**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**PHẠM BÁ TRUNG HIẾU**

**PHÁT TRIỂN HỆ THỐNG THANH TOÁN RESTFUL API TÍCH HỢP ĐA PHƯƠNG THỨC**

**ĐỒ ÁN NGÀNH**

**NGÀNH KHOA HỌC MÁY TÍNH**

**TP. HỒ CHÍ MINH, 2024BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**PHẠM BÁ TRUNG HIẾU**

**PHÁT TRIỂN HỆ THỐNG THANH TOÁN RESTFUL API TÍCH HỢP ĐA PHƯƠNG THỨC**

**Mã số sinh viên: 2151013024**

**ĐỒ ÁN NGÀNH**

**NGÀNH KHOA HỌC MÁY TÍNH**

**Giảng viên hướng dẫn: TRƯƠNG HOÀNG VINH**

**TP. HỒ CHÍ MINH, 2024**

LỜI CẢM ƠN

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

TÓM TẮT ĐỐ ÁN NGÀNH

ABSTRACT

**MỤC LỤC**

[LỜI CẢM ƠN 1](#_Toc176728817)

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN 2](#_Toc176728818)

[TÓM TẮT ĐỐ ÁN NGÀNH 3](#_Toc176728819)

[ABSTRACT 4](#_Toc176728820)

[DANH MỤC TỪ VIẾT TẮT 7](#_Toc176728821)

[DANH MỤC HÌNH VẼ 8](#_Toc176728822)

[DANH MỤC BẢNG 9](#_Toc176728823)

[MỞ ĐẦU 10](#_Toc176728824)

[Chương 1. TỔNG QUAN ĐỀ TÀI 11](#_Toc176728825)

[1.1. Giới thiệu đề tài 11](#_Toc176728826)

[1.1.1. Mục tiêu và phạm vi đề tài 11](#_Toc176728827)

[1.1.2. Phương pháp nghiên cứu 12](#_Toc176728828)

[1.1.3. Bố cục báo cáo 12](#_Toc176728829)

[Chương 2. CƠ SỞ LÝ THUYẾT 13](#_Toc176728830)

[2.1. Giới thiệu Spring Boot 13](#_Toc176728831)

[2.1.1. Đặc điểm chính 13](#_Toc176728832)

[2.1.2. Kiến trúc 13](#_Toc176728833)

[2.1.3. Cách sử dụng 14](#_Toc176728834)

[2.2. Giới thiệu NextJS 16](#_Toc176728835)

[2.2.1. Đặc điểm chính 16](#_Toc176728836)

[2.2.2. Kiến trúc 16](#_Toc176728837)

[2.2.3. Cách sử dụng 16](#_Toc176728838)

[2.3. Cách dịch vụ và công cụ hỗ trợ 17](#_Toc176728839)

[Chương 3. HỆ THỐNG THANH TOÁN RESTFUL API TÍCH HỢP ĐA PHƯƠNG THỨC 18](#_Toc176728840)

[3.1. Giới thiệu bài toán 18](#_Toc176728841)

[3.2. Phân tích hệ thống 18](#_Toc176728842)

[3.2.1. Lược đồ use case 18](#_Toc176728843)

[3.2.2. Đặc tả use case 19](#_Toc176728844)

[3.3. Thiết kế hệ thống 24](#_Toc176728845)

[3.3.1. Sơ đồ tuần tự (sequence diagram) 24](#_Toc176728846)

[3.3.2. Sơ đồ hoạt động (activity diagram) 27](#_Toc176728847)

[3.3.3. Thiết kế dữ liệu 31](#_Toc176728848)

[3.4. Kiến trúc hệ thống 33](#_Toc176728849)

[3.5. Kết quả đề tài 33](#_Toc176728850)

[Chương 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 34](#_Toc176728851)

[4.1. Kết luận 34](#_Toc176728852)

[4.2. Hướng phát triển 34](#_Toc176728853)

[TÀI LIỆU THAM KHẢO 35](#_Toc176728854)

[PHỤ LỤC 36](#_Toc176728855)

DANH MỤC TỪ VIẾT TẮT

DANH MỤC HÌNH VẼ

[Hình 2‑1. Spring initializr 14](#_Toc176728982)

[Hình 2‑2. Cấu trúc dự án Spring Boot 15](#_Toc176728983)

[Hình 2‑3. Controller Spring Boot 16](#_Toc176728984)

[Hình 2‑4. Chương trình Spring Boot đầu tiên 16](#_Toc176728985)

[Hình 3‑1. Lược đồ use case 18](#_Toc176728986)

[Hình 3‑2. Sơ đồ tuần tự tạo thanh toán 24](#_Toc176728987)

[Hình 3‑3. Sơ đồ tuần tự xác nhận giao dịch 24](#_Toc176728988)

[Hình 3‑4. Sơ đồ tuần tự tra cứu trạng thái giao dịch 25](#_Toc176728989)

[Hình 3‑5. Sơ đồ tuần tự hoàn tiền giao dịch 26](#_Toc176728990)

[Hình 3‑6. Sơ đồ hoạt động tạo thanh toán 27](#_Toc176728991)

[Hình 3‑7. Sơ đồ trạng thái xác nhận giao dịch 28](#_Toc176728992)

[Hình 3‑8. Sơ đồ hoạt động tra cứu trạng thái giao dịch 29](#_Toc176728993)

[Hình 3‑9. Sơ đồ hoạt động hoàn tiền giao dịch 30](#_Toc176728994)

[Hình 3‑10. Lược đồ cơ sở dữ liệu 31](#_Toc176728995)

DANH MỤC BẢNG

[Bảng 3‑1. Use case tạo thanh toán 19](#_Toc176728996)

[Bảng 3‑2. Use case xác nhận giao dịch 21](#_Toc176728997)

