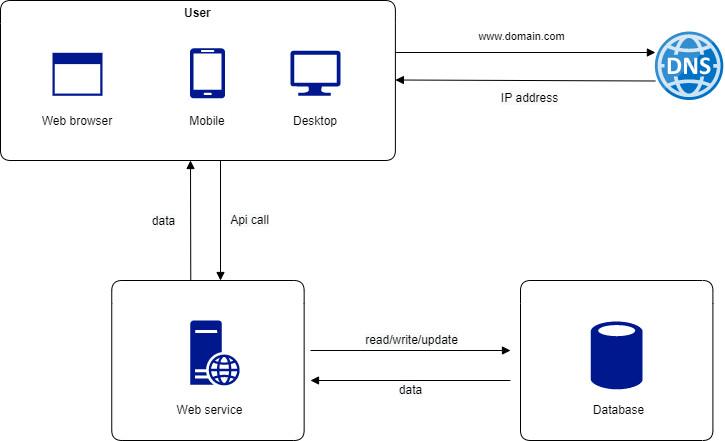
### Mô hình công nghệ (Mục 2.1)

**Kiến trúc phân tầng giao tiếp qua RESTful API**

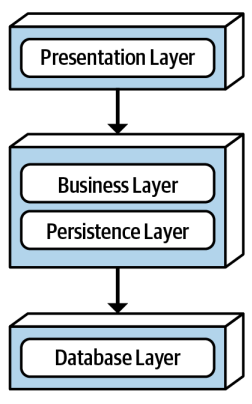


*Mô hình kiến trúc*

**Loại kiến trúc**: Kiến trúc lớp (*layered architecture*).

**Mô tả**:

* Hệ thống được xây dựng theo mỗi lớp đảm nhận chức năng riêng biệt.
* Kiến trúc không có ràng buộc về số lượng các lớp trên hệ thống, thông thường gồm 4 lớp quan trọng: *presentation, business*, *persistence* và *database*.
* Trong một số trường hợp, business và persistence layer có thể kết hợp lại thành 1 lớp chung



*Hình 1 Nguồn: Fundamentals of Software Architecture p.134*

* Về chức năng
  + Tầng presentation là tầng giao diện người dùng, có trách nhiệm hiển thị giao diện người dùng, thúc đẩy sự tương tác giữa người dùng với ứng dụng, chuyển tiếp dữ liệu nhập từ người dùng cho tầng business, hiển thị thông tin từ tầng Business. Tầng này gồm: trình duyệt web, ứng dụng điện thoại, ứng dụng máy tính.
  + Tầng business là Tầng nghiệp vụ, còn được gọi là tầng ứng dụng hoặc tầng logic, chứa các quy tắc và logic nghiệp vụ cốt lõi của ứng dụng. Nó xử lý các yêu cầu của người dùng từ tầng presentation , thực hiện các tính toán cần thiết và điều phối chức năng của ứng dụng. Cấp này chịu trách nhiệm triển khai các quy tắc kinh doanh, xác thực dữ liệu, điều phối quy trình công việc và đảm bảo tính toàn vẹn của ứng dụng. Nó hoạt động như một trung gian hòa giải giữa tầng presentation và tầng persistence.
  + Tầng persistence là tầng truy cập dữ liệu hoặc tầng dữ liệu, chịu trách nhiệm lưu trữ và truy xuất dữ liệu từ cơ sở dữ liệu. Nó xử lý việc lưu trữ và truy xuất dữ liệu ứng dụng, đảm bảo dữ liệu luôn ổn định và sẵn sàng để sử dụng trong tương lai. Tầng này tương tác với hệ thống quản lý cơ sở dữ liệu (DBMS) để thực hiện các hoạt động như truy vấn, cập nhật và xóa dữ liệu. Nó trừu tượng hóa việc lưu trữ dữ liệu cơ bản và cung cấp giao diện (interface) nhất quán cho tầng business, tránh khỏi sự phức tạp của việc lưu trữ dữ liệu
  + Tầng database là tầng cơ sở dữ liệu đề cập đến hệ thống cơ sở dữ liệu thực tế nơi lưu trữ dữ liệu của ứng dụng. Nó có thể là một hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS) như MySQL, Oracle, PostgreSQL hoặc có thể là cơ sở dữ liệu NoSQL như MongoDB hoặc Cassandra, tùy thuộc vào yêu cầu của ứng dụng. Tầng cơ sở dữ liệu chịu trách nhiệm quản lý việc lưu trữ, truy xuất và thao tác dữ liệu dựa trên các hoạt động được thực hiện bởi tầng lưu giữ. Tầng này đảm bảo tính toàn vẹn dữ liệu, thực thi các ràng buộc dữ liệu và cung cấp các cơ chế để truy cập và truy xuất dữ liệu hiệu quả.

