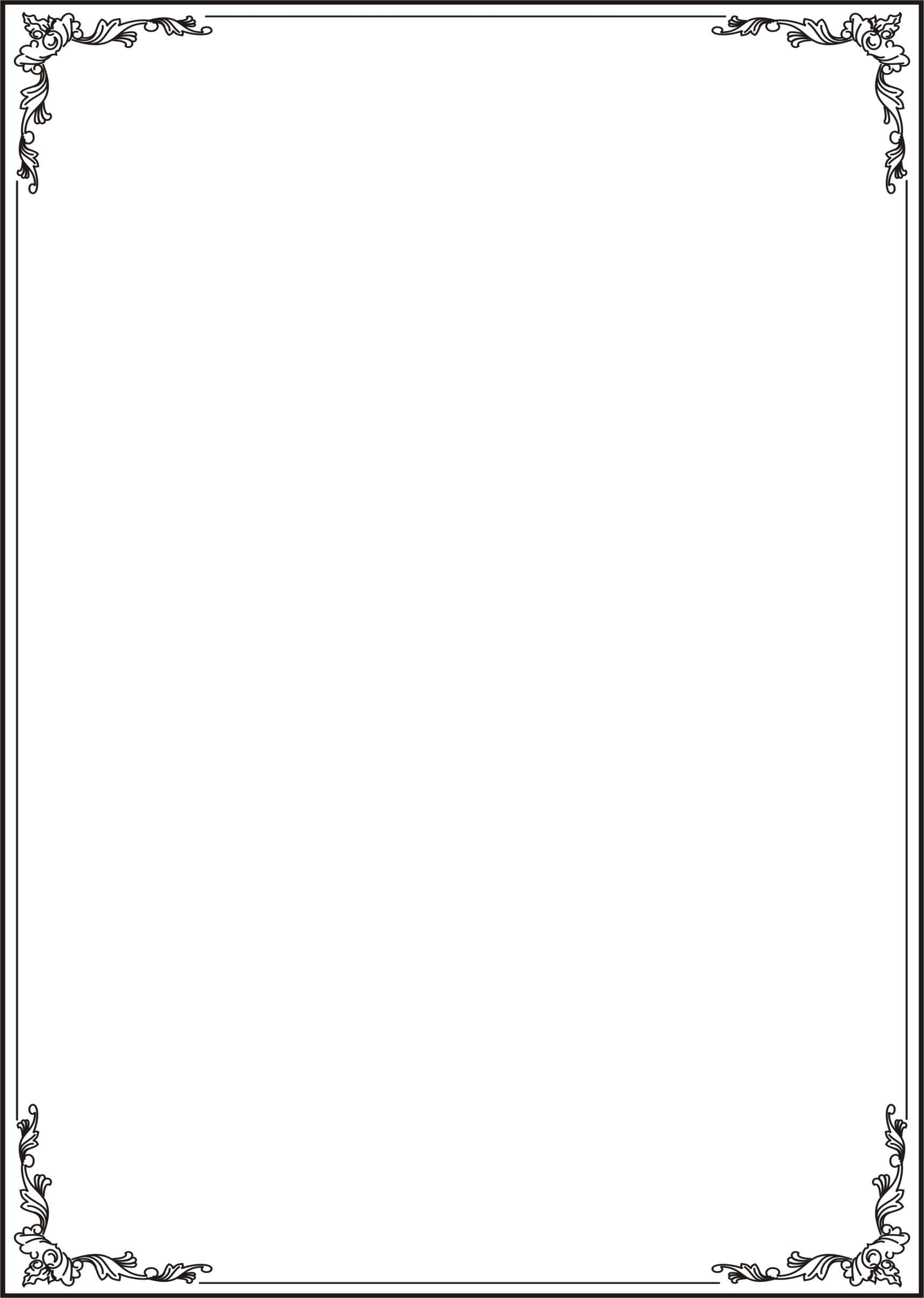
**TRƯỜNG ĐẠI HỌC KINH TẾ - KỸ THUẬT CÔNG NGHIỆP**

**KHOA CÔNG NGHỆ THÔNG TIN**

**\*\*\*\*\*\*\*\*\*\*\*\*\***

****

**Môn Học: Thực Hành Công Nghệ Java**

**BÀI TẬP LỚN**

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ PHÒNG HỌC**

**Nhóm sinh viên thực hiện:**

Phạm Văn Quyết - 22103100057

Đoàn Quốc Trung – 22103100002

Lương Sỹ Anh Tuấn – 22103100041

Đỗ Ngọc Sơn – 22103100020

**Giảng viên hướng dẫn: Cô Trần Thị Huệ**

# **LỜI NÓI ĐẦU**

Trong bối cảnh giáo dục hiện đại, việc quản lý tài nguyên phòng học ngày càng trở nên phức tạp và tốn thời gian khi vẫn dựa nhiều vào thao tác thủ công. Tại các cơ sở đào tạo như trường đại học, cao đẳng, hệ thống lịch giảng dạy, đăng ký phòng thực hành hay phòng thí nghiệm thường xuyên thay đổi theo từng học kỳ, khiến công tác điều phối, phân bổ phòng bị quá tải và dễ dẫn đến chồng chéo, sai sót. Đặc biệt, nhu cầu sử dụng phòng học cho các hoạt động giảng dạy, hội thảo và nhóm nghiên cứu ngày càng đa dạng, đòi hỏi một công cụ hỗ trợ linh hoạt, trực quan và tin cậy.

Xuất phát từ thực tiễn đó, đề tài “Ứng dụng quản lý phòng học bằng Java Swing” được triển khai với mục tiêu xây dựng một phần mềm desktop thân thiện, dễ sử dụng, giúp tự động hóa các quy trình: đăng nhập phân quyền giữa giảng viên và quản trị viên; xem lịch sử sử dụng phòng học theo ngày, tuần; kiểm tra nhanh trạng thái phòng rảnh hoặc đã được đặt; và cung cấp đầy đủ các chức năng CRUD (Thêm, Sửa, Xóa) cho thông tin cơ sở vật chất (cơ sở, dãy nhà, tầng, phòng học). Toàn bộ dữ liệu được lưu trữ và truy vấn trên hệ quản trị MySQL, đảm bảo tính nhất quán, an toàn và dễ dàng sao lưu, mở rộng.

Trong quá trình phát triển, nhóm đã áp dụng các nguyên tắc thiết kế hướng đối tượng, phân tách rõ ràng giữa lớp giao diện (views), lớp xử lý nghiệp vụ (services) và lớp kết nối cơ sở dữ liệu (DAO), đồng thời sử dụng PreparedStatement để phòng ngừa SQL Injection. Giao diện người dùng được xây dựng trên nền Java Swing kết hợp thư viện JCalendar cho việc chọn ngày, đảm bảo tính trực quan và thuận tiện. Ngoài ra, ứng dụng còn tích hợp các thông báo xác nhận và cảnh báo khi lịch đặt trùng, giúp nâng cao trải nghiệm và độ tin cậy.

Kính mong quý thầy cô và các bạn tham khảo, góp ý những hạn chế về giao diện, hiệu năng hoặc mở rộng tính năng như thông báo nhắc nhở qua email, xuất báo cáo thống kê để chúng em ngày càng hoàn thiện và đáp ứng tốt hơn nhu cầu quản lý phòng học trong kỷ nguyên số.

Chúng em xin chân thành cảm ơn!

**MỤC LỤC**

[**LỜI NÓI ĐẦU 2**](#_Toc199662270)

[**ĐỀ BÀI 5**](#_Toc199662271)

[**CHƯƠNG 1 : CƠ SỞ LÝ THUYẾT 12**](#_Toc199662272)

[**1.1.1 NGÔN NGỮ LẬP TRÌNH JAVA 12**](#_Toc199662273)

[**1.1.2 Giới thiệu về Java 12**](#_Toc199662274)

[**1.1.3 Các tính năng của Java 12**](#_Toc199662275)

[**1.1.3 Ưu và nhược điểm của Java 14**](#_Toc199662276)

[**1.1.4 Cách cài đặt một project sử dụng ngôn ngữ Java 14**](#_Toc199662277)

[**1.2 HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU XAMPP 17**](#_Toc199662278)

[**1.2.1 Ưu điểm của xampp 18**](#_Toc199662279)

[**1.2.2 Nhược điểm của xampp 18**](#_Toc199662280)

[**1.3 HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU SQL SERVER 19**](#_Toc199662281)

[**1.3.1 Nhược điểm của SQL Server 21**](#_Toc199662282)

[**CHƯƠNG 2 : PHÂN TÍCH HỆ THỐNG VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU 22**](#_Toc199662283)

[**2.1 MÔ TẢ HỆ THỐNG 22**](#_Toc199662284)

[**2.1.1 Quy trình đăng nhập 22**](#_Toc199662285)

[**2.1.2 Quản lý tài khoản 22**](#_Toc199662286)

[**2.1.3 Quản lý phòng học ( dành cho quản trị viên) 22**](#_Toc199662287)

[**2.1.4 Đăng kí phòng học và lịch học ( dành cho giảng viên ) 22**](#_Toc199662288)

[**2.2 PHÂN TÍCH THIẾT KẾ HỆ THỐNG 22**](#_Toc199662289)

[**2.2.1.Sơ đồ phân rã chức năng 22**](#_Toc199662290)

[**2.2.2 sơ đồ Use case 23**](#_Toc199662291)

[**2.2.2.3 Biểu đồ use case chức năng quản lý cơ sở của admin 25**](#_Toc199662292)

[**2.3 BẢNG CƠ SỞ DỮ LIỆU 30**](#_Toc199662293)

[**2.3.1 Mô tả các thực thể 30**](#_Toc199662294)

[**2.3.2 Các bảng dữ liệu quan hệ 30**](#_Toc199662295)

[**CHƯƠNG 3 : CÀI ĐẶT VÀ THỬ NGHIỆM 35**](#_Toc199662296)

[**3.1 GIAO DIỆN CHÀO MỪNG 35**](#_Toc199662297)

[**3.2 GIAO DIỆN DÀNH CHO ADMIN 36**](#_Toc199662298)

[**3.2.1 Giao diện đăng nhập 36**](#_Toc199662299)

[**3.2.2 Giao diện quản trị 37**](#_Toc199662300)

[**3.2.3 Giao diện tạo tài khoản giảng viên 37**](#_Toc199662301)

[**3.2.4 Giao diện quản lý cơ sở 38**](#_Toc199662302)

[**3.2.5 Giao diện quản lý dãy nhà 38**](#_Toc199662303)

[**3.2.6 Giao diện quản lý tầng 39**](#_Toc199662304)

[**3.3 GIAO DIỆN DÀNH CHO GIẢNG VIÊN 40**](#_Toc199662305)

[**3.3.1 Giao diện đăng nhập 40**](#_Toc199662306)

[**3.3.2 Giao diện menu giảng viên 40**](#_Toc199662307)

[**3.3.3 Giao diện đăng ký phòng học 41**](#_Toc199662308)

[**3.3.4 Giao diện xem và kiểm tra trạng thái phòng học 41**](#_Toc199662309)

[**3.4 KIỂM THỬ HỆ THỐNG 43**](#_Toc199662310)

[**KẾT LUẬN 48**](#_Toc199662311)

[**ĐỊNH HƯỚNG PHÁT TRIỂN 49**](#_Toc199662312)

[**TÀI LIỆU THAM KHẢO 50**](#_Toc199662313)

**ĐỀ BÀI**

**Đề Bài: Xây Dựng Ứng Dụng Quản Lý Phòng Học**

**1.1.Mô tả chung**

Phát triển một ứng dụng desktop sử dụng Java Swing để quản lý phòng học trong một cơ sở giáo dục (ví dụ: trường đại học). Ứng dụng cho phép người dùng (giáo viên và quản trị viên) thực hiện các chức năng như đăng nhập, xem lịch phòng học, kiểm tra trạng thái phòng, và quản lý thông tin phòng học (thêm, sửa, xóa). Ứng dụng sử dụng cơ sở dữ liệu MySQL để lưu trữ thông tin và đảm bảo giao diện thân thiện với người dùng.

**1.2. Yêu cầu chức năng**

**1.2.1. Quản lý tài khoản người dùng**

**Đăng nhập**:

* Người dùng (giáo viên hoặc quản trị viên) đăng nhập bằng email và mật khẩu.
* Ứng dụng kiểm tra thông tin đăng nhập từ bảng nguoidung trong cơ sở dữ liệu.
* Sau khi đăng nhập thành công, chuyển hướng người dùng đến giao diện tương ứng dựa trên vai trò (role):
* role = 0: Quản trị viên (Admin).
* role = 1: Giáo viên (Teacher).

**Tạo tài khoản quản trị viên**:

* Quản trị viên có thể tạo tài khoản mới cho người dùng với các thông tin: mã người dùng (maND), tên người dùng (tenND), email, mật khẩu, và vai trò.
* Email phải được kiểm tra định dạng hợp lệ (sử dụng biểu thức chính quy).

**1.2.2. Xem và kiểm tra thông tin phòng học**

* + **Xem chi tiết phòng học**:
* Người dùng chọn cơ sở (CoSo), dãy nhà (DayNha), tầng (Tang), và phòng học (Phong) thông qua các combo box liên kết (dữ liệu được tải động từ cơ sở dữ liệu).
* Hiển thị lịch sử dụng phòng học trong một bảng (JTable) với các cột: tên phòng, ngày, tiết bắt đầu, tiết kết thúc, và tên giáo viên đăng ký.
* Nếu phòng học chưa được đăng ký, hiển thị thông báo "Phòng học trống".
  + **Kiểm tra trạng thái phòng**:
* Người dùng nhập ngày (thông qua JDateChooser), tiết bắt đầu, và tiết kết thúc để kiểm tra trạng thái phòng (trống hoặc đã đăng ký).
* Kiểm tra tính hợp lệ của tiết học:
* Tiết học nằm trong khoảng 1 đến 16.
* Tiết bắt đầu nhỏ hơn tiết kết thúc.
* Hiển thị thông báo kết quả kiểm tra (phòng trống hoặc đã được đăng ký).
* **Xem lịch phòng trong tuần**:
* Cho phép xem lịch sử dụng phòng học trong một tuần dựa trên phòng được chọn.
* Mở một cửa sổ mới (XemLichPhong\_Form) để hiển thị lịch chi tiết.

