**Product Requirements Document (PRD)**

**---**

**DRONE - FAST FOOD DELIVERY**

**1. Problem Alignment**

Người dùng gặp khó khăn khi đặt món ăn qua nhiều nền tảng khác nhau (web, mobile), trải nghiệm không đồng nhất. Khách hàng thiếu công cụ theo dõi đơn hàng minh bạch, thời gian thực. Các hệ thống giao đồ ăn hiện nay có quá nhiều bước, gây mất thời gian và dễ nhầm lẫn. Quản lý thanh toán và dịch vụ chưa tích hợp đồng bộ, khó mở rộng khi có nhiều đơn hàng.

**1.1. High Level Approach**

Xây dựng ứng dụng web + mobile sử dụng React.js + React Native để người dùng có thể đặt món thuận tiện.

Backend theo kiến trúc 3 lớp và chia thành 6 microservices (User, Product, Order, Payment,Delivery, Branch).

Tích hợp CI/CD pipeline để đảm bảo cập nhật nhanh chóng, ổn định.

Tích hợp giám sát hệ thống real-time để theo dõi hiệu năng từng service.

**1.2. Narrative**

1. Người dùng mở ứng dụng (web/mobile), đăng nhập hoặc tạo tài khoản.
2. Họ duyệt menu, chọn món và thêm vào giỏ hàng.
3. Người dùng xác nhận đơn hàng, chọn phương thức thanh toán (VNPay).
4. Hệ thống xử lý quaPayment Service, xác nhận thành công.
5. Order Service cập nhật trạng thái đơn hàng, người dùng theo dõi tiến trình giao hàng theo thời gian thực.
6. Người dùng nhận được thông báo khi đơn hàng hoàn tất.

**1.3. Goals**

Cung cấp trải nghiệm đặt hàng nhanh, tiện lợi, nhất quán trên web và mobile.

Tích hợp giỏ hàng, thanh toán, và theo dõi đơn hàng trong một hệ thống duy nhất.

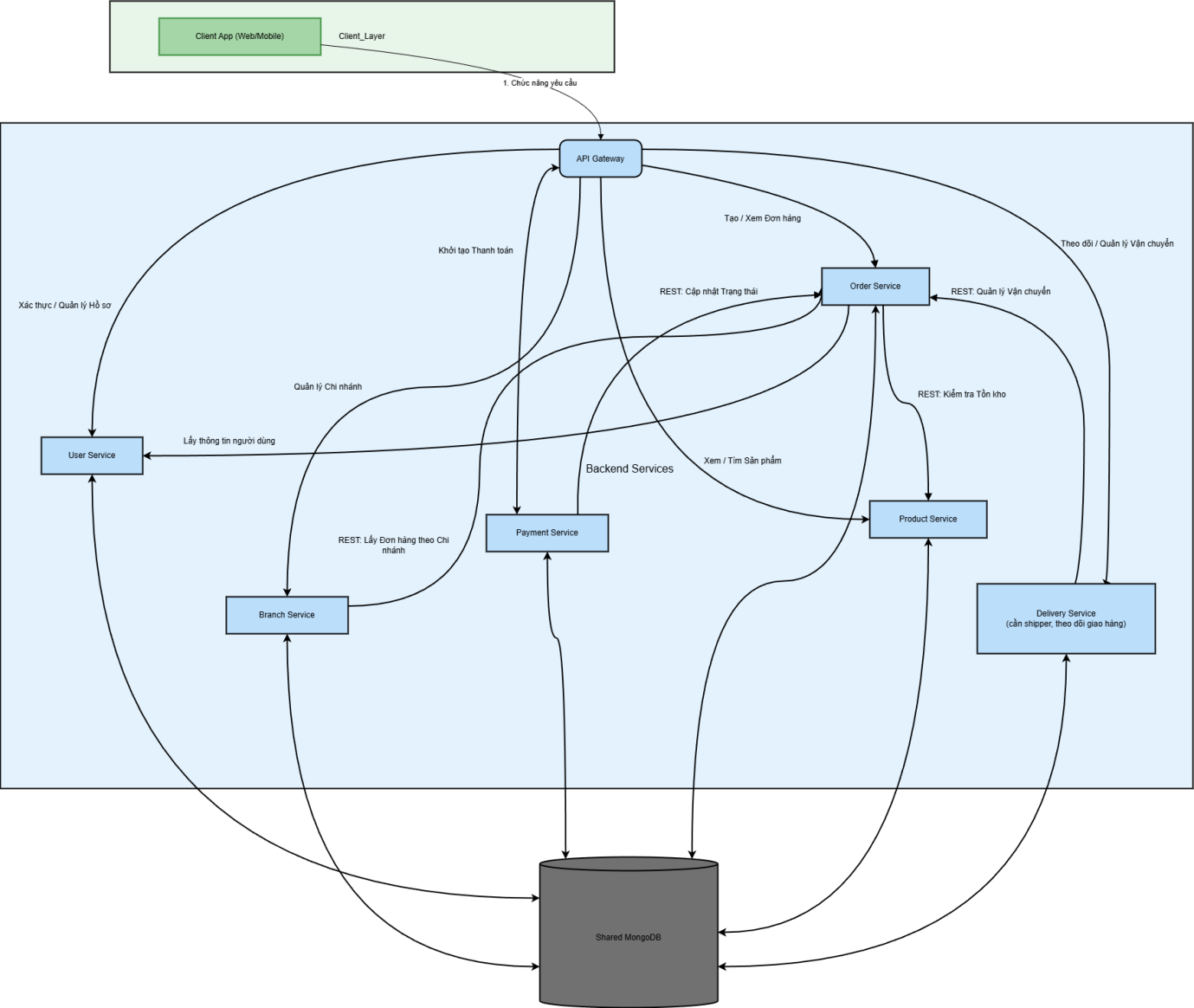
Dễ dàng mở rộng để hỗ trợ nhiều đối tác.

Đảm bảo hệ thống luôn ổn định, có giám sát real-time và khả năng phục hồi khi gặp sự cố.

**1.4. Giả định & Ràng buộc**

* **Giả định về Drone:** Các Drone luôn duy trì kết nối Internet (4G/5G) để gửi tọa độ về hệ thống. Trong trường hợp mất tín hiệu, Drone sẽ tự động quay về điểm xuất phát hoặc hạ cánh an toàn theo lập trình nhúng sẵn.
* **Ràng buộc pháp lý:** Việc vận hành Drone giao hàng tuân thủ các quy định về vùng bay, độ cao và giờ hoạt động của chính quyền sở tại.
* **Thiết bị người dùng:** Người dùng nên bật định vị trên điện thoại để xác định điểm nhận hàng chính xác.

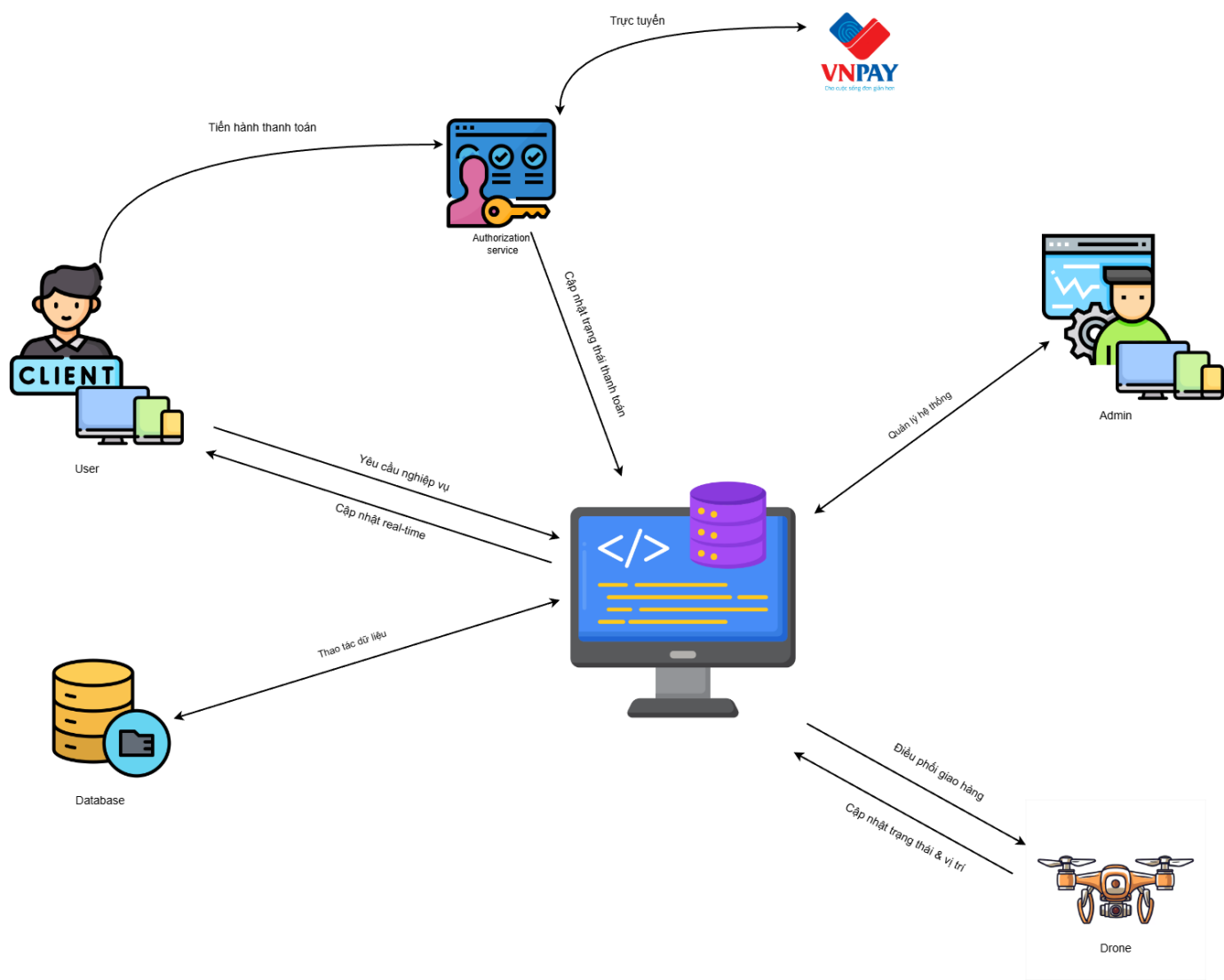
1. **Detailed System Architecture**



Hình 1. Sơ đồ kiến trúc Microservices hướng sự kiện

Mô tả: Sơ đồ kiến trúc Microservices hướng sự kiện minh họa kiến trúc tổng thể của backend, được xây dựng theo mô hình Microservices hướng sự kiện. Nó thể hiện cách Client tương tác với hệ thống thông qua API Gateway, và cách các service cốt lõi giao tiếp với nhau một cách bất đồng bộ qua Message Broker để xử lý một đơn hàng hoàn chỉnh.