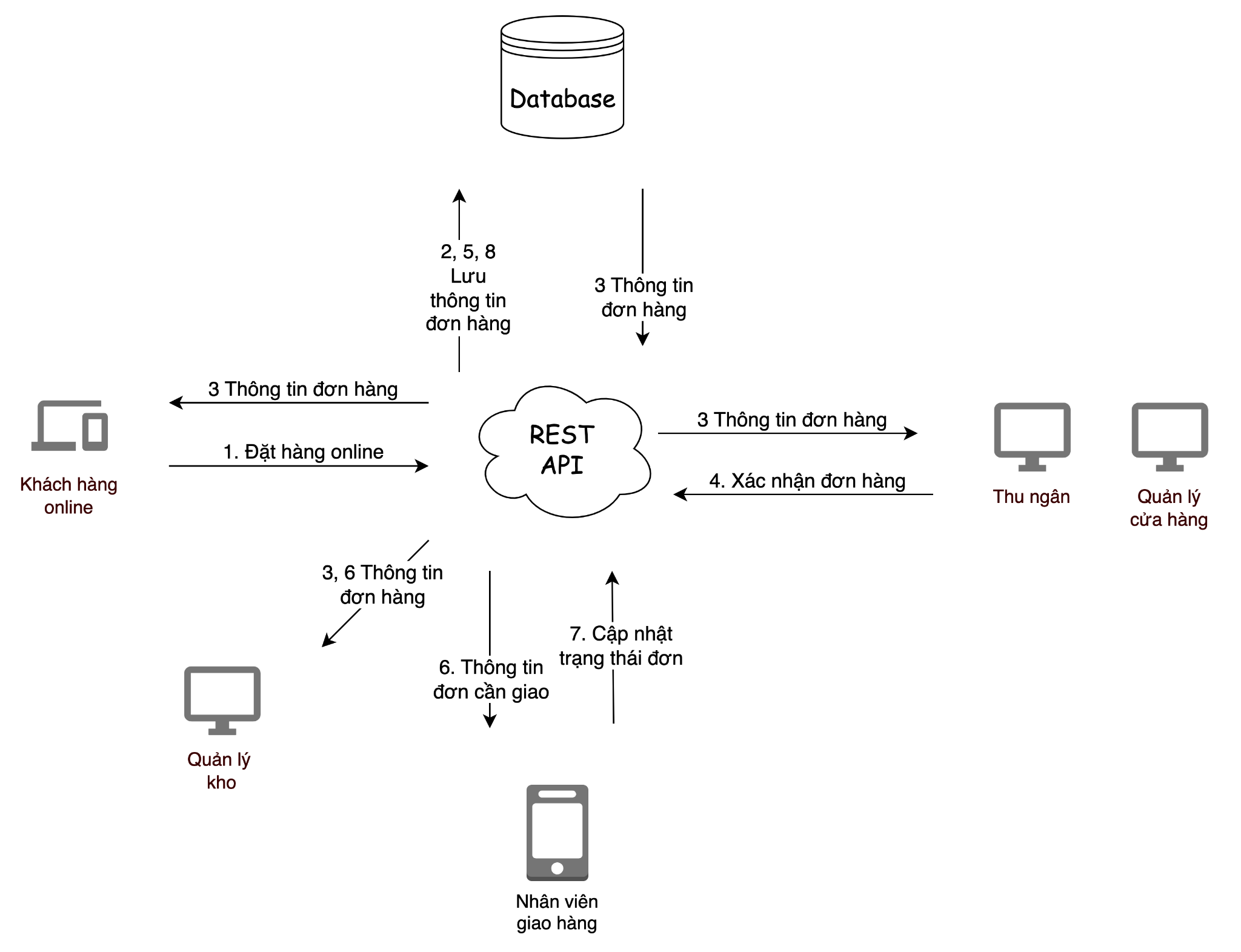
**Lý do chọn kiến trúc lớp trong việc phát triển ứng dụng**:

* Khả năng mở rộng: Kiến trúc ba tầng cho phép khả năng mở rộng bằng cách tách ứng dụng thành các tầng riêng biệt. Mỗi tầng có thể được mở rộng độc lập dựa trên nhu cầu cụ thể của hệ thống.
* Khả năng bảo trì: Mỗi tầng có thể được phát triển, thử nghiệm và duy trì độc lập, giúp cập nhật hoặc thay thế một tầng cụ thể dễ dàng hơn mà không ảnh hưởng đến toàn bộ ứng dụng.
* Khả năng sử dụng lại: Bản chất mô-đun của kiến ​​trúc ba tầng tạo điều kiện cho việc tái sử dụng mã. Các thành phần trong mỗi tầng có thể được thiết kế để có thể tái sử dụng trên các phần khác nhau của ứng dụng hoặc thậm chí trong các ứng dụng khác, giúp tăng hiệu quả phát triển và giảm nỗ lực.
* Bảo mật: Với kiến ​​trúc ba tầng, các biện pháp bảo mật có thể được triển khai ở mỗi tầng. Tầng trình bày có thể xử lý xác thực và ủy quyền của người dùng, tầng doanh nghiệp có thể thực thi các quy tắc kinh doanh và áp dụng kiểm tra bảo mật, còn tầng lưu giữ có thể bảo vệ cơ sở dữ liệu khỏi bị truy cập trái phép.

**Hạn chế của kiến trúc**:

* Độ phức tạp: Kiến trúc ba tầng có độ phức tạp cao hơn so với các kiến ​​trúc đơn giản hơn. Việc tách biệt các mối quan tâm và liên lạc giữa các cấp đòi hỏi phải có sự thiết kế và phối hợp cẩn thận. Do đó có thể dẫn đến tăng thời gian và nỗ lực phát triển.
* Đòi hỏi nhiều nỗ lực phát triển: Phát triển một ứng dụng dựa trên kiến ​​trúc ba tầng có thể đòi hỏi nhiều nỗ lực phát triển hơn so với các kiến ​​trúc đơn giản hơn. Việc thiết kế, triển khai và thử nghiệm nhiều tầng có thể tốn nhiều thời gian và tài nguyên hơn.
* Độ phức tạp của việc triển khai: Việc triển khai và quản lý kiến ​​trúc ba tầng có thể phức tạp hơn việc triển khai một ứng dụng nguyên khối (monolithic). Mỗi tầng có thể yêu cầu triển khai và cấu hình riêng, điều này có thể làm tăng chi phí hoạt động.

### Sơ đồ ứng dụng (Mục 2.2.1)



<Hình a> Sơ đồ ứng dụng tổng quan khi khách mua hàng

Hình a mô tả tổng quan mô tả cách ứng dụng vận hành khi có khách hàng đặt hàng online theo trình tự:

1. Khách hàng tiến hành đặt hàng online thông qua Website của hàng trên máy tính hoặc trên điện thoại

2. REST API sẽ nhận và xử lý yêu cầu đặt hàng của khách, sau đó tạo đơn hàng tương ứng. Thông tin đơn hàng sẽ được lưu lại trong cơ sở dữ liệu

3. Thông tin đơn hàng mới sẽ được thông báo cho các bên liên quan bao gồm Khách hàng, Thu ngân, Quản lý cửa hàng và Quản lý kho.

4. Thu ngân hoặc quản lý cửa hàng xem thông tin đơn hàng và xác nhận đơn

5, 6 Sau khi đơn hàng được xác nhận, REST API sẽ cập nhật thông tin đơn hàng vào cơ sở dữ liệu, sau đó thông tin đơn hàng sẽ được gửi cho Quản lý kho để chuẩn bị hàng và Nhân viên giao hàng để tiến hành giao hàng cho khách.

7,8 Nhân viên giao hàng sau khi giao hàng thành công sẽ chuyển trạng thái đơn hàng. REST API tiếp nhận và cập nhật trạng thái của đơn hàng trên cơ sở dữ liệu.