**SAIGON UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**

**A blue circle with white text

Description automatically generated**

PRODUCT REQUIREMENT DESIGN

**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

**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.
* Xu hướng mới cần giao hàng nhanh, tự động, không phụ thuộc vào con người, tránh kẹt xe → dùng drone để giao.

### Stakeholders:

|  |  |
| --- | --- |
| Vai trò | Quyền hạn chính |
| Khách hàng (User) | Đăng ký, đặt món, thanh toán, theo dõi đơn hàng. |
| Nhà hàng (Restaurant) | Quản lý menu, xử lý đơn, bàn giao cho drone. |
| Admin | Quản lý toàn bộ hệ thống (khách hàng, nhà hàng, món ăn, đơn hàng). |
| Drone System | Thực hiện giao hàng tự động và cập nhật trạng thái hành trình. |

### System Overview

Hệ thống được phát triển theo kiến trúc Microservices, với các thành phần chính:

|  |  |  |
| --- | --- | --- |
| Service | Nhiệm vụ chính | Database riêng |
| User Service | Quản lý tài khoản (khách hàng, nhà hàng, admin) | user\_db |
| Product Service | Quản lý menu món ăn và nhà hàng | product\_db |
| Order Service | Tạo và cập nhật đơn hàng | order\_db |
| Payment Service | Xử lý thanh toán qua VNPay | payment\_db |
| Drone Service *(Future)* | Điều phối, theo dõi hành trình drone | drone\_db |

→ Các service giao tiếp bất đồng bộ thông qua message broker (Kafka) theo mô hình event-driven architecture, và được triển khai, quản lý, scale bằng Kubernetes.

