

SAIGON UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGY



BACKEND DESIGN
MICROSERVICE CHO FOODFAST DELIVERY

Advisor: Dr. Nguyễn Quốc Huy

Class: DCT122C5

Group 11:

- | | |
|------------------------|---------------------|
| 1. Vo Kieu Anh | - 3122411009 |
| 2. Ho Dang Khoa | - 3122411095 |

Ho Chi Minh City, September 20

BACKEND DESIGN

Sub-system:

- User service
- Product service (Restaurant + Dish)
- Order service (Cart + Order)
- Payment service (VNPAY integration)

App:

- Web app (Next.js frontend, NodeJS backend).
- Backend triển khai theo microservices (NodeJS + REST/gRPC).
- Message broker: RabbitMQ để truyền sự kiện giữa Order ↔ Payment ↔ User.
- Authentication: JWT-based (role: customer, restaurant_owner, admin).
- Deployment: Docker containers, có thể mở rộng lên Kubernetes nếu cần.

Data:

- User service: MySQL (User, CustomerProfile, Role).
- Order service: MySQL (Cart, CartItem, Order, OrderItem).
- Payment service: MySQL (Payment log, transaction history).
- Product service: PostgreSQL (Restaurant, Dish, menu info) → linh hoạt schema cho thay đổi món ăn/topping.
- Caching: Redis (giỏ hàng active, session login, recent orders).

Route (API):

- User: /api/user/register, /api/user/login, /api/user/profile
- Product: /api/restaurants, /api/restaurants/:id/dishes
- Order: /api/cart, /api/order, /api/order/:id/status
- Payment: /api/payment/checkout, /api/payment/callback
- Monitoring (Grafana + Prometheus):
- Metrics mặc định (CPU, RAM, request latency, error rate).
- Business metrics: số đơn hàng/ngày, tỷ lệ thanh toán thành công.

- Logging: ELK stack (Elasticsearch + Logstash + Kibana).

Database:

- PostgreSQL: cho User, Order, Payment, Product (ACID, transaction consistency, dễ viết báo cáo thống kê).
- Redis: cho session & cache (tăng tốc độ phản hồi).

```
User
- user_id (PK)
- email
- password_hash
- role (customer, restaurant_owner, admin, etc.)
- created_at
```

```
CustomerProfile
- customer_id (PK, FK -> User.user_id)
- name
- phone
- address
```

```
Restaurant
- restaurant_id (PK)
- owner_id (FK -> User.user_id)
- name
- address
- phone
- status (open/closed)
```

```
Dish
- dish_id (PK)
- restaurant_id (FK -> Restaurant.restaurant_id)
- name
- description
- price
- available (bool)
```

```
Cart
- cart_id (PK)
- customer_id (FK -> User.user_id)
- restaurant_id (FK -> Restaurant.restaurant_id)
- created_at
- status (active/cancelled/checkedout)
```

```
CartItem
- cart_item_id (PK)
- cart_id (FK -> Cart.cart_id)
- dish_id (FK -> Dish.dish_id)
- quantity
```

```
Order
```

- order_id (PK)
- customer_id (FK -> User.user_id)
- restaurant_id (FK -> Restaurant.restaurant_id)
- cart_id (FK -> Cart.cart_id, optional)
- total_amount
- status (pending, confirmed, preparing, completed, cancelled)
- created_at

OrderItem

- order_item_id (PK)
- order_id (FK -> Order.order_id)
- dish_id (FK -> Dish.dish_id)
- quantity
- price

Payment

```

    payment_id      (PK),
    order_id        (FK -> Order.order_id),
    amount          bigint NOT NULL,
    status          enum('pending', 'paid', 'failed', 'refunded') NOT
NULL,
    order_no        varchar UNIQUE, -- Mã giao dịch bạn gửi sang
VNPAY (vnp_TxnRef)
    transaction_no  varchar,         -- Mã giao dịch VNPAY trả về
(vnp_TransactionNo)
    bank_code       varchar,         -- Mã ngân hàng (vnp_BankCode)
    response_code   varchar,         -- Mã phản hồi (vnp_ResponseCode)
    pay_date        timestamp,       -- Ngày giờ thanh toán thành công
(vnp_PayDate)
    created_at      timestamp DEFAULT now(),
    updated_at      timestamp
)

```