1. **Solution Alignment**

****

Hình 2. Sơ đồ tổng quan hệ sinh thái sản phẩm

Mô tả: Sơ đồ tổng quan hệ sinh thái sản phẩm cung cấp cái nhìn tổng quan về toàn bộ hệ sinh thái sản phẩm. Nó xác định các tác nhân chính (User, Admin), hệ thống backend trung tâm, và các luồng tương tác cho những nghiệp vụ cốt lõi như yêu cầu dịch vụ, quản lý hệ thống, xử lý thanh toán và điều phối giao hàng bằng Drone.

* 1. **Key Features**

1. Frontend:

React.js, React Native.

Xử lý đặt món, thanh toán, theo dõi đơn hàng.

1. Backend (BE):

Kiến trúc 3 lớp: Presentation – Business – Data.

Kết nối với DB (MySQL/PostgreSQL).

1. Gồm 5 Microservices:

User Service: quản lý tài khoản.

Product Service: quản lý món ăn, danh mục, tồn kho.

Order Service: xử lý giỏ hàng, trạng thái đơn.

Payment Service: tích hợp cổng thanh toán (VNPay).

Delivery Service: một thành phần quan trọng để thực hiện mục tiêu "theo dõi tiến trình giao hàng theo thời gian thực".

1. CI/CD:

Pipeline tự động build, test, deploy cho cả hệ thống và từng service.

Dùng Docker + Kubernetes để triển khai.

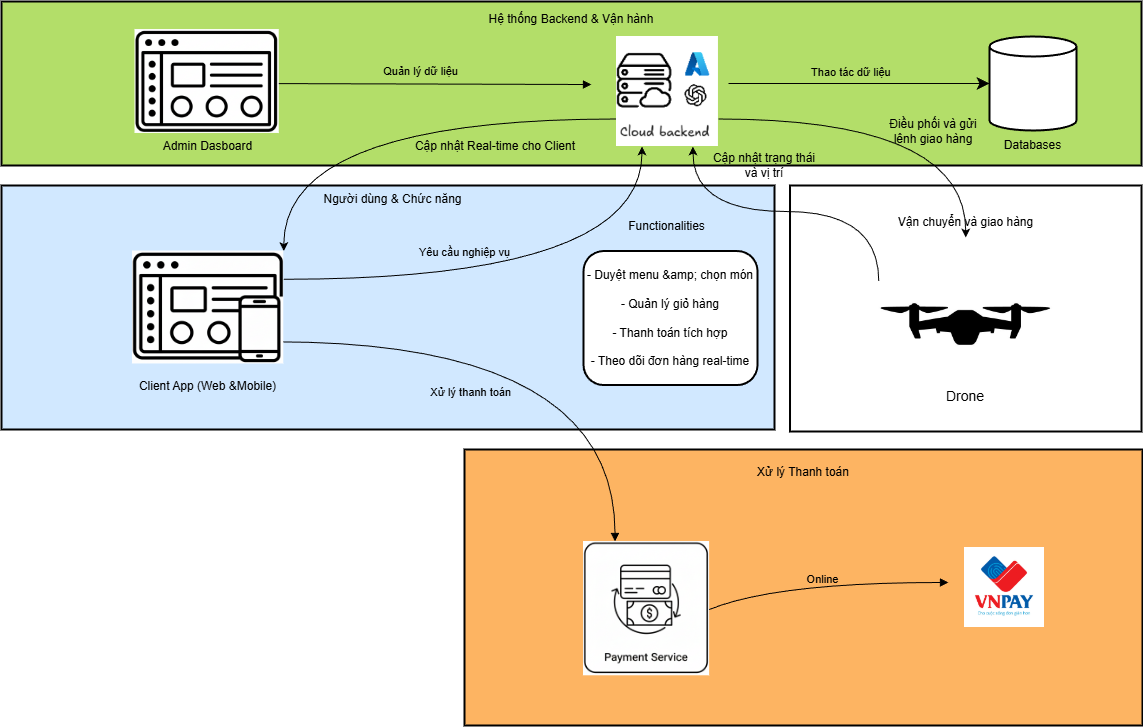
1. Monitoring:

Giám sát real-time bằng Prometheus + Grafana.

Dashboard theo dõi cho từng service (User, Product, Order, Payment).

**3.2. Key Flow**

**Operational Flow**:

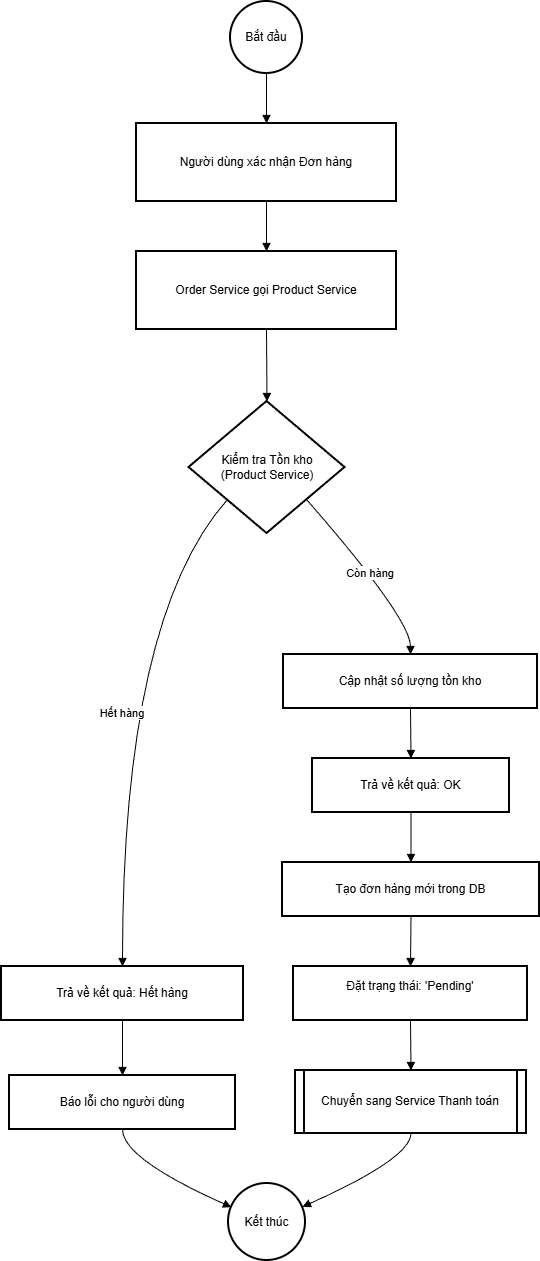


Hình 3. Sơ đồ Luồng Vận hành Hệ thống (Operational Flow Diagram)

Mô tả: Sơ đồ Luồng Vận hành Hệ thống này mô tả bức tranh tổng thể về luồng hoạt động của hệ thống, cho thấy cách các yêu cầu từ người dùng và quản trị viên được Cloud Backend xử lý, sau đó điều phối đến các hệ thống phụ trợ như Cơ sở dữ liệu, Cổng thanh toán, và hệ thống vận hành Drone để hoàn thành một đơn hàng.