**1.2.3. Quản lý cơ sở vật chất**

* + **Quản lý cơ sở (CoSo)**:
* Thêm, sửa, xóa thông tin cơ sở với các thuộc tính: mã cơ sở (maCoSo), tên cơ sở (tenCoSo), địa chỉ (diaChi).
* Lấy danh sách tất cả cơ sở từ cơ sở dữ liệu.
  + **Quản lý dãy nhà (DayNha)**:
* Thêm, sửa, xóa thông tin dãy nhà với các thuộc tính: mã dãy nhà (maDayNha), tên dãy nhà (tenDayNha), mã cơ sở (maCoSo).
* Lấy danh sách dãy nhà theo mã cơ sở.
  + **Quản lý tầng (Tang)**:
* Thêm, sửa, xóa thông tin tầng với các thuộc tính: mã tầng (maTang), tên tầng (tenTang), mã dãy nhà (maDayNha).
* Kiểm tra tầng đã tồn tại trước khi thêm mới.
* Lấy danh sách tầng theo mã dãy nhà.
  + **Quản lý phòng học (Phong)**:
* Thêm, sửa, xóa thông tin phòng học với các thuộc tính: mã phòng (maPhong), tên phòng (tenPhong), mã tầng (maTang).
* Lấy danh sách phòng học theo tên tầng hoặc mã tầng.
* Lấy mã phòng từ tên phòng và ngược lại.

**1.2.4. Quản lý chi tiết phòng học (ChiTietPhongHoc)**

* **Lấy danh sách chi tiết phòng học**:
  + Lấy danh sách chi tiết phòng học theo tên phòng hoặc theo ngày và tên phòng.
  + Dữ liệu bao gồm: mã phòng, mã người dùng, ngày, tiết bắt đầu, tiết kết thúc, tên người dùng, và trạng thái (trống hoặc đã đăng ký).
* **Lấy danh sách tiết trống**:
  + Tạo danh sách chi tiết phòng học bao gồm cả các khoảng tiết trống (từ 1 đến 16) dựa trên danh sách chi tiết phòng học đã đăng ký.
* **Lấy tiết bắt đầu và kết thúc**:
  + Lấy danh sách tiết bắt đầu và tiết kết thúc của một phòng học theo mã phòng và ngày.
* **Lấy chi tiết phòng theo người dùng**:
  + Lấy danh sách chi tiết phòng học đã đăng ký bởi một người dùng (dựa trên maND).
* **Xóa chi tiết phòng học**:
  + Xóa một bản ghi chi tiết phòng học dựa trên mã phòng, ngày, tiết bắt đầu, và tiết kết thúc.

**1.2.5. Giao diện người dùng**

* **Giao diện chính** (XemPhongHoc\_Form):
* Bao gồm các combo box để chọn cơ sở, dãy nhà, tầng, phòng học.
* Bảng hiển thị chi tiết phòng học.
* Nút để xem chi tiết, xem lịch tuần, kiểm tra trạng thái phòng, và quay về trang chủ.
* Panel để nhập thông tin kiểm tra trạng thái phòng (ngày, tiết bắt đầu, tiết kết thúc).
* **Giao diện đăng nhập** (DangNhap\_Form):
* Form nhập email và mật khẩu để đăng nhập.
* **Giao diện menu giáo viên** (MenuGiangVien\_Form):
* Menu chính cho giáo viên sau khi đăng nhập.
* **Giao diện lịch phòng** (XemLichPhong\_Form):
* Hiển thị lịch sử dụng phòng học trong tuần.
* **Tùy chỉnh giao diện**:
* Sử dụng biểu tượng (icon) cho các nút và cửa sổ (ví dụ: logo UNETI)
* Đặt màu nền và kích thước phù hợp cho các thành phần giao diện.
* Sử dụng font chữ thống nhất (ví dụ: Segoe UI, kích thước 14-16).

**1.2.6. Kết nối cơ sở dữ liệu**

* Sử dụng MySQL làm cơ sở dữ liệu với các bảng:
* nguoidung: Lưu thông tin người dùng (maND, tenND, email, password, role).
* coso: Lưu thông tin cơ sở (maCoSo, tenCoSo, diaChi).
* daynha: Lưu thông tin dãy nhà (maDayNha, tenDayNha, maCoSo).
* tang: Lưu thông tin tầng (maTang, tenTang, maDayNha).
* phong: Lưu thông tin phòng học (maPhong, tenPhong, maTang).
* chitietphonghoc: Lưu chi tiết sử dụng phòng (maPhong, maND, ngay, tietHocBatDau, tietHocKetThuc).
* Kết nối cơ sở dữ liệu sử dụng JDBC với thông tin:
* URL: jdbc:mysql://localhost:3306/qlphonghoc?allowPublicKeyRetrieval=true&useSSL=false
* Username: root
* Password: (rỗng)

Xử lý ngoại lệ SQL và đảm bảo đóng kết nối, statement, và result set sau khi sử dụng.

**1.3 Yêu cầu kỹ thuật**

* **Ngôn ngữ lập trình**: Java (sử dụng Java Swing cho giao diện).
* **Cơ sở dữ liệu**: MySQL.
* **Thư viện bên ngoài**:
* com.toedter:jcalendar cho component chọn ngày (JDateChooser).
* JDBC driver cho MySQL (com.mysql.cj.jdbc.Driver).

**Cấu trúc dự án**:

* Gói models: Chứa các lớp mô hình (GiaoVien, CoSo, DayNha, Tang, Phong, ChiTietPhongHoc) để biểu diễn dữ liệu.
* Gói services: Chứa các lớp dịch vụ (AccountService, CoSoService, DayNhaService, TangService, PhongService, ChiTietPhongService, GiaoVienService) để xử lý logic nghiệp vụ và truy vấn cơ sở dữ liệu.
* Gói views: Chứa các lớp giao diện (DangNhap\_Form, XemPhongHoc\_Form, MenuGiangVien\_Form, XemLichPhong\_Form).
* Lớp chính: QuanLyPhongHoc trong gói main để khởi động ứng dụng.
* Gói ConnectDB: Chứa lớp ConnectionDB để quản lý kết nối cơ sở dữ liệu.

**Xử lý lỗi**:

* Hiển thị thông báo lỗi bằng JOptionPane khi có ngoại lệ (ví dụ: thông tin nhập không hợp lệ, lỗi kết nối cơ sở dữ liệu).
* Ghi log lỗi SQL bằng Logger hoặc in ra console.

**Bảo mật**:

* Sử dụng PreparedStatement để ngăn chặn SQL Injection.
* Kiểm tra định dạng email bằng biểu thức chính quy.

**1.4 Yêu cầu giao diện**

* Giao diện phải trực quan, dễ sử dụng.
* Các combo box phải tải dữ liệu động (ví dụ: chọn cơ sở sẽ cập nhật danh sách dãy nhà tương ứng).
* Bảng hiển thị chi tiết phòng học phải có kích thước cột cố định và không cho phép chỉnh sửa trực tiếp.
* Sử dụng các biểu tượng (icon) cho các nút hành động (ví dụ: xem, kiểm tra, trang chủ).
* Đặt tiêu đề cửa sổ và biểu tượng ứng dụng phù hợp.

**1.5 .Đầu ra mong đợi**

* Một ứng dụng Java Swing chạy được, kết nối với cơ sở dữ liệu MySQL.
* Người dùng có thể đăng nhập, xem và kiểm tra trạng thái phòng học, quản lý cơ sở vật chất (nếu là quản trị viên).
* Ứng dụng xử lý lỗi tốt và cung cấp phản hồi rõ ràng qua thông báo.

**Lưu ý**

* Đảm bảo cơ sở dữ liệu qlphonghoc được tạo với cá c bảng và cấu trúc như mô tả.
* Kiểm tra kỹ các truy vấn SQL để đảm bảo tính đúng đắn và hiệu suất.
* Tối ưu hóa việc đóng kết nối cơ sở dữ liệu để tránh rò rỉ tài nguyên.

**1.6 Phân Chia Công Việc**

| **Thành viên** | **Nhiệm vụ chính** | **Chức năng đảm nhiệm** | **Ghi chú** |
| --- | --- | --- | --- |
| **Sơn** | Xử lý giao diện người dùng (UI cơ bản) | - Giao diện đăng nhập- Giao diện menu giáo viên- Giao diện xem phòng học | Công việc tương đối đơn giản, chủ yếu xử lý form và luồng người dùng |
| **Tuấn** | Quản lý cơ sở vật chất | - Quản lý cơ sở- Quản lý dãy nhà- Quản lý tầng- Quản lý phòng học | Sử dụng nhiều form, CRUD với CSDL nhưng không logic quá phức tạp |
| **Trung** | Quản lý tài khoản & phân quyền | - Đăng nhập- Tạo tài khoản quản trị viên- Lấy thông tin tài khoản- Regex kiểm tra email | Phải kết nối CSDL, xử lý bảo mật, kiểm tra đầu vào, tránh SQLi |
| **Quyết** | Quản lý chi tiết phòng học & lịch tuần | - Xem lịch phòng theo tuần- Kiểm tra tiết học trống- Xoá, thêm, sửa tiết học | Công việc logic phức tạp, phải xử lý kiểm tra tiết học, xung đột lịch |

# **CHƯƠNG 1 : CƠ SỞ LÝ THUYẾT**

* + 1. NGÔN NGỮ LẬP TRÌNH JAVA
    2. **Giới thiệu về Java**

Java là ngôn ngữ lập trình đa nền tảng (cross-platform), được phát triển bởi James Gosling tại Sun Microsystems (nay là Oracle Corporation). Ngôn ngữ lập trình này ra đời vào năm 1995 và được thiết kế để có thể chạy trên các nền tảng khác nhau, từ máy tính cá nhân đến thiết bị di động, các máy chủ và thiết bị nhúng.

Java sử dụng cấu trúc lập trình hướng đối tượng (object-oriented programming - OOP) và được xây dựng trên cơ sở của ngôn ngữ lập trình C++. Nó cung cấp một môi trường chạy ảo (virtual machine) gọi là Java Virtual Machine (JVM), giúp các chương trình Java có thể chạy trên nhiều nền tảng khác nhau mà không cần phải biên dịch lại.

Hiện tại, Java là một trong những ngôn ngữ lập trình phổ biến nhất hiện nay, được sử dụng rộng rãi trong các ứng dụng di động, phát triển web, lập trình trên các thiết bị nhúng, máy tính cá nhân, máy chủ, game và nhiều lĩnh vực khác.

* + 1. **Các tính năng của Java**

**Đa nền tảng**

Java được thiết kế để có thể chạy trên nhiều nền tảng khác nhau, vì vậy nó rất phù hợp cho việc phát triển các ứng dụng đa nền tảng. Java sử dụng một máy ảo (JVM - Java Virtual Machine) để chạy mã nguồn, vì vậy mã nguồn được viết một lần và có thể chạy trên nhiều hệ điều hành khác nhau mà không cần thay đổi.

**Quản lý bộ nhớ tự động**

Java có tính năng tự động quản lý bộ nhớ, tức là nó tự động thu dọn các vùng nhớ không sử dụng nữa để giảm thiểu các lỗi bộ nhớ. Điều này giúp cho các ứng dụng được viết bằng Java có thể chạy ổn định và tránh các lỗi liên quan đến bộ nhớ.**Hỗ trợ đa luồng**

Java có thể xử lý đa luồng, cho phép chương trình thực hiện nhiều tác vụ cùng một lúc. Điều này giúp cho các ứng dụng có thể chạy nhanh và hiệu quả hơn, đặc biệt là khi phải xử lý nhiều tác vụ cùng một lúc.

**Tính bảo mật cao**

Java có các tính năng bảo mật như kiểm tra kiểu tĩnh và kiểm tra lỗi trên đường dẫn. Java được thiết kế để giảm thiểu các lỗ hổng bảo mật và các vấn đề liên quan đến an ninh.

**Hỗ trợ các thư viện và công cụ phát triển mạnh mẽ**

Java có các thư viện tiêu chuẩn và các công cụ phát triển như IDE (Integrated Development Environment) để giúp các nhà phát triển tạo ra các ứng dụng dễ dàng hơn. Các thư viện và công cụ này cung cấp các tính năng như: tạo giao diện người dùng, xử lý dữ liệu, kết nối cơ sở dữ liệu và nhiều hơn nữa,...

**Tính di động**

Java được sử dụng rộng rãi trong lĩnh vực di động bao gồm: các thư viện hỗ trợ việc phát triển ứng dụng di động, đóng gói ứng dụng thành các file .jar hoặc .apk, cung cấp các tính năng như xử lý đa nhiệm, kết nối mạng và tích hợp với các thiết bị phần cứng như máy ảnh.

**Tính độc lập với nền tảng**

Java có thể chạy trên nhiều nền tảng khác nhau và không phụ thuộc vào bất kỳ nền tảng cụ thể nào. Điều này giúp cho các ứng dụng Java có thể được triển khai trên nhiều hệ thống khác nhau mà không cần sửa đổi mã nguồn.

**Tính kế thừa và đa hình**

Java là ngôn ngữ lập trình hướng đối tượng (OOP), vì vậy nó có các tính năng như kế thừa, đa hình và đóng gói. Các tính năng này giúp cho mã nguồn được tái sử dụng và giảm thiểu sự trùng lặp trong mã nguồn.

**Tính mở rộng**

Java có tính năng mở rộng, cho phép các nhà phát triển thêm các tính năng mới vào ngôn ngữ bằng cách tạo các thư viện và API riêng. Điều này giúp cho Java có thể được sử dụng trong nhiều lĩnh vực khác nhau và được phát triển theo các hướng khác nhau.

