



SOFTWARE ENGINEERING (CO3001)

Assignment

Restaurant Pos 2.0

(Week 4: Task 3)

Group: Tran Nhat Huy - 1913566
Tran Minh Tan - 1915076
Ho Vu Thuan -1915378
Do Chi Khai -1913769
Ha Viet Hoang -1913419



Member list:

No	Name	ID Number
1	Tran Nhat Huy	1913566
2	Tran Minh Tan	1915076
3	Ho Vu Thuan	1915378
4	Do Chi Khai	1913769
5	Ha Viet Hoang	1913419

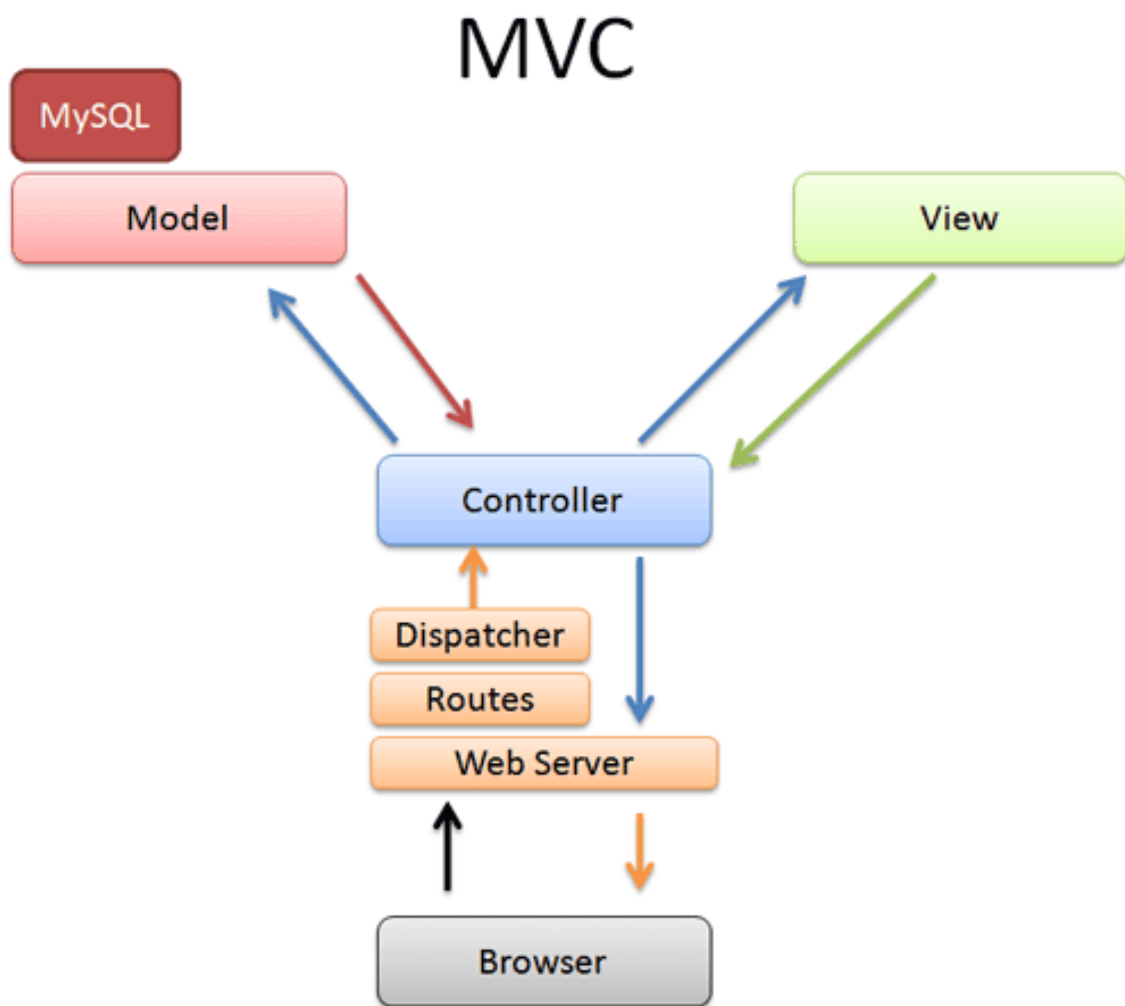
Contents

1	Mô hình MVC	3
1.1	Giới thiệu về mô hình MVC	3
1.2	Luồng đi trong mô hình MVC	4
1.3	Ưu và nhược điểm của mô hình MVC	5
	Ưu điểm	5
	Nhược điểm	5
2	Ứng dụng mô hình MVC vào POS	6
3	Component diagram	6
3.1	User management	6
3.2	Food ordering	7
3.3	Table Reservation	7
3.4	Online payment	8
4	References	9

1 Mô hình MVC

1.1 Giới thiệu về mô hình MVC

MVC là từ viết tắt bởi 3 từ **Model – View – Controller**. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. Mô hình source code thành 3 phần, tương ứng mỗi từ. Mỗi từ tương ứng với một hoạt động tách biệt trong một mô hình. Để hiểu rõ hơn, sau đây chúng ta sẽ cùng đi phân tích từng thành phần:



Mô hình MVC và các thành phần bên trong của MVC:

- **Model (M):** 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,...

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