**Business Logic Flow:**

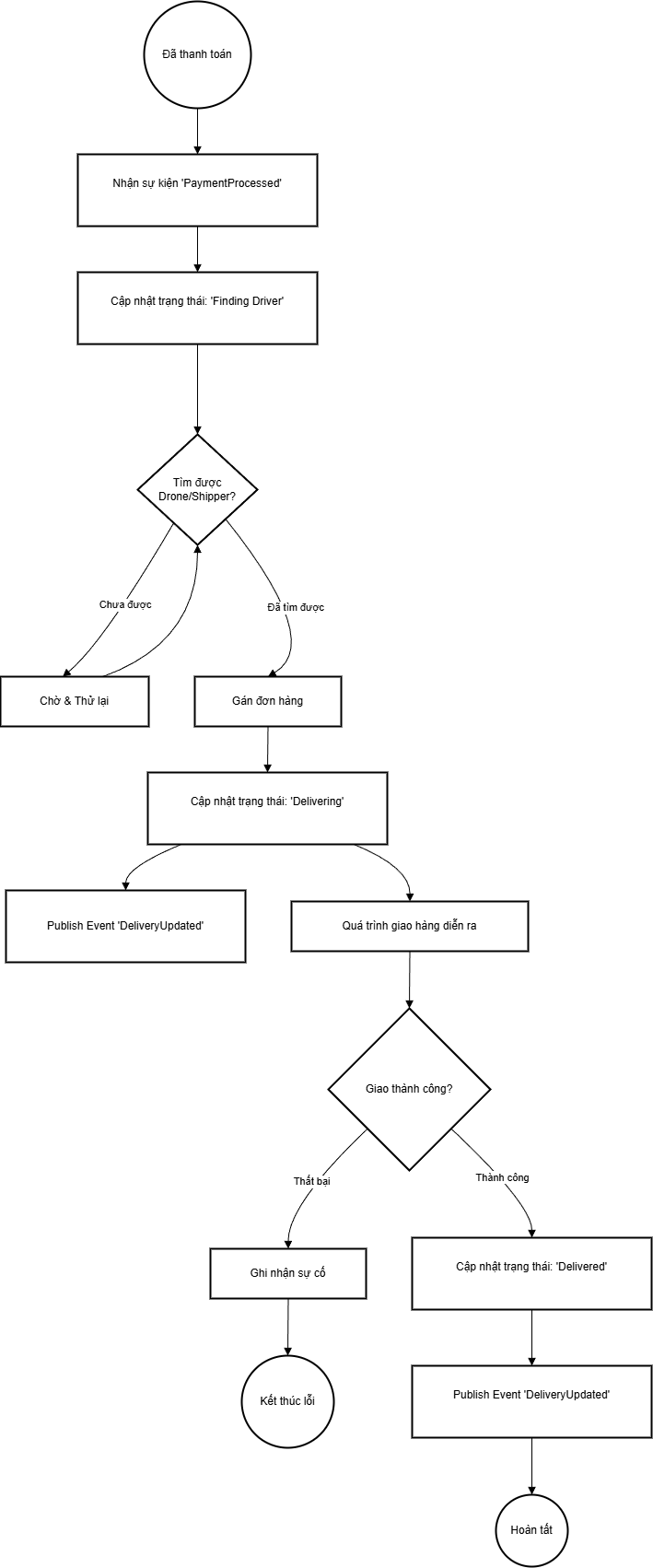
**Inventory Check**

****

Hình 4. Quy trình Kiểm tra Tồn Kho

Mô tả: Sơ đồ hoạt động (Activity Diagram) này mô tả logic nghiệp vụ cốt lõi khi khởi tạo đơn hàng. Nó minh họa quy trình rẽ nhánh quan trọng giữa Order Service và Product Service: hệ thống sẽ kiểm tra số lượng tồn kho trước; nếu còn hàng thì tiến hành "giữ hàng" (reserve stock) và tạo đơn trạng thái chờ, ngược lại sẽ báo lỗi ngay lập tức để đảm bảo tính toàn vẹn dữ liệu.

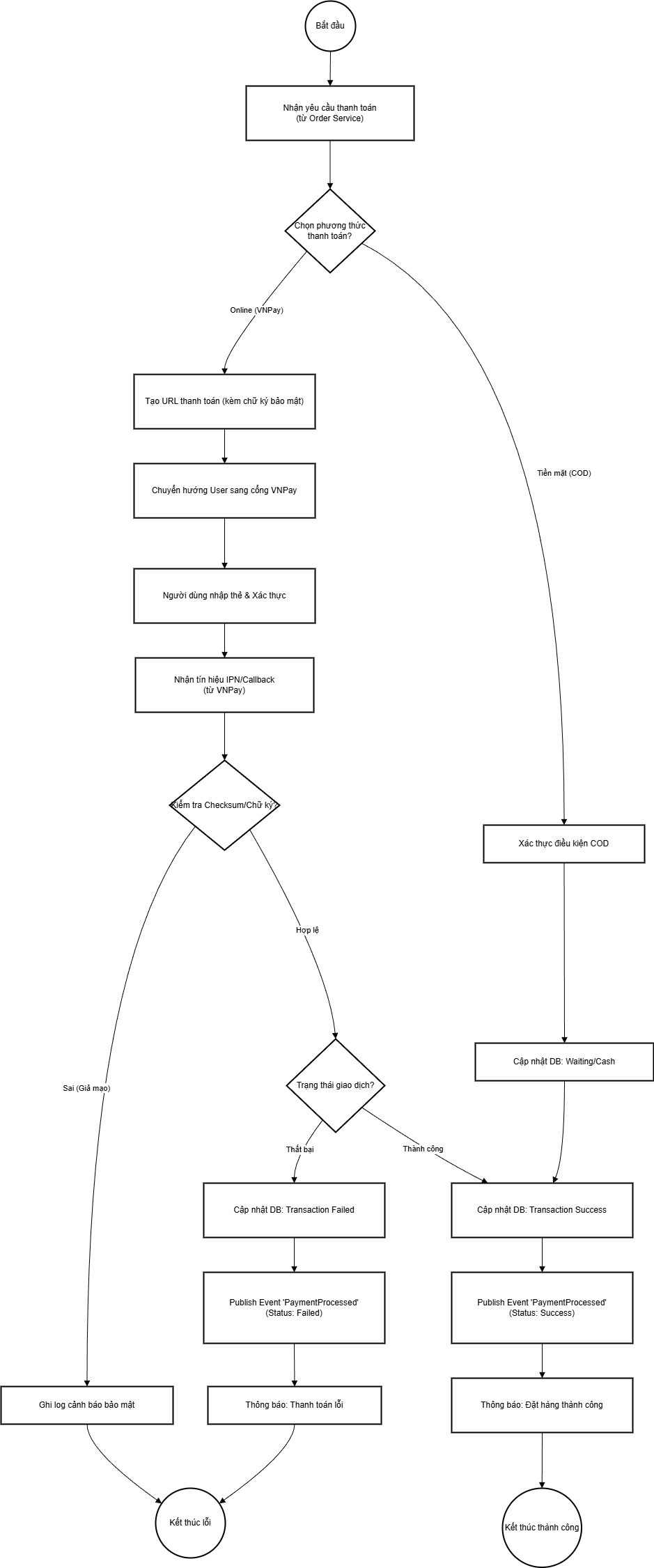
**Delivery Lifecycle**

****

Hình 5. Quy trình Vận Hành Giao Hàng

Mô tả: Đây là sơ đồ hoạt động chi tiết cho nghiệp vụ thanh toán, bao quát cả hai trường hợp: thanh toán Online (qua VNPay) và Tiền mặt (COD). Sơ đồ làm rõ các bước xác thực bảo mật (như kiểm tra Checksum/Chữ ký điện tử) và logic cập nhật trạng thái giao dịch vào Cơ sở dữ liệu trước khi kích hoạt các quy trình tiếp theo.

