Họ và tên: Đặng Việt Anh Mã sinh viên: 20224066

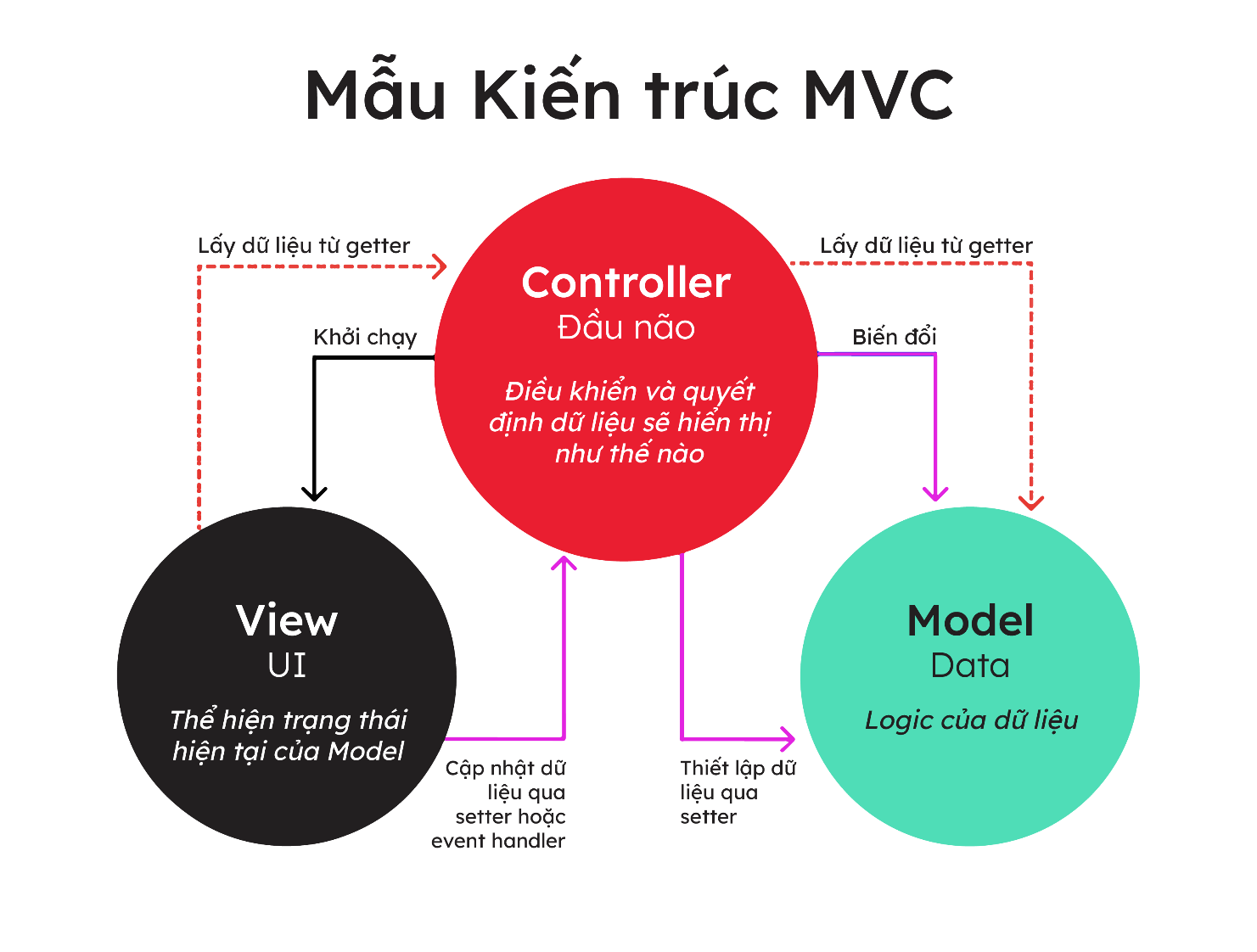
Lớp: DCCNTT13.10.21

**Báo Cáo Chi Tiết Về Mô Hình MVC**

***Giới Thiệu Về Mô Hình MVC***

Mô hình MVC (Model-View-Controller) là một mẫu kiến trúc phần mềm được phát triển từ những năm 1970, ban đầu cho giao diện người dùng trên máy tính để bàn, và sau này trở thành tiêu chuẩn trong phát triển web hiện đại. Nó được Trygve Reenskaug giới thiệu lần đầu vào năm 1979 và phổ biến qua ngôn ngữ Smalltalk vào năm 1987. MVC chia ứng dụng thành ba thành phần chính, mỗi thành phần có vai trò riêng, giúp tách biệt logic kinh doanh, giao diện, và xử lý đầu vào, từ đó tăng tính mô-đun và dễ bảo trì.

Nghiên cứu từ nhiều nguồn, như [Wikipedia MVC](https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller) và [GeeksforGeeks MVC Design](https://www.geeksforgeeks.org/mvc-design-pattern/), cho thấy MVC không chỉ áp dụng cho desktop mà còn được sử dụng rộng rãi trong các framework web như ASP.NET Core , Ruby on Rails, và Django. Các nguồn tiếng Việt như [Vietnix MVC](https://vietnix.vn/tim-hieu-mo-hinh-mvc-la-gi/) và [Viblo MVC](https://viblo.asia/p/tat-tan-tat-ve-mo-hinh-mvc-Rk74avjAJeO) cũng nhấn mạnh vai trò của MVC trong lập trình web tại Việt Nam, đặc biệt trong các dự án cần mở rộng và bảo trì lâu dài.