[Bảng 3‑3. Use case tra cứu trạng thái 22](#_Toc176728998)

[Bảng 3‑4. Use case hoàn tiền giao dịch 23](#_Toc176728999)

MỞ ĐẦU

# TỔNG QUAN ĐỀ TÀI

## Giới thiệu đề tài

Trong bối cảnh phát triển mạnh của xã hội, các ví điện tử đã và đang được nhiều người tin tưởng và sử dụng để thanh toán cho các giao dịch mua sắm. Chúng đem lại sự tiện lợi, nhanh chóng và hợp xu hướng với giới trẻ. Các dịch vụ mua sắm cũng đang dần tích hợp để theo kịp sự phát triển của thời đại, góp phần tiện lợi cho khách hàng. Tuy nhiên, để có được các dịch vụ thanh toán thông qua ví điện tử đôi khi sẽ lại là khó khăn, nhất là trong việc tích hợp, triển khai và bảo trì, bên cạnh đó cũng tốn nhiều nguồn lực và nhân công.

Đối với các ví điện tử, doanh nghiệp phải tham khảo tài liệu kỹ thuật riêng biệt, tuân thủ các chuẩn và bảo mật của bên thanh toán. Tùy thuộc vào ngôn ngữ và kiến trúc hệ thống, việc hiện thực lại có thể khác nhau. Điều này gây ra khó khăn trong việc xây dựng, bảo trì và tiêu tốn nguồn lực đối với các doanh nghiệp.

Nắm bắt được nhu cầu, đề tài này đề xuất phát triển hệ thống thanh toán Restful API tích hợp đa phương thức. Hệ thống sẽ đóng vai trò là lớp trung gian, cung cấp bộ API chuẩn hóa cho phép doanh nghiệp dễ dàng tích hợp nhiều phương thức thanh toán khác nhau thông qua một hệ thống duy nhất.

### Mục tiêu và phạm vi đề tài

#### Mục tiêu

Xây dựng được hệ thống thanh toán RestFul API tích hợp đa phương thức, cụ thể:

* Phân quyền và chứng thực hai vai trò quan trọng là merchant và admin
* Bộ API chuẩn hóa gồm 4 loại: Tạo thanh toán, xác nhận thanh toán, tra cứu trạng thái và hoàn tiền giao dịch
* Doanh nghiệp có thể gửi yêu cầu tạo, xác nhận, tra cứu và hoàn tiền thanh toán tới các ví điện tử
* Trang Dashboard, cho phép merchant theo dõi, quản lý và thanh toán các giao dịch
* Trang Admin, cho phép quản lý các merchant và phương thức thanh toán

#### Phạm vi

Để đảm bảo được yêu cầu và tiến độ, đề tài sẽ tập trung vào các mục sau:

* Xây dựng hệ thống có kiến trúc có thể mở rộng và có khả năng thêm mới các hình thức thanh toán khác trong tương lai
* Tích hợp các ví điện tử phổ biến (MoMo, ZaloPay, VnPay và PayPal)
* Cung cấp bộ API thanh toán chuẩn với nhiều phương thức thanh toán cho doanh nghiệp
* Trang Dashboard giúp cho doanh nghiệp theo dõi và kiểm soát các giao dịch

### Phương pháp nghiên cứu

#### Phân tích yêu cầu

* Khảo sát thị trường và xu hướng của ví điện tử
* Xác định nhu cầu thanh toán hiện có

#### Xác định nghiệp vụ

#### Vẽ sơ đồ use case và đặc tả, mô tả các tính năng cần thiết

#### Vẽ sơ đồ sequence, mô tả quá trình chi tiết cách hệ thống hoạt động và tương tác

#### Vẽ sơ đồ activity, mô tả luồng đi của hệ thống

#### Thiết kế lược đồ cơ sở dữ liệu, cách hệ thống lưu trữ, tương tác

#### Lựa chọn công nghệ

* Spring Boot: triển khai theo kiến trúc microservices, đáp ứng được khả năng nâng cấp, mở rộng và phát triển
* NextJS: giao diện trực quan, linh hoạt và hiệu suất cao
* PostgreSQL: lưu trữ dữ liệu có cấu trúc, đảm bảo sự nhất quán và ổn định
* MongoDB: lưu trữ dữ liệu phi cấu trúc, đáp ứng khả năng mở rộng và hiệu suất cao.

### Bố cục báo cáo

Bao gồm 4 chương:

* Chương 1: Tổng Quan Đề Tài
* Chương 2: Cơ Sở Lý Thuyết
* Chương 3: Hệ Thống Thanh Toán RESTFUL API tích hợp đa phương thức
* Chương 4: Kết Luận Và Hướng Phát Triển

# CƠ SỞ LÝ THUYẾT

## Giới thiệu Spring Boot

Spring Boot là một framework giúp phát triển các ứng dụng trên nền Java. Nó cung cấp công cụ mạnh mẽ để phát triển các ứng dụng web và microservices một cách nhanh chóng. Spring Boot là một phần mở rộng được phát triển lên từ spring framework, cho nên được thừa hưởng toàn bộ lợi ích của nó.

Bên cạnh đó, cộng đồng Spring rất mạnh mẽ, hỗ trợ nhiều tiện ích khác nhau, có thể kể như Spring Security, Spring Cloud, … Hoàn toàn tương thích và dễ dàng tích hợp.