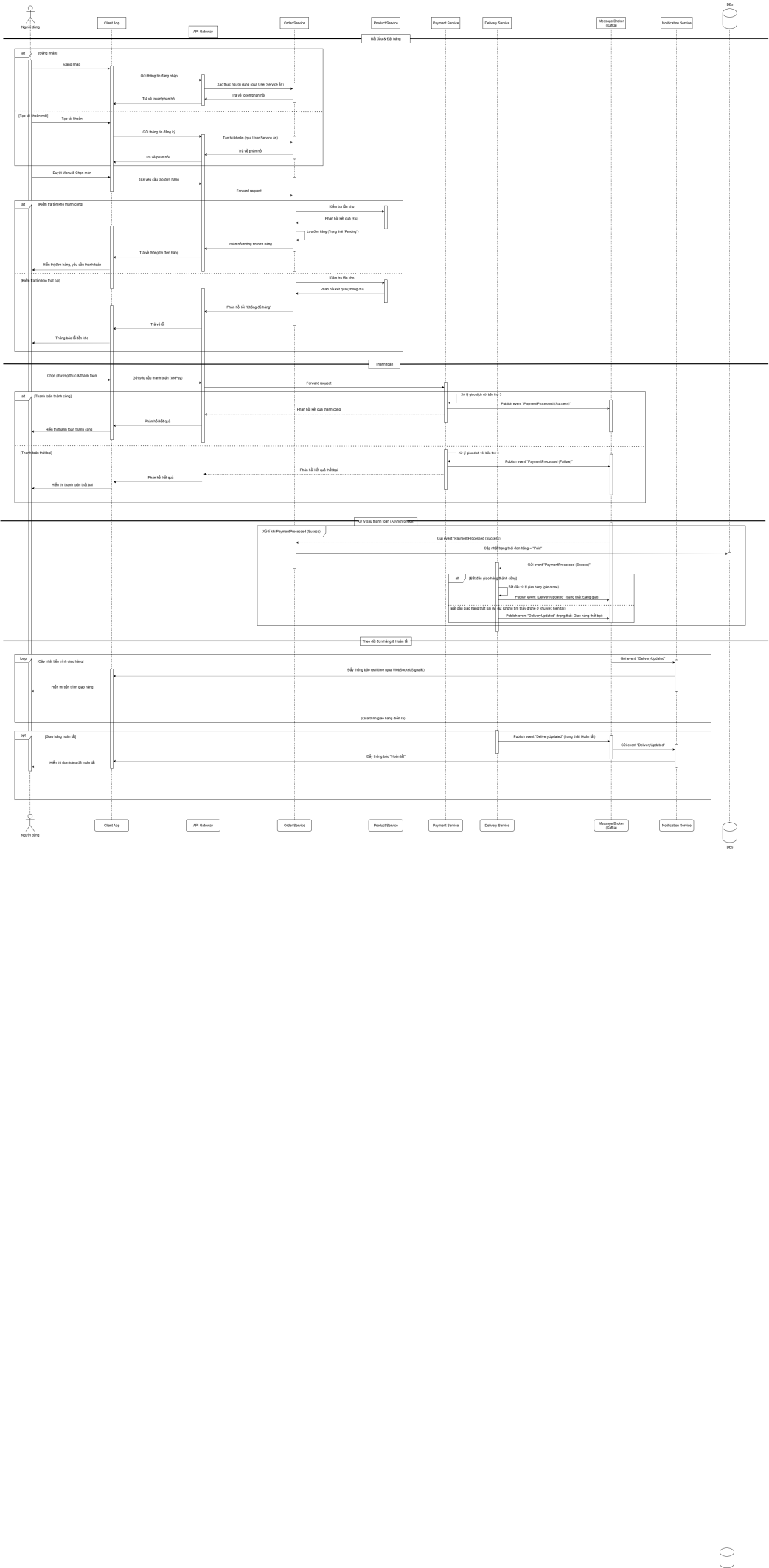
**Payment**

****

Hình 6. Quy trình Thanh Toán

Mô tả: Đây là sơ đồ hoạt động chi tiết cho nghiệp vụ thanh toán, bao quát cả hai trường hợp: thanh toán Online (qua VNPay) và Tiền mặt (COD). Sơ đồ làm rõ các bước xác thực bảo mật (như kiểm tra Checksum/Chữ ký điện tử) và logic cập nhật trạng thái giao dịch vào Cơ sở dữ liệu trước khi kích hoạt các quy trình tiếp theo.

**Detailed Interaction Flow:**

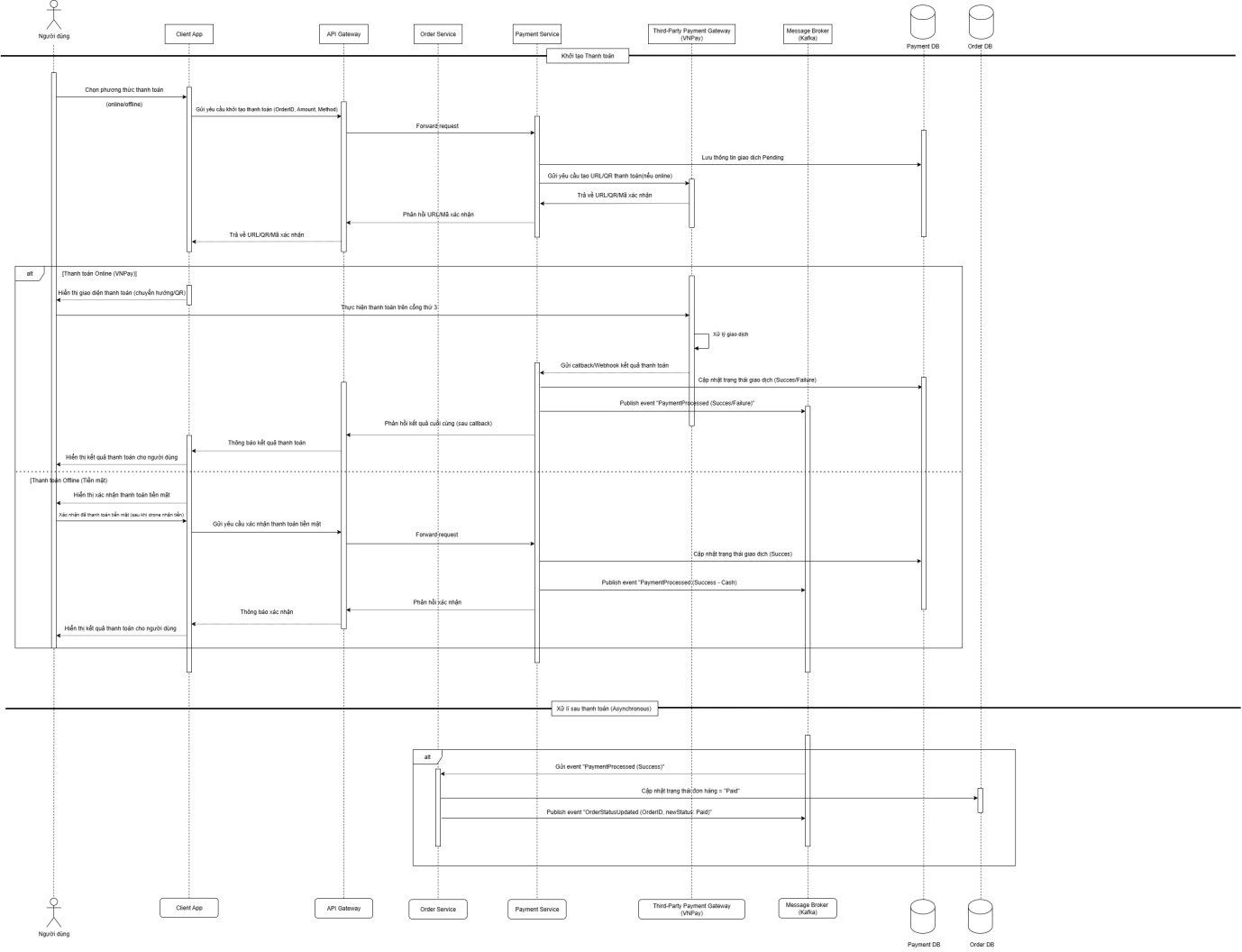


Hình 7. Luồng Tương tác Chi tiết

Mô tả: Sơ đồ Tuần tự này đi sâu vào chi tiết kỹ thuật, minh họa thứ tự các thông điệp và lời gọi hàm giữa các thành phần phần mềm (từ Client App đến các Microservices) để thực hiện một kịch bản đặt hàng thành công từ đầu đến cuối.

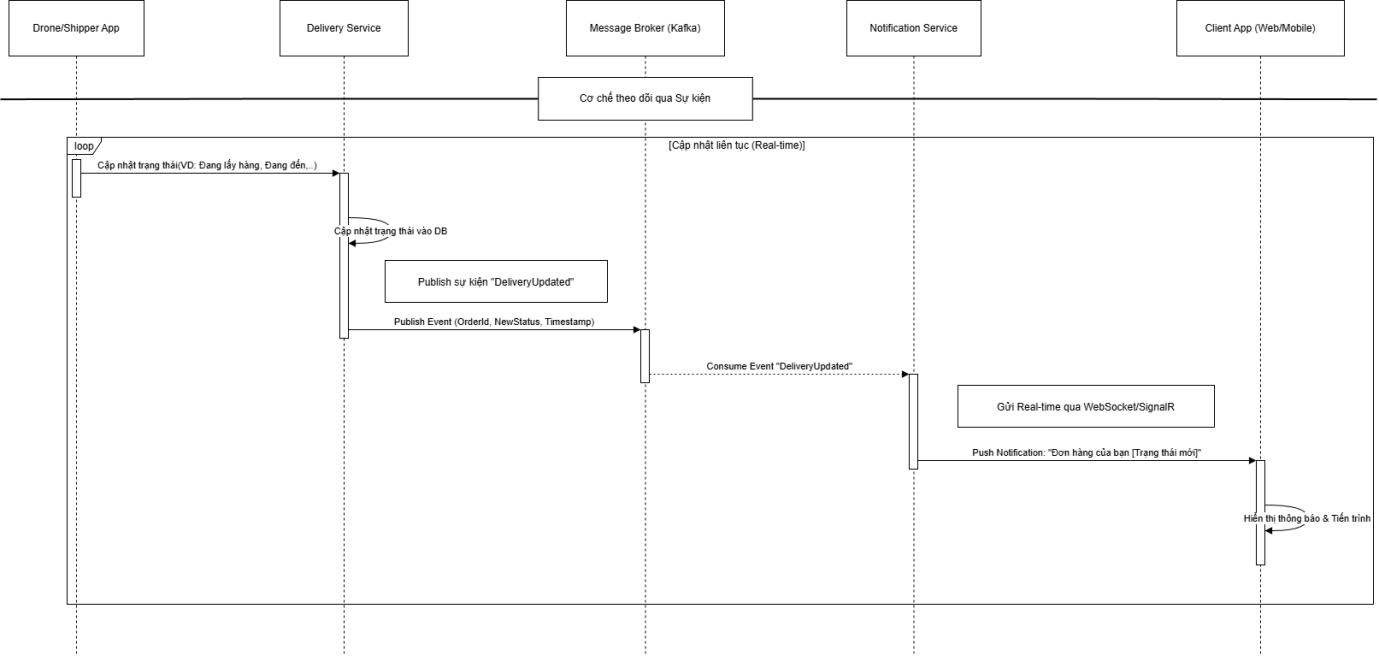
**Payment Flow:**

Người dùng xác nhận đơn → Payment Service xử lý → cập nhật Order Service.

Hình 8. Luồng Thanh toán

Mô tả: Sơ đồ Tuần tự này tập trung làm rõ các bước kỹ thuật phức tạp trong quy trình thanh toán. Nó thể hiện chi tiết luồng tương tác giữa Payment Service với cổng thanh toán bên thứ ba (VNPay), cách hệ thống xử lý callback, và cách nó phát đi sự kiện PaymentProcessed lên Message Broker sau khi giao dịch hoàn tất.

