

Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology



WEB BÁN HÀNG ONLINE

Nhóm 32 – Môn Thực Tập Công Nghệ Phần Mềm - Tuần 4

GVHD: Nguyễn Hữu Vũ
Nguyễn Phương Vương – 1614186
Đoàn Văn Tuyển – 1613942
Trần Thị Kim Ngân – 1612176
Chống Đức Anh – 1610032
Mai Văn Hào – 1610878



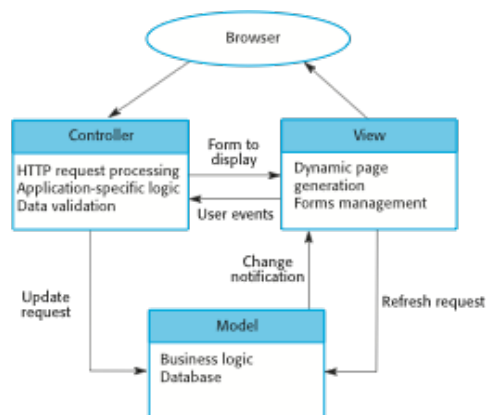
Ngày	Bản	Công việc	Người thực hiện
27/3/2019	4.0	Tạo báo cáo, phân công công việc, mô tả các chức năng cho các thành viên.	Nguyễn Phương Vương
29/3/2019	4.1	- Thực hiện tổng hợp, hệ thống lại các báo cáo của các thành viên.	Nguyễn Phương Vương
		- Mô tả đặc điểm, tích chất, ví dụ cụ thể của Controller.	Trần Thị Kim Ngân
		- Mô tả đặc điểm, tích chất, ví dụ cụ thể của Modal.	Chống Đức Anh
		- Mô tả đặc điểm, tích chất, ví dụ cụ thể của MVC.	Mai Văn Hào
		- Mô tả đặc điểm, tích chất, ví dụ cụ thể của View.	Đoàn Văn Tuyển
30/3/2019	4.2	- Chỉnh sửa tổng hợp các nhiệm vụ.	Nguyễn Phương Vương

1. Giới thiệu

Qua việc quan sát và tìm hiểu về các nhu cầu trong cuộc sống, nhóm đã nhận ra được tầm quan trọng của trang web đối với các lĩnh vực khác nhau như kinh doanh, học tập, sức khỏe, giải trí, ... Đặc biệt, đối với kinh doanh, web giúp người bán hàng và người mua dễ dàng kết nối được với nhau, dễ thanh toán và giao nhận sản phẩm. Do đó, nhóm quyết định tập trung xây dựng một trang web bán hàng với hiệu suất tốt, đảm bảo được các nhu cầu cơ bản của người dùng.

2. MVC

Name	MVC(Model-View-Controller)
Description	MVC là viết tắt của Model – View – Controller. Là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. Nói cho dễ hiểu, nó là mô hình phân bố source code thành 3 phần, mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác.
Example	Hình 1.1 ở dưới biểu diễn kiến trúc một hệ thống ứng dụng web được sử dụng mẫu MVC
When used	Được sử dụng khi có nhiều cách để xem và tương tác với dữ liệu. Cũng như được sử dụng khi các yêu cầu về tác tác và trình bày dữ liệu chưa biết có thể xảy ra.
Advantages	Thể hiện tính chuyên nghiệp trong lập trình, phân tích thiết kế. Do được chia thành cá thành phần độc lập nên giúp phát triển ứng dụng nhanh, đơn giản, dễ nâng cấp, bảo trì..
Disadvantages	Đối với dự án nhỏ việc áp dụng mô hình MVC gây công kênh, tốn thời gian trong quá trình phát triển. Tốn thời gian trung chuyển dữ liệu của các thành phần.



Hình 1.1

3. Các thành phần trong MVC

3.1. Model

3.1.1. Khái niệm Model

Model : là nơi chứa những nghiệp vụ tương tác với dữ liệu hoặc hệ quản trị cơ sở dữ liệu (mysql, mssql...); nó sẽ bao gồm các class/function xử lý nhiều nghiệp vụ như kết nối database, truy vấn dữ liệu, thêm – xóa – sửa dữ liệu... và gửi trả lại phản hồi dựa trên hướng dẫn của “Controller”.