**1.1.3 Ưu và nhược điểm của Java**

**Ưu điểm của Java**

* Độ tin cậy cao
* Tính đa nền tảng
* Quản lý bộ nhớ tự động
* Công cụ phát triển phong phú
* Hỗ trợ đa luồng

**Nhược điểm của Java**

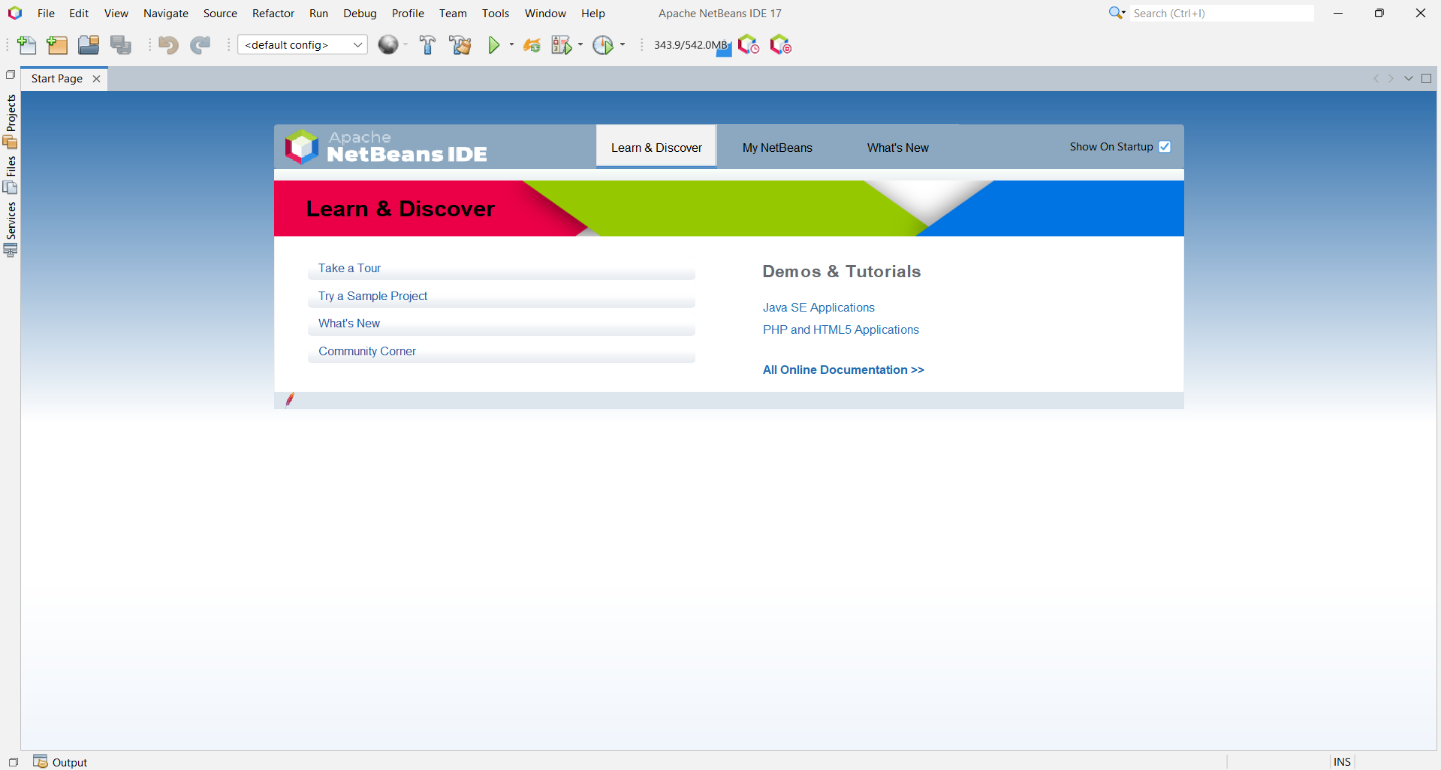
* Tốc độ chậm hơn so với các ngôn ngữ lập trình gần sát với phần cứng, chẳng hạn như C hoặc C++.
* Java có thể chạy trên nhiều nền tảng khác nhau, nhưng ứng dụng này có thể cần đến một trình biên dịch hoặc máy ảo Java riêng biệt để có thể chạy trên các thiết bị di động.
* Sử dụng bộ nhớ lớn hơn so với một số ngôn ngữ lập trình khác.
* Cú pháp phức tạp hơn so với một số ngôn ngữ lập trình khác
  + 1. **Cách cài đặt một project sử dụng ngôn ngữ Java**



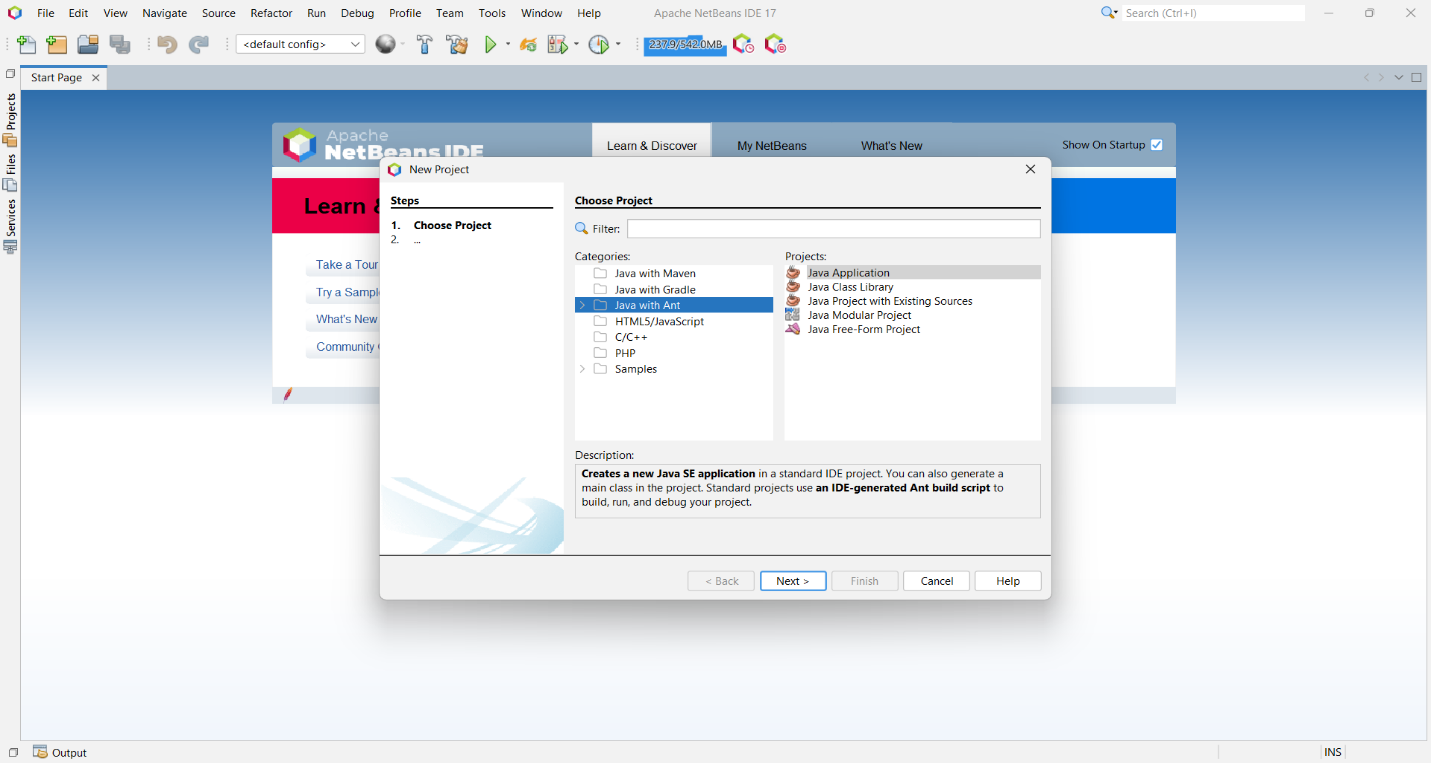
Để tạo một project sử dụng ngôn ngũ java cần đáp ứng các yêu cầu sau :

* Hệ điều hành Windows và Apache NetBeans IDE 17 trở lên
* Đã được cài đặt JDK 17 trở lên

**Bước 1 :** Mở Apache Netbeans lên và nhận được giao diện như sau

Hình 1.1 Giao diện khởi động Apache Netbeans

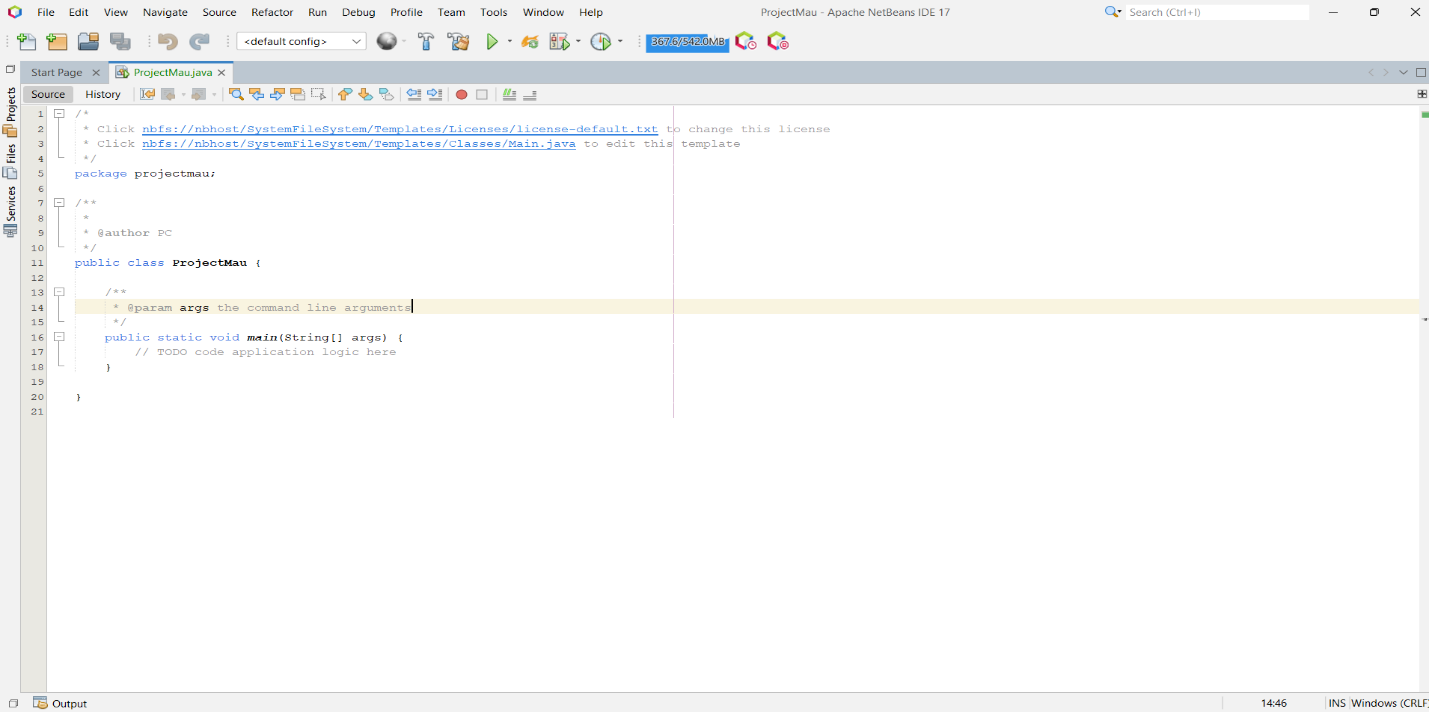
**Bước 2 :** chọn  **File -> New project -> Java with Ant -> Java Application**



Hình 1.2 : Tạo project Java

**Bước 3 :** Tại đây, chúng ta có thể đặt tên cho project và vị trí lưu

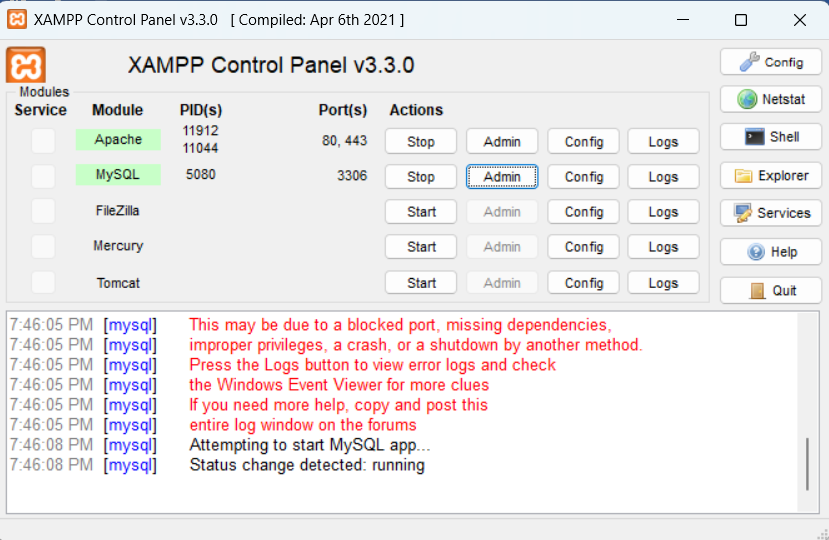
Hình 1.3 Đặt tên dự án

**Bước 4 :** Nhấp vào **Finish** ra màn hình như sau

Hình 1.4 Hoàn thành Project Java

* 1. **HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU XAMPP**

**XAMPP** là một phần mềm cho phép **giả lập** môi trường server hosting ngay trên máy tính của bạn, cho phép bạn **chạy demo website** mà **không**cần phải mua**hosting**hay[**VPS**](https://www.thegioididong.com/hoi-dap/vps-la-gi-dung-de-lam-gi-co-uu-nhuoc-diem-nhu-the-nao-1337214). Chính vì vậy, XAMPP hay được phục vụ cho hoạt động học tập giảng dạy thực hành và phát triển web.

**XAMPP** được viết tắt của **X**+**Apache**+**MySQL**+**PHP**+**Perl** vì nó được tích hợp sẵn [Apache](https://vi.wikipedia.org/wiki/Apache_(HTTP)), [MySQL](https://www.thegioididong.com/game-app/huong-dan-cach-tai-cai-dat-mysql-ban-moi-nhat-chi-tiet-tung-1299084), [PHP](https://vi.wikipedia.org/wiki/PHP), FTP server, Mail Server. Còn X thể hiện cho sự đa nền tảng của XAMPP vì nó có thể dùng được cho 4 hệ điều hành khác nhau: [Windows](https://www.thegioididong.com/phan-mem/windows-10-home-32-bit-64-bit-all-languages-kw9-0), [MacOS](https://www.thegioididong.com/hoi-dap/he-dieu-hanh-macos-la-gi-956642), [Linux](https://www.thegioididong.com/hoi-dap/he-dieu-hanh-linux-la-gi-uu-nhuoc-diem-cua-he-dieu-hanh-1312530) và [Solaris](https://vi.wikipedia.org/wiki/Solaris_(h%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh)" \t "_blank).Hình 1.5 : Giao diện Xampp

### **1.2.1 Ưu điểm của xampp**

Có thể chạy được linh hoạt được trên mọi hệ điều hành: [Cross-platform](https://vi.wikipedia.org/wiki/%C4%90a_n%E1%BB%81n_t%E1%BA%A3ng), Windows, MacOS, Linux.