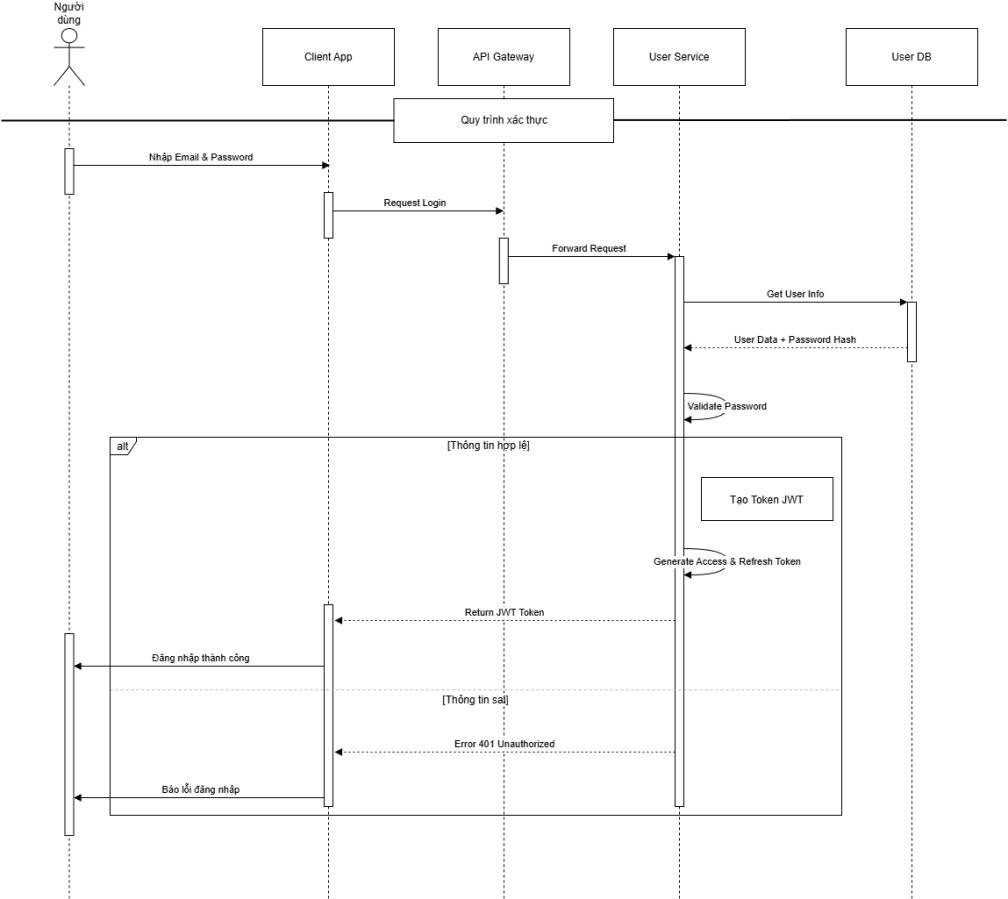
**Real-time Delivery Tracking Flow:**

****

Hình 9. Luồng Theo dõi Giao hàng Thời gian thực

Mô tả: Sơ đồ tuần tự này minh họa cơ chế cập nhật vị trí thời gian thực (Real-time tracking) dựa trên kiến trúc hướng sự kiện. Nó cho thấy luồng dữ liệu từ Drone/Shipper được gửi về Delivery Service, sau đó publish sự kiện qua Kafka và được Notification Service đẩy xuống ứng dụng người dùng thông qua WebSocket/SignalR, đảm bảo độ trễ thấp nhất mà không cần tải lại trang.

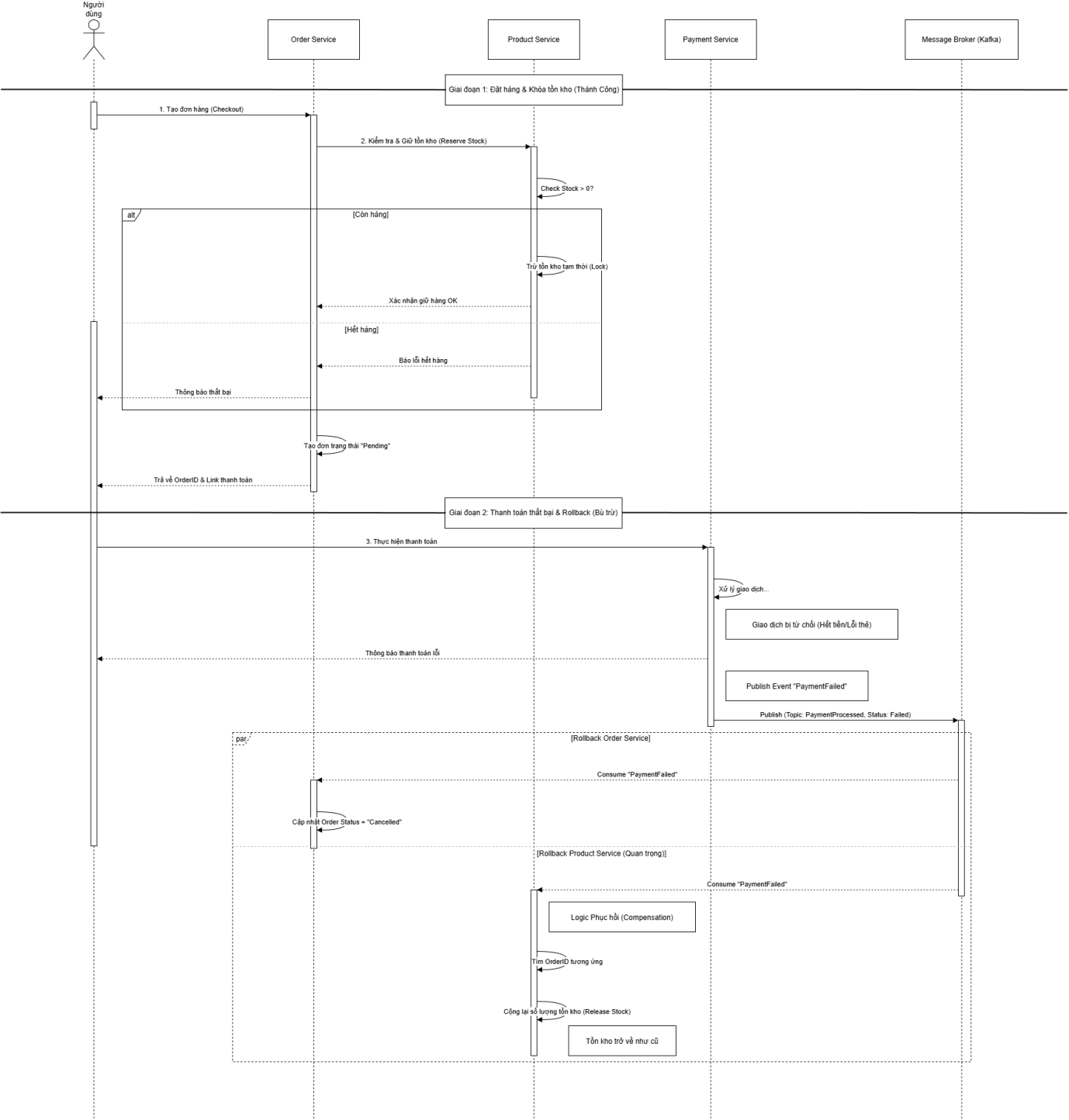
**Authentication Flow:**

****

Hình 10. Luồng Xác thực & Đăng nhập

Mô tả: Sơ đồ này mô tả quy trình bảo mật và cấp quyền truy cập hệ thống. Nó thể hiện các bước tương tác từ khi người dùng nhập thông tin xác thực, hệ thống kiểm tra đối chiếu trong User DB, và quy trình cấp phát chuỗi mã hóa JWT (bao gồm Access Token và Refresh Token) để xác thực cho các phiên làm việc an toàn sau này.

**Compensation/Rollback**

****

Hình 11. Luồng Phục hồi Tồn Kho

Mô tả: Sơ đồ này minh họa cơ chế Bù trừ hay còn gọi là Rollback nghiệp vụ trong kiến trúc Microservices. Nó mô tả kịch bản khi giao dịch thanh toán thất bại sau khi đã giữ hàng: các service liên quan (Order, Product) sẽ lắng nghe sự kiện lỗi từ Payment Service để tự động hủy đơn và hoàn trả lại số lượng tồn kho, đảm bảo tính nhất quán cuối cùng của dữ liệu.

1. **Yêu cầu nghiệp vụ**

**4.1. Mục tiêu nghiệp vụ**

* Cung cấp một trải nghiệm đặt hàng nhất quán, nhanh chóng và tiện lợi trên cả hai nền tảng Web và Mobile.
* Tích hợp liền mạch các chức năng từ duyệt menu, quản lý giỏ hàng, thanh toán đến theo dõi đơn hàng trong một hệ thống duy nhất để tăng sự hài lòng của khách hàng.
* Xây dựng hệ thống có khả năng mở rộng linh hoạt để dễ dàng tích hợp thêm các đối tác nhà hàng và dịch vụ vận chuyển trong tương lai.
* Đảm bảo hệ thống hoạt động ổn định, có khả năng phục hồi cao và được giám sát theo thời gian thực để giảm thiểu gián đoạn dịch vụ.

**4.2. User Stories**

**Dành cho Khách hàng:**

* **Story 1: Đặt hàng nhanh chóng**
  + As a: Người dùng bận rộn.
  + I want to: Duyệt menu và hoàn thành đặt hàng trong vài bước đơn giản.
  + So that: Tiết kiệm thời gian và có trải nghiệm tiện lợi.
  + **Acceptance Criteria (Tiêu chí chấp nhận):**

1. Người dùng có thể tìm kiếm món ăn theo tên hoặc danh mục.
2. Thao tác thêm món vào giỏ hàng không quá 2 lần chạm/click.
3. Quy trình Checkout (từ Giỏ hàng -> Thanh toán thành công) diễn ra dưới 2 phút.
4. Thông tin đơn hàng hiển thị rõ ràng tổng tiền và phí vận chuyển trước khi xác nhận.