### Đặc điểm chính

* Auto Configurations
* Standalone
* Opinionated
* Production-ready

### Kiến trúc

Spring Boot theo mô hình MVC (Model -View – Controller) trong việc phát triển các ứng dụng

Client: sẽ gửi các Https request lên hệ thống

Controller: Nơi xử lý các request từ user và gửi xuống service layer

Service Layer: Nơi xử lý các nghiệp vụ, sẽ bao gồm hai phần, Service và Repository

Model: Các thực thể đại diện cho các thuộc tính trong cơ sở dữ liệu

### Cách sử dụng



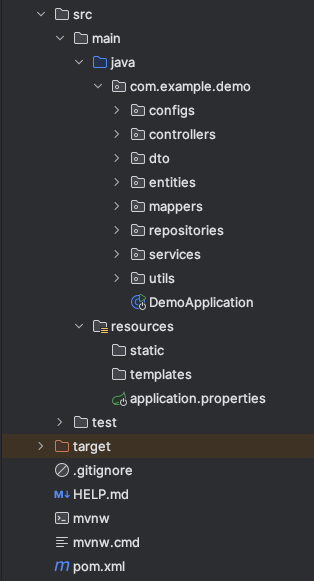
Hình ‑1. Spring initializr

Spring Initializr là công cụ giúp khởi tạo nhanh cấu trúc dự án Spring Boot một cách đơn giản.

Ở đây, ta có thể cấu hình phiên bản Spring Boot sẽ sử dụng và thêm các dependencies cần thiết.

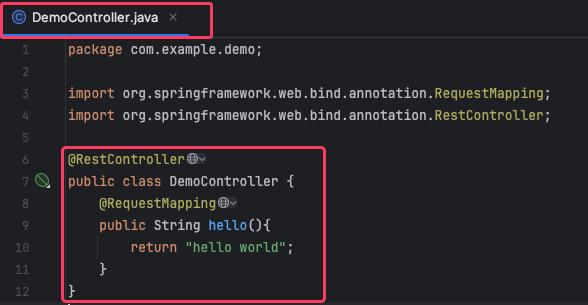
Sau khi hoàn thành cấu hình, ta bấm generate để tải dự án.

Cấu trúc một dự án sử dụng Spring Boot bao gồm:



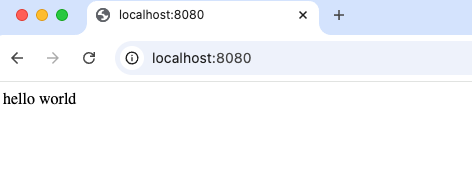
Hình ‑2. Cấu trúc dự án Spring Boot

Ta tạo một controller đơn giản



Hình ‑3. Controller Spring Boot

Chạy dự án và truy cập ***localhost:8080***



Hình ‑4. Chương trình Spring Boot đầu tiên

## Giới thiệu NextJS

### Đặc điểm chính

### Kiến trúc

### Cách sử dụng

## Cách dịch vụ và công cụ hỗ trợ

# HỆ THỐNG THANH TOÁN RESTFUL API TÍCH HỢP ĐA PHƯƠNG THỨC

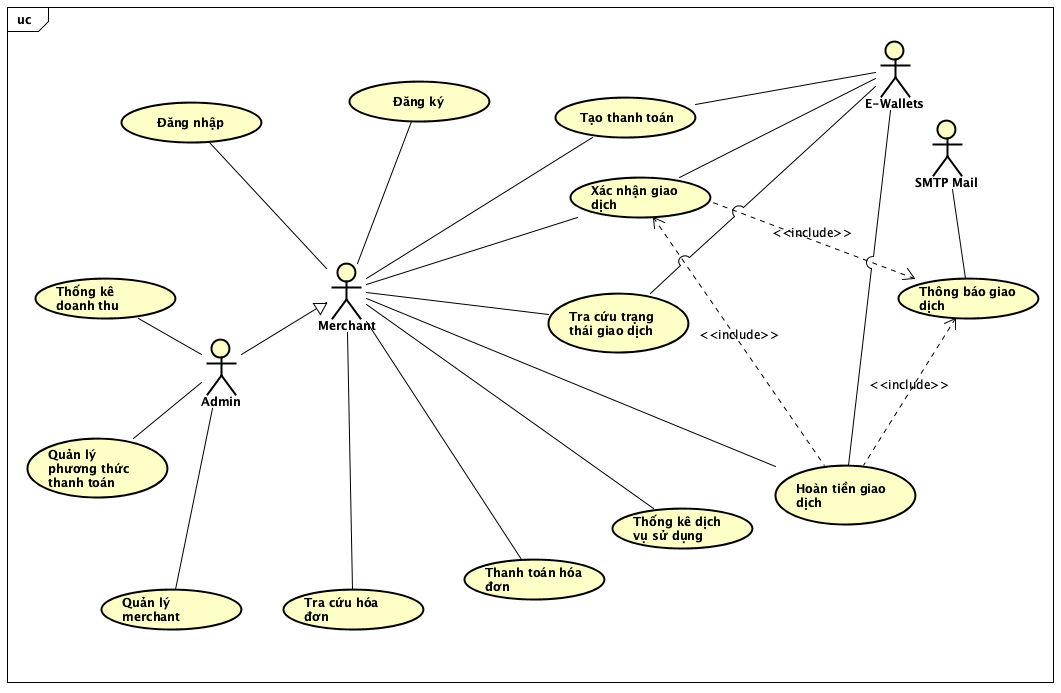
## Giới thiệu bài toán

Trong bối cảnh xã hội phát triển, các ví điện tử đang ngày càng phổ biến, đặc biệt là giới trẻ. Các dịch vụ mua sắm hiện nay đang dần sử dụng để bắt kịp xu hướng của xã hội. Tuy nhiên, để có thể tích hợp nhiều ví điện tử vào một hệ thống đôi khi sẽ rất tốn kém. Ngoài ra, mỗi ví điện tử sẽ có tài liệu kỹ thuật và chuẩn riêng, điều năng gây phức tạp hóa quy trình tích hợp và triển khai.