3.1.2. Đặc điểm của Model

- Model là thành phần chủ yếu được sử dụng để thao tác xử lý dữ liệu. Trong các framework, Model vẫn thường sử dụng theo phương thức Active Record. Một trong những design pattern. Chúng có tác dụng rút ngắn thời gian viết câu truy vấn cho người sử dụng. Biến những câu truy vấn phức tạp trở nên gần gũi và thân thiện với người sử dụng thông qua các thư viện được định nghĩa sẵn.

- Model thường sẽ là các phương thức có trách nhiệm xử lý các tác vụ như: select, insert, update, delete các record trong database. Ứng với các lấy dữ liệu, model thường sử dụng mảng để gửi trả kết quả về. Vì mảng có thể cho phép model lưu trữ nhiều thông tin hơn, nên thường các record khi bóc tách chúng sẽ mang các dữ liệu của database một cách chi tiết.

- Khi sử dụng models, ta cũng cần tuân theo nguyên tắc chính của chúng là không xuất giá trị trực tiếp trong model. Mà tất cả những dữ liệu ấy, phải đưa vào mảng và trả về theo phương thức. Và tiếp tục ở view ta sẽ sử dụng nó để lấy dữ liệu ra.

3.1.3. Ví dụ về Model

```
function getuser($id){  
    $sql="select * from info where id='$id'";  
    $this->query($sql);  
    $row=$this->fetch();  
    return $row;  
}
```

- Đây là một function trả về thông tin của user thông qua id của họ, khi Controller gửi yêu cầu đến Model, function sẽ được thực thi và gửi dữ liệu về lại Controller.

3.2. View

3.1.1. Khái niệm View

- View : là nơi chứa những giao diện như một nút bấm, khung nhập, menu, hình ảnh... nó đảm nhiệm nhiệm vụ hiển thị dữ liệu và giúp người dùng tương tác với hệ thống.

3.1.2. Đặc điểm của View

- View chịu trách nhiệm định dạng lại dữ liệu được truyền ra từ model. Dữ liệu được truyền ra có thể có nhiều định dạng khác nhau tùy vào cách người lập trình xử lý như là xml, json, array,.. Lưu ý là bạn đừng nên nhập nhằng, khó hiểu giữa view và template. Khi nhận được dữ liệu với định dạng nhất định từ hệ thống, tiếp tục view sẽ làm thao tác chuyển đổi dữ liệu thành một cấu trúc html cho người dùng thấy được. Thông thường 1 controller sẽ chỉ định gửi dữ liệu đến 1 view cụ thể. Chẳng hạn với công việc hiển thị tài khoản người dùng, thì Controller “display

account” sẽ gọi đến lớp view “display account”. Tại đây, lớp view sẽ sử dụng 1 template có sẵn trong hệ thống để render ra các trang html. Mẫu template có sẵn này xử dụng lại những phần cố định của website như header, footer, menu,...