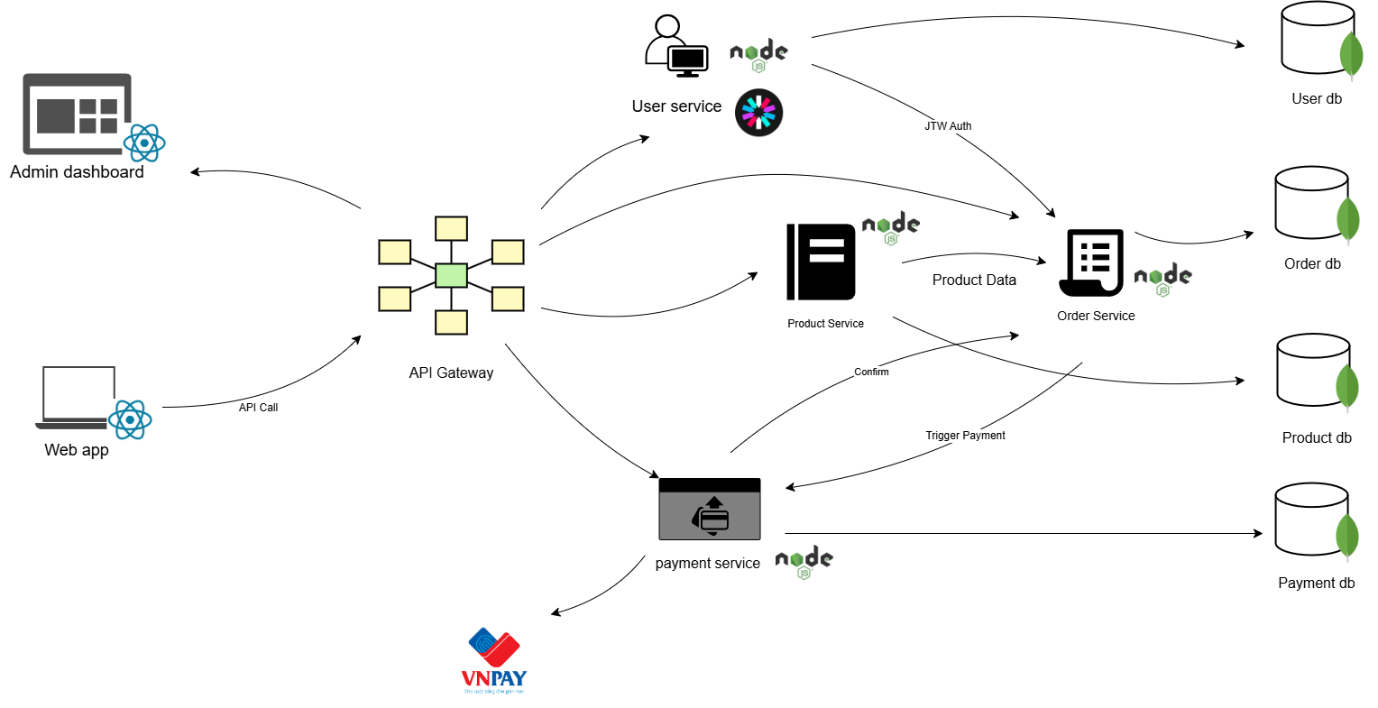
### High Level Approach

* Xây dựng hệ thống microservices với các dịch vụ chính:
  + User Service (nhà hàng và khách hàng)
  + Product Service (nhà hàng quản lý món ăn, admin quản lý nhà hàng)
  + Order Service
  + Payment Service
* Dùng API Gateway để route request từ client.
* Dùng Kafka để giao tiếp bất đồng bộ giữa các services.
* Mỗi service có database riêng → độc lập, dễ scale.
* Hỗ trợ thanh toán online qua VNPay, 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: đăng ký mở nhà hàng và đợi duyệt, quản lý menu, cập nhật trạng thái món ăn, cập nhật trạng thái nhà hàng (đóng/mở), xử lý đơn hàng.
* Admin có thể: quản lý khách hàng, quản lý danh sách nhà hàng, phê duyệt hoặc khóa nhà hàng/khách hàng.
* Hệ thống có thể: điều phối drone cho từng đơn hàng.
* Đem lại trải nghiệm theo dõi đơn hàng realtime bằng công nghệ drone.

### Solution Alignment

****

**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
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 product

AI-generated content may be incorrect.**

**1. Customer Flow**

* **Customer** register/login qua (User Service) *→ lưu user info (bcrypt, JWT).*
* **Customer** view menu (Product Service)
* Duyệt danh sách nhà hàng & món ăn (Product Service).
* **Customer** add item & create order (Order Service)
* **Customer** thanh toán VNPay (Payment Service).
* **Customer** có thể view order status (Order Service).

**2. Order & Payment Flow**

* **Order Service**:
  + create order → publish order.created.
* **Payment Service**:
  + subscribe order.created → process payment → xử lý thanh toán VNPay → publish payment.event → notification gửi email/sms.
* **Order Service**:
  + subscribe payment.event → update order status (paid/failed).
* **Drone Service:** (nếu có) nhận đơn đã thanh toán → bắt đầu hành trình giao hàng.
* **Refund Flow (mới)**:
  + nếu order.canceled → Payment Service xử lý refund → publish refund.event → notification gửi email/sms cho User.

**3. Restaurant Flow**

* Order Service gửi order.confirm.request đến Restaurant System.
* Nhà hàng nhận đơn tự động trên app → trạng thái mặc định “preparing”
* Nếu xác nhận “Completed” → Order Service cập nhật trạng thái “delivering” để drone giao hàng → Notification Service gửi email/sms cho User.
* Khi hoàn tất drone quay về và nhà hàng xác nhận “Completed” → trạng thái “Done”.

**4. Admin Flow**

* Admin login (với role admin).
* Admin có thể:
  + Theo dõi đơn hàng (Order Service).
  + Quản lý khách hàng, nhà hàng (User Service).
  + Kiểm soát thanh toán (Payment Service).
  + Theo dõi món ăn (Product Service).
* Giám sát log hệ thống.
* Có thể khóa / mở tài khoản hoặc can thiệp lỗi.