Giải quyết vấn đề trên, đề tài đề xuất giải pháp đóng vai trò là hệ thống trung gian giữa các doanh nghiệp và các ví điện tử. Hệ thống sẽ cung cấp giải pháp API thanh toán đa phương thức, cho phép doanh nghiệp dễ dàng kết nối với nhiều phương thức thanh toán khác nhau. Giải pháp sẽ giúp doanh nghiệp đơn giản hóa và tiết kiệm chi phí, đồng thời vẫn đảm bảo được hiệu năng và bảo mật.

## Phân tích hệ thống

### Lược đồ use case



Hình ‑1. Lược đồ use case

### Đặc tả use case

#### Use case tạo thanh toán

|  |  |
| --- | --- |
| **Use case Id** | UC-01 |
| **Tên use case** | Tạo thanh toán |
| **Mô tả** | Merchant gửi yêu cầu tạo thanh toán với phương thức tương ứng |
| **Actor chính** | Merchant |
| **Actor Phụ** | E-Wallets, SMTP Mail |
| **Tiền điều kiện** | Không |
| **Hậu điều kiện** | Nhận thông tin giao dịch thanh toán thông qua hệ thống |
| **Luồng hoạt động** | 1. Merchant gửi yêu cầu tạo thanh toán 2. Hệ thống xác thực thông tin merchant 3. Hệ thống lưu giao dịch mới 4. Hệ thống xác định ví điện tự theo yêu cầu của merchant 5. Hệ thống gửi yêu cầu thanh toán đến ví điện tử tương ứng 6. Ví điện tử xử lý yêu cầu và trả kết quả 7. Hệ thống cập nhật trạng thái giao dịch 8. Hệ thống trả thông tin thanh toán cho merchant |
| **Luồng thay thế** | 2a. Xác thực thất bại   1. Hệ thống báo lỗi cho merchant 2. Use case kết thúc   6a.Ví điện tử không phản hồi   1. Hệ thống thử lại 2. Chuyển sang luồng xử lý ngoại lệ   6b. Tạo thanh toán thất bại   1. Hệ thống cập nhật trạng thái giao dịch 2. Tiếp tục bước 8 |
| **Luồng ngoại lệ** | 1. Lưu request log 2. Thông báo cho merchant về trạng thái giao dịch |

Bảng ‑1. Use case tạo thanh toán

#### Use case xác nhận giao dịch

|  |  |
| --- | --- |
| **Use case Id** | UC-02 |
| **Tên use case** | Xác nhận giao dịch |
| **Mô tả** | Merchant xác nhận đã thanh toán với giao dịch của ví điện tử |
| **Actor chính** | Merchant |
| **Actor Phụ** | E-Wallets, SMTP Mail |
| **Tiền điều kiện** | Không |
| **Hậu điều kiện** | Xác nhận thành công thanh toán |
| **Luồng hoạt động** | 1. Merchant gửi yêu cầu xác nhận giao dịch 2. Hệ thống xác thực thông tin merchant 3. Hệ thống xác định ví điện tự theo yêu cầu của merchant 4. Hệ thống gửi yêu cầu xác nhận thanh toán 5. Ví điện tử xử lý yêu cầu và trả kết quả 6. Hệ thống cập nhật trạng thái giao dịch 7. Hệ thống trả kết quả thanh toán cho merchant 8. Hệ thống gửi thông báo về kết quả giao dịch thông qua mail |
| **Luồng thay thế** | 2a. Xác thực thất bại   1. Hệ thống báo lỗi cho merchant 2. Use case kết thúc   6a.Ví điện tử không phản hồi   1. Hệ thống thử lại 2. Chuyển sang luồng xử lý ngoại lệ   6b. Tạo thanh toán thất bại   1. Hệ thống cập nhật trạng thái giao dịch 2. Tiếp tục bước 8 |
| **Luồng ngoại lệ** | 1. Lưu request log 2. Thông báo cho merchant về trạng thái giao dịch |

Bảng ‑2. Use case xác nhận giao dịch

#### Use case tra cứu trạng thái giao dịch

|  |  |
| --- | --- |
| **Use case Id** | UC-03 |
| **Tên use case** | Tra cứu trạng thái giao dịch |
| **Mô tả** | Cho phép merchant tra cứu trạng thái giao dịch |
| **Actor chính** | Merchant |
| **Actor Phụ** | E-Wallets |
| **Tiền điều kiện** | Không |
| **Hậu điều kiện** | Nhận kết quả trạng thái của giao dịch |
| **Luồng hoạt động** | 1. Merchant gửi yêu cầu tra cứu 2. Hệ thống xác thực thông tin merchant 3. Hệ thống xác định ví điện tự theo yêu cầu của merchant 4. Hệ thống gửi yêu cầu tra cứu đến ví điện tử tương ứng 5. Ví điện tử xử lý yêu cầu và trả kết quả 6. Hệ thống trả kết quả tra cứu cho merchant |
| **Luồng thay thế** | 2a. Xác thực thất bại   1. Hệ thống báo lỗi cho merchant 2. Use case kết thúc   6a.Ví điện tử không phản hồi   1. Hệ thống thử lại 2. Chuyển sang luồng xử lý ngoại lệ |
| **Luồng ngoại lệ** | Lưu request log |

