TaiKhoan", username);  command.Parameters.AddWithValue("@MatKhau", password);  try  {  SqlDataReader reader = command.ExecuteReader();  if (reader.Read())  {  string HoTen = reader["HoTen"].ToString();  string MaKhoa = reader["MaKHoa"].ToString();  string SDT = reader["Sdt"].ToString();  string HocHam = reader["HocHam"].ToString();  string DiaChi = reader["DiaChi"].ToString();  reader.Close();  GiangVienDTO gv = new GiangVienDTO(HoTen, username, MaKhoa, SDT, HocHam, DiaChi);  ChucNangGiangVien f\_GV = new ChucNangGiangVien();  DatabaseConnection db\_G = new DatabaseConnection(username, password);  f\_GV.Update\_Form(gv, "HK1\_22-23", db\_G);  f\_GV.ShowDialog();  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.CloseConnection();  }  return true;  }  else  {  MessageBox.Show("Lỗi kết nối đến cơ sở dữ liệu.", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return false;  }  } |

Code trong sql

|  |
| --- |
| CREATE PROCEDURE proc\_GV\_DN --Giảng viên đăng nhập  @TaiKhoan CHAR(10),  @MatKhau CHAR(18)  AS  BEGIN  IF EXISTS (SELECT 1 FROM TAIKHOAN, GIANGVIEN WHERE MaGiangVien = @TaiKhoan and TaiKhoan = @TaiKhoan and MatKhau = @MatKhau )  BEGIN  SELECT \*  FROM GIANGVIEN  WHERE MaGiangVien = @TaiKhoan  END  ELSE  BEGIN  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Sai Tài khoản hoặc Mật khẩu'  RAISERROR(@err, 16, 1)  END  END; |

## Form đổi mật khẩu

A screenshot of a login form

Description automatically generated

## Form dành cho người quản lý

### Form mở đầu

A screenshot of a computer

Description automatically generated

### Form tạo tài khoản cho người quản lý, giảng viên hoặc học viên

A screenshot of a computer

Description automatically generated

Tạo tài khoản:

Code C#:

|  |
| --- |
| private void bt\_Create\_ACCADMIN\_Click(object sender, EventArgs e)  {  Admin\_TaoTaiKhoan ad = new Admin\_TaoTaiKhoan(db);  ad.ShowDialog();  } |

|  |
| --- |
| private void bt\_Create\_ACC\_Click(object sender, EventArgs e)  {  string u = tb\_TaiKhoan.Text;  string p = tb\_MatKhau.Text;  hva\_DAO.TaoTaiKhoan(db, u, p);  } |

|  |
| --- |
| public void TaoTaiKhoan(DatabaseConnection db, string u, string p)  {  try  {  db.OpenConnection();  string query = "proc\_TaoTK";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@TaiKhoan", u);  cmd.Parameters.AddWithValue("@MatKhau", p);  cmd.ExecuteNonQuery();  MessageBox.Show("Thông báo: Tạo tài khoản thành công.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.CloseConnection();  }  } |

Code SQL:

|  |
| --- |
| CREATE PROCEDURE proc\_TaoTK(@TaiKhoan NVARCHAR(20), @MatKhau NVARCHAR(20))  AS  BEGIN  INSERT INTO TaiKhoan(TaiKhoan, MatKhau) VALUES (@TaiKhoan, @MatKhau)  END;  GO |

### Form quản lý thông tin giảng viên

A screenshot of a computer

Description automatically generated

Code C# gọi danh mục:

|  |
| --- |
| public DataTable load\_DSGiangVien(DatabaseConnection db) // load danh sách lớp học cho học viên đăng ký (khoa nào đăng ký khoa đó)  {  db.OpenConnection();  string query = "SELECT \* FROM fu\_load\_GV()";  SqlCommand cmd = new SqlCommand(query, db.connection);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt;  } |

Code tạo Function trên SQL:

|  |
| --- |
| CREATE FUNCTION [dbo].[func\_getGiangVienByMaGiangVien] (@maGiangVien CHAR(6))  RETURNS TABLE  AS  RETURN  (  SELECT MaGiangVien, HoTen, HocVi, HocHam, Diachi, Sdt, MaKhoa, MatKhau  FROM GIANGVIEN LEFT JOIN TAIKHOAN ON TaiKhoan = MaGiangVien  WHERE MaGiangVien = @maGiangVien  ); |