- Có cấu hình đơn giản nhưng mang đến nhiều chức năng hữu ích như: Lập [Server](https://www.thegioididong.com/hoi-dap/may-chu-server-la-gi-co-may-loai-co-vai-tro-nhu-the-nao-1335540) giả định, lập Mail Server giả định và hỗ trợ SSL trên [localhost](https://vi.wikipedia.org/wiki/Localhost).

- Tích hợp được nhiều tính năng với các thành phần quan trọng như: Apache, PHP, MySql. Vì thế, người dùng không cần cài đặt từng phần trên riêng lẻ mà chỉ cần cài XAMPP là có 1 web server hoàn chỉnh.

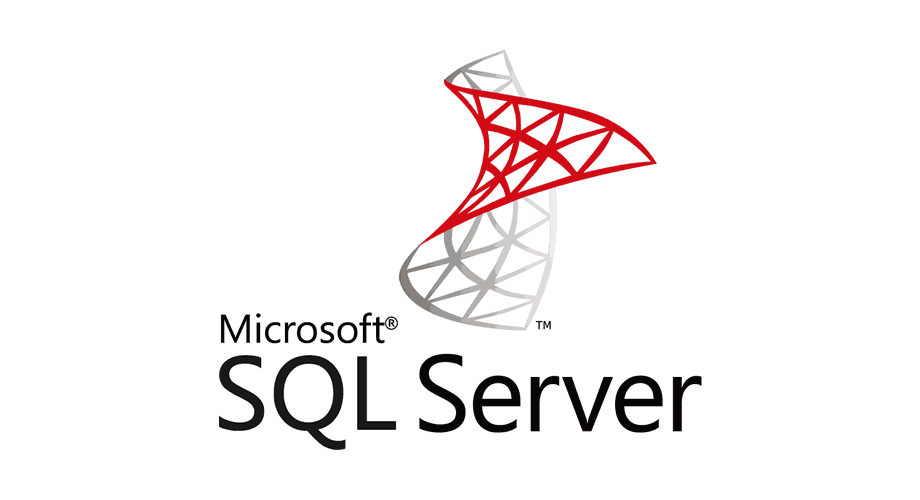
 -Tạo mã nguồn mở: Giao diện quản lý dễ dàng và tiện lợi giúp người dùng luôn chủ động được trong chế độ khởi động lại hay bật/ tắt đối với máy chủ theo thời gian phù hợp nhất.

**1.2.2 Nhược điểm của xampp**

- Do cấu hình khá đơn giản cho nên Xampp hoàn toàn không nhận được sự hỗ trợ về cấu hình [Module](https://www.thegioididong.com/game-app/module-la-gi-dinh-nghia-cua-module-trong-cac-linh-vuc-1336435), đồng thời cũng không có cả MySQL.

- Dung lượng tương đối nặng (141Mb).

- Không có nhiều phiên bản cho từng thành phần của server như PHP, Apache mà phải cài đặt riêng.

**1.3 HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU SQL SERVER**

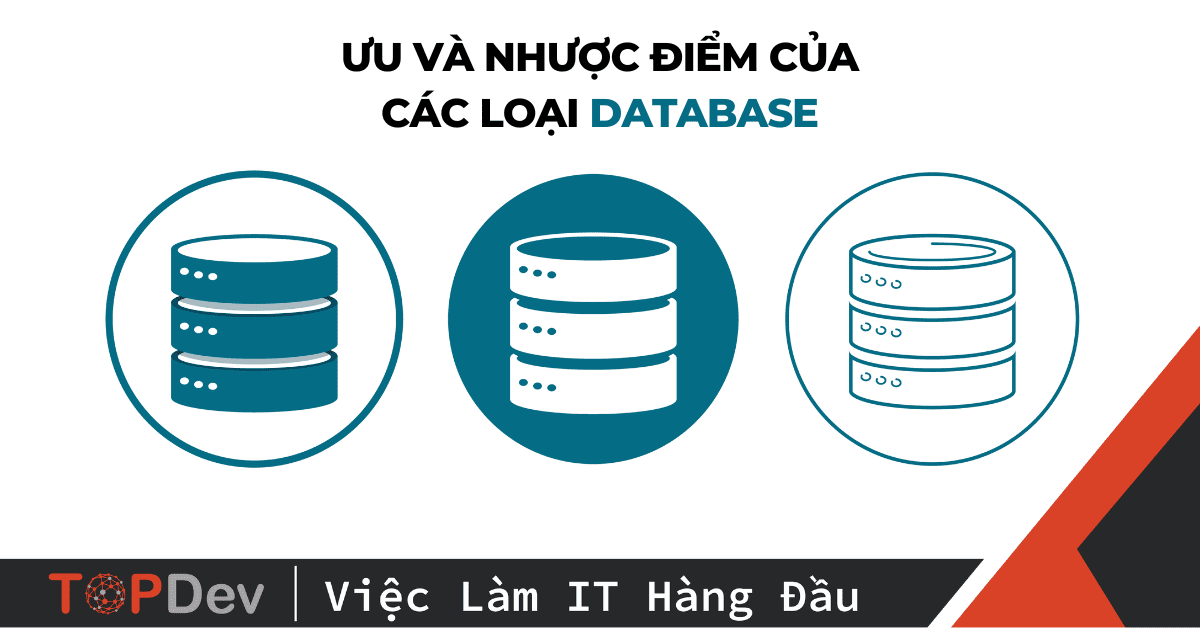
SQL Server hay Microsoft SQL Server là một hệ thống quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) được phát triển bởi Microsoft vào năm 1988. Nó được sử dụng để tạo, duy trì, quản lý và triển khai hệ thống RDBMS.

Được thiết kế để quản lý và lưu trữ dữ liệu, SQL Server cho phép người dùng truy vấn, thao tác và quản lý dữ liệu một cách hiệu quả và an toàn. SQL Server là một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất trên thế giới và được sử dụng rộng rãi trong các doanh nghiệp.

Phần mềm SQL Server được sử dụng khá rộng rãi vì nó được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn lên đến Tera – Byte cùng lúc phục vụ cho hàng ngàn user. Bên cạnh đó, ứng dụng này cung cấp đa dạng kiểu lập trình SQL từ ANSI SQL (SQL truyền thống) đến [SQL](https://topdev.vn/blog/sql-la-gi/) và cả T-SQL (Transaction-SQL) được sử dụng cho cơ sở dữ liệu quan hệ nâng cao.

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) do Microsoft phát triển. Nó được thiết kế để lưu trữ và quản lý dữ liệu một cách hiệu quả, hỗ trợ việc truy vấn, cập nhật, và bảo mật dữ liệu trong các ứng dụng từ nhỏ đến lớn. SQL Server sử dụng ngôn ngữ T-SQL (Transact-SQL) – một phần mở rộng của chuẩn SQL – để thực hiện các thao tác với dữ liệu.

SQL Server là một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất hiện nay và thường được sử dụng trong các hệ thống quản lý doanh nghiệp, phần mềm thương mại điện tử, hệ thống học trực tuyến và nhiều lĩnh vực khác. SQL Server có nhiều phiên bản như: Express, Standard, Enterprise,… phù hợp với quy mô và nhu cầu của từng loại dự án.Ưu điểm của SQL Server

* ***Hiệu suất cao:*** SQL Server được tối ưu hóa để xử lý các truy vấn lớn với tốc độ nhanh, đặc biệt là trong các hệ thống cần xử lý dữ liệu nhiều và phức tạp.
* ***Tính bảo mật cao:*** Hệ thống bảo mật tích hợp mạnh mẽ với các tính năng như mã hóa dữ liệu, xác thực bằng Active Directory, phân quyền chi tiết giúp kiểm soát quyền truy cập.
* ***Dễ sử dụng và quản lý:*** SQL Server Management Studio (SSMS) cung cấp giao diện đồ họa trực quan, dễ thao tác ngay cả với người mới bắt đầu.
* ***Khả năng tích hợp cao:*** Tích hợp tốt với các công nghệ khác của Microsoft như .NET, Azure, Power BI và Visual Studio.
* ***Sao lưu và khôi phục dữ liệu mạnh mẽ:*** Hỗ trợ các cơ chế sao lưu định kỳ, toàn bộ hoặc từng phần, đảm bảo an toàn dữ liệu khi xảy ra sự cố.
* ***Hỗ trợ lưu trữ lớn và nhiều người dùng:*** Phù hợp với các hệ thống lớn cần nhiều kết nối đồng thời và dữ liệu lớn.
  + 1. Nhược điểm của SQL Server
* *Chi phí bản quyền cao:* Các phiên bản như Standard và Enterprise yêu cầu phí bản quyền, có thể gây khó khăn cho các doanh nghiệp nhỏ hoặc sinh viên.
* *Yêu cầu cấu hình cao:* Để vận hành ổn định với hiệu suất cao, SQL Server đòi hỏi phần cứng và cấu hình hệ thống mạnh mẽ.
* *Hạn chế về nền tảng:* Chủ yếu tối ưu cho hệ điều hành Windows, mặc dù các phiên bản gần đây đã hỗ trợ Linux nhưng vẫn còn một số giới hạn.
* *Tài nguyên tiêu tốn nhiều:* Khi triển khai hệ thống lớn, SQL Server có thể tiêu tốn nhiều tài nguyên CPU, RAM nếu không được tối ưu truy vấn và cấu hình đúng cách.

***Kết luận:*** SQL Server là lựa chọn phù hợp cho các ứng dụng đòi hỏi độ ổn định, bảo mật cao và khả năng xử lý dữ liệu mạnh mẽ. Tuy nhiên, người dùng cần cân nhắc giữa hiệu suất và chi phí khi quyết định sử dụng, đặc biệt trong các dự án có ngân sách hạn chế hoặc yêu cầu triển khai trên nền tảng đa dạng.

# **CHƯƠNG 2 : PHÂN TÍCH HỆ THỐNG VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU**

## 2.1 MÔ TẢ HỆ THỐNG

### **2.1.1 Quy trình đăng nhập**

Khi người dùng muốn sử dụng hệ thống, người dùng sẽ phải truy cập vào trang web của hệ thống và tiến hành đăng nhập. Người dùng đăng nhập vào hệ thống bằng email cùng mật khẩu đã đăng ký bởi admin. Hệ thống sẽ tự động phân quyền và hiển thị giao diện riêng biệt, đảm bảo mỗi vai trò có trải nghiệm và các chức năng được tối ưu hóa phù hợp với nhu cầu của người dùng.

### **2.1.2 Quản lý tài khoản**

Chức năng Quản lý tài khoản cho phép Admin tạo, chỉnh sửa, xóa tài khoản người dùng. Admin có thể chỉnh sửa thông tin của người dùng. Nếu cần, Admin có thể xóa tài khoản khi cần thiết. Hệ thống đảm bảo mọi thay đổi được cập nhật trong cơ sở dữ liệu và thông báo kết quả cho Admin

### **2.1.3 Quản lý phòng học ( dành cho quản trị viên)**

Sau khi đăng nhập , quản trị viên sẽ được chuyển đến giao diện Menu quản trị . Tại đây , quản trị viên có thể chỉnh sửa ( thêm , sửa , xóa ) dữ liệu về cơ sở , dãy nhà , tầng , phòng học .

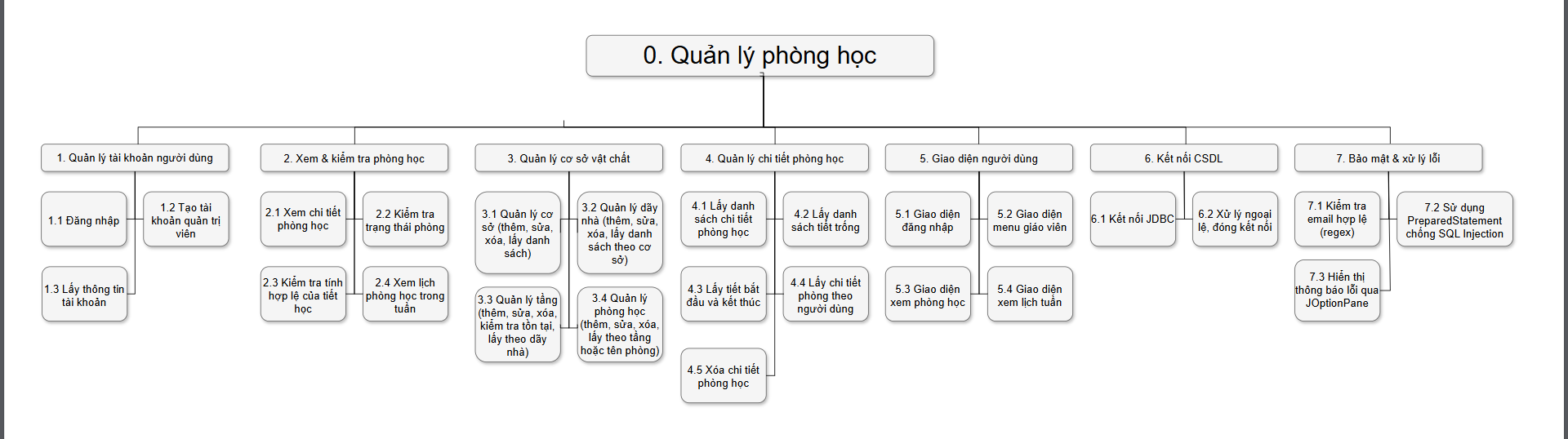
### **2.1.4 Đăng kí phòng học và lịch học ( dành cho giảng viên )**

Sau khi đăng nhập , giảng viên sẽ được chuyển đến giao diện Menu giảng viên . Tại đây , giảng viên có thể đăng kí phòng học bằng cách nhấp vào mục “ Đăng kí phòng học “ và điền các thông tin như cơ sở , dãy nhà , tầng , phòng , ngày , tiết bắt đầu , tiết kết thúc .

