**1. Mô hình kiến trúc được chọn: MVC (Model – View – Controller)**

**Lý do lựa chọn:**

* Hệ thống có giao diện người dùng (web/app) và nhiều nhóm người dùng khác nhau như **Ban Giám Hiệu, Phòng Đào Tạo, Giảng viên, Quản trị viên**.
* **MVC** giúp tách biệt rõ ràng giữa:
  + **View:** giao diện người dùng.
  + **Controller:** xử lý và điều phối yêu cầu.
  + **Model:** dữ liệu và nghiệp vụ.
* Việc phân tách này giúp **dễ mở rộng, bảo trì, kiểm thử và tái sử dụng** mã nguồn.
* Mô hình phù hợp với các ứng dụng web hiện đại phát triển bằng **Spring Boot** hoặc **.NET MVC**.

**2. Mô tả các thành phần kiến trúc**

| **Thành phần** | **Vai trò** | **Mô tả chi tiết** |
| --- | --- | --- |
| **Model** | Dữ liệu & Nghiệp vụ | Chứa các lớp đại diện cho đối tượng trong hệ thống như: GiangVien, Khoa, LichDay, PhanCong, PhanHoi, TaiKhoan. Thực hiện nghiệp vụ và giao tiếp với cơ sở dữ liệu. |
| **View** | Giao diện người dùng | Hiển thị thông tin và nhận thao tác từ người dùng. Ví dụ: trang **“Quản lý hồ sơ giảng viên”**, **“Phân công lịch dạy”**, **“Xem báo cáo thống kê”**. |
| **Controller** | Điều phối & Xử lý yêu cầu | Tiếp nhận yêu cầu từ người dùng, gọi đến Model hoặc Service để xử lý, sau đó trả kết quả về View. Ví dụ: GiangVienController, LichDayController, BaoCaoController. |
| **Database Layer** | Lưu trữ dữ liệu | Cơ sở dữ liệu quan hệ (MySQL/PostgreSQL) chứa các bảng: GiangVien, Khoa, LichDay, PhanHoi, TaiKhoan, VaiTro. |
| **Service Layer (tuỳ chọn)** | Tách logic nghiệp vụ | Đặt giữa Controller và Model, dùng trong hệ thống có nghiệp vụ phức tạp nhằm giảm tải Controller và tăng khả năng tái sử dụng. |

**3. Sơ đồ kiến trúc tổng quan**

**Mô tả luồng hoạt động:**

1. Người dùng (BGH, Giảng viên, Phòng Đào Tạo, Admin) truy cập hệ thống qua giao diện web.
2. Yêu cầu được gửi đến **Controller** tương ứng.
3. **Controller** gọi **Service** hoặc **Model** để xử lý nghiệp vụ và truy xuất dữ liệu từ **Database**.
4. Kết quả được trả về **View** để hiển thị cho người dùng.

**Sơ đồ tổng quan :**

[Người dùng]

│

▼

[View (Giao diện Web)]

│

▼

[Controller] ───> [Service Layer] ───> [Model] ───> [Database]

│ ▲

└─────────────── Nhận dữ liệu hiển thị ─┘

A screenshot of a computer

AI-generated content may be incorrect.

*Ví dụ* khi **Giảng viên đăng nhập**, yêu cầu được gửi đến LoginController, sau đó gọi TaiKhoanService để kiểm tra thông tin trong Database. Kết quả được gửi lại cho LoginView hiển thị thông báo đăng nhập thành công hoặc thất bại.

**4. Công nghệ đề xuất**

| **Thành phần** | **Công nghệ sử dụng** |
| --- | --- |
| **Frontend** | HTML, CSS, JavaScript, hoặc framework như **Vue.js / React** |
| **Backend** | **Spring Boot (Java)** theo mô hình **MVC** |
| **Database** | **MySQL** hoặc **PostgreSQL** |
| **Triển khai** | Server nội bộ hoặc nền tảng Cloud (**Heroku, Render, AWS**) |

**5. Ưu điểm của mô hình MVC**

* Tách biệt rõ ràng giữa các tầng: dễ dàng **phát triển song song** và **bảo trì**.
* **Tăng khả năng mở rộng** khi thêm tính năng mới.
* **Dễ kiểm thử** và tái sử dụng mã nguồn.
* Phù hợp cho các **ứng dụng web hiện đại** và **đa người dùng**.