* **Story 2: Theo dõi đơn hàng minh bạch**
  + As a: Khách hàng.
  + I want to: Theo dõi trạng thái và vị trí đơn hàng theo thời gian thực.
  + So that: Chủ động sắp xếp thời gian nhận hàng và yên tâm hơn.
  + **Acceptance Criteria:**

1. Hiển thị bản đồ với icon Drone di chuyển tương ứng với vị trí thực tế.
2. Vị trí/Trạng thái cập nhật tự động mỗi 3-5 giây mà không cần tải lại trang (sử dụng WebSocket).
3. Hiển thị thông báo Pop-up hoặc Push Notification ngay khi đơn hàng chuyển trạng thái (VD: Đang giao -> Đã đến).

* **Story 3: Trải nghiệm nhất quán**
  + As a: Người dùng đa nền tảng.
  + I want to: Trải nghiệm đồng bộ trên cả Web và Mobile.
  + So that: Đặt hàng mọi lúc mọi nơi không gặp trở ngại.
  + **Acceptance Criteria:**

1. Giỏ hàng được đồng bộ: Thêm món trên Web thì mở App Mobile cũng phải thấy món đó.
2. Lịch sử đơn hàng hiển thị đầy đủ trên cả hai nền tảng.
3. Giao diện (UI) sử dụng chung Design System trên cả React.js và React Native.

**Dành cho Quản trị viên (Admin):**

* **Story 4: Quản lý sản phẩm hiệu quả**
  + As a: Quản trị viên cửa hàng.
  + I want to: Dễ dàng thêm/sửa/xóa món ăn và cập nhật tồn kho.
  + So that: Thực đơn luôn chính xác với thực tế.
  + **Acceptance Criteria:**

1. Admin có thể upload ảnh món ăn, nhập giá và mô tả chi tiết.
2. Khi cập nhật số lượng tồn kho = 0, món ăn tự động ẩn hoặc hiện nhãn "Hết hàng" trên App người dùng ngay lập tức.

* **Story 5: Giám sát hoạt động hệ thống**
  + As a: Quản trị viên hệ thống .
  + I want to: Xem dashboard hiệu năng các microservice.
  + So that: Phát hiện và xử lý sự cố nhanh chóng.
  + **Acceptance Criteria:**

1. Dashboard hiển thị biểu đồ độ trễ và lưu lượng request của từng Service.
2. Hệ thống gửi cảnh báo qua Email/Slack nếu tỷ lệ lỗi vượt quá 1% trong 5 phút.

**4.3. Thực trạng & Vấn đề**

* Người dùng hiện tại gặp khó khăn do trải nghiệm không đồng nhất khi đặt món ăn qua các nền tảng khác nhau.
* Khách hàng thiếu một công cụ theo dõi đơn hàng minh bạch và cập nhật theo thời gian thực, gây ra sự không chắc chắn.
* Quy trình đặt hàng hiện tại có nhiều bước phức tạp, dễ gây nhầm lẫn và tốn thời gian.
* Việc quản lý thanh toán và các dịch vụ khác chưa được tích hợp đồng bộ, gây khó khăn trong việc mở rộng quy mô.

**4.4. Nghiệp vụ chính**

* Đăng ký & Đăng nhập: Người dùng có thể tạo tài khoản hoặc đăng nhập một cách dễ dàng.
* Đặt hàng & Quản lý giỏ hàng: Người dùng duyệt menu, chọn món và quản lý giỏ hàng của mình.
* Thanh toán: Người dùng xác nhận đơn hàng và thực hiện thanh toán qua các phương thức trực tuyến tích hợp (VNPay).
* Theo dõi đơn hàng: Người dùng theo dõi trạng thái và tiến trình giao hàng của đơn hàng theo thời gian thực sau khi thanh toán thành công.
* Thông báo & Hoàn tất: Người dùng nhận được thông báo tự động khi trạng thái đơn hàng thay đổi và khi đơn hàng được giao thành công.

**4.5. Success Metrics / KPIs**

**1. Chỉ số Nghiệp vụ & Tài chính :**

* Tỷ lệ Hoàn thành Đơn hàng : Tỷ lệ phần trăm đơn hàng được tạo và thanh toán thành công.
  + Mục tiêu: > 98%.
* Giá trị Đơn hàng Trung bình : Tổng doanh thu chia cho tổng số đơn hàng.
  + Mục tiêu: Tăng 10% sau 3 tháng ra mắt.

**2. Chỉ số Trải nghiệm Người dùng:**

* Thời gian Hoàn thành Tác vụ (Task Completion Time): Thời gian trung bình từ lúc người dùng bắt đầu chọn món đến khi đặt hàng thành công.
  + Mục tiêu: < 2 phút, nhằm đáp ứng yêu cầu về trải nghiệm nhanh và tiện lợi.
* Tỷ lệ Giữ chân Người dùng (User Retention Rate): Tỷ lệ phần trăm người dùng quay lại đặt hàng trong tháng tiếp theo.
  + Mục tiêu: > 40%.
* Điểm hài lòng của khách hàng (Customer Satisfaction - CSAT): Đo lường qua khảo sát trong ứng dụng.
  + Mục tiêu: Đạt 4/5 sao trở lên.

**3. Chỉ số Hiệu năng & Vận hành:**

* Thời gian Phản hồi API (API Response Time): Thời gian phản hồi của các API chính (lấy sản phẩm, tạo đơn hàng).
  + Mục tiêu: ≤ 500ms, đúng theo yêu cầu phi chức năng.
* Độ sẵn sàng của Hệ thống (System Uptime): Tỷ lệ phần trăm thời gian hệ thống hoạt động ổn định.
  + Mục tiêu: 99.9%, nhằm đảm bảo tính sẵn sàng và độ tin cậy cao.
* Thời gian Trung bình để Phát hiện Sự cố (Mean Time to Detect - MTTD): Thời gian trung bình từ khi sự cố xảy ra đến khi hệ thống giám sát phát hiện được.
  + Mục tiêu: < 5 phút, thể hiện hiệu quả của hệ thống Monitoring.

**5. Yêu cầu chức năng**

**5.1. User Service**

* Cho phép người dùng tạo tài khoản mới và đăng nhập bằng email/mật khẩu.
* Quản lý thông tin hồ sơ và địa chỉ giao hàng của người dùng.

**5.2. Product Service**

* Cung cấp API để lấy danh sách món ăn và thông tin chi tiết của từng món.
* Cho phép quản trị viên thêm, xóa, sửa thông tin món ăn (giá, mô tả, hình ảnh).
* Quản lý và cập nhật khi có đơn hàng được tạo.

**5.3. Order Service**

* Xử lý logic giỏ hàng: thêm, xóa, cập nhật số lượng sản phẩm.
* Tạo đơn hàng mới với trạng thái ban đầu là "Pending" sau khi người dùng xác nhận.
* Cung cấp API để người dùng có thể xem trạng thái và lịch sử các đơn hàng đã đặt.
* Cập nhật trạng thái đơn hàng dựa trên các sự kiện từ Payment Service và Delivery Service.

**5.4. Payment Service**

* Tích hợp với các cổng thanh toán bên thứ ba như VNPay để xử lý giao dịch trực tuyến.
* Xử lý các callback/webhook từ cổng thanh toán để xác nhận giao dịch thành công hay thất bại.
* Publish sự kiện PaymentProcessed lên Message Broker (Kafka) sau khi xử lý xong một giao dịch.

**5.5. Delivery Service**

* Tiếp nhận thông tin đơn hàng đã được thanh toán thành công để bắt đầu quy trình giao hàng.
* Quản lý và cập nhật các trạng thái giao hàng (ví dụ: đang tìm Drone, đang giao, đã giao thành công).
* Cung cấp dữ liệu để người dùng có thể theo dõi tiến trình giao hàng theo thời gian thực.

**5.6. Notification Service**

* Lắng nghe các sự kiện thay đổi trạng thái đơn hàng từ Message Broker.
* Gửi thông báo đẩy hoặc cập nhật real-time qua WebSocket/SignalR đến Client App khi có cập nhật mới về đơn hàng.

**5.7. Admin Portal (Trang quản trị)**

* **Dashboard tổng quan:** Hiển thị các chỉ số kinh doanh chính (Tổng đơn hàng trong ngày, Doanh thu, Số Drone đang hoạt động)(nếu có đủ thời gian).
* **Bản đồ nhiệt (Heatmap):** Hiển thị vị trí thời gian thực của tất cả các Drone đang bay trên bản đồ số.
* **Quản lý sự cố:** Nhận cảnh báo ngay lập tức khi có Drone gặp sự cố hoặc đơn hàng bị lỗi thanh toán/giao vận. (nếu có đủ thời gian)
* **Quản lý Menu:** Giao diện CRUD (Thêm, Đọc, Sửa, Xóa) thân thiện để quản lý danh mục và món ăn (tương tác với Product Service).

**6. Yêu cầu phi chức năng**

**6.1. Bảo mật:** Giao tiếp giữa client và server phải được mã hóa bằng HTTPS.

**6.2. Hiệu năng:**

* Thời gian phản hồi của các API chính (lấy danh sách sản phẩm, tạo đơn hàng) phải nhanh chóng ( ≤ 100ms) để đảm bảo trải nghiệm người dùng mượt mà.
* Hệ thống phải có khả năng xử lý đồng thời một lượng lớn người dùng và đơn hàng trong giờ cao điểm.