Thông thường, các ứng dụng web 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 *Controller*. Tuy nhiên, View không có mối quan hệ trực tiếp với *Controller*, cũng không được lấy dữ liệu từ *Controller* mà chỉ hiển thị yêu cầu chuyển cho *Controller* mà thôi.

Ví dụ: Nút “delete” được tạo bởi View khi người dùng nhấn vào nút đó sẽ có một hành động trong *Controller*.

- **Controller (C):** Bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua *View*. Từ đó, C đưa ra dữ liệu phù hợp với người dùng. Bên cạnh đó, *Controller* còn có chức năng kết nối với *Model*.

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

Ví dụ: Chức năng thêm bài viết mới trong trang quản trị website. Nơi đây có 2 trường nhập về tiêu đề và nội dung bài viết, thì trong đó:

- View sẽ hiển thị ra phần nhập form tiêu đề và nội dung.
- Controller lấy dữ liệu từ 2 trường và gửi tới Model.
- Model lúc này sẽ nhận dữ liệu từ Controller để lưu trữ trong cơ sở dữ liệu.

Mỗi bộ phận thực hiện chức năng nhất định, nhưng chúng có sự thống nhất, liên kết với nhau tạo nên mô hình *MVC*.

1.2 Luồng đi trong mô hình MVC

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.

Khi 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.

