**SAIGON UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**

**A blue circle with white text

Description automatically generated**

REPORT

**MICROSERVICE CHO FOODFAST DELIVERY**

**Advisor: Dr. Nguyễn Quốc Huy**

**Class: DCT122C5**

**Group 11:**

1. **Vo Kieu Anh - 3122411009**
2. **Ho Đang Khoa - 3122411095**

Ho Chi Minh City, September 2025

**MỤC LỤC**

[**A. PRODUCT REQUIREMENT (PRD)** 4](#_Toc210292625)

[I. Product Requirements Document (PRD) – FoodFast Delivery System 4](#_Toc210292626)

[Problem Alignment 4](#_Toc210292627)

[High Level Approach 4](#_Toc210292633)

[Goals 4](#_Toc210292644)

[II. Services & Requirements 4](#_Toc210292649)

[III. Solution Alignment 6](#_Toc210292661)

[IV. Key Flows 7](#_Toc210292688)

[V. LAUNCH PLAN 9](#_Toc210292695)

[VI. APPENDIX 9](#_Toc210292696)

[**B. BACKEND** 9](#_Toc210292697)

[**C. TEST-CASE** 9](#_Toc210292698)

# **A. PRODUCT REQUIREMENT (PRD)**

## Product Requirements Document (PRD) – FoodFast Delivery System

### Problem Alignment

* Người dùng muốn đặt đồ ăn nhanh nhưng các ứng dụng hiện tại giao bằng shipper, thường bị kẹt xe → chậm trễ.
* Quá trình thanh toán và theo dõi đơn hàng chưa thật sự hiện đại và minh bạch.
* Cửa hàng khó quản lý đơn hàng và trạng thái giao hàng trong thời gian thực.
* Xu hướng mới cần giao hàng nhanh, tự động, không phụ thuộc vào con người → dùng drone để giao.

### Các thành phần chính

* **User (Khách hàng)**:
  + Đăng ký, đăng nhập.
  + Chọn nhà hàng → chọn món ăn → đặt đơn hàng.
  + Thanh toán và theo dõi tình trạng đơn hàng & drone.
* **Restaurant (Nhà hàng)**:
  + Nhận đơn từ hệ thống.
  + Chuẩn bị món ăn.
  + Bàn giao cho Drone để giao.
* **Admin**:
  + Quản lý nhà hàng, món ăn.
  + Giám sát các giao dịch, trạng thái drone, hệ thống.
  + Có thể can thiệp nếu lỗi (manual override).
* **Drone**:
  + Nhận đơn đã thanh toán và xác nhận từ hệ thống.
  + Thực hiện hành trình giao hàng: **ordering → processing → delivering → done.**
  + Mỗi giao dịch kết thúc khi Drone hoàn thành chuyến giao.
* **Server (Microservices)**:
  + **User Service**: quản lý người dùng.
  + **Product/Restaurant Service**: quản lý nhà hàng & món ăn.
  + **Order Service**: tạo và xử lý đơn hàng.
  + **Payment Service**: xử lý thanh toán.
  + **Drone Service**: phân công drone, cập nhật trạng thái bay.
  + **Monitoring/Transaction**: giám sát toàn bộ quy trình, lưu log giao dịch.

### High Level Approach

* Xây dựng hệ thống microservices với các dịch vụ chính:
  + User Service
  + Restaurant Service (quản lý nhà hàng & món ăn, có Admin quản lý)
  + Order Service
  + Payment Service
  + Drone Service (quản lý drone, lịch bay, trạng thái giao hàng)
* Dùng API Gateway để route request từ client.
* Dùng RabbitMQ để giao tiếp bất đồng bộ giữa các services.
* Mỗi service có database riêng → độc lập, dễ scale.
* Hỗ trợ thanh toán online, quản lý đơn hàng & trạng thái giao bằng drone real-time.

### Goals

* Người dùng có thể: đăng ký, đăng nhập, chọn nhà hàng → chọn món ăn → đặt hàng → thanh toán → theo dõi trạng thái drone giao hàng.
* Nhà hàng có thể: quản lý menu, cập nhật tồn kho, xử lý đơn hàng.
* Admin có thể: quản lý danh sách nhà hàng, phê duyệt hoặc khóa nhà hàng.
* Hệ thống có thể: điều phối drone cho từng đơn hàng, cập nhật trạng thái drone theo lộ trình (đang bay, đã giao, kết thúc).

## Services & Requirements

* + - 1. **User Service**

**Functions** :

* Đăng ký, đăng nhập (JWT).
* Quản lý thông tin người dùng (tài khoản, địa chỉ giao hàng).
* Cung cấp API cho Order & Payment Service khi cần.

**Database** :

* Users(id, name, email, password\_hash, address, phone, created\_at).

**2. Product Service**

**Functions** :

* CRUD nhà hàng (Admin).
* CRUD menu/món ăn (chủ nhà hàng).
* Quản lý tồn kho theo món ăn.

