



GIỚI THIỆU MỘT SỐ MÔ HÌNH LẬP TRÌNH (Lập trình trực quan)

Ths. Nguyễn Thu Hường

NỘI DUNG

❖ Mô hình 3 lớp

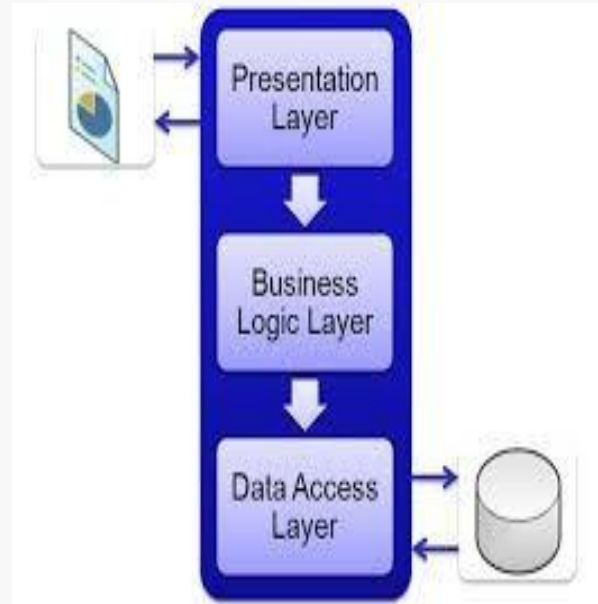
❖ Mô hình MVC

MÔ HÌNH 3 LỚP

Khái niệm mô hình 3 lớp



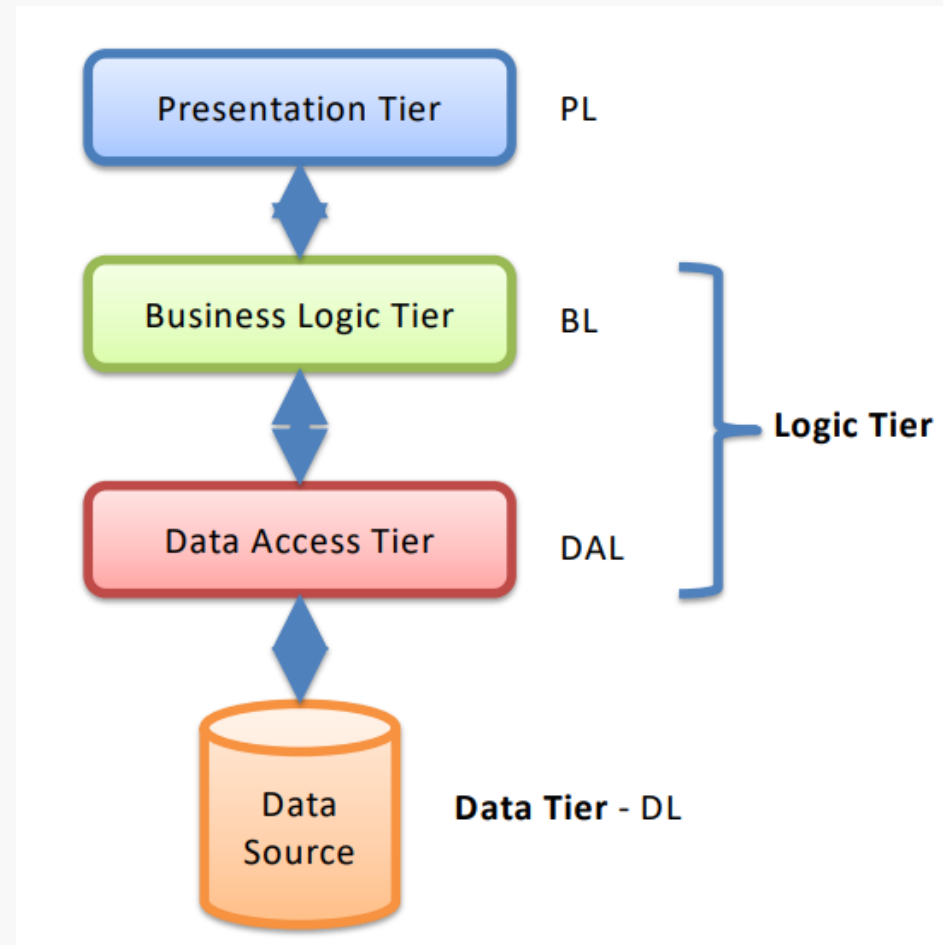
- ❖ Mô hình 3 lớp hay Kiến trúc 3 lớp (**3 Tier/Layer Architecture**)
- ❖ Là một kiến trúc phần mềm chia ứng dụng thành ba lớp (layer) riêng biệt:
 - Lớp trình bày (Presentation Layer)
 - Lớp logic nghiệp vụ (Business Logic Layer)
 - Lớp truy cập dữ liệu (Data Access Layer).