3.1.3. Ví dụ về View

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8"/>
  <title>Chi tiet hoc sinh</title>
</head>
<body>
  <h2>Chi tiet hoc sinh:</h2>
  <?php
    echo "<p><b>" . $student->name . "</b></p>";
    echo "<p> Age: " . $student->age . ", School: " . $student->university . "</p><br>";
  ?>
  <p><a href="javascript:history.back()">Back</a></p>
</body>
</html>
```

Đoạn code về View – chi tiết học sinh.

- Controller sẽ truyền các biến dữ liệu vào các View để chúng có thể dùng các dữ liệu này để hiển thị lên. Việc này được thực hiện bằng cách nhúng mã PHP vào trong mã HTML.

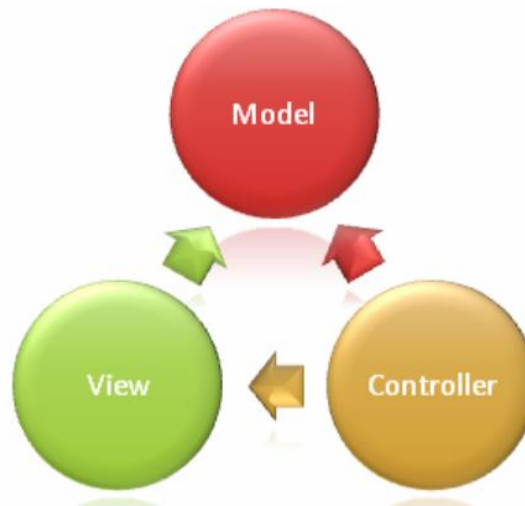
3.3. Controller

3.1.1. Khái niệm Controller

- Là phần điều hướng các request tới những tác vụ tương ứng. Controller là một phần không thể thiếu ở bất cứ framework nào. Vì nó có trách nhiệm gọi và nhận request từ hệ thống tới người sử dụng.

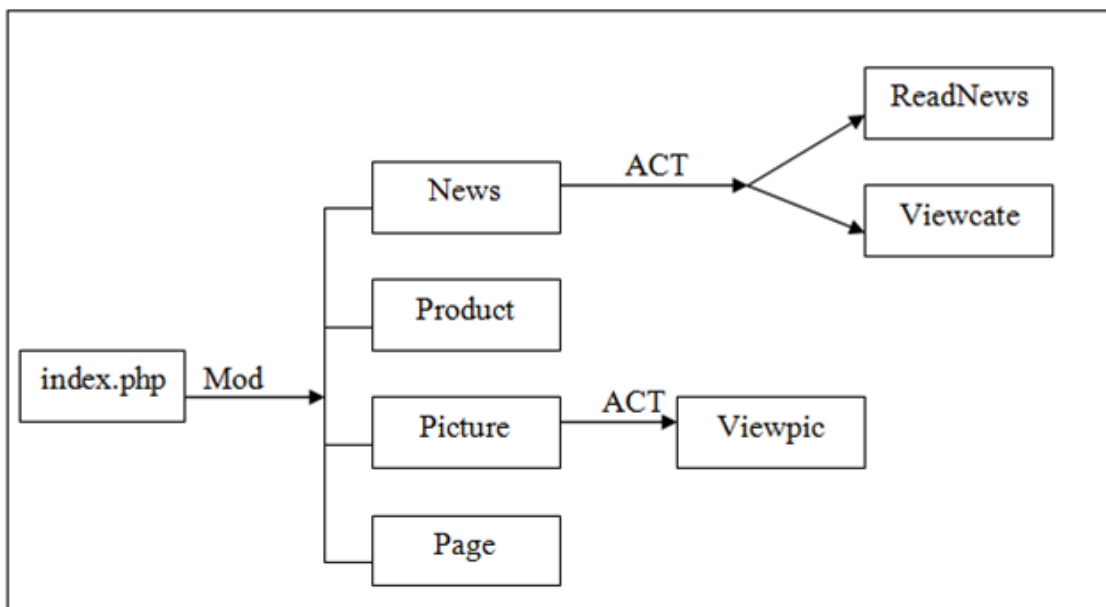
3.1.2. Đặc điểm của View

- Controller đóng vai trò trung gian giữa Model và View. Nó có nhiệm vụ tiếp nhận yêu cầu từ client sau đó xử lý request, load model tương ứng và gửi data qua view tương ứng rồi trả kết quả về cho client.



- Như vậy trên toàn ứng dụng của ta, tất cả các request đều sẽ phải đi tới controller. Và tại đây, ứng với các tham số người sử dụng truyền mà ta đưa họ đến một tác vụ nào đó trên ứng dụng.
- Tại các tác vụ này, chúng sẽ thông qua lớp model để làm việc và trả kết quả trở về controller. Cuối cùng controller sẽ đẩy dữ liệu thao tác tới view. View là thành phần cuối cùng mà người sử dụng nhận được khi họ gửi request tới ứng dụng. Có thể hiểu controller, giống với kỹ thuật đa cấp với các tác vụ chạy ứng dụng phân cấp theo từng nhánh riêng biệt như: Module, action,....