**6.3. Tính sẵn sàng & Độ tin cậy:**

* Hệ thống phải được thiết kế để có tính sẵn sàng cao, đảm bảo hoạt động ổn định và có khả năng phục hồi khi một trong các microservice gặp sự cố.
* Sử dụng cơ sở dữ liệu có cơ chế sao lưu và phục hồi định kỳ.

**6.4. Khả năng mở rộng:**

* Kiến trúc microservices phải cho phép mỗi dịch vụ có thể được mở rộng quy mô một cách độc lập tùy theo tải.
* Sử dụng Kubernetes để tự động hóa việc triển khai và mở rộng các container.

**6.5. Khả năng giám sát:**

* Hệ thống phải được giám sát hiệu năng theo thời gian thực bằng các công cụ như Prometheus và Grafana.
* Phải có các dashboard giám sát riêng cho từng microservice để dễ dàng theo dõi và phát hiện sự cố.

**6.6. Khả năng triển khai:** Toàn bộ quy trình từ build, test đến deploy phải được tự động hóa thông qua CI/CD pipeline.

**7. Backend Design**

**7.1. Sub-system**

Payment Service: Tích hợp với cổng thanh toán của bên thứ ba là VNPay để xử lý các giao dịch trực tuyến.

**7.2. App**

* **Frontend:**

Web App: React.js.

Mobile App: React Native.

* **Mô tả Giao diện:**
  + **Màn hình Trang chủ:**

- Thanh tìm kiếm món ăn trên cùng.

- Danh sách "Món ngon gợi ý" và "Danh mục món ăn" dạng lưới (Grid).

- Thanh điều hướng gồm: Trang chủ, Đơn hàng, Giỏ hàng, Tài khoản.

* **Màn hình Theo dõi đơn hàng:**

**-** Bản đồ chiếm 2/3 màn hình phía trên, hiển thị lộ trình và icon Drone.

- Thẻ thông tin phía dưới: Trạng thái hiện tại, Tên Shipper/Mã Drone, Thời gian dự kiến.

- Nút "Liên hệ hỗ trợ" hoặc "Chi tiết đơn hàng".

* **Màn hình Giỏ hàng & Thanh toán:**

- Danh sách món đã chọn.

- Khu vực chọn phương thức thanh toán.

- Tổng cộng tiền hàng + Phí ship.

- Nút "Đặt hàng" nổi bật.

* **Backend:**

Framework: Spring Boot.

Kiến trúc: Microservices.

* Database: PostgreSQL.

**7.3. Real-time Delivery Tracking**

Hệ thống không sử dụng GPS của drone nhưng thực hiện theo dõi đơn hàng theo thời gian thực thông qua cơ chế sự kiện.

* Cung cấp dữ liệu:

Delivery Service chịu trách nhiệm quản lý và cập nhật các trạng thái giao hàng.

Mỗi khi có thay đổi, Delivery Service sẽ publish một sự kiện lên Message Broker.

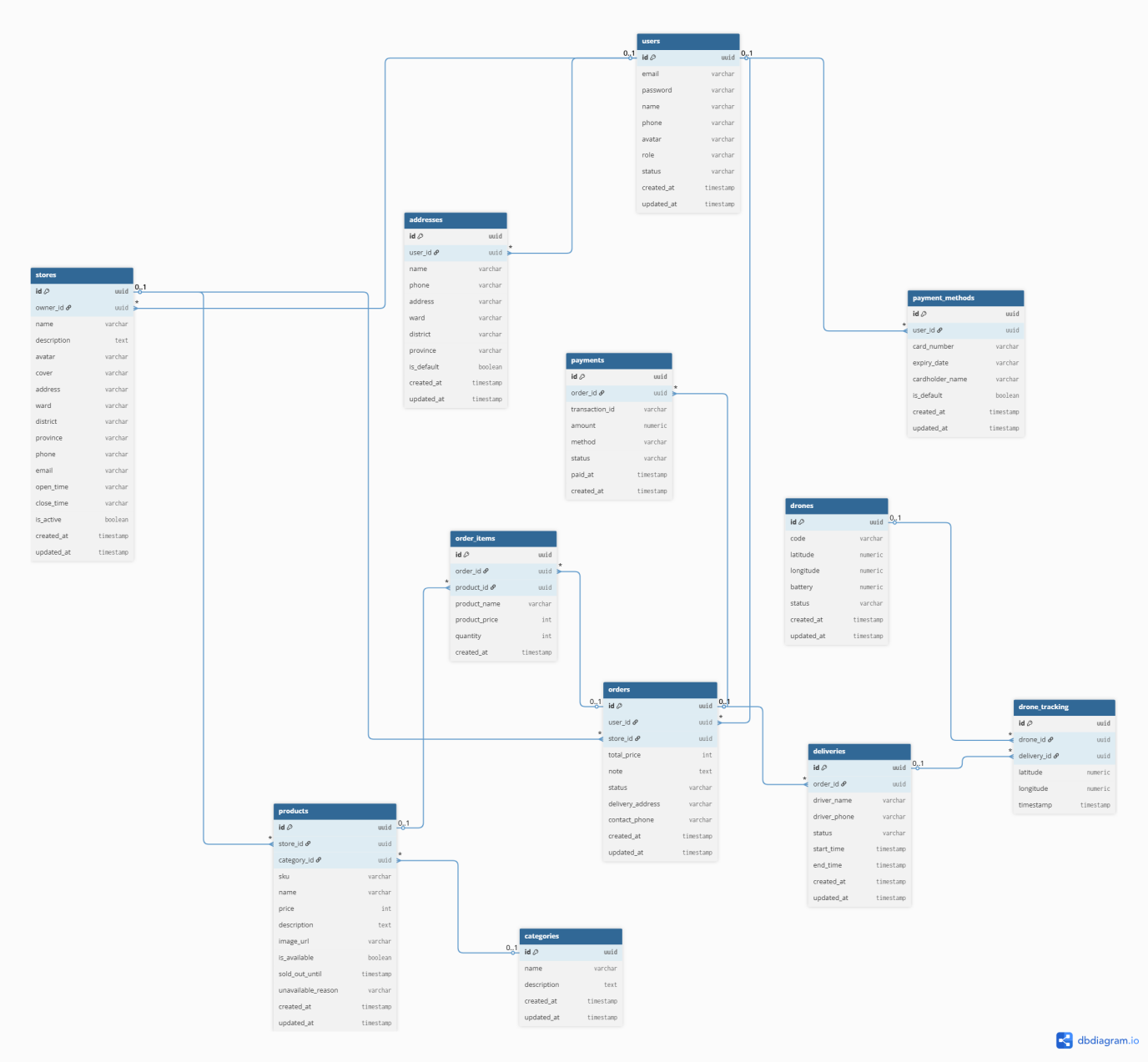
* Hiển thị dữ liệu:

Notification Service sẽ lắng nghe các sự kiện DeliveryUpdated từ Kafka.

Ngay khi nhận được sự kiện, Notification Service sẽ gửi thông báo đẩy hoặc cập nhật real-time qua WebSocket/SignalR đến ứng dụng của người dùng .

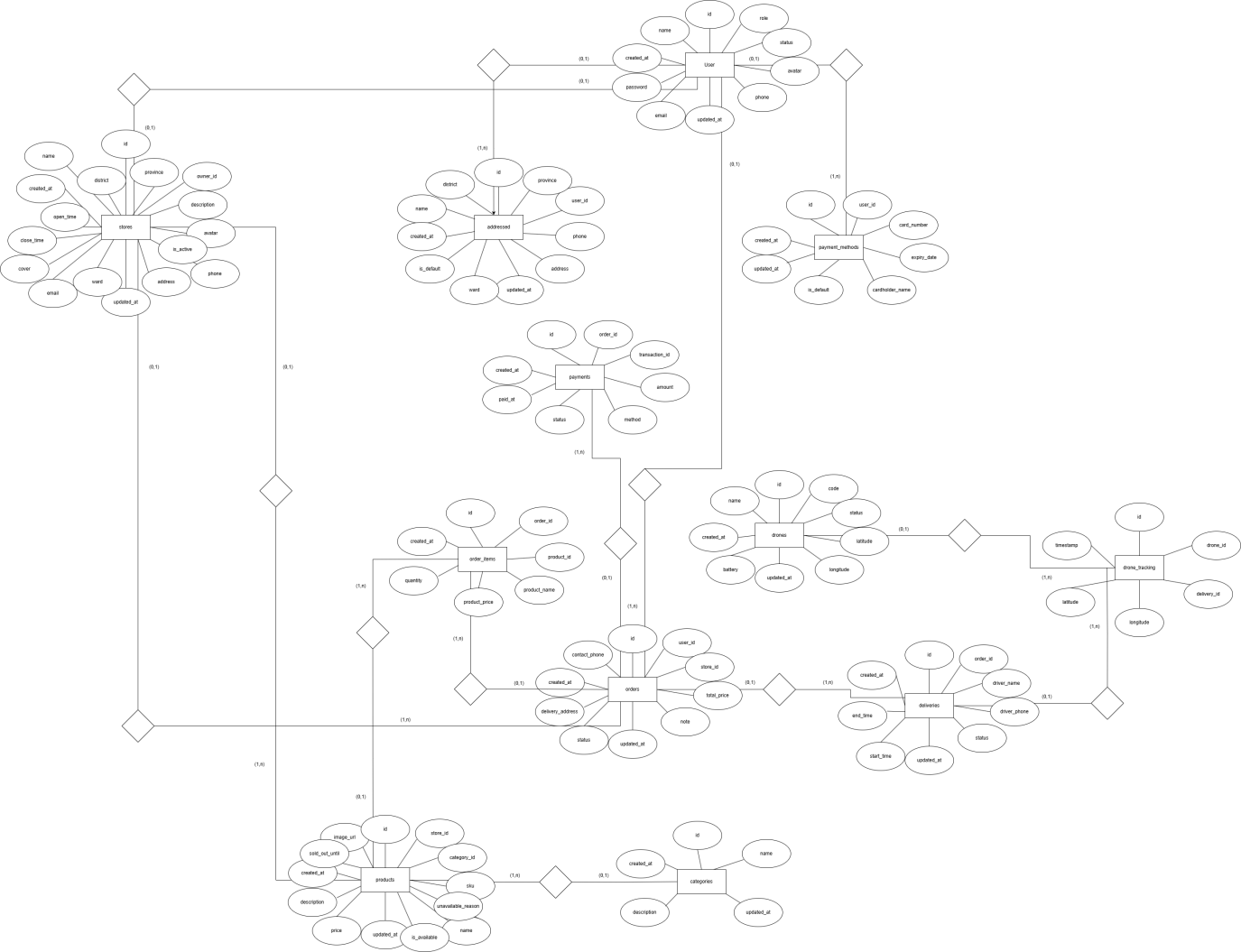
Người dùng từ đó có thể theo dõi trạng thái và tiến trình giao hàng của đơn hàng một cách minh bạch và tức thì.

