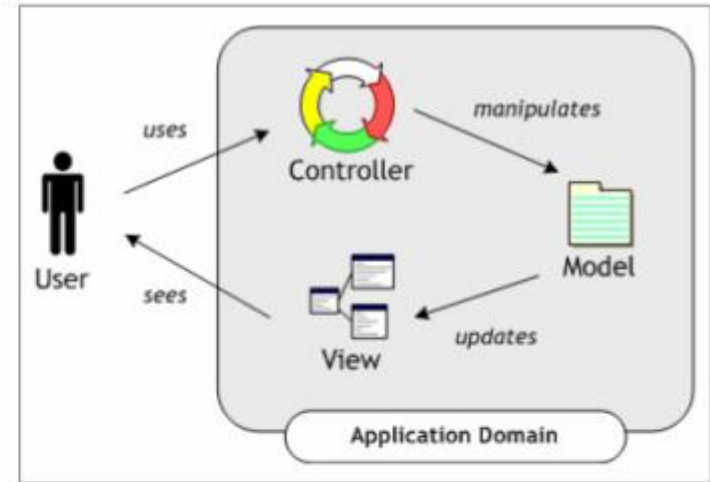
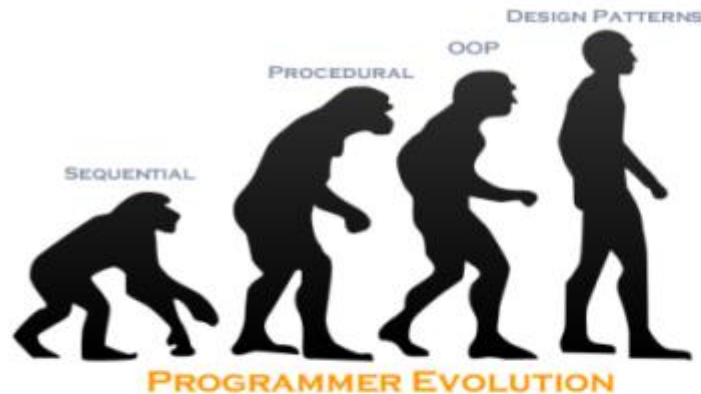
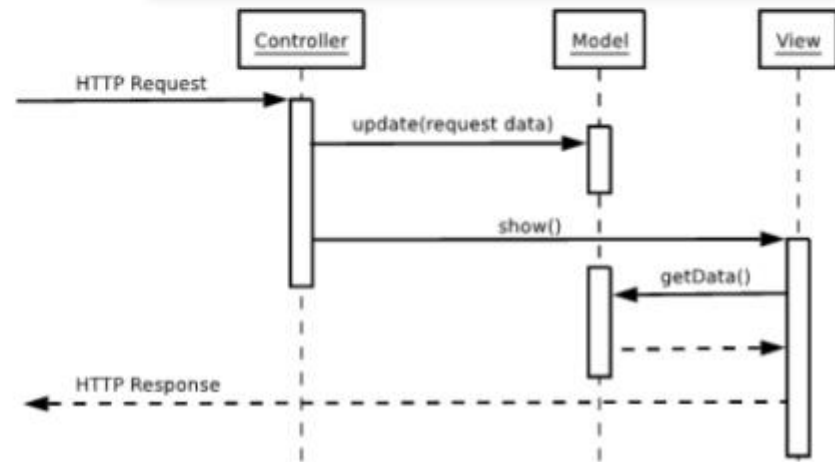


Giới thiệu về ASP.NET MVC

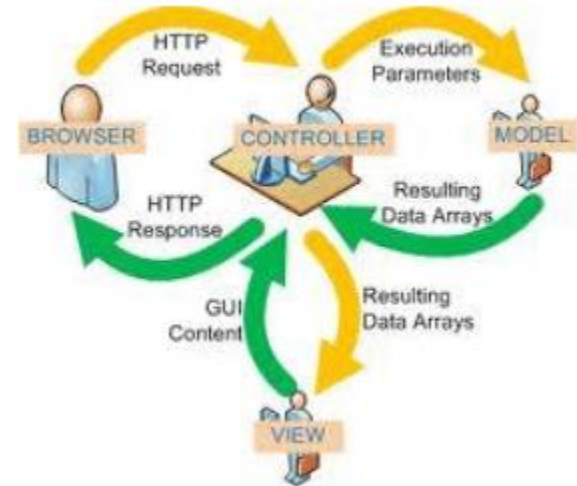
ASP.NET MVC



- MVC Patterns
 - Model
 - View
 - Controller
- MVC Frameworks
 - Java: Struct, Spring
 - PHP: Zend
 - ASP.NET MVC



ASP.NET MVC



■ Controller

- Nhận yêu cầu từ user
- Xử lý và xây dựng model phù hợp
- Chuyển Model cho View

■ View: Tiếp nhận Model từ Controller để sinh giao diện phù hợp

■ Model: Chứa dữ liệu chia sẻ chung giữa Controller và View

ASP.NET MVC - Models

■ Models

- Lưu trữ thông tin, trạng thái của các đối tượng, là 1 lớp được ánh xạ từ 1 bảng trong **CSDL**.
- Chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất database, các Class, hàm xử lý..