Giảng viên có thể tra cứu các phòng học đã được đăng ký bằng cách nhấp vào mục “ Xem chi tiết phòng học “ sau đó chọn cơ sở , dãy nhà , tầng , phòng để kiểm tra xem phòng học đã được đăng kí vào ngày nào ,tiết nào bằng cách chọn “ Xem chi tiết” hoặc đã được đăng ký trong tuần khi chọn “ Xem lịch phòng trong tuần “.

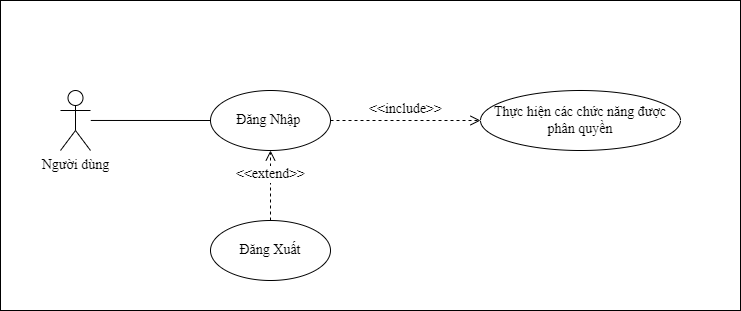
## 2.2 PHÂN TÍCH THIẾT KẾ HỆ THỐNG

**2.2.1.Sơ đồ phân rã chức năng**



Hình 2.1 sơ đồ phân rã chức năng

**2.2.2 sơ đồ Use case**

2.2.2.1 . Biểu đồ chức năng đăng nhập

Hình 2.2 Biểu đồ use case đăng nhập

**Đặc tả use case:**

|  |  |
| --- | --- |
| **Hoạt động của tác nhân** | **Phản ứng của hệ thống** |
| 1: Chọn đăng nhập |  |
|  | 2: Hiển thị giao diện đăng nhập |
| 3: Nhập thông tin đăng nhập |  |
| 4: Gửi thông tin đến hệ thống |  |
|  | 5: Kiểm tra thông tin và xác nhận |
|  | 6: Đúng thì đưa người dùng về giao diện được phân quyền |
|  | 7: Sai thì yêu cầu nhập lại thông tin |

**Tên use case:** Đăng Nhập

**Tác nhân:** Người dùng

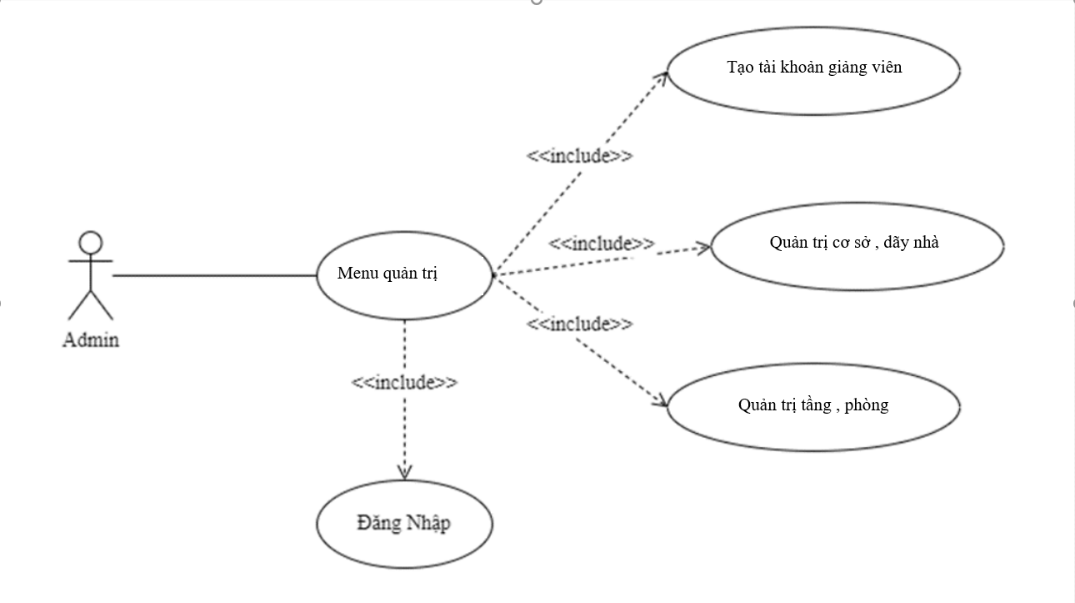
**Chức năng của use case:** Đăng nhập để sử dụng các chức năng được hệ thống phân quyền

**Dòng sự kiện chính:**

* Người dùng nhập thông tin đăng nhập để đăng nhập vào hệ thống
* Sau khi đăng nhập thành công có thể sử dụng các chức năng được hệ thống phân quyền

**Dòng sự kiện thay thế:** Nếu người dùng nhập sai hoặc thiếu thông tin thì sẽ được yêu cầu nhập lại

2.2.2.2 Biểu đồ use case chức năng của Admin



Hình 2.3 Biểu đồ case chức năng của Admin

**Đặc tả use case:**

|  |  |
| --- | --- |
| **Hoạt động của tác nhân** | **Phản ứng của hệ thống** |
| 1: Đăng nhập vào tài khoản admin |  |
|  | 2: Hiển thị giao diện Menu quản trị |
| 3: Thực hiện các chức năng của admin |  |
|  | 4: Hiển thị giao diện tương ứng với chức năng admin chọn |

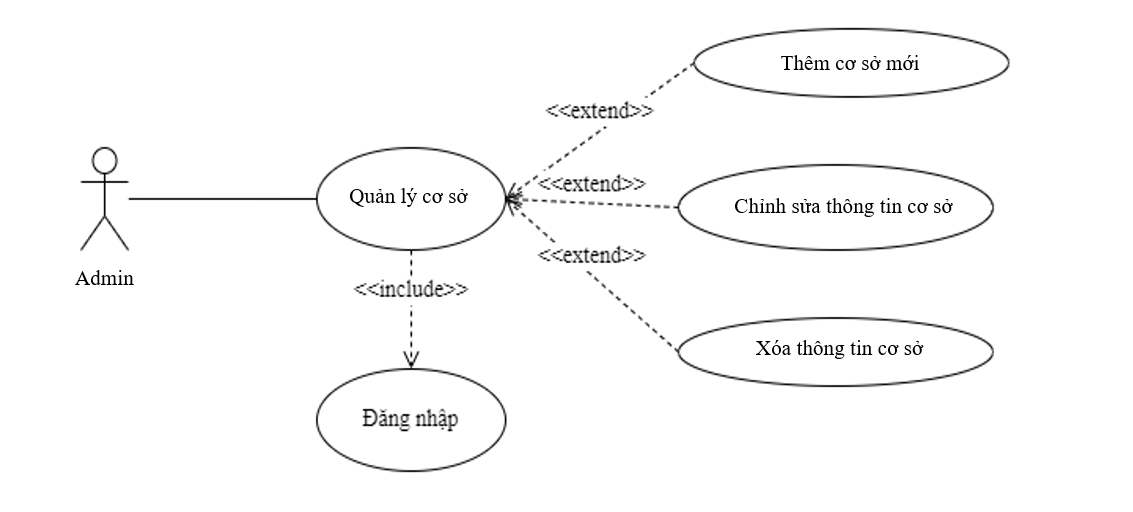
**Tác nhân:** Admin

**Chức năng của use case:** Admin có thể tạo tài khoản mới cho giảng viên , quản lý, thêm, sửa, xóa dữ liệu cơ sở , dãy nhà , tầng , phòng học .

**Dòng sự kiện chính:**

* Người dùng đăng nhập vào hệ thống với phân quyền là admin
* Admin có thể thực hiện các thêm tài khoản của giảng viên , thêm , sửa, xóa dữ liệu cơ sở , dãy nhà , tầng , phòng học.

### 2.2.2.3 Biểu đồ use case chức năng quản lý cơ sở của admin



Hình 2.4 : Biểu đồ use case chức năng quản lý cơ sở

**Đặc tả use case:**

|  |  |
| --- | --- |
| **Hoạt động của tác nhân** | **Phản ứng của hệ thống** |
| 1: Đăng nhập vào tài khoản admin |  |
|  | 2: Hiển thị giao diện Menu quản trị |
| 3: Chọn chức năng quản trị cơ sở |  |
|  | 4: Hiển thị giao diện tương ứng với chức năng admin chọn |
| 5: Chọn chức năng mong muốn (Thêm , sửa , xóa ) |  |
|  | 6: Thực hiện chức năng tương ứng |

**Tác nhân:** Admin

**Chức năng của use case:** Admin có thể thêm, sửa, xóa dữ liệu cơ sở

**Dòng sự kiện chính:**

* Người dùng đăng nhập vào hệ thống với phân quyền là admin
* Admin có thể thực hiện thêm , sửa , xóa dữ liệu cơ sở .

2.2.2.4 Biểu đồ use case chức năng quản lý dãy nhà của admin



Hình 2.5 : Biểu đồ use case chức năng quản lý dãy nhà của admin

**Đặc tả use case:**

|  |  |
| --- | --- |
| **Hoạt động của tác nhân** | **Phản ứng của hệ thống** |
| 1: Đăng nhập vào tài khoản admin |  |
|  | 2: Hiển thị giao diện Menu quản trị |
| 3: Chọn chức năng quản trị dãy nhà |  |
|  | 4: Hiển thị giao diện tương ứng với chức năng admin chọn |
| 5: Chọn chức năng mong muốn (Thêm , sửa , xóa ) |  |
|  | 6: Thực hiện chức năng tương ứng |

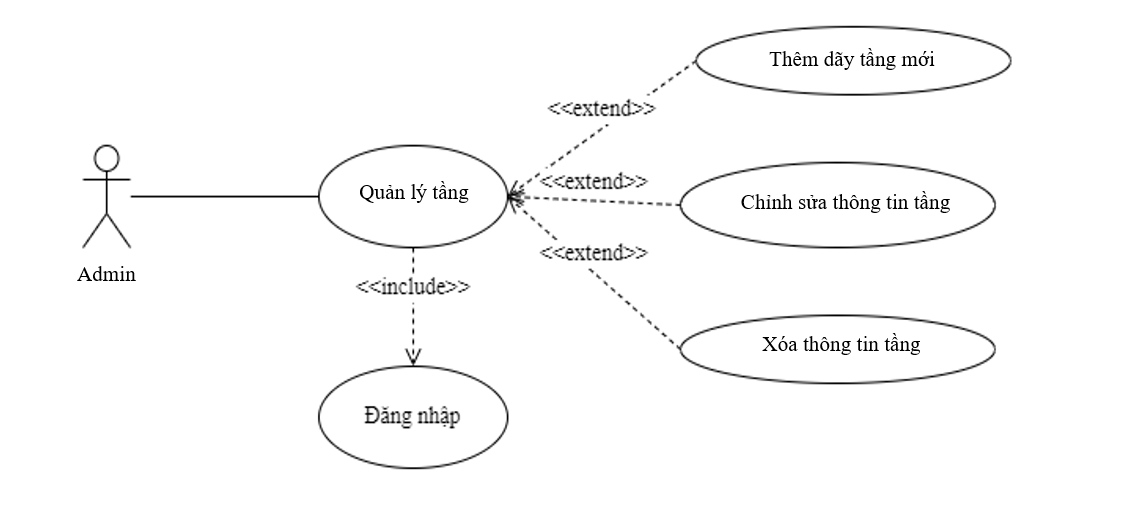
**Tác nhân:** Admin

**Chức năng của use case:** Admin có thể thêm, sửa, xóa dữ liệu dãy nhà

**Dòng sự kiện chính:**

* Người dùng đăng nhập vào hệ thống với phân quyền là admin
* Admin có thể thực hiện thêm , sửa , xóa dữ liệu dãy nhà .

2.2.2.5 Biểu đồ use case chức năng quản lý tầng của admin



Hình 2.6 : Biểu đồ use case chức năng quản lý tầng của admin

**Đặc tả use case:**

|  |  |
| --- | --- |
| **Hoạt động của tác nhân** | **Phản ứng của hệ thống** |
| 1: Đăng nhập vào tài khoản admin |  |
|  | 2: Hiển thị giao diện Menu quản trị |
| 3: Chọn chức năng quản trị tầng |  |
|  | 4: Hiển thị giao diện tương ứng với chức năng admin chọn |
| 5: Chọn chức năng mong muốn (Thêm , sửa , xóa ) |  |
|  | 6: Thực hiện chức năng tương ứng |

**Tác nhân:** Admin

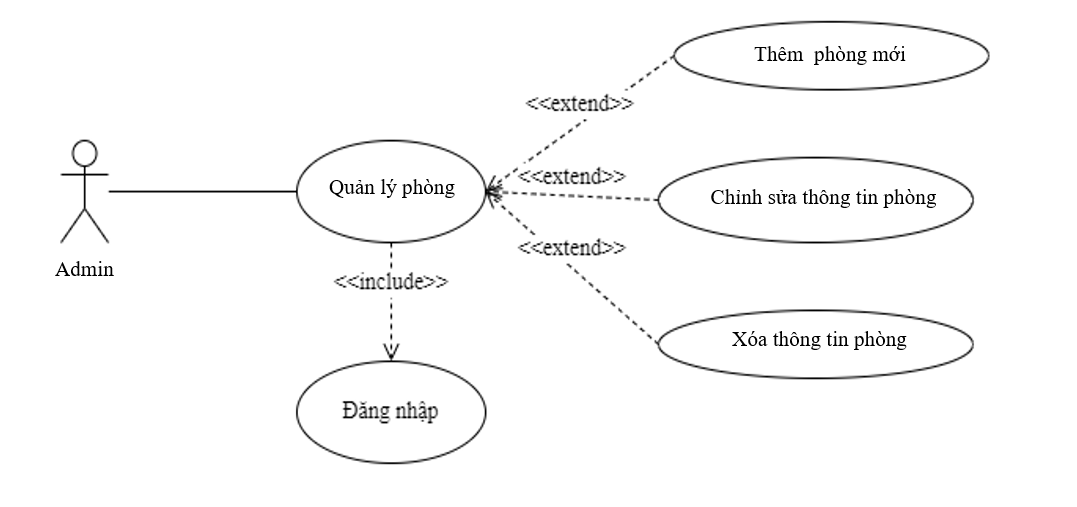
**Chức năng của use case:** Admin có thể thêm, sửa, xóa dữ liệu tầng

**Dòng sự kiện chính:**

Người dùng đăng nhập vào hệ thống với phân quyền là admin

Admin có thể thực hiện thêm , sửa , xóa dữ liệu tầng.