Bảng ‑3. Use case tra cứu trạng thái

#### Use case hoàn tiền giao dịch

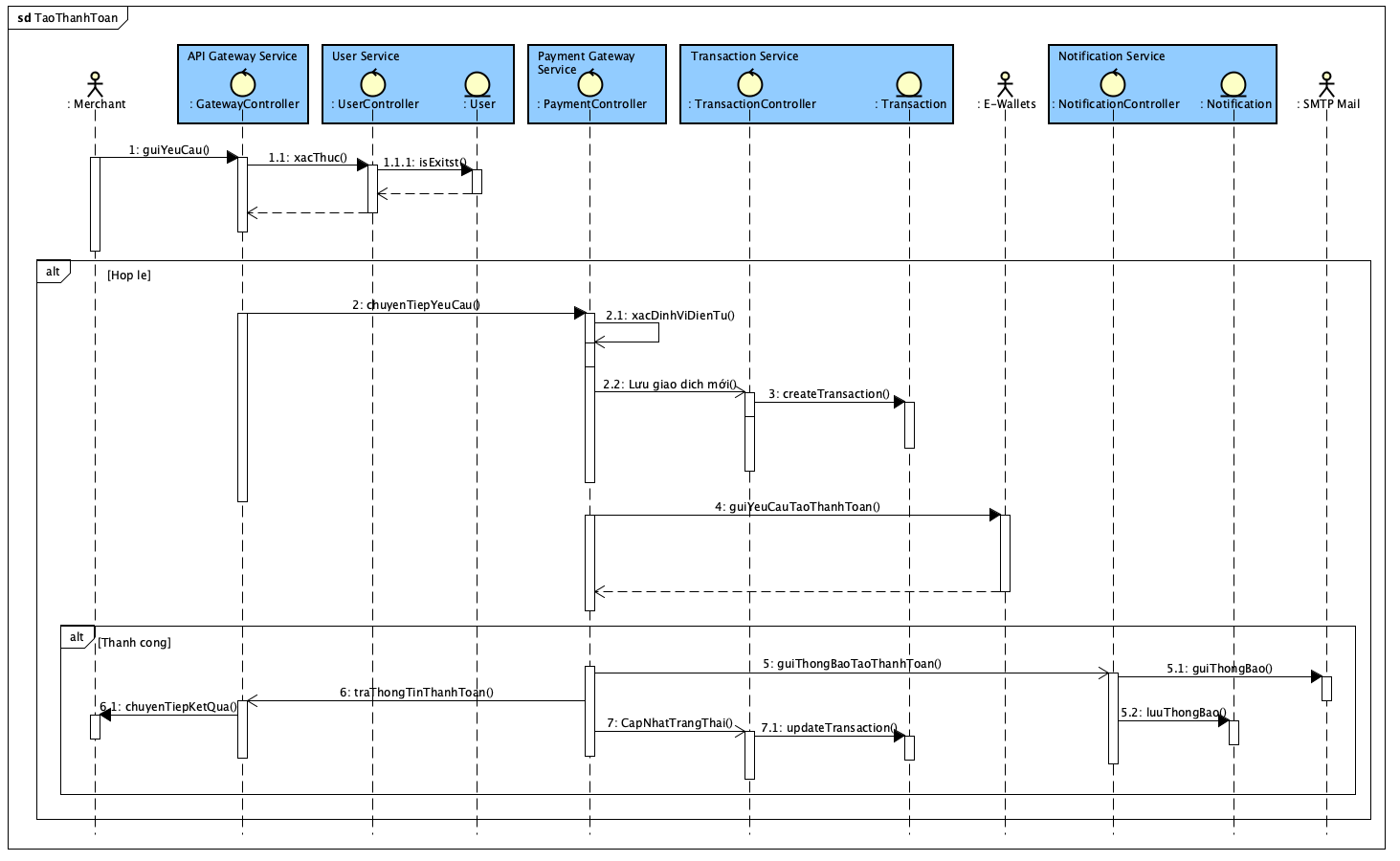
|  |  |
| --- | --- |
| **Use case Id** | UC-04 |
| **Tên use case** | Hoàn tiền giao dịch |
| **Mô tả** | Merchant gửi yêu cầu hoàn tiền giao dịch |
| **Actor chính** | Merchant |
| **Actor Phụ** | E-Wallets, SMTP Mail |
| **Tiền điều kiện** | Không |
| **Hậu điều kiện** | Merchant nhận được tiền hoàn |
| **Luồng hoạt động** | 1. Merchant gửi yêu cầu hoàn tiền 2. Hệ thống xác thực thông tin merchant 3. Hệ thống xác định ví điện tự theo yêu cầu của merchant 4. Hệ thống kiểm tra trạng thái giao dịch được lưu trong cơ sở dữ liệu 5. Hệ thống gửi yêu cầu hoàn tiền đến ví điện tử tương ứng 6. Ví điện tử xử lý yêu cầu và trả kết quả 7. Hệ thống cập nhật trạng thái giao dịch 8. Hệ thống trả kết quả thanh toán cho merchant 9. Hệ thống gửi thông báo về kết quả hoàn tiền |
| **Luồng thay thế** | 2a. Xác thực thất bại   1. Hệ thống báo lỗi cho merchant 2. Use case kết thúc   4a. Có trạng thái là “chưa thanh toán”   1. Hệ thống báo lỗi cho merchant 2. Use case kết thúc   6a.Ví điện tử không phản hồi   1. Hệ thống thử lại 2. Chuyển sang luồng xử lý ngoại lệ   6b. Hoàn tiền thất bại   1. Hệ thống cập nhật trạng thái giao dịch 2. Tiếp tục bước 8 |
| **Luồng ngoại lệ** | 1. Lưu request log 2. Thông báo cho merchant về trạng thái giao dịch |