**Database:**

* Restaurants(id, name, address, owner\_id, status).
* Dishes(id, restaurant\_id, name, description, price, stock).

**3. Order Service**

**Functions** :

* + Tạo đơn hàng mới.
  + Gửi sự kiện order.created.
  + Nhận sự kiện payment.event để cập nhật trạng thái.
  + Gọi Drone Service để phân công drone giao hàng cho mỗi đơn.

**Database** :

* Orders(id, user\_id, items, total\_amount, status, created\_at).
* OrderItems(id, order\_id, product\_id, quantity, price).

**4. Payment Service**

**Functions** :

* Subscribe order.created để xử lý thanh toán.
* Kết quả thanh toán → publish payment.event.
* Quản lý lịch sử thanh toán.

**Database** :

* Payments(id, order\_id, user\_id, amount, status, method, created\_at).

***(Optional Future) Notification Service***

* Subscribe các sự kiện payment.event, drone.updated → gửi thông báo cho người dùng.

## Solution Alignment

**A diagram of a computer network

AI-generated content may be incorrect.**

**Key Features (FoodFast Delivery – Drone)**

1. User Experience & Ordering
   * Người dùng đăng ký, đăng nhập.
   * Duyệt danh sách nhà hàng, menu món ăn từ Product API.
   * Tạo đơn hàng thông qua Order API, theo dõi trạng thái real-time.
2. Restaurant Management
   * Nhà hàng quản lý menu, tồn kho qua Product API.
   * Tiếp nhận đơn đặt từ Order API và xác nhận chuẩn bị món.
3. Payment & Transaction Tracking
   * Thanh toán online thông qua Payment API (kết nối ví điện tử/PayPal/Ngân hàng).
   * Toàn bộ giao dịch được ghi lại trong Payment Database, đảm bảo minh bạch.
4. Drone Delivery Integration *(future extension)*
   * Sau khi thanh toán thành công, hệ thống sẽ phân công Drone giao đơn hàng.
   * Trạng thái Drone (ready → delivering → completed) được đồng bộ với Order API để hiển thị cho User.
5. Admin Dashboard
   * Admin có thể quản lý toàn bộ hệ thống thông qua User API và Product API.
   * Bao gồm: thêm/xóa nhà hàng, giám sát đơn hàng, trạng thái thanh toán, trạng thái drone.
6. Monitoring & Logging
   * Tất cả service (User, Product, Order, Payment) đều có log tập trung.
   * Monitoring dashboard giúp theo dõi health-check và hiệu suất hệ thống theo thời gian thực.

**Future Considerations**

* Drone Service: tích hợp hệ thống điều khiển drone, tracking GPS.
* Recommendation Engine: gợi ý nhà hàng/món ăn dựa trên lịch sử đặt hàng.
* Caching Layer: giảm độ trễ khi load menu & trạng thái đơn hàng.
* Scalability: hỗ trợ nhiều drone, nhiều đơn hàng đồng thời**Top of Form**

## Key Flows

**A diagram of a customer service

AI-generated content may be incorrect.**

**1. User Flow**

* User register/login qua User Service → lưu user info (bcrypt, JWT).
* User view menu (Product Service → product DB).
* User add item & create order (Order Service → order DB).
* User checkout & payment (Payment Service → update order status + payment DB).
* User receive notifications (Notification Service → email/sms).
* User có thể track order status (Order Service).

**2. Order & Payment Flow**

* **Order Service**:
  + create order → publish order.created.
* **Payment Service**:
  + subscribe order.created → process payment → publish payment.event.
* **Order Service**:
  + subscribe payment.event → update order status (paid/failed).
* **Notification Service**:
  + subscribe payment.event → gửi email invoice hoặc payment failed.
* **Refund Flow (mới)**:
  + nếu order.canceled → Payment Service xử lý refund → publish refund.event → Notification Service gửi thông báo hoàn tiền.

**3. Restaurant Flow (bổ sung)**

* Order Service gửi order.confirm.request đến Restaurant System.
* Restaurant confirm/reject → gửi lại order.confirmed / order.rejected.
* Nếu confirmed → Order Service cập nhật trạng thái processing → Notification Service gửi email/sms cho User.

**4. Admin Flow**

* Admin login (User Service với role admin).
* Admin có thể:
  + Quản lý menu (CRUD qua Product Service).
  + Theo dõi đơn hàng (Order Service).
  + Kiểm soát thanh toán (Payment Service).
* Tất cả hành động được log → Centralized Logging (ELK stack)

**5. Security & Monitoring Flow**

* Tất cả request đi qua API Gateway (JWT validation, HTTPS).
* AWS KMS quản lý secret key & encrypt DB connection.
* Monitoring: Health check API từ mỗi service + centralized logging (ELK).
* Alert: Nếu downtime > 1 phút → gửi cảnh báo cho admin.

## LAUNCH PLAN

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

## APPENDIX

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

# **B. BACKEND**

# **C. TEST-CASE**