### Technical Architecture

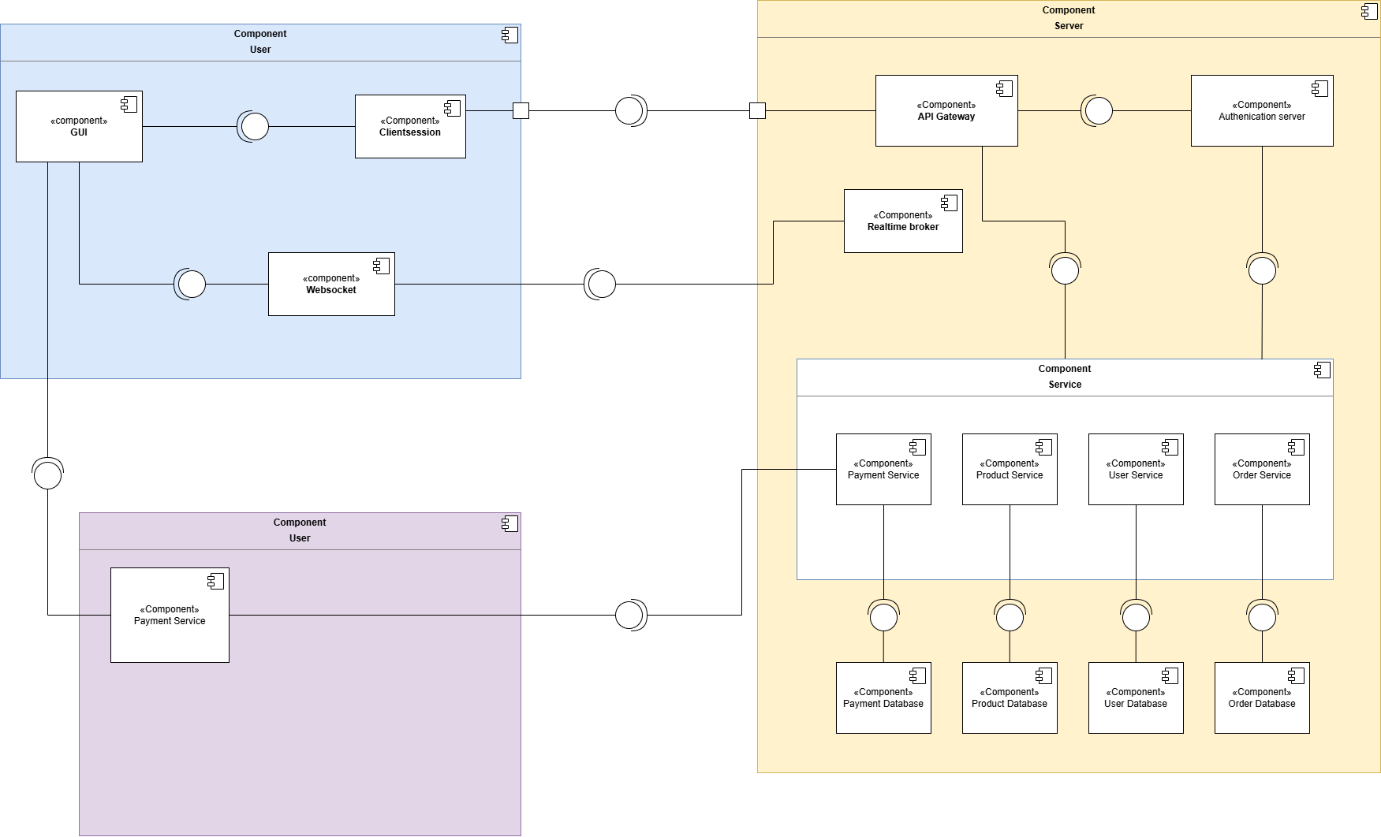
* **API Gateway:** Route request & validate JWT.
* **Event Bus (Kafka):** Giao tiếp bất đồng bộ giữa services.
* **Kubernetes Deployment:** Tự động scale, rolling update.
* **Database per service:** Đảm bảo tính độc lập và tránh coupling.
* **Security:** HTTPS, JWT, VNPay SDK, AES encryption cho dữ liệu nhạy cảm.