### Form thêm, cập nhật thông tin giảng viên

A screenshot of a computer

Description automatically generated

### Form quản lý thông tin học viên

A screenshot of a computer

Description automatically generated

Code C# gọi danh mục:

|  |
| --- |
| public DataTable load\_DSHocVien(DatabaseConnection db)//load danh sach lop học cho học viên đk(khoa nào đăng kí khao đó)  {  db.OpenConnection();  string query = "SELECT \* FROM fu\_load\_HocVien()";  SqlCommand cmd = new SqlCommand(query, db.connection);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt;  } |

Code tạo Function trên SQL:

|  |
| --- |
| CREATE FUNCTION fu\_load\_HocVien ( )  RETURNS TABLE  AS RETURN (  SELECT MaHocVien, HoTen, NgaySinh, DiaChi, QueQuan, MaKhoa, MatKhau  FROM HOCVIEN LEFT JOIN TAIKHOAN ON MaHocVien = TaiKhoan  ); |

### Form thêm, cập nhật thông tin học viên

A computer screen shot of a person

Description automatically generated

### Form quản lý thông tin lớp học

A screenshot of a computer

Description automatically generated

Code C# gọi danh mục:

|  |
| --- |
| public DataTable load\_lopHoc(DatabaseConnection db)  {  db.OpenConnection();  string query = "SELECT \* FROM DanhSachLopHoc()";  SqlCommand cmd = new SqlCommand(query, db.connection);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt;  } |

Code tạo Function trên SQL:

|  |
| --- |
| CREATE FUNCTION [dbo].[DanhSachLopHoc] ( )  RETURNS TABLE  AS  RETURN  (  SELECT MaLop, TenLop, MaGiangVien, MaMon, MaHocKy, SoHocVienDangKy, MaPhong, Thu, Tiet  FROM LOPHOC  ); |

### Form thêm, cập nhật thông tin lớp học

A screenshot of a computer

Description automatically generated

## Form chức năng giảng viên

### Form mở đầu dành cho giảng viên

A screenshot of a computer

Description automatically generated

Code C# gọi danh mục:

|  |
| --- |
| public DataTable load\_DSLopDay(DatabaseConnection db, GiangVienDTO gv, string MaHocKy)  {  db.OpenConnection();  string query = "SELECT \* FROM v\_DSLopDay WHERE MaGiangVien = @MaGiangVien AND MaHocKy = @MaHocKy";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@MaGiangVien", gv.Magiangvien);  cmd.Parameters.AddWithValue("@MaHocKy", MaHocKy);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt;  } |

Code tạo view trên SQL:

|  |
| --- |
| CREATE VIEW v\_DSLopDay AS  SELECT lh.MaGiangVien, mh.TenMonHoc, lh.MaLop, mh.MaMon, lh.Tiet, lh.Thu, lh.MaPhong Phong, lh.MaHocKy  FROM dbo.LOPHOC lh  JOIN dbo.MONHOC mh ON mh.MaMon = lh.MaMon; |

### Form xem danh sách học viên trong một lớp

A screenshot of a computer

Description automatically generated

Code C# gọi danh mục:

|  |
| --- |
| public DataTable load\_DSHocVien(DatabaseConnection db, string LopHoc)  {  db.OpenConnection();  string query = "SELECT \* FROM v\_DSHV WHERE MaLop = @MaLop";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@MaLop", LopHoc);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt;  } |

Code tạo view trên SQL:

|  |
| --- |
| CREATE VIEW v\_DSHV AS  SELECT lh.MaLop, hv.MaHocVien, hv.HoTen  FROM dbo.DKMH dk  JOIN dbo.LOPHOC lh ON dk.MaLop = lh.MaLop  JOIN dbo.HOCVIEN hv ON dk.MaHocVien = hv.MaHocVien; |

### Form đề xuất mở thêm lớp

A screenshot of a computer

Description automatically generated

Code C# gọi danh mục:

|  |
| --- |
| public void GV\_GuiDeXuat(DatabaseConnection db, GiangVienDTO gv, string MaMon, int SoHV)  {  db.OpenConnection();  SqlCommand cmd = new SqlCommand("proc\_GuiYeuCau", db.connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaGiangVien", gv.Magiangvien);  cmd.Parameters.AddWithValue("@MaMon", MaMon);  cmd.Parameters.AddWithValue("@SoHocVienDuocDangKy", SoHV);  try  {  cmd.ExecuteNonQuery();  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.CloseConnection();  } |

## Form dành cho học viên

### Form đăng ký môn học

A screenshot of a computer

Description automatically generated

Code C# hiển thị danh sách môn có thể đăng kí

|  |
| --- |
| public DataTable load\_MonDK(DatabaseConnection db, HocVienDTO hv)  {  db.OpenConnection();  string query = "SELECT \* FROM fu\_load\_MonDK(@MaKhoa)";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@MaKhoa", hv.Makhoa);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt;  } |