Kiến trúc mô hình 3 lớp



- ❖ Các lớp khác nhau có thể nằm trên cùng một máy tính, hoặc trên các máy tính khác nhau.



Lớp Presentation



- ❖ Là nơi giao tiếp với người dùng.
- ❖ Nhận dữ liệu từ người dùng chuyển vào lớp Business logic
- ❖ Hiển thị kết quả xử lý
- ❖ **Thông thường là:** Windows form, wpf, web form, HTML/CSS..

- ❖ Chứa các quy tắc xử lý nghiệp vụ: tính toán, kiểm tra điều kiện, xử lý dữ liệu.
- ❖ Là cầu nối giữa giao diện và dữ liệu
- ❖ Giúp tách biệt giao diện và cơ sở dữ liệu
- ❖ **Ví dụ:** Kiểm tra đăng nhập hợp lệ, tính tiền đơn hàng,...

- ❖ Giao tiếp trực tiếp với cơ sở dữ liệu (SQL Server, MySQL...).
- ❖ Dùng để kết nối cơ sở dữ liệu. Truy vấn, thêm, sửa, xóa dữ liệu
- ❖ **Thường dùng:** ADO.NET, Entity framework...

- ❖ Dễ phân chia công việc cho các thành viên trong nhóm phát triển phần mềm.
- ❖ Tái sử dụng
- ❖ Dễ bảo trì, sửa đổi.
- ❖ Dễ phát triển, mở rộng.
- ❖ Dễ kiểm thử.
- ❖ Tăng tính bảo mật do lớp giao diện không truy cập trực tiếp vào dữ liệu

- ❖ Đối với dự án nhỏ thì phát triển theo mô hình 3 lớp sẽ **làm tăng tính phức tạp**
- ❖ Hiệu năng bị ảnh hưởng nếu thiết kế không tốt, do dữ liệu phải đi qua nhiều lớp.
- ❖ Tốn thời gian phát triển ban đầu.
- ❖ Việc chia nhỏ thành nhiều lớp đòi hỏi thiết kế và tổ chức mã nguồn tốt. → Gây khó khăn cho những lập trình viên mới.