### Key Functional Requirements

|  |  |  |
| --- | --- | --- |
| Service | Chức năng chính | Actor |
| **User Service** | Đăng ký, đăng nhập, quản lý hồ sơ | User, Admin |
| **Product Service** | CRUD món ăn, tồn kho | Restaurant, Admin |
| **Order Service** | Tạo đơn, cập nhật trạng thái | User, Restaurant |
| **Payment Service** | Thanh toán VNPay, lịch sử giao dịch | User |
| **Drone Service** (Future) | Phân công & tracking drone | System |

**Non-Functional Requirements**

* **Scalability:** Mỗi service có thể scale độc lập.
* **Resilience:** Có cơ chế retry, circuit breaker.
* **Security:** Bảo mật token, dữ liệu người dùng, và kết nối VNPay.
* **Monitoring:** Real-time metrics & alerting.
* **Extensibility:** Dễ mở rộng với Drone Service, Recommendation Engine.
* **Component Diagram**



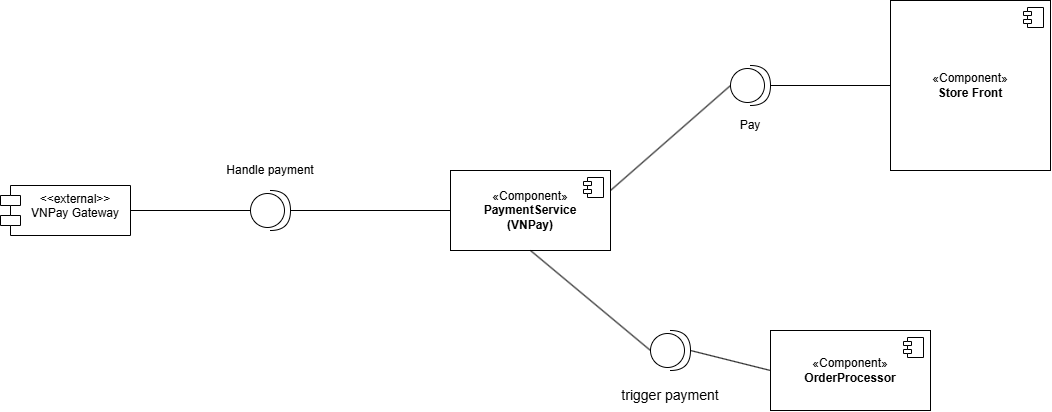
*Hình1: Lược đồ Component*

* **Component xử lý đặt hàng**



*Hình 2: Component thành phần xử lý đặt hàng*

* **Component thanh toán**



*Hình 3: Component thanh toán*

* **ERD:**

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 4: ERD tổng quan toàn hệ thống

