Name	**The layered architecture pattern**

Description	Tổ chức hệ thống thành các lớp với chức năng liên quan được liên kết với mỗi lớp. Một lớp cung cấp các dịch vụ cho lớp phía trên nó nên các lớp cấp thấp nhất sẽ đại diện cho các dịch vụ cốt lõi có khả năng được sử dụng trong toàn hệ thống.
	User interface: hiển thị hệ thống bằng đồ họa để người dùng thao tác các chức năng.
	• Display các element UI của các trang khác nhau:
	  ○ Các thành phần menu, giỏ hàng,... của trang order.
	  ○ Form, button,... của trang payment.
	  ○ Form để thực hiện các hành động CRUD đối với các trang quản lý thực đơn, đơn đặt hàng,...
	  ○ Form đăng nhập hay đăng ký của người dùng.
	  ○ Hình ảnh, trạng thái, thông tin của các món ăn trong menu.
	• User tương tác với hệ thống bằng các quá trình xử lý sự kiện ở trong view trên các trang:
	  ○ Hiển thị Dropdown menu khi xử lý sự kiện click vào menu của user.
	  ○ Hiển thị slide tự động
	• Gửi các tín hiệu xử lý sự kiện cho lớp tiếp theo.
	• Truyền data vào các request của người dùng và gửi cho lớp tiếp theo
	• Khi được thông báo các thay đổi trạng thái từ lớp dưới, thực hiện truy xuất dữ liệu để thực hiện refresh lại view và update view mới.
	User interface management/ authentication and authorization: nhận các request cùng data từ lớp phía trên. Sau đó kiểm tra và xác nhận yêu cầu của request. Từ đó truyền request và data xuống lớp tiếp theo để gọi function tương ứng và thao tác với dữ liệu đó. Cuối cùng thông báo cho lớp trên khi có sự thay đổi trạng thái cùng với truyền dữ liệu đã được xử lý tương ứng với view được đề xuất.
	Core business logic/ application functionality
	System utilities:
	• Trả về reponse từ các request được gửi tới.
	• Gọi các hàm tương ứng để thực hiện các thao tác trên dữ liệu:
	  ○ CRUD: món ăn, đơn hàng, thông tin khách hàng, giao dịch thanh toán.

	◦ Sử dụng các phương thức để thực hiện các thao tác đăng ký, đăng nhập, quản lý tài khoản người dùng. • Cung cấp các phương thức tương tác với database. • Cung cấp các phương thức để xử lý gọi API để thực hiện xác thực cho việc thực hiện thanh toán với các hệ thống thanh toán. **System support (OS, database):** • Cung cấp môi trường để vận hành ứng dụng. • Cơ sở dữ liệu để lưu trữ và truy vấn dữ liệu.
Advantages	• Dễ dàng thay thế toàn bộ các lớp miễn sao giao diện được duy trì. • Có thể xây dựng các cơ sở dự phòng ở mỗi lớp để tăng độ tin cậy cũng như an toàn cho hệ thống. • Quản lý và quản trị rủi ro đơn giản do mỗi lớp có các module riêng và được phân chia rõ ràng.
Disadvantages	• Việc cung cấp sự phân tách rõ ràng giữa các lớp thường rất khó và một lớp cấp cao có thể phải tương tác trực tiếp với các lớp cấp thấp hơn thay vì thông qua lớp ngay bên dưới nó. • Hiệu năng là một vấn đề do hiệu suất giữa các lớp khác nhau là khác nhau do sự yêu cầu xử lý khác nhau. • Cần đảm bảo tương tác hiệu quả giữa các lớp.