MÔ HÌNH MVC

Khái niệm mô hình MVC

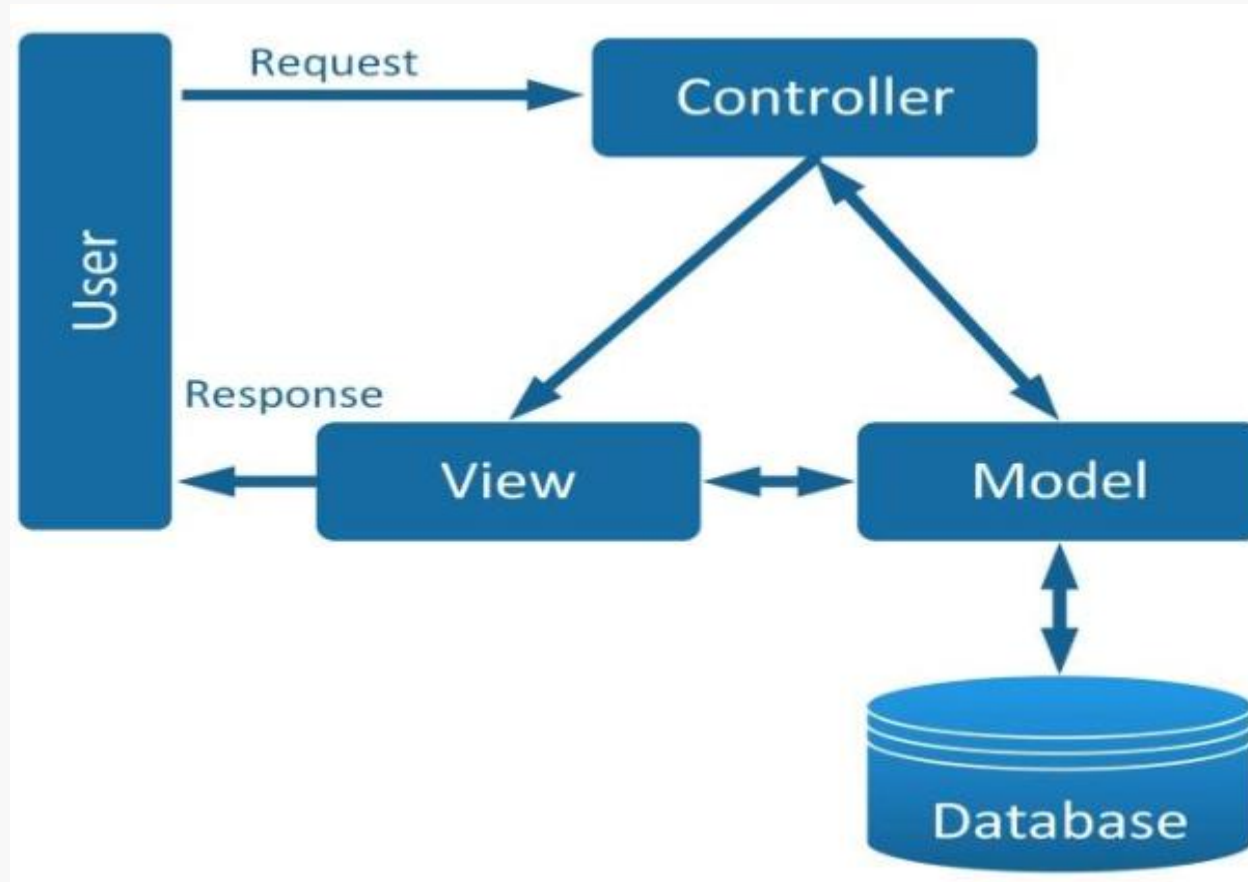


- ❖ MVC (Model – View – Controller).
- ❖ Là một mô hình kiến trúc phần mềm được sử dụng phổ biến trong **phát triển ứng dụng web và desktop** (thường sử dụng phổ biến cho ứng dụng Web)
- ❖ Nhằm **tách biệt rõ ràng các thành phần chức năng** trong ứng dụng.

Kiến trúc mô hình MVC



- ❖ **MVC** chia thành *ba phần* được *kết nối với nhau* và mỗi thành phần đều có một nhiệm vụ **riêng** của nó và **độc lập** với các thành phần khác.



❖ Model

- Xử lý nghiệp vụ, tính toán ...
- Có nhiệm vụ thao tác với **Database**
- Nó chứa **tất cả các hàm, các phương thức truy vấn trực tiếp** với dữ liệu
- **Controller** sẽ thông qua các hàm, phương thức đó để **lấy dữ liệu** rồi gửi qua **View**

❖ View

- Là giao diện người dùng (**User Interface**)
- Chứa các thành phần tương tác với người dùng như menu, button, image, text,...
- Nơi nhận dữ liệu từ **Controller** và hiển thị

❖ Controller

- Là thành phần **trung gian** giữa **Model** và **View**
- Đảm nhận vai trò tiếp nhận yêu cầu từ người dùng, thông qua **Model** để **lấy dữ liệu** sau đó thông qua **View** để **hiển thị** cho người dùng.

Luồng xử lý trong MVC



- ❖ Đầu tiên là **Request** từ người dùng được **gửi từ client đến server**
- ❖ Sau đó **Controller** dựa vào yêu cầu của người dùng tiến hành **giao tiếp với Model** để lấy data từ database
- ❖ Cuối cùng **Controller** gửi dữ liệu vừa lấy được về **View** và hiển thị ra cho người dùng trên trình duyệt

Ưu điểm



- ❖ Phân tách rõ ràng giữa giao diện và nghiệp vụ → Dễ dàng phân chia nhiệm vụ trong nhóm, hỗ trợ tương tác làm việc nhóm
- ❖ Tái sử dụng
- ❖ Dễ bảo trì, dễ mở rộng
- ❖ Phù hợp với ứng dụng web

Nhược điểm



- ❖ Phức tạp hóa cấu trúc dự án cho những dự án nhỏ.
- ❖ Khó lập trình cho những lập trình viên mới vào nghề
- ❖ Tăng số lượng file, lớp trong dự án
- ❖ Quản lý liên kết phức tạp
- ❖ Controller dễ quá tải

Bài tập



- ❖ So sánh mô hình 3 lớp (three layer) và mô hình MVC.
- ❖ Triển khai bài tập lớn dựa trên kiến trúc 3 lớp

Trân trọng cảm ơn!