Bảng ‑4. Use case hoàn tiền giao dịch

## Thiết kế hệ thống

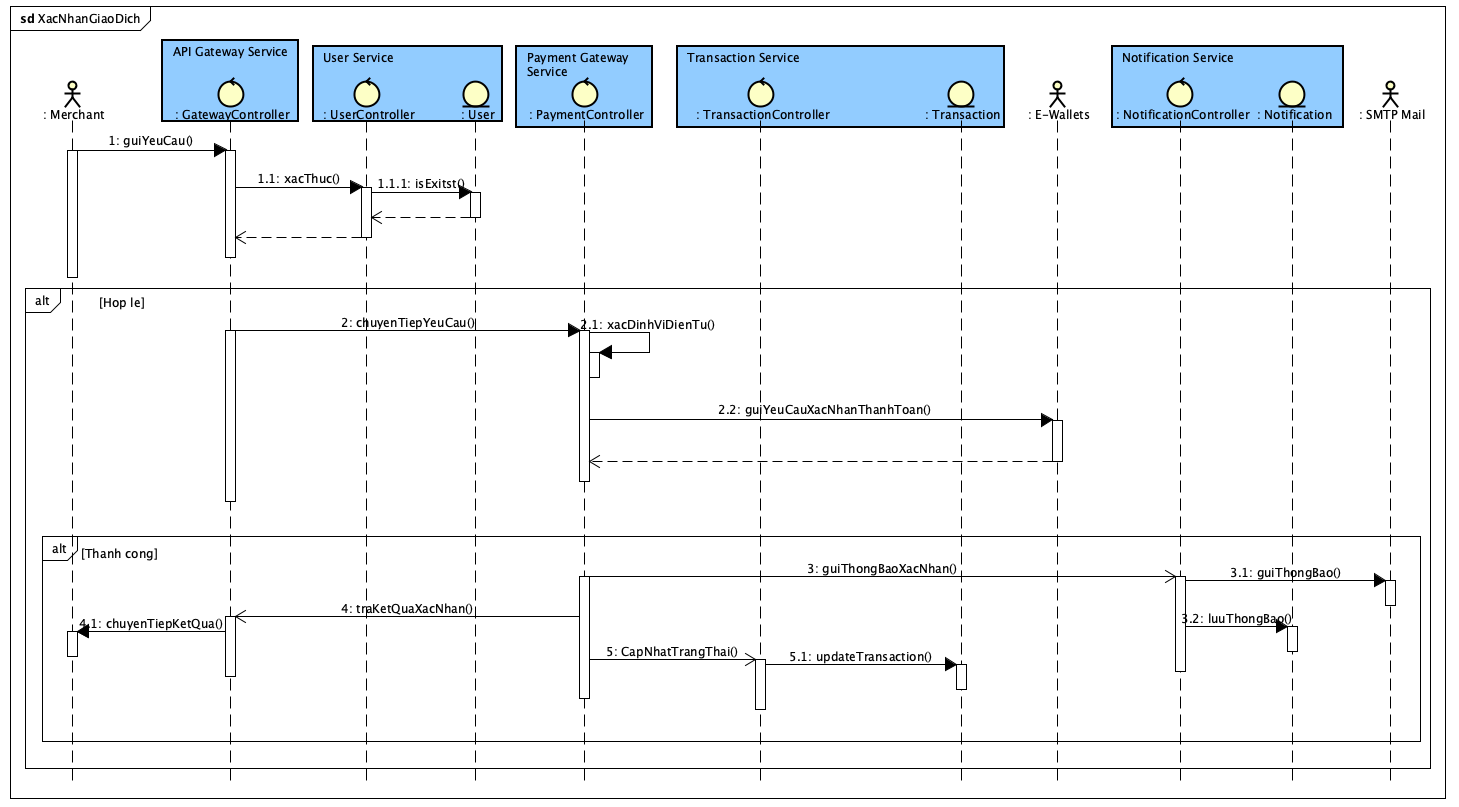
### Sơ đồ tuần tự (sequence diagram)

