TRƯỜNG ĐẠI HỌC CÔNG NGHỆ GIAO THÔNG VẬN TẢI

**KHOA CÔNG NGHỆ THÔNG TIN**

A logo of a university of transport technology

Description automatically generated**-----o0o-----**

**MÔN HỌC**

**KIẾN TRÚC VÀ THIẾT KẾ PHẦN MỀM**

**Tên đề tài :**

**Tìm hiểu về mô hình MVC**

|  |  |
| --- | --- |
| **GIẢNG VIÊN HƯỚNG DẪN:** | **NGUYỄN VĂN CƯỜNG** |
| **NHÓM SINH VIÊN SỐ 5:** | **PHAN MINH KHOA**  **NGUYỄN VĂN KHÁNH**  **NGUYỄN MINH VŨ**  **VŨ DOÃN HÀ**  **NGUYỄN MINH HIẾU**  **LƯƠNG MẠNH TIẾN** |
| **LỚP:** | **72DCHT21** |

**HÀ NỘI 2-2024**

**MỤC LỤC**

[**MỞ ĐẦU** 2](#_Toc160011476)

[**I. GIỚI THIỆU MÔ HÌNH MVC** 3](#_Toc160011477)

[**1.1. Định nghĩa Mô hình MVC** 3](#_Toc160011478)

[**1.2. Lịch sử phát triển và nguồn gốc của MVC** 4](#_Toc160011479)

[**II. CÁC THÀNH PHẦN CỦA MÔ HÌNH MVC VÀ LUỒNG ĐI CỦA MVC** 5](#_Toc160011480)

[**2.1.** **Model** 5](#_Toc160011481)

[**2.2.** **View** 5](#_Toc160011482)

[**2.3.** **Controller** 6](#_Toc160011483)

[**2.4.** **Luồng tương tác giữa các thành phần MVC** 6](#_Toc160011484)

[**III. ƯU ĐIỂM VÀ NHƯỢC ĐIỂM CỦA MÔ HÌNH MVC** 7](#_Toc160011485)

[**3.1. Ưu điểm** 7](#_Toc160011486)

[**3.2. Nhược điểm** 7](#_Toc160011487)

[**IV. ỨNG DỤNG THỰC TẾ CỦA MÔ HÌNH MVC** 9](#_Toc160011488)

[**V. KẾT LUẬN** 11](#_Toc160011489)

[**VI. TÀI LIỆU THAM KHẢO** 12](#_Toc160011490)

# **MỞ ĐẦU**

Trong bối cảnh ngày nay, khi sự phức tạp của ứng dụng phần mềm ngày càng gia tăng, việc chọn lựa một kiến trúc phù hợp là quyết định quan trọng đối với các nhà phát triển. Trong hành trình này, Mô hình MVC (Model-View-Controller) nổi lên như một kiến trúc mạnh mẽ và linh hoạt, đóng vai trò quan trọng trong việc tạo ra những ứng dụng phần mềm hiệu quả và dễ bảo trì.

Mục đích của đề tài này là tìm hiểu sâu sắc về Mô hình MVC và hiểu rõ về ý nghĩa của nó trong phát triển phần mềm. Chúng ta sẽ đàm phán về cơ bản của Mô hình MVC, làm thế nào nó hoạt động, và tại sao nó trở thành một trong những lựa chọn phổ biến cho việc xây dựng ứng dụng từ những dự án nhỏ đến những hệ thống lớn.

Qua việc phân tích chi tiết về cấu trúc và nguyên lý hoạt động của Mô hình MVC, chúng ta sẽ cảm nhận được lợi ích mà nó mang lại, từ khả năng tái sử dụng mã nguồn, dễ bảo trì đến sự phân chia rõ ràng giữa logic và giao diện người dùng. Đồng thời, qua việc nghiên cứu các ứng dụng thực tế và frameworks phổ biến sử dụng Mô hình MVC, chúng ta sẽ thấy rõ sức mạnh và linh hoạt của nó trong nhiều lĩnh vực phát triển phần mềm.

Cuối cùng, đề tài cũng sẽ đặt ra những câu hỏi về triển vọng của Mô hình MVC trong tương lai, liệu nó có vẫn giữ vững vị thế của mình trong ngành công nghiệp phần mềm đang ngày càng chuyển đổi và đòi hỏi sự linh hoạt và mở rộng?

# **I. GIỚI THIỆU MÔ HÌNH MVC**

## **1.1. Định nghĩa Mô hình MVC**

MVC (Model – View – Controller) là một mô hình thiết kế hay kiến trúc được sử dụng phổ biến trong kỹ thuật phần mềm. Nó có nhiệm vụ phân bổ các source code thành 3 phần chính. Mỗi thành phần có một nhiệm vụ riêng biệt và xử lý độc lập với các thành phần khác.