2.2.2.6 Biểu đồ use case chức năng quản lý phòng của admin



Hình 2.7 : Biểu đồ use case chức năng quản lý phòng của admin

**Đặc tả use case:**

|  |  |
| --- | --- |
| **Hoạt động của tác nhân** | **Phản ứng của hệ thống** |
| 1: Đăng nhập vào tài khoản admin |  |
|  | 2: Hiển thị giao diện Menu quản trị |
| 3: Chọn chức năng quản trị phòng |  |
|  | 4: Hiển thị giao diện tương ứng với chức năng admin chọn |
| 5: Chọn chức năng mong muốn (Thêm , sửa , xóa ) |  |
|  | 6: Thực hiện chức năng tương ứng |

**Tác nhân:** Admin

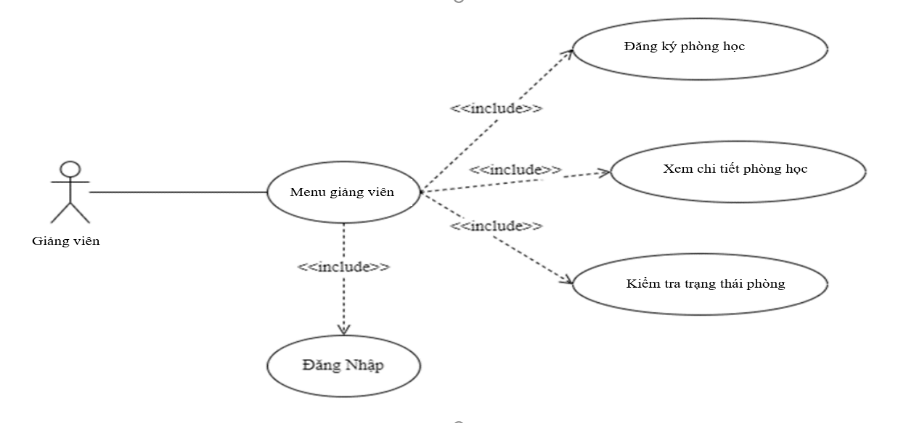
**Chức năng của use case:** Admin có thể thêm, sửa, xóa dữ liệu phòng

**Dòng sự kiện chính:**

* Người dùng đăng nhập vào hệ thống với phân quyền là admin

Admin có thể thực hiện thêm , sửa , xóa dữ liệu phòng.

2.2.2.7 Biểu đồ use case chức năng của Giảng viên



Hình 2.8 : Biểu đồ case chức năng Giảng viên .

**Đặc tả use case:**

|  |  |
| --- | --- |
| **Hoạt động của tác nhân** | **Phản ứng của hệ thống** |
| 1: Đăng nhập vào tài khoản Giảng viên |  |
|  | 2: Hiển thị giao diện Menu giảng viên |
| 3: Thực hiện các chức năng của giảng viên |  |
|  | 4: Hiển thị giao diện tương ứng với chức năng giảng viên chọn |

**Tác nhân:** Giảng viên

**Chức năng của use case:** Giảng viên có thể đăng kí phòng học hoặc xem chi tiết thông tin phòng học như lịch theo tuần , các tiết học đã đăng kí của 1 phòng .

**Dòng sự kiện chính:**

* Người dùng đăng nhập vào hệ thống với phân quyền là Giảng viên
* Giảng viên có thể đăng ký hoặc xem thông tin phòng học .

**2.2.3 . Sơ đồ lớp**

Hình 2.9 Sơ đồ lớp của hệ thống

## **2.3 BẢNG CƠ SỞ DỮ LIỆU**

### **2.3.1 Mô tả các thực thể**

**NguoiDung**: Lưu thông tin người dùng như tên, email, mật khẩu và vai trò (quản trị viên hoặc giáo viên).

**QuanTriVien**: Thực thể kế thừa từ NguoiDung, có chức năng tạo và xoá người dùng.

**CoSo**: Lưu thông tin các cơ sở (chi nhánh) của trường học.

**DayNha**: Lưu thông tin các dãy nhà trong từng cơ sở.

**Tang**: Lưu thông tin tầng trong từng dãy nhà.

**PhongHoc**: Lưu thông tin phòng học thuộc từng tầng.

**ChiTietPhongHoc**: Lưu thông tin sử dụng phòng học (lịch, tiết học, người đăng ký).

### **2.3.2 Các bảng dữ liệu quan hệ**

Từ mô hình quan hệ của hệ thống chuyển thành bảng dữ liệu quan hệ như sau :

* **NguoiDung** (maND, tenND, email, matKhau)
* **QuanTriVien** (maND)
* **CoSo** (maCoSo, tenCoSo, diaChi)
* **DayNha** (maDayNha, tenDayNha, maCoSo)
* **Tang** (maTang, tenTang, maDayNha)
* **PhongHoc** (maPhong, tenPhong, maTang)
* **ChiTietPhongHoc** (maPhong, maND, ngay, tietBatDau, tietKetThuc, trangThai)

Bảng: NguoiDung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Khóa | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | PK | maND | Nvarchar(20) | Mã người dùng |
| 2 |  | tenND | Nvarchar(100) | Tên người dùng |
| 3 |  | email | Nvarchar(100) | Email đăng nhập |
| 4 |  | matKhau | Nvarchar(100) | Mật khẩu |

Bảng: QuanTriVien

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Khóa | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | PK | maND | Nvarchar(20) | Khóa chính tham chiếu từ bảng NguoiDung |

Bảng: CoSo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Khóa | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | PK | maCoSo | Nvarchar(20) | Mã cơ sở |
| 2 |  | tenCoSo | Nvarchar(100) | Tên cơ sở |
| 3 |  | diaChi | Nvarchar(255) | Địa chỉ cơ sở |

Bảng: DayNha

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Khóa | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | PK | maDayNha | Nvarchar(20) | Mã dãy nhà |
| 2 |  | tenDayNha | Nvarchar(100) | Tên dãy nhà |
| 3 | FK | maCoSo | Nvarchar(20) | Mã cơ sở liên kết |

Bảng: Tang

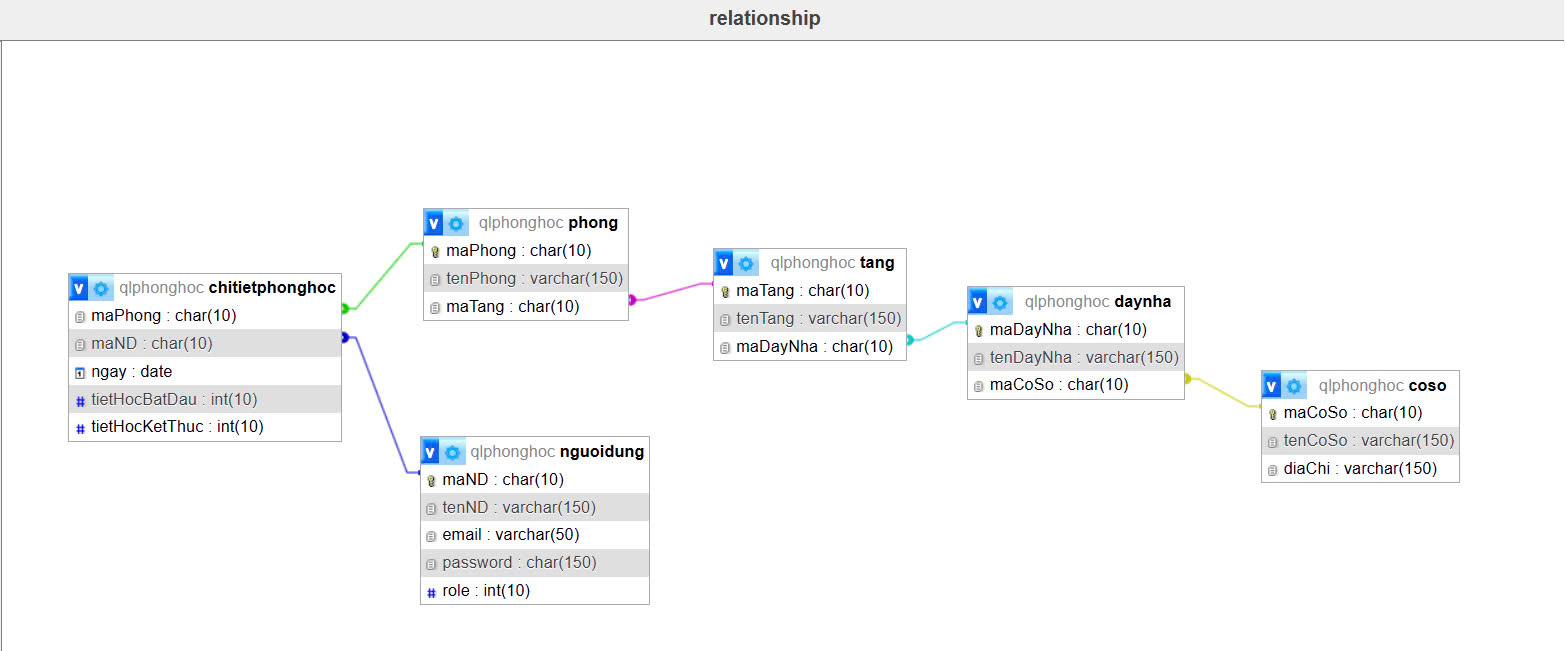
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Khóa | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | PK | maTang | Nvarchar(20) | Mã tầng |
| 2 |  | tenTang | Nvarchar(50) | Tên tầng |
| 3 | FK | maDayNha | Nvarchar(20) | Mã dãy nhà liên kết |

Bảng: PhongHoc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Khóa | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | PK | maPhong | Nvarchar(20) | Mã phòng học |
| 2 |  | tenPhong | Nvarchar(100) | Tên phòng học |
| 3 | FK | maTang | Nvarchar(20) | Mã tầng liên kết |

Bảng: ChiTietPhongHoc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Khóa | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | PK | maPhong | Nvarchar(20) | Mã phòng học |
| 2 | PK | ngay | Date | Ngày sử dụng |
| 3 | PK | tietBatDau | Int | Tiết bắt đầu |
| 4 | PK | tietKetThuc | Int | Tiết kết thúc |
| 5 |  | maND | Nvarchar(20) | Mã người dùng |
| 6 |  | trangThai | Nvarchar(50) | Trạng thái phòng |