#### Tạo thanh toán



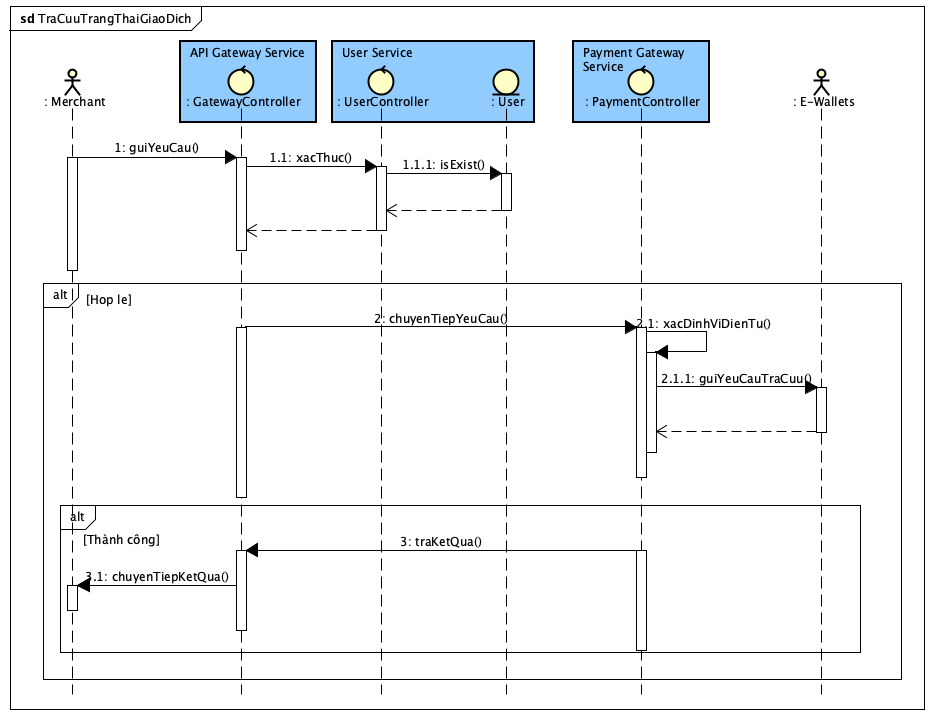
Hình ‑2. Sơ đồ tuần tự tạo thanh toán

#### Xác nhận giao dịch



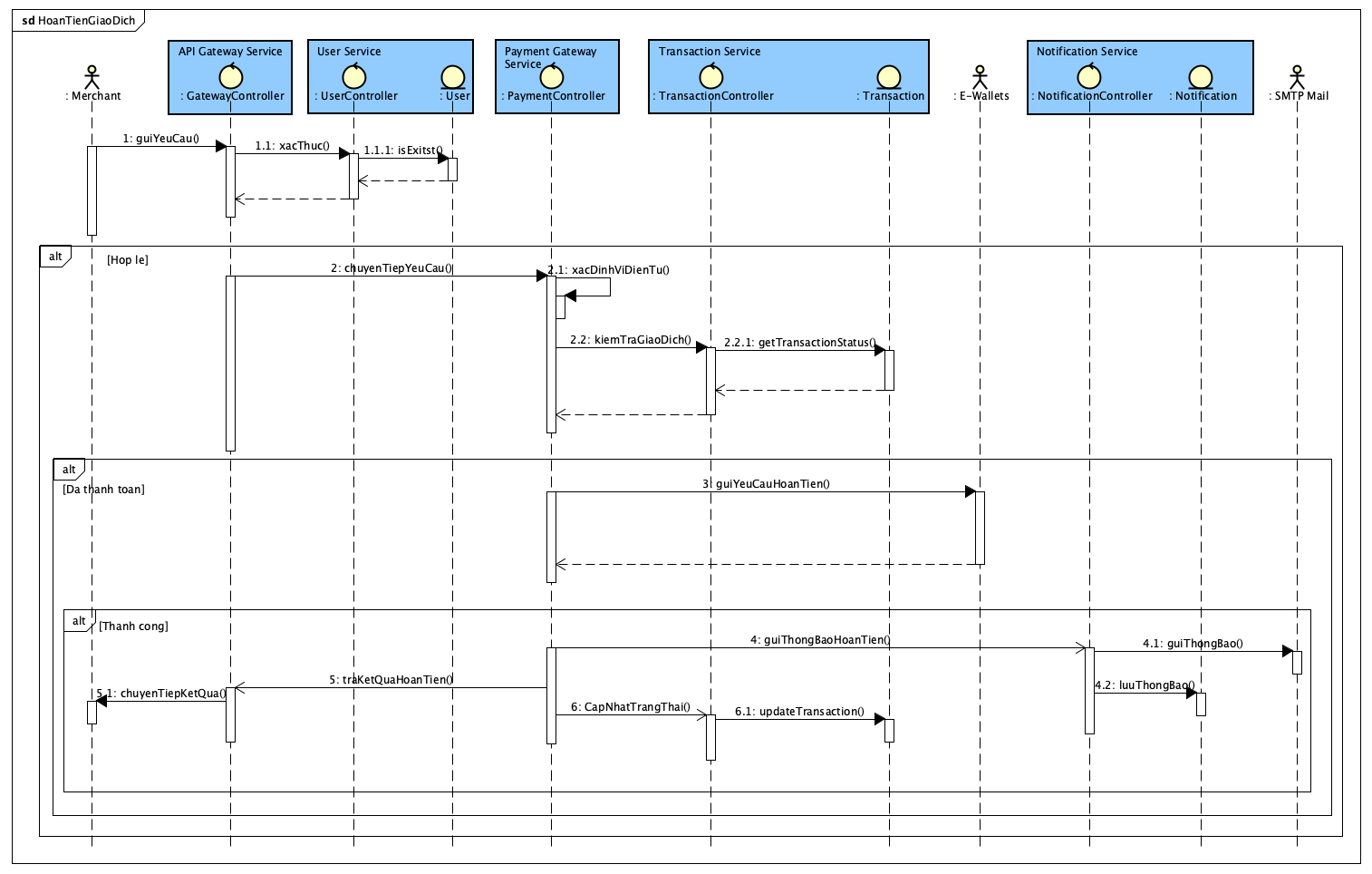
Hình ‑3. Sơ đồ tuần tự xác nhận giao dịch