Code tạo function trên sql

|  |
| --- |
| CREATE FUNCTION fu\_load\_MonDK (@MaKhoa VARCHAR(5)) --Hàm load các môn học cho học viên đăng ký  RETURNS TABLE  AS RETURN (  SELECT MaMon, TenMonHoc, SoTinChi  FROM MonHoc  WHERE MaKhoa = @MaKhoa  ); |

Code C# hiển thị danh sách các lớp đã đăng ký

|  |
| --- |
| public DataTable load\_MonDaDK(DatabaseConnection db, HocVienDTO hv)  {  db.OpenConnection();  string query = "SELECT \* FROM v\_DSDaDangKi WHERE MaHocVien = @MaHocVien";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@MaHocVien", hv.Mahv);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt;  } |

Code tạo view trên sql

|  |
| --- |
| CREATE VIEW v\_DSDaDangKi AS  SELECT lh.MaLop, mh.TenMonHoc, mh.SoTinChi, lh.Tiet, lh.Thu, p.MaPhong Phong,  gv.HoTen AS GiangVien, dk.MaHocVien, lh.MaMon  FROM dbo.MONHOC mh  JOIN dbo.LOPHOC lh ON lh.MaMon = mh.MaMon  JOIN dbo.GIANGVIEN gv ON lh.MaGiangVien = gv.MaGiangVien  JOIN dbo.PHONGHOC p ON lh.MaPhong = p.MaPhong  JOIN dbo.DKMH dk ON dk.MaLop = lh.MaLop; |

### Form xem danh sách lớp học

A screenshot of a computer

Description automatically generated

Code C# hiển thị danh sách các lớp đã đăng ký

|  |
| --- |
| public DataTable load\_DSLop(DatabaseConnection db, string MaMon, string MaHK)  {  db.OpenConnection();  string query = "SELECT \* FROM v\_DSLopHoc WHERE MaMon = @MaMon AND MaHocKy = @MaHocKy";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@MaMon", MaMon);  cmd.Parameters.AddWithValue("@MaHocKy", MaHK);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt; |

Code tạo view trên sql

|  |
| --- |
| CREATE VIEW v\_DSLopHoc AS  SELECT lh.MaLop, mh.TenMonHoc, mh.SoTinChi, lh.Tiet, lh.Thu, p.MaPhong Phong,  gv.HoTen AS 'Giảng viên', mh.MaKhoa, mh.MaMon, lh.SoHocVienDangKy,lh.MaHocKy  FROM dbo.MONHOC mh  JOIN dbo.LOPHOC lh ON lh.MaMon = mh.MaMon  JOIN dbo.GIANGVIEN gv ON lh.MaGiangVien = gv.MaGiangVien  JOIN dbo.PHONGHOC p ON lh.MaPhong = p.MaPhong; |

### Form chi tiết học phí

A screen shot of a computer

Description automatically generated

Code C# hiển thị danh sách học phí từng môn

|  |
| --- |
| public DataTable load\_DSHP(DatabaseConnection db, HocVienDTO hv)  {  db.OpenConnection();  string query = "SELECT \* FROM fu\_load\_DSHocPhi(@MaHocVien, @MaKhoa)";  SqlCommand cmd = new SqlCommand(query, db.connection);  cmd.Parameters.AddWithValue("@MaHocVien", hv.Mahv);  cmd.Parameters.AddWithValue("@MaKhoa", hv.Makhoa);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  db.CloseConnection();  return dt;  } |

Code sql tạo fu\_load\_DSHocPhi

|  |
| --- |
| CREATE FUNCTION fu\_load\_DSHocPhi (@MaHocVien CHAR(10), @MaKhoa VARCHAR(5)) --Hàm danh sách học phí  RETURNS TABLE  AS  RETURN  (  SELECT lh.MaLop, mh.TenMonHoc, lh.Tiet, lh.Thu, p.MaPhong Phong,  gv.HoTen AS GiangVien, dk.MaHocVien, lh.MaMon, mh.SoTinChi,  CASE  WHEN @MaKhoa = 'FHQ' THEN mh.SoTinChi \* 800000  ELSE mh.SoTinChi \* 600000  END AS HocPhi  FROM dbo.MONHOC mh  JOIN dbo.LOPHOC lh ON lh.MaMon = mh.MaMon  JOIN dbo.GIANGVIEN gv ON lh.MaGiangVien = gv.MaGiangVien  JOIN dbo.PHONGHOC p ON lh.MaPhong = p.MaPhong  JOIN dbo.DKMH dk ON dk.MaLop = lh.MaLop  WHERE dk.MaHocVien = @MaHocVien  ); |