* **Chia theo service:**

**User-service:**

**A diagram of a diagram

AI-generated content may be incorrect.**

**Product-service:**

**A diagram of a diagram

AI-generated content may be incorrect.**

**Order-service:**

**A diagram of a diagram

AI-generated content may be incorrect.**

**Payment\_service:**

**A diagram of a diagram

AI-generated content may be incorrect.**

* **Delpoyment:**
* **Class Diagram:**

**A diagram of a restaurant

AI-generated content may be incorrect.**

*Hình 5: Sơ đồ lớp*

* **FocusUsecase:**

|  |  |  |
| --- | --- | --- |
|  | | |
| Use Case Name: | Place order proccessing | |
| Actor (s): | Customer | |
| Maturity: | Focused | |
| Summary: | Khách hàng duyệt menu, chọn món ăn, đặt đơn hàng và thanh toán trực tuyến thông qua cổng VNPAY. | |
| Basic Course of Events: | Actor Action | System Response |
| 1. Customer mở ứng dụng/web và duyệt menu món ăn từ các nhà hàng. |  |
|  | 2. Hệ thống hiển thị danh sách món ăn và thông tin chi tiết (giá, mô tả, trạng thái còn hàng).**A1** |
| 3. Customer chọn món và thêm vào giỏ hàng. **E1** |  |
|  | 4. Hệ thống cập nhật giỏ hàng. |
| 5. Customer chọn xem giỏ hàng. |  |
|  | 6. Hệ thống hiển thị thông tin giỏ hàng và tổng tiền tạm tính. |
| 7.Customer nhận “Tiến hành đặt hàng”. |  |
|  | 8.Hệ thống hiển thị đơn hàng tạm tính. |
| 7. Customer nhấn “Đặt hàng ”để xác nhận đơn hàng.**A3** |  |
|  | 8. Hệ thống tạo đơn hàng ở trạng thái *pending* tính tổng tiền và chuyển hướng đến cổng thanh toán VNPay. |
| Alternative Paths: | **A1**. Customer có thể tìm kiếm món ăn khác của nhà hàng đang chọn. | |
| Actor Action | System Response |
| 1. Khách hàng nhấn vào biểu tượng “kính lúp” ở trang món ăn của nhà hàng và tìm món. |  |
|  | 2. Hệ thống hiển thị danh sách món ăn liên quan hoặc món khách hàng cần. |
| **A2**. Khách hàng hủy đơn trước khi thanh toán | |
| Actor Action | System Response |
| 1. Customer nhấn “Hủy” tại bước thanh toán. |  |
|  | 2. Hệ thống hủy đơn hàng, cập nhật trạng thái *cancelled* và quay về trang giỏ hàng. |
| Exception Paths: | **E1:** Chưa xác thực đăng nhập  Khách hàng thêm vào giỏ hàng nhưng chưa đăng kí/đăng nhập thì hệ thống yêu cầu xác thực đăng kí đăng nhập. | |
| Extension Points: |  | |
| Preconditions: | Khách hàng đã đăng nhập vào hệ thống.  Nhà hàng có món ăn đang hoạt động (*status = open*). | |
| Post Conditions: |  | |
| A diagram of a flowchart  AI-generated content may be incorrect. | | |

|  |  |  |
| --- | --- | --- |
|  | | |
| Use Case Name: | Payment proccessing | |
| Actor (s): | Customer, VNPay gateway | |
| Maturity: | Focused | |
| Summary: | Sau khi thanh toán thành công, hệ thống xác nhận đơn hàng và thông báo cho nhà hàng chuẩn bị món. | |
| Basic Course of Events: | Actor Action | System Response |
| 1. Customer chọn “Đặt hàng” và chuyển đến bước thanh toán. |  |
|  | 2. Frontend gửi yêu cầu tạo liên kết thanh toán đến Backend |
|  | 3. Backend tạo mã xác thực (auth\_code). |
|  | 4. Backend gửi yêu cầu đến VNPay để tạo URL thanh toán. |
|  | 5. Backend trả lại URL thanh toán cho Frontend và giao diện chuyển hướng sang thanh toán VNPay. |
| 6. Customer nhập thông tin thẻ / tài khoản và xác nhận thanh toán. |  |
|  | 7. VNPay xử lý giao dịch và gửi phản hồi (Webhook) về hệ thống.**E1** |
|  |  | 8. Backend kiểm tra xác thực và cập nhật trạng thái đơn hàng = “paid”. |
|  |  | 9. Hệ thống hiển thị thông báo “Thanh toán thành công” và lưu lịch sử giao dịch. |
| Exception Paths: | **E1**. Thanh toán thất bại  1. VNPay trả về thông báo lỗi hoặc giao dịch không thành công.  2. Hệ thống cập nhật đơn hàng ở trạng thái *failed*, hiển thị thông báo lỗi và cho phép người dùng thử lại và quay về bước chi tiết đơn hàng. | |
| Extension Points: | **EP1**: Apply Voucher  Có thể mở rộng để khách hàng nhập mã giảm giá trước khi thanh toán. | |
| Preconditions: | Cổng thanh toán VNPay khả dụng. | |
| Post Conditions: | Nếu thanh toán thành công:   * Đơn hàng được ghi nhận với trạng thái *paid*. * Hệ thống gửi thông báo đến nhà hàng. * Biên lai thanh toán được lưu trong lịch sử giao dịch của khách hàng.   Nếu thanh toán thất bại hoặc bị hủy:   * Đơn hàng ở trạng thái *failed* hoặc *cancelled*. | |
|  | | |