**2.3.3 Mô hình quan hệ (Database Diagrams)**

Hình 2.10 : mô hình quan hệ

# **CHƯƠNG 3 : CÀI ĐẶT VÀ THỬ NGHIỆM**

## 3.1 GIAO DIỆN CHÀO MỪNG



Hình 3.1 : Giao diện chào mừng

## 3.2 GIAO DIỆN DÀNH CHO ADMIN

### **3.2.1 Giao diện đăng nhập**

Hình 3.2 Giao diện đăng nhập

### **3.2.2 Giao diện quản trị**

Hình 3.3 :Giao diện khi đăng nhập tài khoản dành cho admin

### **3.2.3 Giao diện tạo tài khoản giảng viên**

Hình 3.4 : Chức năng đăng ký tài khoản giảng viên

### **3.2.4 Giao diện quản lý cơ sở**

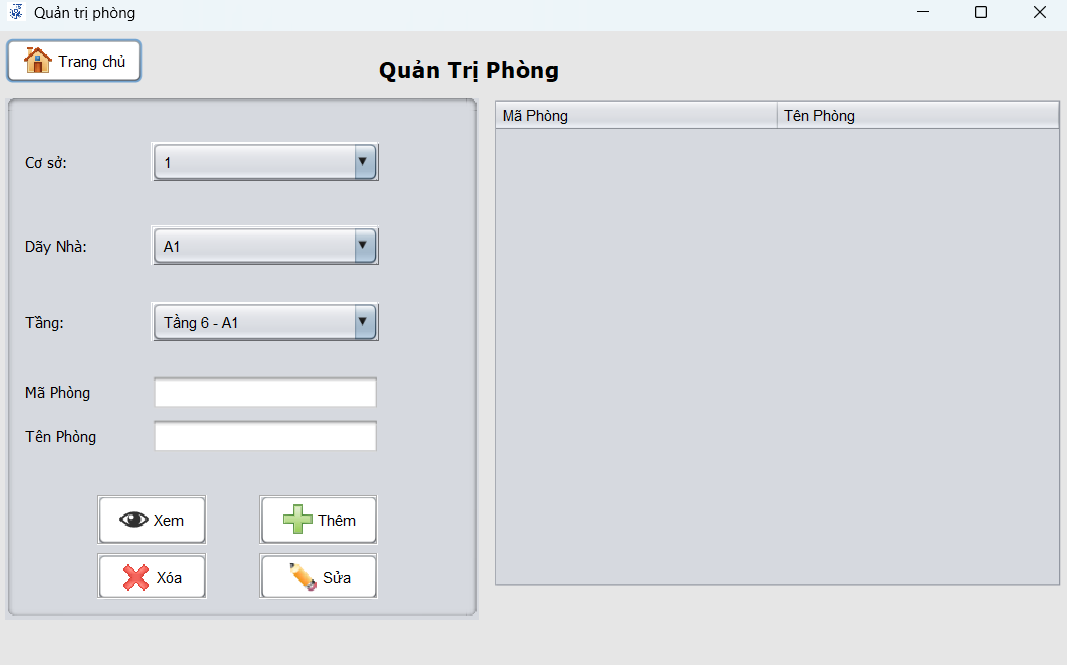
Hình 3.5 : Chức năng quản lý cơ sở

### **3.2.5 Giao diện quản lý dãy nhà**

Hình 3.6 : Chức năng quản lý dãy nhà

### **3.2.6 Giao diện quản lý tầng**

Hình 3.7: Chức năng quản lý tầng

**3.2.7 Giao diện quản lý phòng**

Hình 3.8 : Chức năng quản lý phòng

## 3.3 GIAO DIỆN DÀNH CHO GIẢNG VIÊN

### **3.3.1 Giao diện đăng nhập**

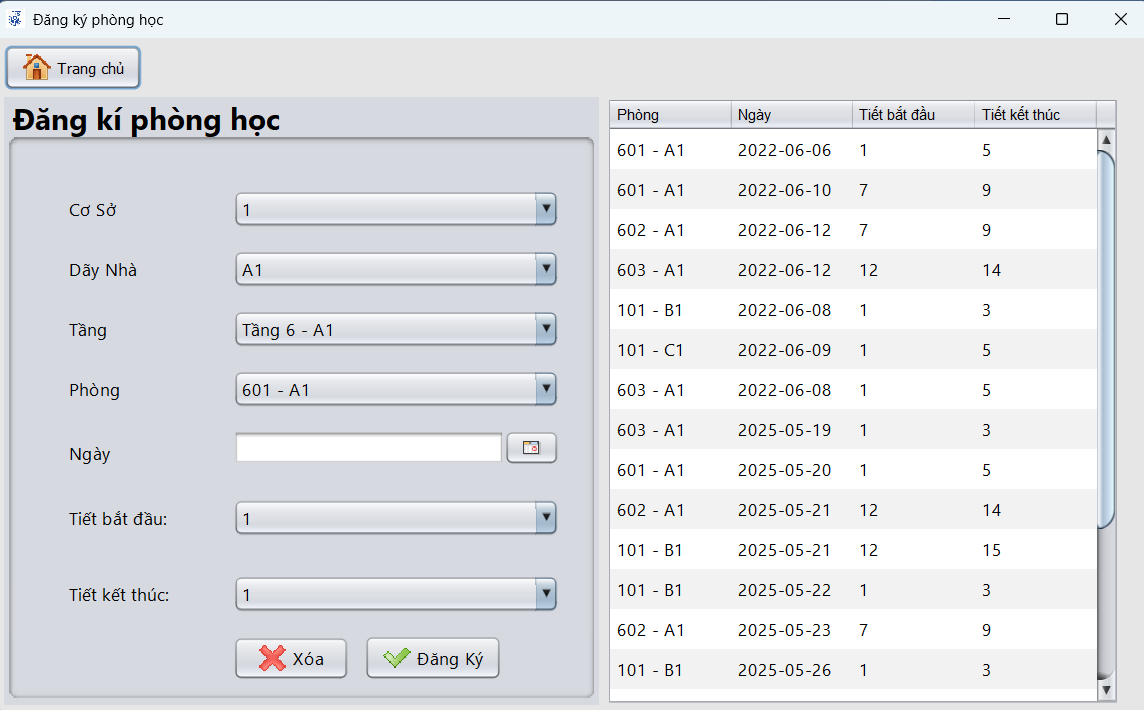
****

Hình 3.9 : Giao diện đăng nhập giảng viên

### **3.3.2 Giao diện menu giảng viên**

**Hình 3.10 : Giao diện menu giảng viên**

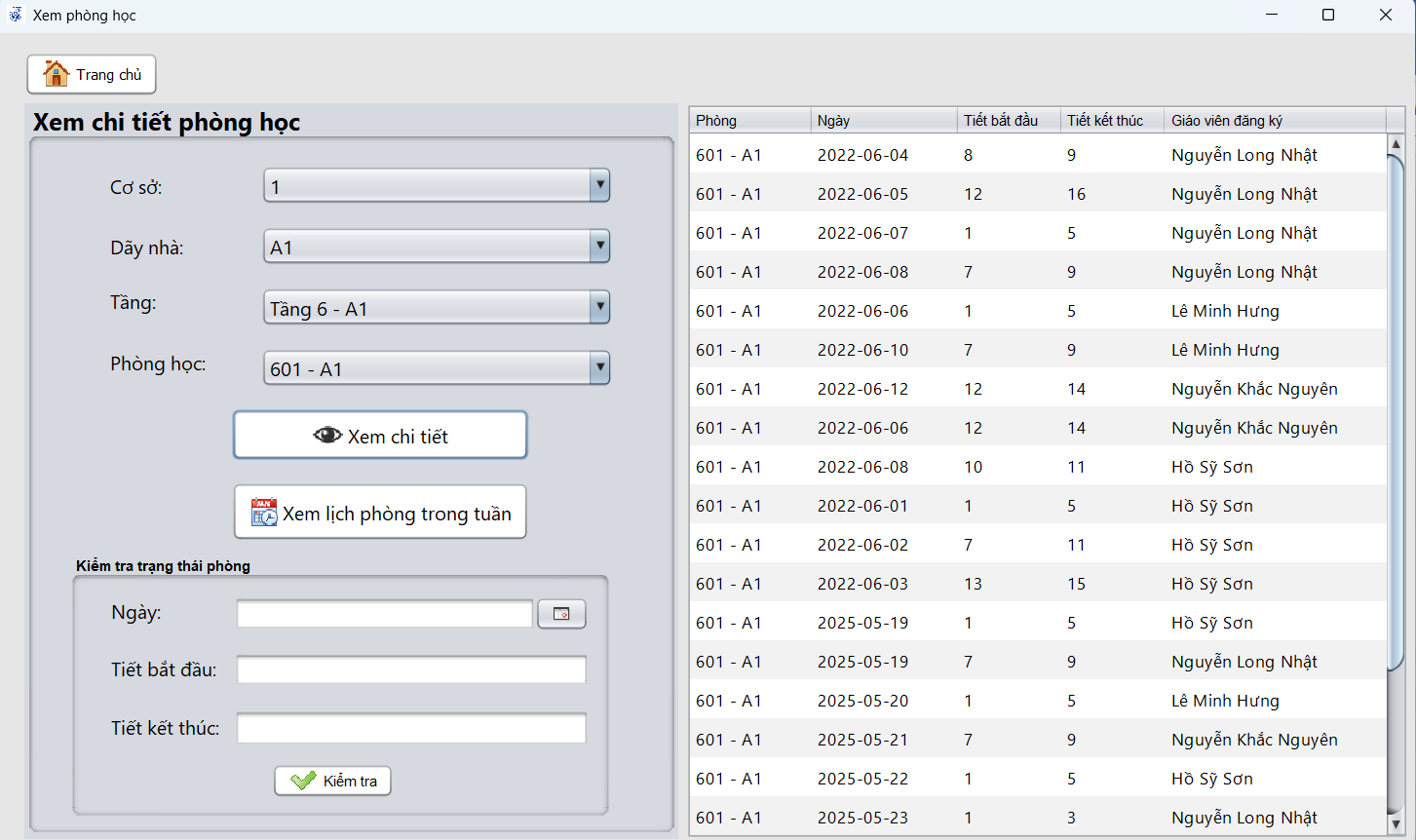
### **3.3.3 Giao diện đăng ký phòng học**

****

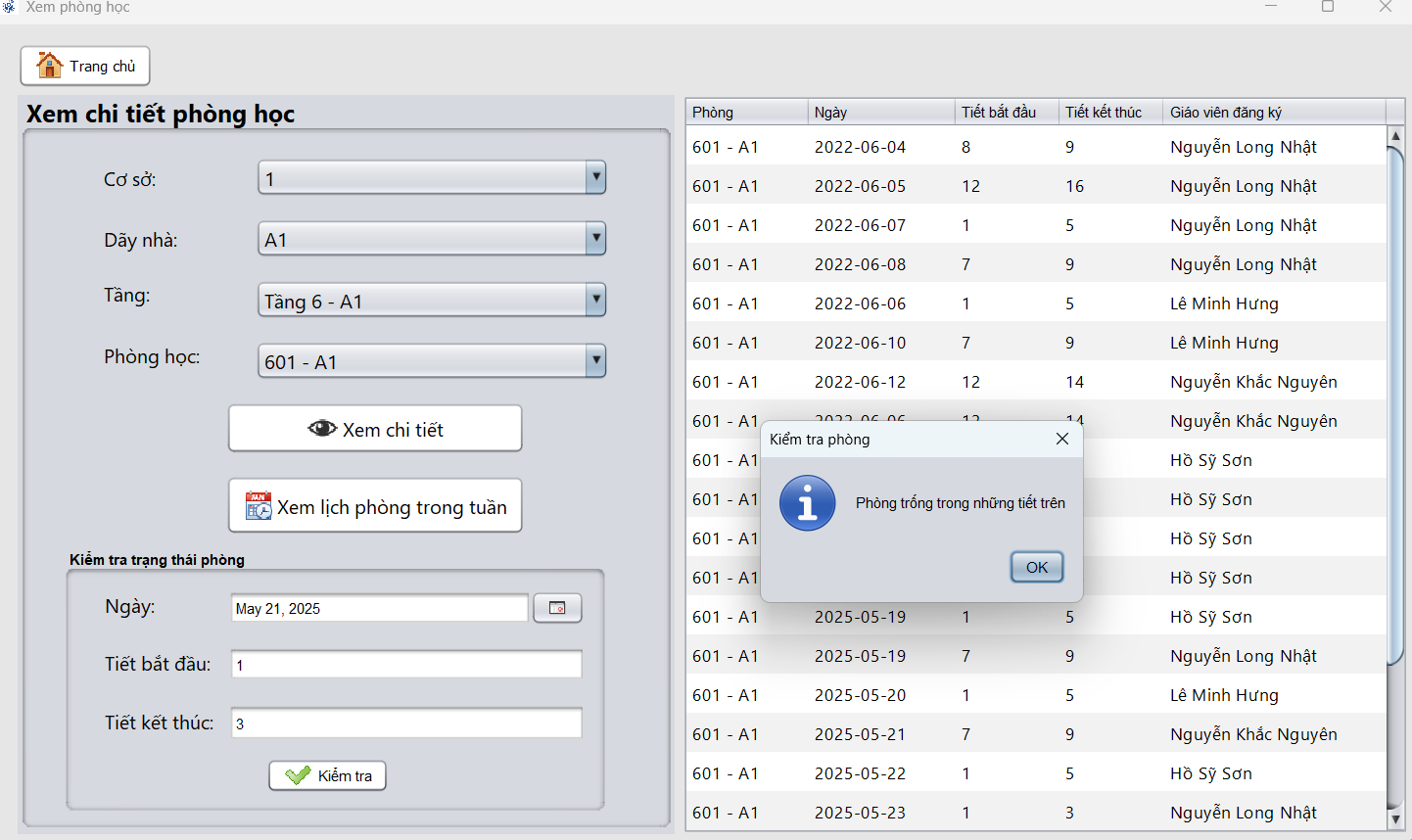
**Hình 3.11 : Giao diện đăng ký phòng học**

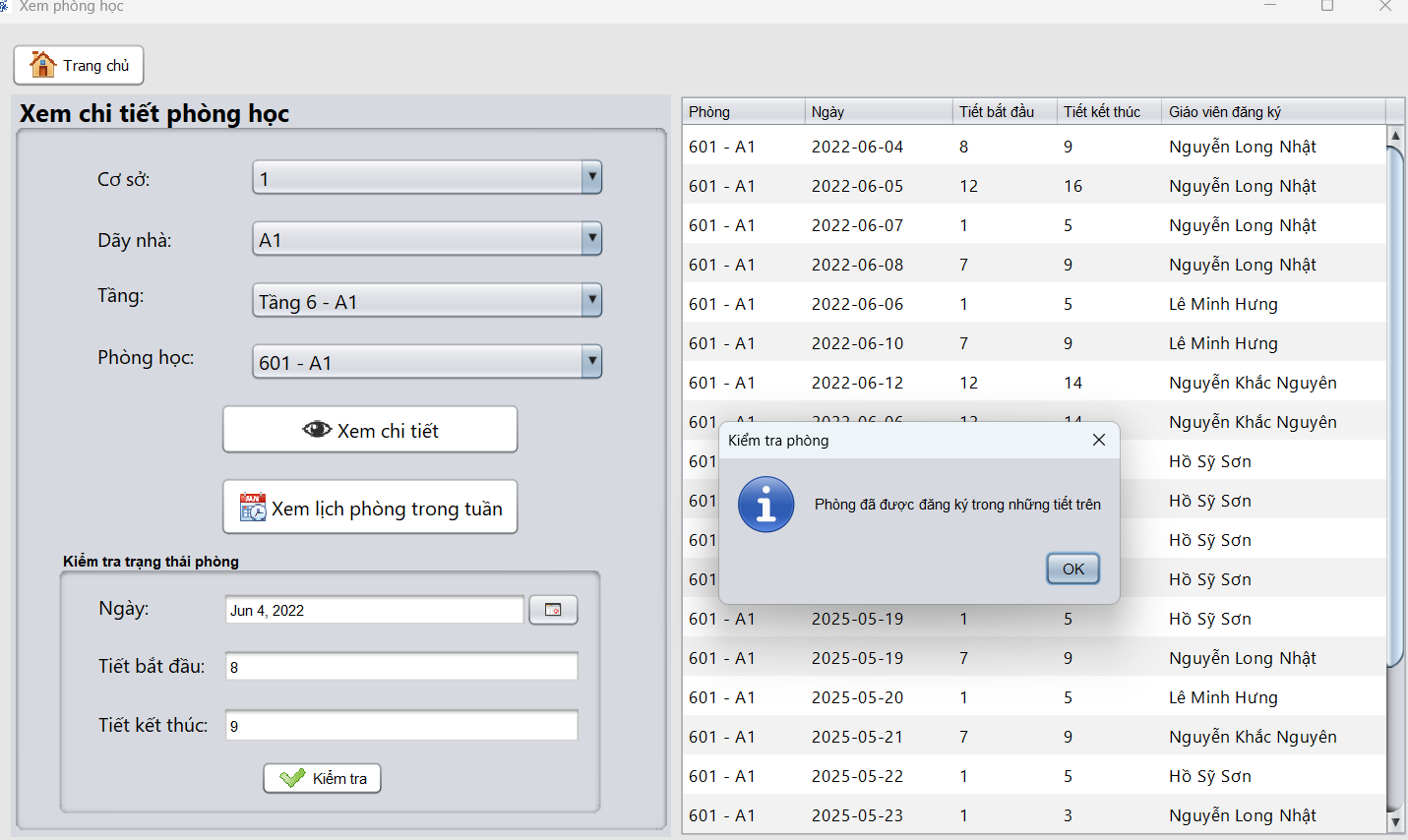
### **3.3.4 Giao diện xem và kiểm tra trạng thái phòng học**