**7.4. Database**

****Hình 12. Database

Mô tả: Sơ đồ CSDL (Database Schema) thể hiện cấu trúc logic của cơ sở dữ liệu vật lý, bao gồm tất cả các bảng, các cột dữ liệu, kiểu dữ liệu, và các mối quan hệ khóa chính - khóa ngoại giữa chúng. Đây là bản thiết kế chi tiết cho việc triển khai database trên PostgreSQL.

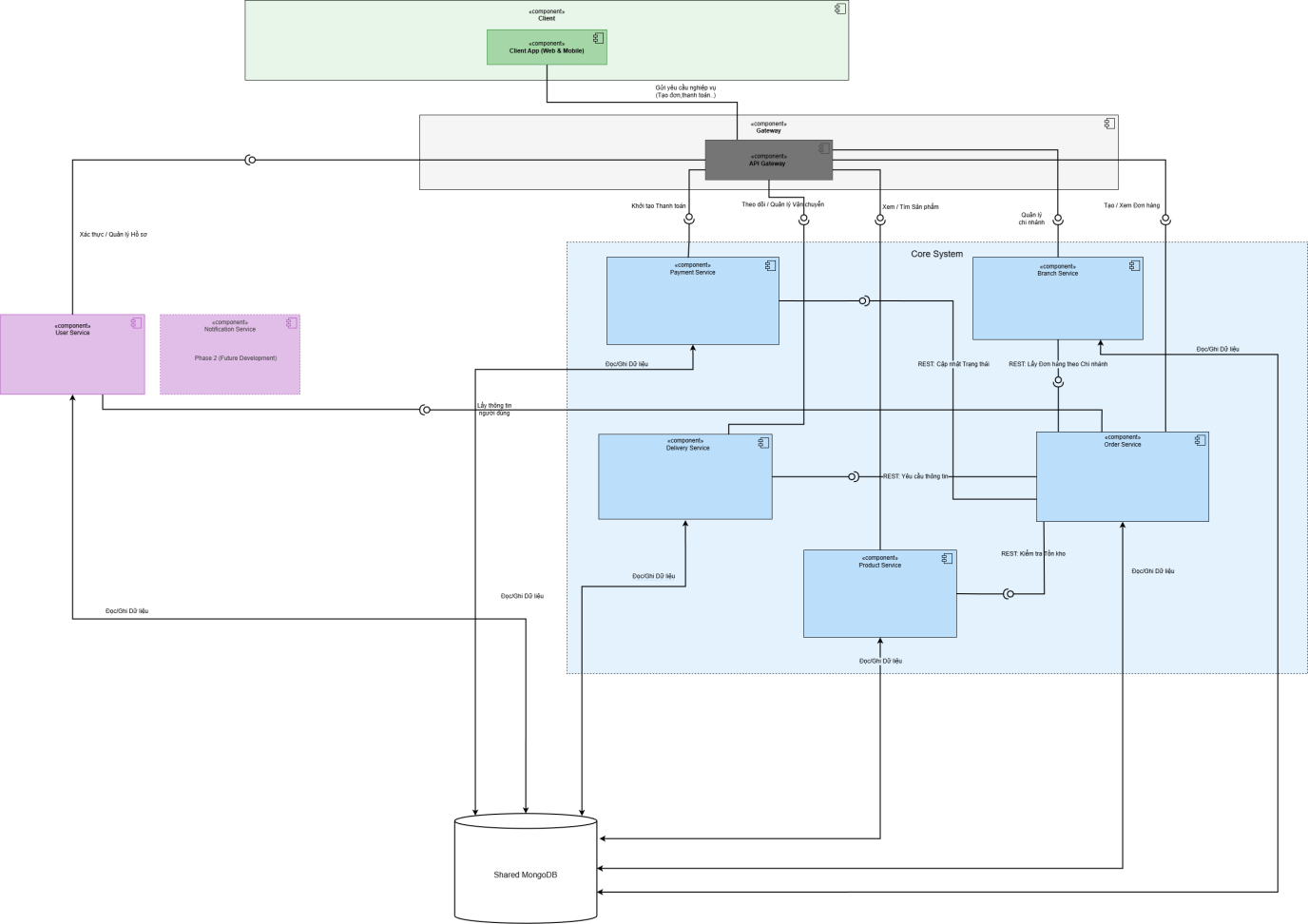
**7.5. ERD**

****

Hình 13. Sơ đồ ERD

Mô tả: Sơ đồ Quan hệ Thực thể này mô tả cơ sở dữ liệu ở mức độ khái niệm, tập trung vào các thực thể nghiệp vụ chính (User, Order, Product...), các thuộc tính của chúng, và mối quan hệ (một-một, một-nhiều, nhiều-nhiều) giữa các thực thể đó.

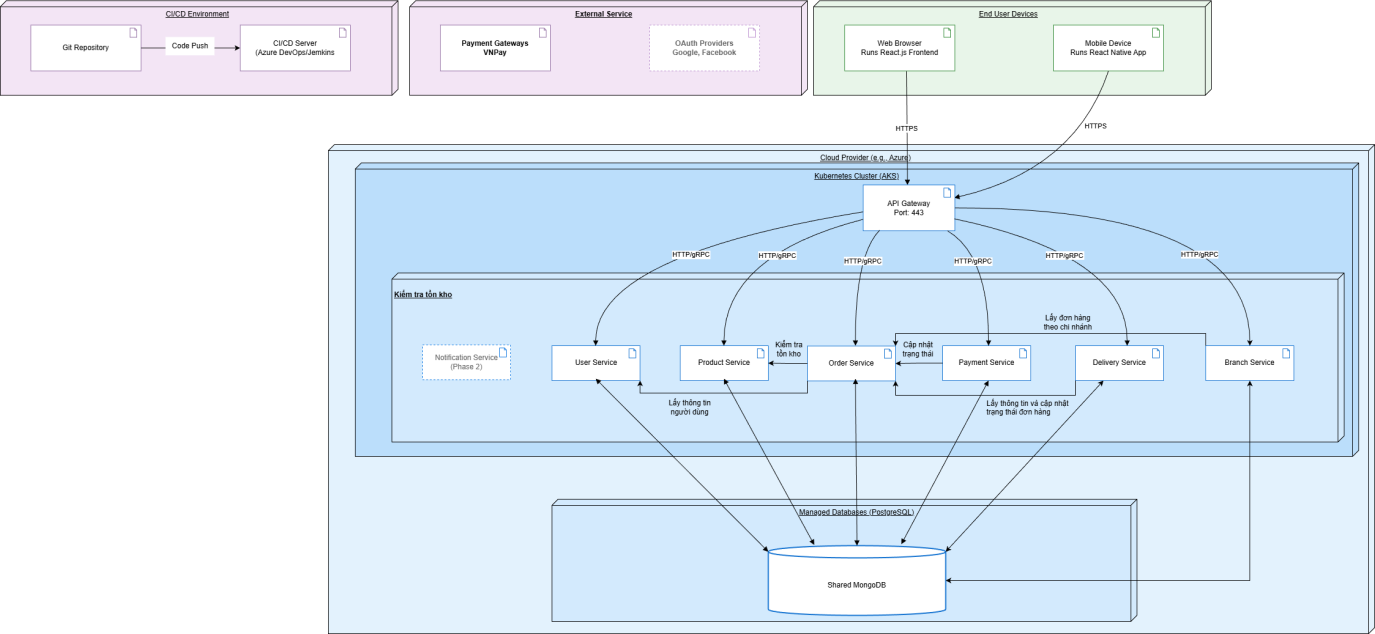
**7.6. Component**

****

Hình 14. Sơ đồ Component

Mô tả: thể hiện các thành phần phần mềm chính và mối quan hệ tương tác, phụ thuộc giữa chúng. Nó làm rõ kiến trúc Microservices RESTful, nơi các dịch vụ giao tiếp đồng bộ qua API Gateway và HTTP Call trực tiếp, giúp hệ thống có tính độc lập cao và khả năng triển khai linh hoạt.

**7.7. Deployment**

****

Hình 15. Sơ đồ Triển khai

Mô tả: kiến trúc vật lý và môi trường vận hành của hệ thống, chỉ rõ cách các thành phần phần mềm được triển khai trên các hạ tầng như Cloud Provider (Kubernetes Cluster, Managed Databases) và cách chúng kết nối với các dịch vụ bên ngoài và môi trường CI/CD.

------------------------------------------------------------------------------------------------