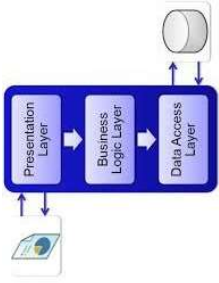


## Phân biệt mô hình 3 layer và MVC

3 Layer và MVC là 2 mô hình phổ biến trong lập trình và có nhiều điểm giống nhau, mà không phải lập trình viên nào cũng phân biệt được một cách rõ ràng. Vậy thì hôm nay chúng ta sẽ cùng nhau tìm hiểu về sự giống và khác nhau giữa 2 mô hình này. Trước khi phân biệt chúng thì tôi và các bạn hãy tìm hiểu về chúng trước đã nhé. **1. Mô hình 3 layer** ( tiếng Việt gọi nôm na là 3 lớp, cần phân biệt với 3 Tier, nhiều khi bị nhầm lẫn nghiêm trọng, các bạn có thể tìm hiểu thêm ngoài phạm vi bài này 😊 ). **Mô hình 3 layer được cấu thành như hình minh họa dưới đây.**

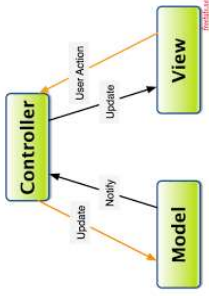


Như bạn thấy thì 3 layer gồm có 3 thành phần: Presentation Layers, Business Logic Layers, và Data Access Layers.

- **Presentation Layers:** Lớp này làm nhiệm vụ giao tiếp với người dùng cuối để thu thập dữ liệu và hiển thị kết quả/dữ liệu thông qua các thành phần trong giao diện người sử dụng.
- **Business Logic Layers:** Đây là layer xử lý chính các dữ liệu trước khi được đưa lên hiển thị trên màn hình hoặc xử lý các dữ liệu trước khi chuyển xuống Data Access Layer để lưu dữ liệu xuống cơ sở dữ liệu. Đây cũng là nơi để kiểm tra ràng buộc, các yêu cầu nghiệp vụ, tính toán, xử lý các yêu cầu và lựa chọn kết quả trả về cho Presentation Layers.
- **Data Access Layers:** Lớp này thực hiện các nghiệp vụ liên quan đến lưu trữ và truy xuất dữ liệu của ứng dụng như đọc, lưu, cập nhật cơ sở dữ liệu. **Cách thức vận hành như sau:**
- Đầu tiên User giao tiếp với Presentation Layers (GUI) để gửi đi thông tin và yêu cầu. Tại layer này, các thông tin sẽ được kiểm tra, nếu OK chúng sẽ được chuyển xuống Business Logic Layer (BLL).
- Tại BLL, các thông tin sẽ được nhào nặn, tính toán theo đúng yêu cầu đã gửi, nếu không cần đến Database thì BLL sẽ gửi trả kết quả về GUI, ngược lại nó sẽ đẩy dữ liệu (thông tin đã xử lý) xuống Data Access Layer (DAL).
- DAL sẽ thao tác với Database và trả kết quả về cho BLL, BLL kiểm tra và gửi nó lên GUI để hiển thị cho người dùng.

- Một khi gặp lỗi (các trường hợp không đúng dữ liệu) thì đang ở layer nào thì quăng lên trên layer cao hơn nó 1 bậc cho tới GUI thì sẽ quăng ra cho người dùng biết – Các dữ liệu được trung chuyển giữa các Layer thông qua một đối tượng gọi là Data Transfer Object (DTO), đơn giản đây chỉ là các Class đại diện cho các đối tượng được lưu trữ trong Database. **Tổ chức mô hình 3 layer:** Có rất nhiều cách đặt tên cho các thành phần của 3 lớp như: Cách 1: GUI, BUS, DAL Cách 2: GUI, BLL, DAO, DTO Cách 3: Presentation, BLL, DAL \*\* II. Mô hình MVC. \*\* \*\* Mô hình MVC là gì? \*\* Mô hình MVC (Model – View – Controller) là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. Nó giúp cho các developer tách ứng dụng của họ ra 3 thành phần khác nhau Model, View và Controller. Mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác. **Các thành phần của MVC:**

- Model: Đây là thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất database, đối tượng mô tả dữ liệu như các Class, hàm xử lý...
- View: Đảm nhận việc hiển thị thông tin, tương tác với người dùng, nơi chứa tất cả các đối tượng GUI như textbox, images...Hiểu một cách đơn giản, nó là tập hợp các form hoặc các file HTML
- Controller: Giữ nhiệm vụ nhận điều hướng các yêu cầu từ người dùng và gọi đúng những phương thức xử lý chúng... Chẳng hạn thành phần này sẽ nhận request từ url và form để thao tác trực tiếp với Model. Đây là hình ảnh minh họa:



**MVC làm việc như thế nào?** Đây là một cách đơn giản để mô tả lại luồng sự kiện được xử lý trong MVC: – User tương tác với View, bằng cách click vào button, user gửi yêu cầu đi. – Controller nhận và điều hướng chúng đến đúng phương thức xử lý ở Model. – Model nhận thông tin và thực thi các yêu cầu. – Khi Model hoàn tất việc xử lý, View sẽ nhận kết quả từ Model và hiển thị lại cho người dùng.

**Ưu điểm và nhược điểm của MVC:** 1. *Ưu điểm* Thể hiện tính chuyên nghiệp trong lập trình, phân tích thiết kế. Do được chia thành các thành phần độc lập nên giúp phát triển ứng dụng nhanh, đơn giản, dễ nâng cấp, bảo trì.. 2. *Nhược điểm* Đối với dự án nhỏ việc áp dụng mô hình MVC gây cồng kềnh, tốn thời gian trong quá trình phát triển. Tốn thời gian trung chuyển dữ liệu của các thành phần.

Sau khi đã tìm hiểu về 2 mô hình trên, chúng ta nhận thấy được điểm giống và khác nhau giữa 2 mô hình trên như sau: *Điểm giống nhau:*

- Cả hai đều là mô hình kiến trúc áp dụng trong lập trình được tách thành 3 thành phần.
- Tách biệt sự phụ thuộc giữa ngôn ngữ lập trình, môi trường phát triển, xử lý logic.
- Các thành phần ở mỗi mô hình có nhiều điểm tương đồng như Presentation Layer và View... *Điểm khác biệt:*
- Khác biệt rõ nhất là cách xử lý yêu cầu (workflow) của 2 mô hình (Tham khảo cách làm việc của 2 mô hình).
- Ở MVC, thành phần Model giữ chức năng của cả lớp Business và Data Access trong mô hình 3 Layer.

**Vậy khi nào nên sử dụng 3-Layer, còn khi nào nên sử dụng MVC?**

- 3 Layer thường được sử dụng trong lập trình ứng dụng.
- Trong khi đó, MVC được ưu chuộng nhiều trong lập trình website vì tính linh hoạt và không bắt buộc các DTO như 3-Layer .