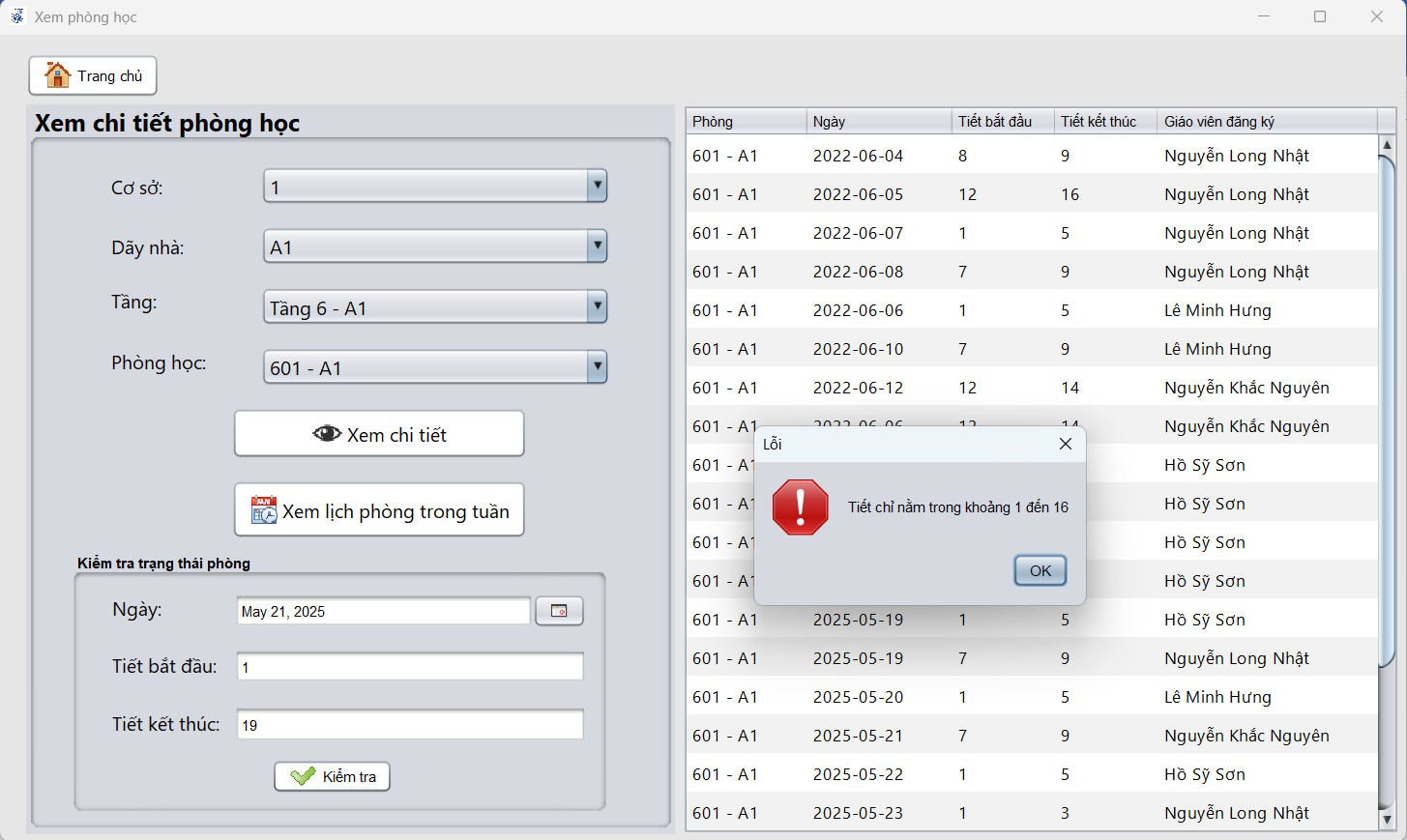
**Hình 3.12 : Giao diện trạng thái phòng học**

**3.3.4.1 Xem chi tiết phòng học**

Hình 3.13 : Xem chi tiết phòng học

**3.3.4.2 Kiểm tra trạng thái phòng học**

Hình 3.14 : trạng thái phòng đã đăng ký

**3.3.4.3** . Kiểm tra tính hợp lệ của tiết học

Hình 3.15: Kiếm tra tính hợp lệ của tiết học

## 3.4 KIỂM THỬ HỆ THỐNG

Kiểm thử hệ thống nhằm đảm bảo rằng phần mềm hoạt động đúng theo yêu cầu đặt ra , phát hiện và sửa chữa lỗi trước khi triển khai chính thức .

**Môi trường kiểm thử**

* Hệ điều hành windows 11
* Ứng dụng : Apache Netbeans IDE 21

**Quy trình kiểm thử**

* Viết kịch bản kiểm thử (Test Case)
* Tiến hành chạy thử các tính năng theo kịch bản kiểm thử
* Ghi nhận kết quả thực tế và so sánh với kết quả mong đợi
* Đánh giá Fail/Pass

**Kịch bản kiểm thử (Test Case ) cho hệ thống :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Mô tả | Input | Kỳ vọng | Kết quả |
| ĐĂNG NHẬP | | | | |
| TC\_DN\_01 | Đăng nhập thành công | Email: admin@gmail.com, Pass: 123 | Đăng nhập thành công → mở MenuQuanTri\_Form | Pass |
| TC\_DN\_02 | Email sai định dạng | Email: admin.com, Pass: 123 | Hiện thông báo lỗi định dạng email | Pass |
| TC\_DN\_03 | Mật khẩu sai | Email đúng, Pass sai | Hiện thông báo 'Sai thông tin đăng nhập' | Pass |
| TC\_DN\_04 | Tài khoản không tồn tại | Email lạ + pass bất kỳ | Hiện thông báo 'Không tìm thấy tài khoản' | Pass |
| TC\_DN\_05 | Trống email hoặc mật khẩu | Email hoặc mật khẩu rỗng | Hiện cảnh báo 'Không được để trống' | Pass |
| ĐĂNG KÝ PHÒNG | | | | |
| TC\_DKP\_01 | Đăng ký phòng thành công | Chọn phòng, nhập ngày, tiết 1-3 | Thêm vào CSDL, refresh bảng lịch phòng | Pass |
| TC\_DKP\_02 | Trùng tiết đã đăng ký | Chọn phòng, ngày, tiết trùng lịch cũ | Hiện thông báo 'Phòng đã được đăng ký' | Pass |
| TC\_DKP\_03 | Nhập tiết không hợp lệ | Tiết bắt đầu = 10, tiết kết thúc = 3 | Hiện cảnh báo 'Tiết không hợp lệ' | Pass |
| TC\_DKP\_04 | Không chọn phòng | Để trống combo box phòng học | Hiện cảnh báo | Pass |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | | Mô tả | | Input | Kỳ vọng | | Kết quả | |
| MENU GIÁO VIÊN | | | | | | | | |
| TC\_MGV\_01 | Mở form xem phòng học | | Click nút tương ứng | | | Form XemPhongHoc\_Form hiển thị | | Pass |
| TC\_MGV\_02 | Đăng xuất | | Click nút logout | | | Quay lại DangNhap\_Form | | Pass |
| QUẢN LÝ CƠ SỞ | | | | | | | | |
| TC\_CS\_01 | Thêm cơ sở hợp lệ | | Nhập tên, địa chỉ | | | Cơ sở được thêm và hiển thị trong bảng | | Pass |
| TC\_CS\_02 | Trùng mã cơ sở | | Nhập mã đã tồn tại | | | Hiện thông báo lỗi | | Pass |
| TC\_CS\_03 | Xóa cơ sở đang được dùng | | Xóa cơ sở có dãy nhà liên quan | | | Cảnh báo không cho xóa | | Pass |
| XEM LỊCH SỬ DỤNG | | | | | | | | |
| TC\_XPH\_01 | Load đúng dữ liệu | | Chọn Cơ sở → Dãy → Tầng → Phòng | | | Hiển thị đúng thông tin phòng và lịch sử dụng | | Pass |
| TC\_XPH\_02 | Phòng chưa có lịch | | Chọn phòng trống | | | Hiện thông báo 'Phòng trống' | | Pass |
| TC\_XPH\_03 | Kiểm tra trạng thái phòng | | Nhập ngày và tiết | | | Hiển thị đúng trạng thái | | Pass |
| TC\_XPH\_04 | Mở form lịch tuần | | Click nút 'Xem lịch tuần' | | | Mở XemLichPhong\_Form | | Pass |
| LỊCH THEO TUẦN | | | | | | | | |
| TC\_LT\_01 | Xem đúng lịch tuần | | Chọn phòng bất kỳ | | | Hiển thị lịch theo từng ngày, đúng dữ liệu | | Pass |
| TC\_LT\_02 | Phòng không có lịch | | Phòng trống cả tuần | | | Hiển thị lịch rỗng hoặc thông báo | | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testcase ID | Mô tả | Input | Kỳ vọng | Kết quả |
| QUẢN LÝ DÃY NHÀ | | | | |
| TC\_DNHA\_01 | Thêm dãy nhà thành công | Nhập mã, tên dãy, mã cơ sở hợp lệ | Dãy nhà được thêm vào DB và hiển thị | Pass |
| TC\_DNHA\_02 | Thêm trùng mã dãy nhà | Nhập mã đã tồn tại | Hiện thông báo lỗi | Pass |
| TC\_DNHA\_03 | Xóa dãy nhà đang chứa tầng | Chọn dãy nhà có tầng liên quan | Hiện cảnh báo, không cho xóa | Pass |
| QUẢN LÝ TẦNG | | | | |
| TC\_TANG\_01 | Thêm tầng thành công | Nhập mã, tên tầng, mã dãy nhà hợp lệ | Tầng được thêm vào DB | Pass |
| TC\_TANG\_02 | Thêm trùng mã tầng | Nhập mã tầng đã tồn tại | Hiện cảnh báo lỗi | Pass |
| TC\_TANG\_03 | Xóa tầng có phòng | Tầng đang chứa phòng học | Không cho phép xóa | Pass |
| QUẢN LÝ TẦNG | | | | |
| TC\_PHONG\_01 | Thêm phòng học hợp lệ | Nhập mã, tên phòng, mã tầng đúng | Thêm thành công và hiển thị trong bảng | Pass |
| TC\_PHONG\_02 | Thêm trùng mã phòng | Nhập mã phòng đã tồn tại | Thông báo lỗi | Pass |
| TC\_PHONG\_03 | Xóa phòng có lịch sử dụng | Chọn phòng đã từng đăng ký tiết | Không cho xóa, hiện thông báo | Pass |

# **KẾT LUẬN**

Sau quá trình nghiên cứu và phát triển, dưới sự hướng dẫn tận tình của giảng viên – cô Trần Thị Huệ, nhóm chúng em đã hoàn thành hệ thống **ứng dụng quản lý phòng học** sử dụng công nghệ **Java Swing** kết nối với cơ sở dữ liệu **MySQL**.

Hệ thống đã đáp ứng được các chức năng chính như: quản lý cơ sở vật chất (cơ sở, dãy nhà, tầng, phòng học), đăng ký và kiểm tra tình trạng phòng học theo thời gian thực, phân quyền người dùng (giảng viên và quản trị viên), kiểm tra lịch phòng theo tuần, đảm bảo đúng yêu cầu đặt ra trong đề tài.

**Kết quả đạt được:**

* Giao diện người dùng rõ ràng, dễ sử dụng, xử lý mượt mà các chức năng tương tác như đăng nhập, đăng ký phòng, kiểm tra trạng thái phòng.
* Quản trị viên có thể dễ dàng tạo tài khoản, quản lý thông tin cơ sở vật chất.
* Giảng viên có thể đăng ký phòng học và kiểm tra lịch phòng thuận tiện, tiết kiệm thời gian và tăng hiệu quả sử dụng phòng học.
* Kiểm thử hệ thống cho thấy các chức năng hoạt động ổn định, đúng yêu cầu, và xử lý được nhiều trường hợp ngoại lệ.

**Hạn chế còn tồn tại:**

* Giao diện tuy đầy đủ chức năng nhưng vẫn còn đơn giản về mặt thẩm mỹ, chưa thực sự trực quan, chưa tối ưu trải nghiệm người dùng.
* Việc kiểm tra xung đột lịch vẫn còn sơ khai, chưa có tính năng cảnh báo nâng cao hoặc hiển thị trực quan lịch biểu.
* Một số phần code có thể chưa được tối ưu về hiệu suất truy vấn hoặc tổ chức lớp chưa chuẩn hóa hoàn toàn theo mô hình MVC.

# **ĐỊNH HƯỚNG PHÁT TRIỂN**

Trong thời gian tới, để hệ thống hoàn thiện và thân thiện hơn, nhóm đề xuất một số định hướng như sau:

* **Tích hợp tính năng video call / chat thời gian thực**, giúp giảng viên và sinh viên có thể tương tác trực tiếp trong những lớp học đặc biệt hoặc khi cần trao đổi thông tin.
* **Nâng cấp giao diện người dùng** bằng cách sử dụng các framework UI hiện đại (như JavaFX hoặc tích hợp WebView), cải thiện bố cục, màu sắc, biểu tượng nhằm tạo trải nghiệm trực quan hơn.
* **Cải thiện tính năng lịch học:** hiển thị lịch học bằng dạng bảng trực quan (ví dụ: dạng lịch tuần), hỗ trợ kéo – thả để chỉnh sửa, kiểm tra xung đột tốt hơn.
* **Bổ sung tính năng thông báo (notification)** khi phòng bị đăng ký trùng hoặc khi có cập nhật phòng học.
* **Phát triển phiên bản Web hoặc Mobile** để mở rộng khả năng truy cập và quản lý từ xa cho giảng viên và sinh viên.

# **TÀI LIỆU THAM KHẢO**

1. **Oracle Java Documentation** – <https://docs.oracle.com/javase/>
2. **Java Swing Tutorial** – https://www.javatpoint.com/java-swing
3. **MySQL Documentation** – <https://dev.mysql.com/doc/>
4. Nguyễn Hữu Tuấn – Giáo trình Java căn bản, NXB Giao thông Vận tải, 2021.
5. Nguyễn Văn Hiển – Giáo trình Phân tích và Thiết kế hệ thống thông tin, NXB Bách Khoa Hà Nội, 2020.
6. Tài liệu giảng dạy môn Thực hành Công nghệ Java, Khoa CNTT – Trường Đại học Kinh tế Kỹ thuật Công nghiệp.
7. Stack Overflow – Hệ thống hỗ trợ lập trình viên: <https://stackoverflow.com/>
8. YouTube – Các kênh hướng dẫn Java Swing & MySQL (Lập trình Java Cơ bản đến Nâng cao).