1.3 Ưu và nhược điểm của mô hình MVC

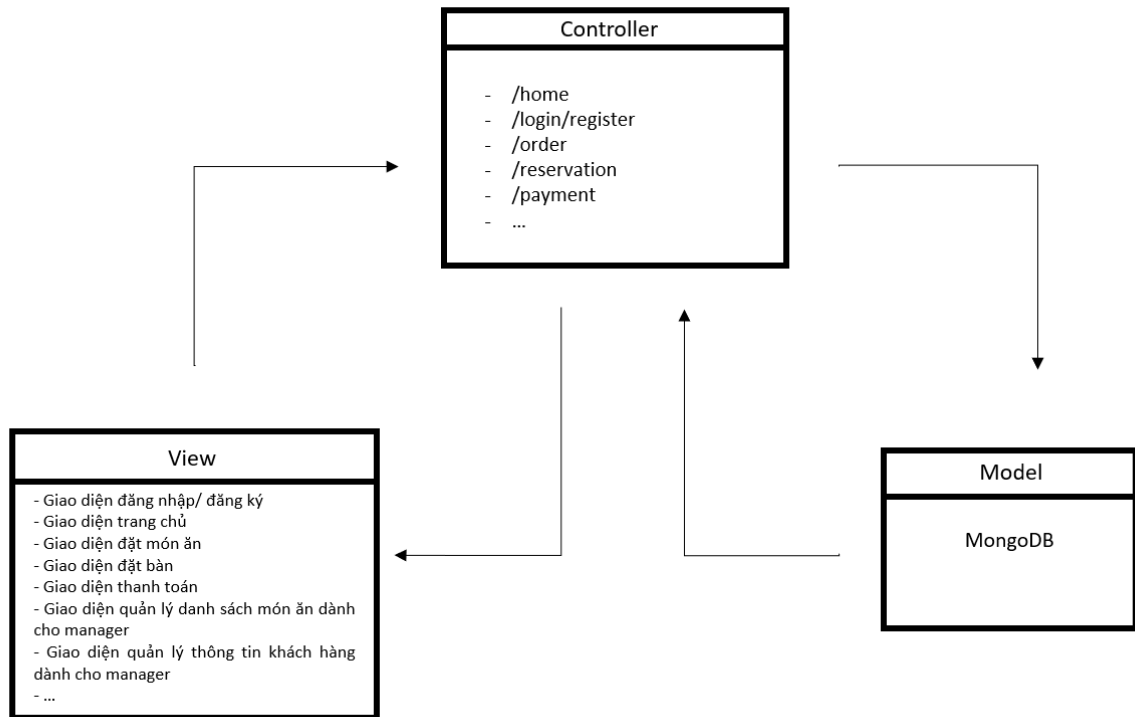
Ưu điểm

- 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.
- 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.

Nhược điểm

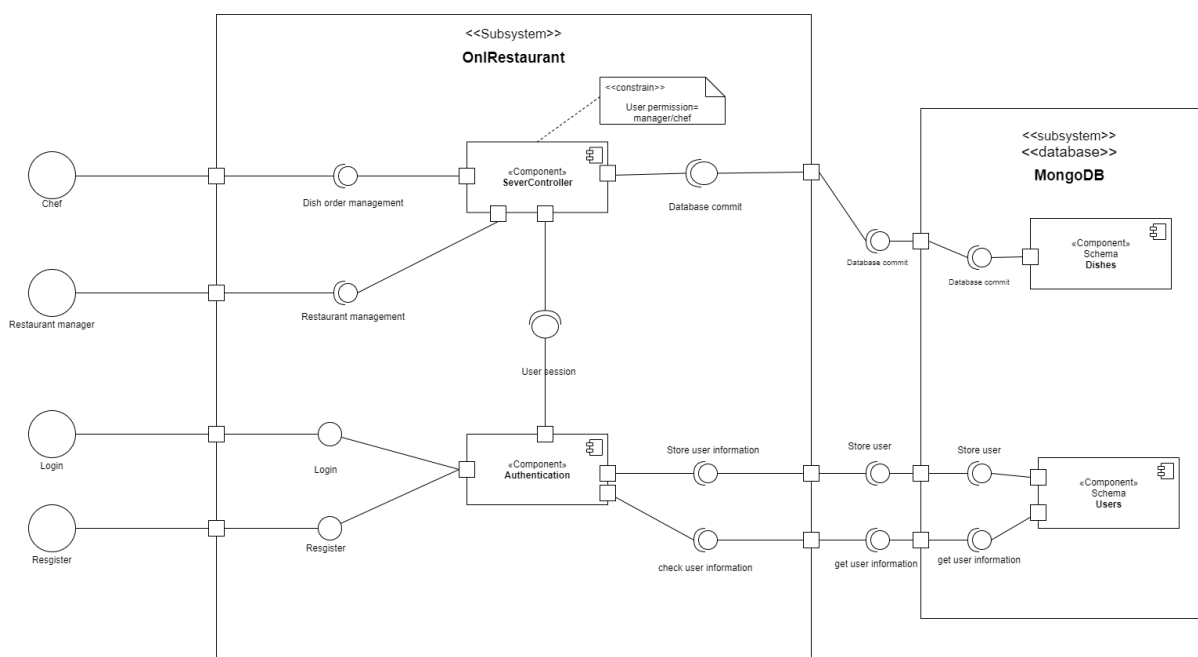
MVC thường được sử dụng vào những dự án lớn. Do đó, với các dự án nhỏ, mô hình MVC có thể gây cồng kềnh, tốn thời gian trong quá trình phát triển cũng như thời gian trung chuyển dữ liệu.