3.1.3. Ví dụ về Controller

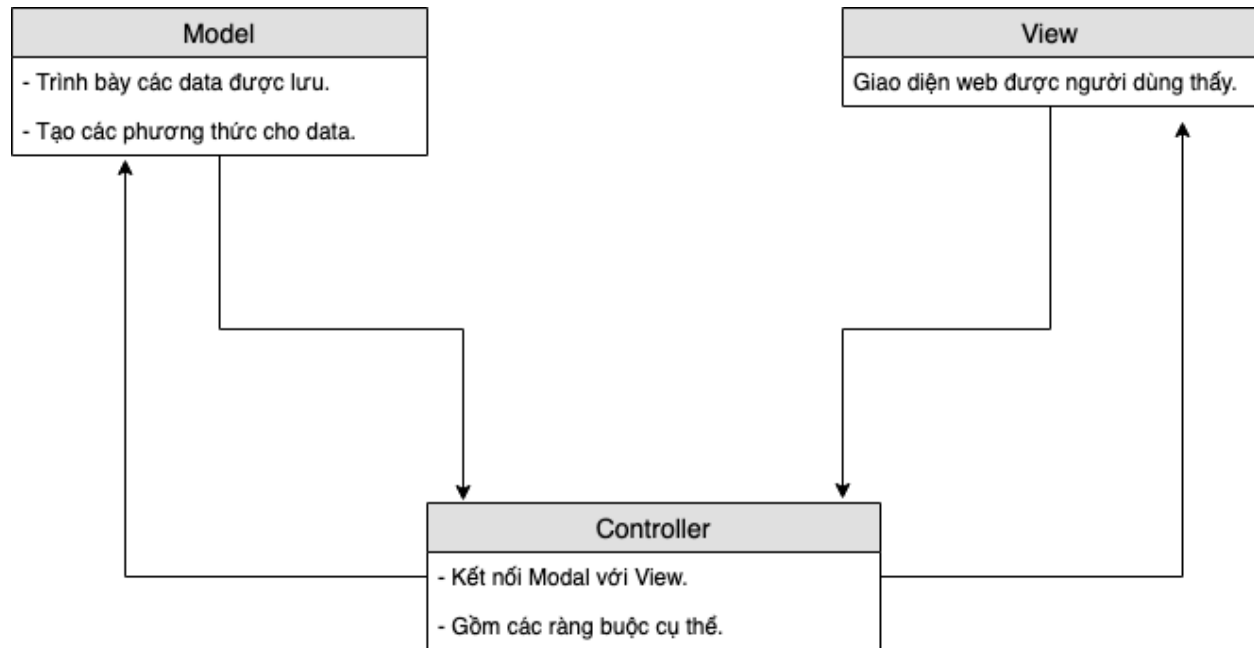


- Qua hình vẽ này, ta hiểu rằng. Để có thể thao tác với các action (hành động) ta cần phải đi qua file index.php. Lúc này file index đóng vai trò như một controller được dùng để điều hướng các

request.

- Ứng với các request thì nó sẽ trả về một controller khác để xử lý tác vụ một cách cụ thể. Tại controller con, nó sẽ gọi các action riêng biệt.
- Ở đó, thông qua các action mà nó gọi tới các file xử lý giản đơn.

4. Mô hình MVC trong hệ thống web



- Model:

+ Chứa các data và phương thức của nó nếu cần thiết.

+ Ví dụ: có class product chứa các thông tin cơ bản là id, name, price và phương thức thao tác của nó như save(), fetchAll().

- View:

+ Chứa các giao diện mà người dùng thấy được.

+ Gồm các logic về việc gửi và nhận dữ liệu.

+ Ví dụ: dùng các template engine như EJS để thao tác gửi và nhận dữ liệu.

- Controller:

+ Chứa các ràng buộc cụ thể. Cụ thể, các đường link mà người dùng truy cập được Controller xử lý. Controller cần những phương thức từ Model để xử lý.

+ Ví dụ: người dùng vào link home, controller được gọi và xử lý link bằng phương thức get. Các thông tin, thao tác product được tạo trong Model. Model lấy dữ liệu từ database rồi truyền vào lại cho Controller. Từ đó, Controller trả ngược về cho người dùng.

5. Kế hoạch phát triển

Sau khi thực hiện xong các chức năng cơ bản của website. Nhóm cố gắng tiếp tục phát triển web đạt hiệu suất tốt hơn (Chẳng hạn: web quản lý hơn 1000 khách và không bị sập khi nhiều truy cập



cùng lúc). Từ đó, nhóm sẽ có khả năng định hình hướng phát triển một sản phẩm thực tế phục vụ cho khách hàng sau này.

Tài liệu tham khảo