 Tên gọi 3 thành phần:

* **Model** (dữ liệu): Quản lí xử lí các dữ liệu.
* **View** (giao diện): Nới hiển thị dữ liệu cho người dùng.
* A diagram of a model view controller

  Description automatically generated**Controller** (bộ điều khiển): Điều khiển sự tương tác của hai thành phần Model và View.

Mô hình MVC (MVC pattern) thường được dùng để phát triển giao diện người dùng. Nó cung cấp các thành phần cơ bản để thiết kế một chương trình cho máy tính hoặc điện thoại di động, cũng như là các ứng dụng web.

## **1.2. Lịch sử phát triển và nguồn gốc của MVC**

MVC được tiến sĩ Trygve Reenskaug đưa vào ngôn ngữ lập trình Smalltalk-76 khi ông đến trung tâm Nghiên cứu Xerox Palo Alto (PARC) vào giữa năm 1970. Sau đó, việc triển khai trở nên phổ biến trong các phiên bản khác của Small- Talk. Năm 1988, các bài báo “The Journal of Object Technology” – JOT mang lại bước tranh toàn cảnh về MVC mang liệu sự hiệu quả tốt nhất.

# **II. CÁC THÀNH PHẦN CỦA MÔ HÌNH MVC VÀ LUỒNG ĐI CỦA MVC**

A diagram of a computer system

Description automatically generated

* 1. **Model**

Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller. Model thể hiện dưới hình thức là một cơ sở dữ liệu hoặc có khi chỉ đơn giản là một file XML bình thường. Model thể hiện rõ các thao tác với cơ sở dữ liệu như cho phép xem, truy xuất, xử lý dữ liệu, …

* 1. **View**

Đây là phần giao diện (theme) dành cho người dùng. Nơi mà người dùng có thể lấy được các thông tin dữ liệu của MVC thông qua thao tác truy vấn như tìm kiếm hoặc thông qua các website.

Thông thường, các ứng dụng web sẽ sử dụng MVC View như một phần của hệ thống, nơi các thành phần HTML được tạo ra. Bên cạnh đó, View cũng có chức năng ghi nhận hoạt động của người dùng để tương tác với phần Controller.

Tuy nhiên, View lại không có mối quan hệ trực tiếp với Controller. Đồng thời, nó cũng không lấy được dữ liệu từ Controller mà chỉ có chức năng hiển thị yêu cầu và chuyển cho Controller.

* 1. **Controller**

Bộ phận này có nhiệm vụ xử lý các yêu cầu của người dùng đưa đến thông qua View. Từ đó, C sẽ đưa ra những dữ liệu phù hợp nhất cho người dùng. Ngoài ra, Controller còn có chức năng kết nối với model.

* 1. **Luồng tương tác giữa các thành phần MVC**

Bạn có thể hình dung, khi một yêu cầu từ máy client gửi tới server, Controller sẽ thực hiện nhiệm vụ của mình đó là tiếp nhận và xử lý yêu cầu. Trong trường hợp cần thiết, nó có thể liên hệ Model – bộ phận làm việc với database để hỗ trợ.

A diagram of a model-view controller pattern

Description automatically generatedKhi xử lý xong yêu cầu, kết quả sẽ được trả về View. Tại View sẽ tạo mã HTML thành giao diện và trả về hiển thị trên trình duyệt.

# **III. ƯU ĐIỂM VÀ NHƯỢC ĐIỂM CỦA MÔ HÌNH MVC**

## **3.1. Ưu điểm**

**Nhẹ, tiết kiệm băng thông:** MVC không sử dụng viewstate nên khá tiết kiệm diện tích băng thông. Khi sử dụng, người dùng có thể sử dụng ứng dụng trên web cần tương tác gửi và nhận dữ liệu một cách liên tục. Do đó, việc giảm băng thông giúp cho website hoạt động tốt và ổn định hơn.

**Kiểm tra dễ dàng:** Với MVC, bạn có thể dễ dàng kiểm tra, rà soát lỗi phần mềm trước khi tới tay người tiêu dùng, đảm bảo chất lượng và độ uy tín cao hơn.

**Chức năng control:** Trên các nền website thì ngôn ngữ lập trình như CSS, HTML, Javascript có một vai trò vô cùng quan trọng. Việc sử dụng mô hình MVC sẽ giúp bạn có một bộ control ưu việt trên nền tảng các ngôn ngữ hiện đại với nhiều hình thức khác nhau.

**View và size:** View sẽ là nơi lưu trữ các dữ liệu. Càng nhiều yêu cầu được thực hiện thì kích thước càng tệp càng lớn. Khi đó, đường truyền mạng cũng giảm tốc độ load. Việc sử dụng mô hình MVC sẽ giúp bạn tiết kiệm được diện tích băng thông một cách tối ưu.