2 Ứng dụng mô hình MVC vào POS

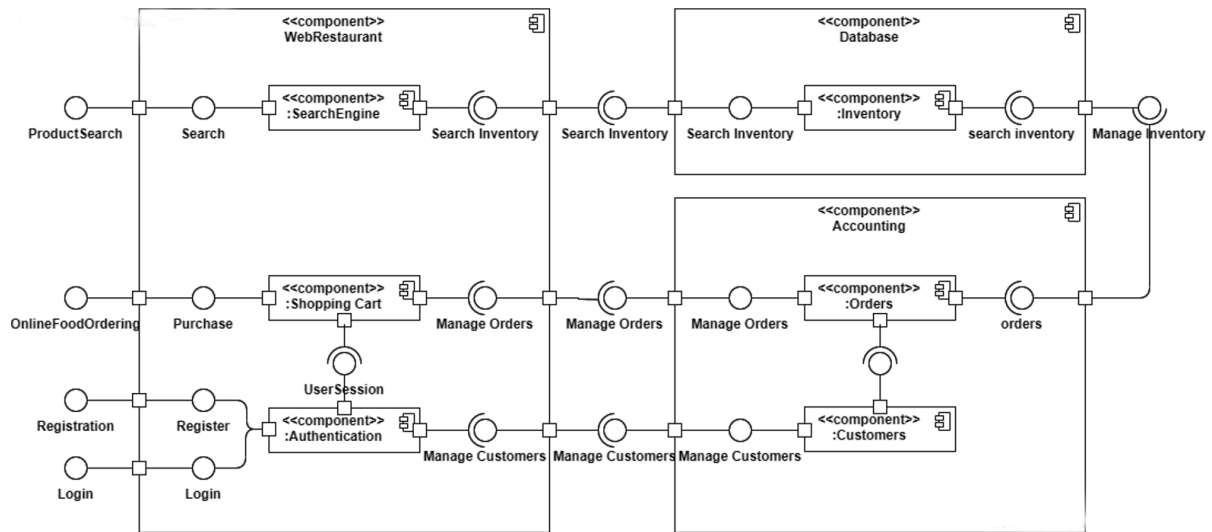


3 Component diagram

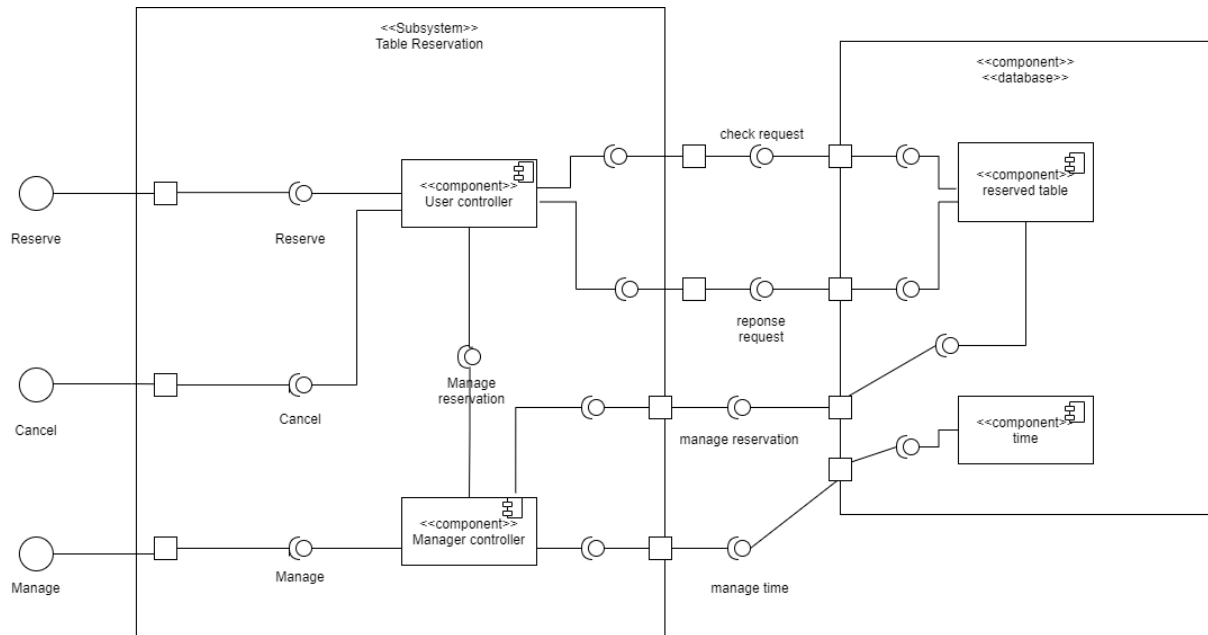
3.1 User management



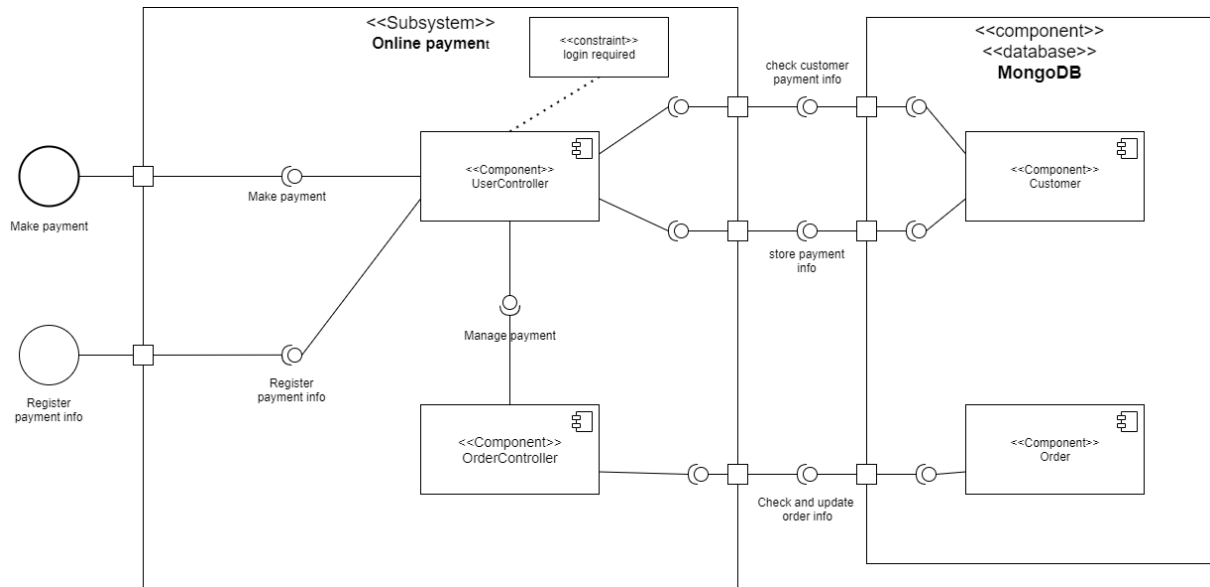
3.2 Food ordering



3.3 Table Reservation



3.4 Online payment



4 References

- [1] <https://vietnix.vn/tim-hieu-mo-hinh-mvc-la-gi/model-du-lieu>
- [2] [https://www.visual-paradigm.com/guide/uml-unified-modeling-language/...
what-is-component-diagram/](https://www.visual-paradigm.com/guide/uml-unified-modeling-language/...what-is-component-diagram/)
- [3] Một số tài liệu và báo cáo mẫu được thầy upload lên BKeL