Ví dụ: lớp *Product* được sử dụng để mô tả dữ liệu từ bảng *Products*, bao gồm *ProductID*, *OrderDate*

ASP.NET MVC - Views

■ Views

- Chịu trách nhiệm hiển thị các thông tin lên cho người dùng thông qua giao diện.
- Chứa các đối tượng GUI(Textbox, images...).
- Các thông tin cần hiển thị được lấy từ thành phần **Models**.

■ Ví dụ: Đối tượng *Product* có "*Edit*" view bao gồm các textboxes, các dropdowns và checkboxes để chỉnh sửa các thuộc tính của sản phẩm

■

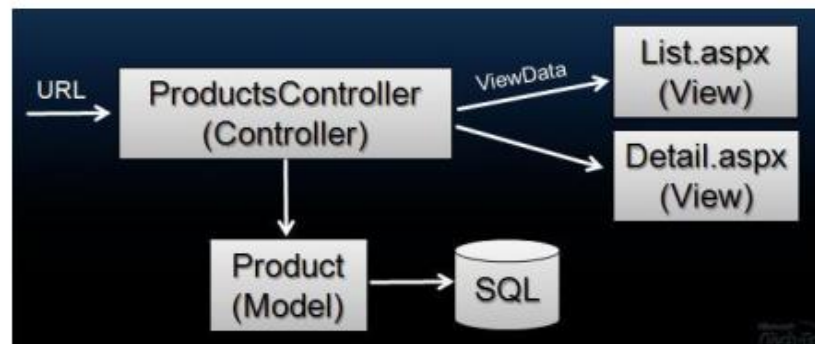
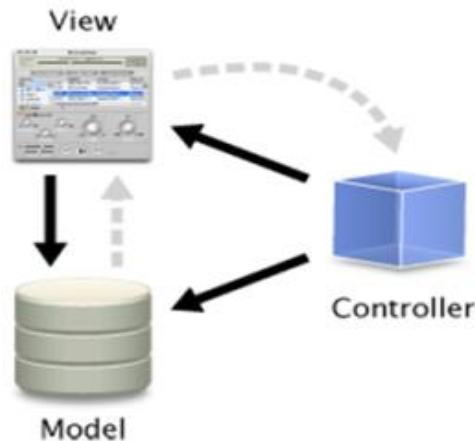
ASP.NET MVC - Controllers

■ Controllers

- Xử lý các tác động về mặt giao diện, các thao tác đối với models, và chọn view để hiển thị ra màn hình.
- Điều hướng các yêu cầu từ người dùng và gọi phương thức xử lý
- Trong **MVC**, view chỉ có tác dụng hiển thị giao diện, còn điều khiển vẫn do **Controllers** đảm trách.

Cách làm việc của MVC

- User tương tác với View, bằng cách click vào button, gửi yêu cầu đi.
- Controller nhận và điều hướng đến phương thức xử lý ở Model.
- Model nhận thông tin và thực thi các yêu cầu, View sẽ nhận kết quả từ Model và hiển thị lại cho người dùng.



Ưu điểm MVC

- Thể hiện tính chuyên nghiệp trong lập trình, PTTK.
- Được chia thành các thành phần độc lập nên giúp phát triển ứng dụng nhanh, dễ nâng cấp, bảo trì..
- Ứng dụng tạo ra chạy ổn định trên Windows
- Đáp ứng nhiều loại thiết bị truy cập
- An toàn, dễ tích hợp

Nhược điểm MVC

- 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.

Web sử dụng MVC

- Dễ dàng quản lý sự phức tạp của ứng dụng bằng cách chia ứng dụng thành Model, View, Controller
- Không sử dụng view state hoặc server-based form. Điều này tốt cho những lập trình viên muốn quản lý hết các khía cạnh của một ứng dụng.
- Sử dụng mẫu Front Controller, mẫu này giúp quản lý các requests (yêu cầu) chỉ thông qua một Controller. Hỗ trợ tốt hơn cho mô hình phát triển ứng dụng hướng kiểm thử (TDD)
- Hỗ trợ tốt cho các ứng dụng được xây dựng bởi những đội có nhiều lập trình viên và thiết kế mà vẫn quản lý được tính năng của ứng dụng

Thanks for your attention!