Hình 1 . Kiến trúc mô hình MVC

***Đặc Điểm Chính Của Mô Hình MVC***

Dưới đây là bảng tổng hợp các đặc điểm chính của từng thành phần trong MVC, dựa trên các nguồn tham khảo:

| Thành Phần | Đặc Điểm Chính | Ví Dụ |
| --- | --- | --- |
| Model | - Quản lý dữ liệu và logic kinh doanh, độc lập với giao diện. | - Trong ứng dụng danh sách công việc, Model định nghĩa "nhiệm vụ" và "danh sách". |
|  | - Đảm bảo dữ liệu nhất quán, thường tương ứng với cơ sở dữ liệu (database table). | - Truy xuất dữ liệu từ database, xử lý logic nghiệp vụ như tính toán giá. |
| View | - Hiển thị giao diện người dùng, có thể có nhiều View cho cùng một Model. | - HTML templates trong Rails/Django, hiển thị danh sách công việc trên trình duyệt. |
|  | - Nhận dữ liệu từ Controller và hiển thị, không xử lý logic đầu vào (trong một số triển khai). | - Các nút bấm, menu, hình ảnh trên giao diện. |
| Controller | - Xử lý đầu vào từ người dùng, kết nối Model và View. | - Trong Rails, nhận yêu cầu từ router, xử lý dữ liệu và chuẩn bị phản hồi. |
|  | - Chuyển đổi yêu cầu thành lệnh cho Model hoặc View, có thể phức tạp trong ứng dụng lớn. | - Xử lý sự kiện như nhấn nút, cập nhật dữ liệu và hiển thị kết quả. |

***Lợi Ích Và Ưu Điểm***

MVC mang lại nhiều lợi ích, đặc biệt trong phát triển web:

Tách Biệt Các Loại Logic: Theo [ITViec MVC](https://itviec.com/blog/mvc-la-gi/), MVC giúp tách logic nghiệp vụ (Model), logic giao diện (View), và logic đầu vào (Controller), cho phép tập trung vào từng khía cạnh mà không phụ thuộc lẫn nhau.

Phát Triển Song Song: Nhiều lập trình viên có thể làm việc cùng lúc, ví dụ, một người làm View, người khác làm Controller, như được đề cập trong [Monamedia MVC](https://monamedia.co/mvc-la-gi-ung-dung-cua-mo-hinh-mvc-trong-lap-trinh/).

Tiết Kiệm Băng Thông: Theo [Vietnix MVC](https://vietnix.vn/tim-hieu-mo-hinh-mvc-la-gi/), MVC không sử dụng viewstate, giúp giảm băng thông, cải thiện hiệu suất website, đặc biệt trong các ứng dụng cần tương tác liên tục.

Tái Sử Dụng Mã: Các thành phần có thể được tái sử dụng, như View có thể hiển thị nhiều Model khác nhau, theo [TechTarget MVC](https://www.techtarget.com/whatis/definition/model-view-controller-MVC).

Dễ Kiểm Tra: Tính mô-đun giúp dễ dàng kiểm tra đơn vị (unit testing), như được nhấn mạnh trong [FreeCodeCamp MVC](https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/).

***Thách Thức Và Hạn Chế***

Mặc dù có nhiều lợi ích, MVC cũng đối mặt với một số thách thức:

Luồng Dữ Liệu Phức Tạp: Theo [FPTShop MVC](https://fptshop.com.vn/tin-tuc/danh-gia/mo-hinh-mvc-183671), việc duy trì dòng chảy dữ liệu rõ ràng giữa Model, View, và Controller có thể khó khăn, đặc biệt trong ứng dụng lớn.

Controller Quá Tải: [TopDev MVC](https://topdev.vn/blog/kien-truc-model-view-controller/) chỉ ra rằng Controller có thể trở nên phức tạp, xử lý nhiều nhiệm vụ, dẫn đến khó bảo trì.

Hạn Chế Linh Hoạt: Một số nguồn, như [MarketEnterprise MVC](https://www.marketenterprise.vn/blog/mo-hinh-mvc-trong-lap-trinh.html), cho rằng cấu trúc rõ ràng của MVC có thể hạn chế sự sáng tạo trong một số trường hợp.

Không Hỗ Trợ Xác Thực Chính Thức: Theo [TopDev MVC](https://topdev.vn/blog/kien-truc-model-view-controller/), MVC không cung cấp hỗ trợ xác thực chính thức, đòi hỏi phải thực hiện rõ ràng.

Ứng Dụng Thực Tế

MVC được áp dụng rộng rãi trong các framework hiện đại. Ví dụ, trong ASP.NET Core

***So Sánh Với Các Mô Hình Khác***

MVC là một trong nhiều mẫu kiến trúc, và có các biến thể như MVVM (Model-View-ViewModel) hoặc MVP (Model-View-Presenter), như được đề cập trong [MDN MVC Glossary](https://developer.mozilla.org/en-US/docs/Glossary/MVC). So với các mô hình khác, MVC nổi bật nhờ tính phổ biến và hỗ trợ từ nhiều framework, nhưng cũng có thể phức tạp hơn trong một số trường hợp so với các mô hình đơn giản hơn.