**Chức năng Soc (Separation of Concern):** Chức năng này cho phép bạn phân tách rõ ràng các phần như Model, giao diện, data, nghiệp vụ.

**Tính kết hợp:** Việc tích hợp ở mô hình MVC cho phép bạn thoải mái viết code trên nền tảng website. Khi đó, server của bạn sẽ được giảm tải khá nhiều.

**Đơn giản:** Đây là một mô hình với kết cấu tương đối đơn giản. Dù bạn không có quá nhiều chuyên môn cũng có thể sử dụng được.

## **3.2. Nhược điểm**

Ngoài các ưu điểm nổi bật bên trên, mô hình MVC cũng có những hạn chế nhất định, cụ thể như:

Cần nhiều thời gian để tìm hiểu nếu muốn áp dụng đúng đắn.

Tốn kém thời gian và công sức để quản lý tổ chức file.

Xây dựng quy trình tương đối phức tạp, bởi vậy không cần thiết áp dụng mô hình này cho các dự án nhỏ.

# 

# **IV. ỨNG DỤNG THỰC TẾ CỦA MÔ HÌNH MVC**

Mô hình MVC được ứng dụng trong nhiều ngôn ngữ lập trình khác nhau, nhưng phổ biến nhất là ứng dụng ASP.NET MVC hay PHP MVC.

MVC đang là mô hình được ứng dụng rất nhiều trong lập trình.

Hệ thống MVC phát triển tốt sẽ cho phép phát triển front – end, back – end cùng trên hệ thống mà không có sự can thiệp, chia sẻ, chỉnh sửa các tập tin trong khi một hoặc hai bên vẫn đang làm việc.

Việc sử dụng mô hình tương đối đơn giản. Chỉ cần hiểu rõ quy trình vận hành, nắm được các chức năng của từng bộ phận thì việc triển khai mô hình MVC tương đối dễ dàng.

Dưới đây là một số ví dụ về việc sử dụng **mô hình MVC** trong phát triển ứng dụng:

* **Ứng dụng Blog**: Mô hình MVC được sử dụng để phân chia logic xử lý (Controller), hiển thị giao diện (View) và lưu trữ dữ liệu (Model). Điều này giúp cho việc quản lý bài viết, tương tác với người dùng và lưu trữ dữ liệu trở nên dễ dàng và rõ ràng hơn.
* **Hệ thống quản lý nhân sự**: Mô hình MVC được áp dụng để quản lý thông tin nhân viên (Model), hiển thị danh sách nhân viên, biểu đồ và báo cáo (View). Và thực hiện các chức năng như thêm, sửa, xóa nhân viên (Controller).
* **Ứng dụng thương mại điện tử**: Mô hình MVC được sử dụng để quản lý danh mục sản phẩm (Model), hiển thị giao diện sản phẩm, giỏ hàng và thanh toán (View), và xử lý các yêu cầu của người dùng như thêm sản phẩm vào giỏ hàng, xem chi tiết sản phẩm (Controller).

# **V. KẾT LUẬN**

Tóm lại, Mô hình MVC không chỉ là một khái niệm kiến trúc mà là một công cụ thực sự hữu ích và linh hoạt trong phát triển phần mềm. Qua quá trình tìm hiểu, chúng ta có thể nhận thức rõ về ý nghĩa và lợi ích của Mô hình MVC.

Mô hình này không chỉ giúp phân chia trách nhiệm rõ ràng giữa các thành phần mà còn tạo ra sự tách biệt linh hoạt giữa logic kinh doanh và giao diện người dùng. Điều này mang lại những lợi ích quan trọng như dễ bảo trì, tái sử dụng mã nguồn, và kiểm thử hiệu quả.

Ứng dụng thực tế của Mô hình MVC đã được chứng minh thông qua sự thành công của nhiều frameworks và dự án phần mềm nổi tiếng. Đối với các lập trình viên và nhóm phát triển, Mô hình MVC không chỉ là một nguyên lý thiết kế, mà còn là một hướng dẫn hữu ích giúp họ tổ chức mã nguồn một cách hiệu quả.

Mặc dù Mô hình MVC đã tồn tại từ lâu, nhưng với sự tiến bộ của công nghệ, nó vẫn giữ vai trò quan trọng trong việc phát triển phần mềm hiện đại. Tiếp tục nghiên cứu và áp dụng Mô hình MVC sẽ giúp chúng ta hiểu rõ hơn về tiềm năng và giới hạn của nó trong các dự án tương lai.

# **VI. TÀI LIỆU THAM KHẢO**

1. <https://viblo.asia/p/tat-tan-tat-ve-mo-hinh-mvc-Rk74avjAJeO>
2. <https://mona.media/mo-hinh-mvc/>
3. <https://codegym.cc/vi/groups/posts/vi.303.phan-7-gioi-thieu-mau-mvc-model-view-controller->