* **Screen:**

**A screenshot of a food menu

AI-generated content may be incorrect.**

**A screenshot of a food order

AI-generated content may be incorrect.**

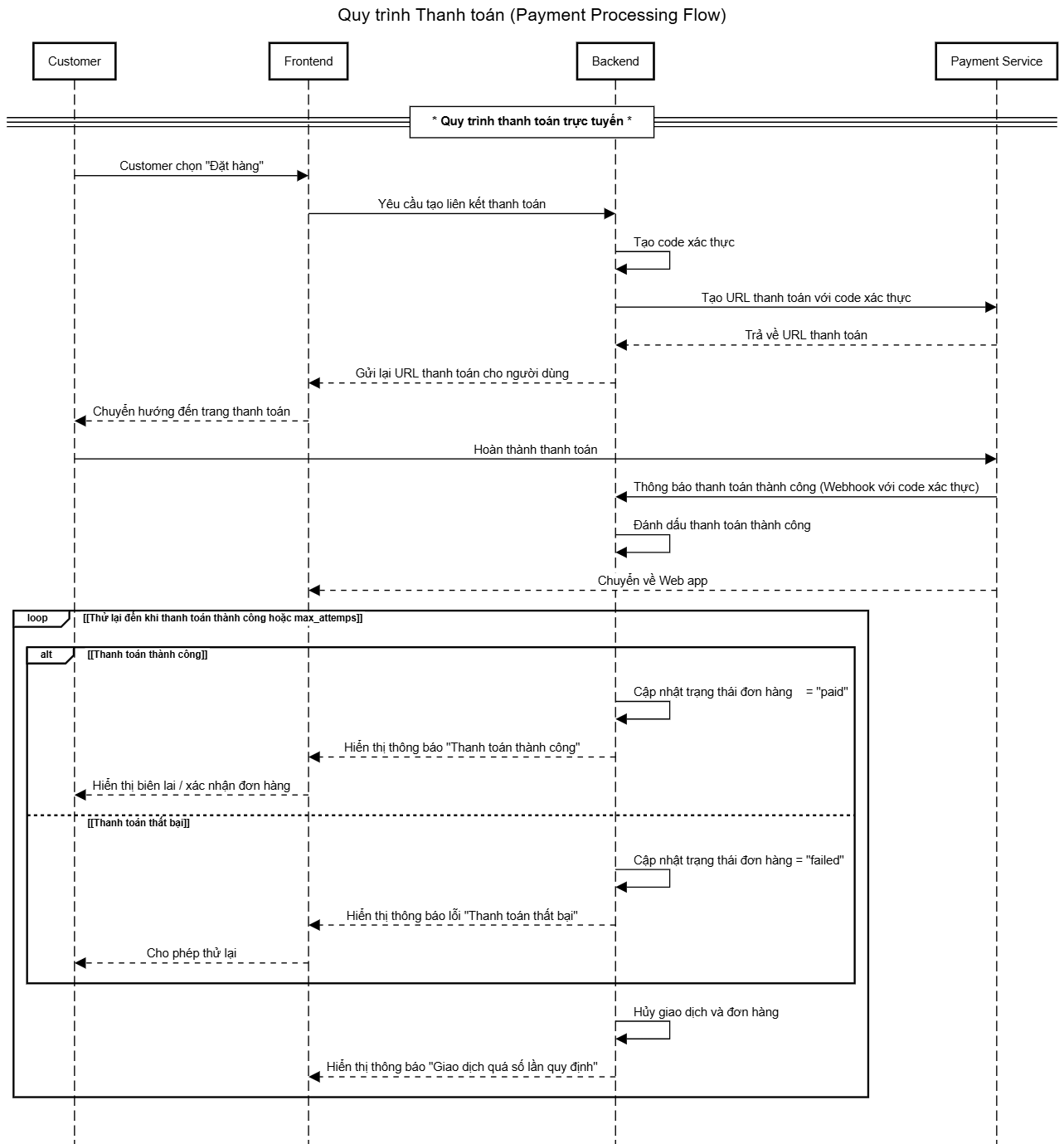
**A screenshot of a login form

AI-generated content may be incorrect.**

* **Sequence:**

A white sheet of paper with black lines

AI-generated content may be incorrect.



## LAUNCH PLAN

|  |  |  |
| --- | --- | --- |
| **TARGET DATE** | **MILESTONE** | **DESCRIPTION** |
| **Sprint 1** |  |  |
|  |  |  |

## APPENDIX

|  |  |
| --- | --- |
| **DATE** | **DESCRIPTION** |
|  |  |
|  |  |