#### Tra cứu trạng thái giao dịch



Hình ‑4. Sơ đồ tuần tự tra cứu trạng thái giao dịch

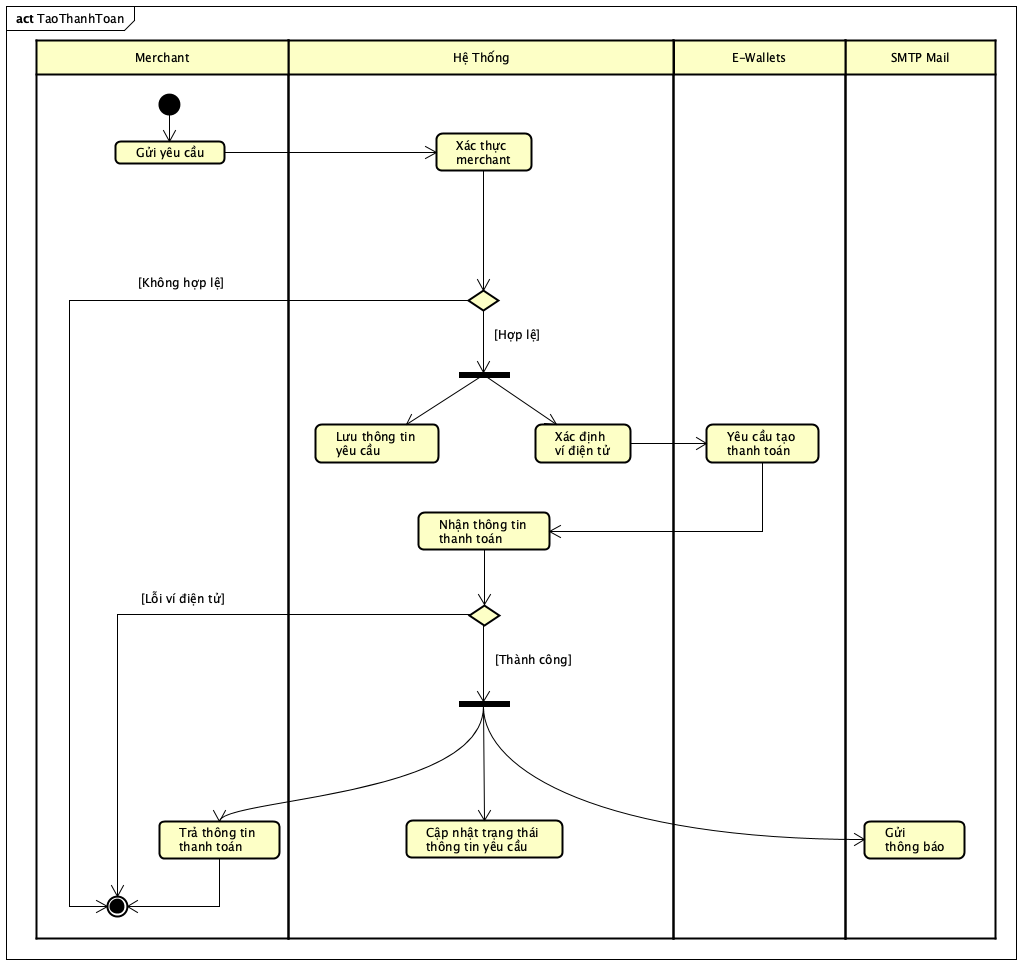
#### Hoàn tiền giao dịch



Hình ‑5. Sơ đồ tuần tự hoàn tiền giao dịch

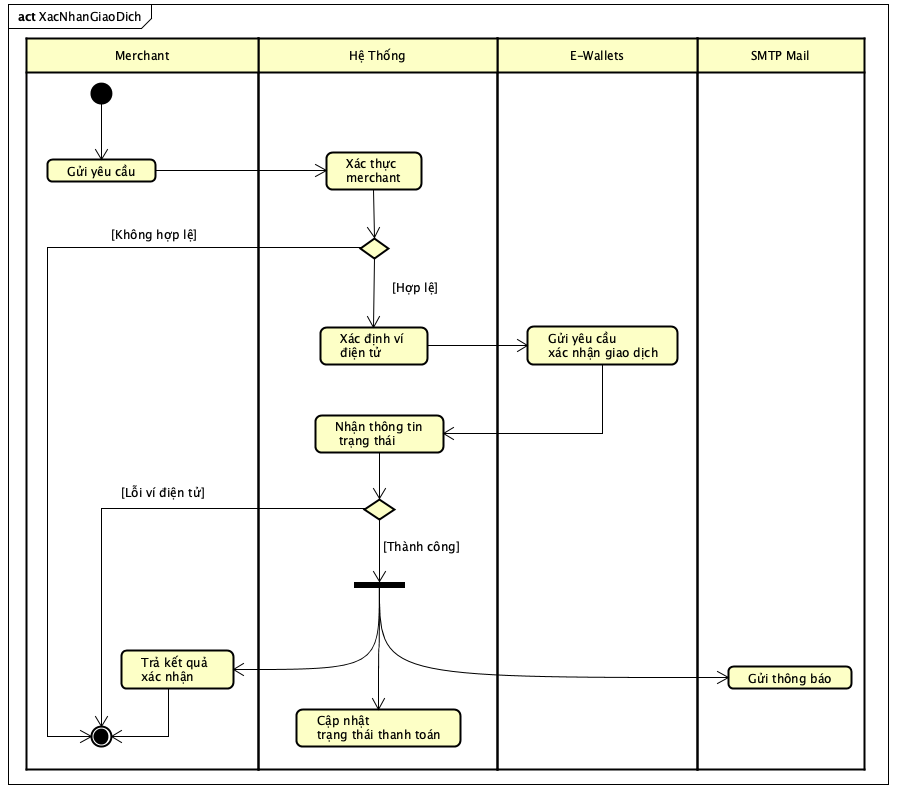
### Sơ đồ hoạt động (activity diagram)

#### Tạo thanh toán



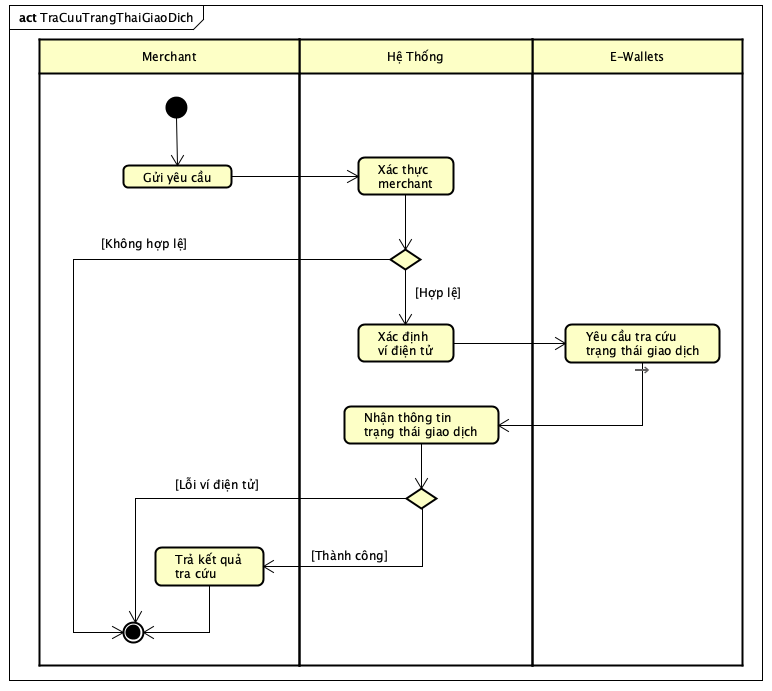
Hình ‑6. Sơ đồ hoạt động tạo thanh toán

#### Xác nhận giao dịch



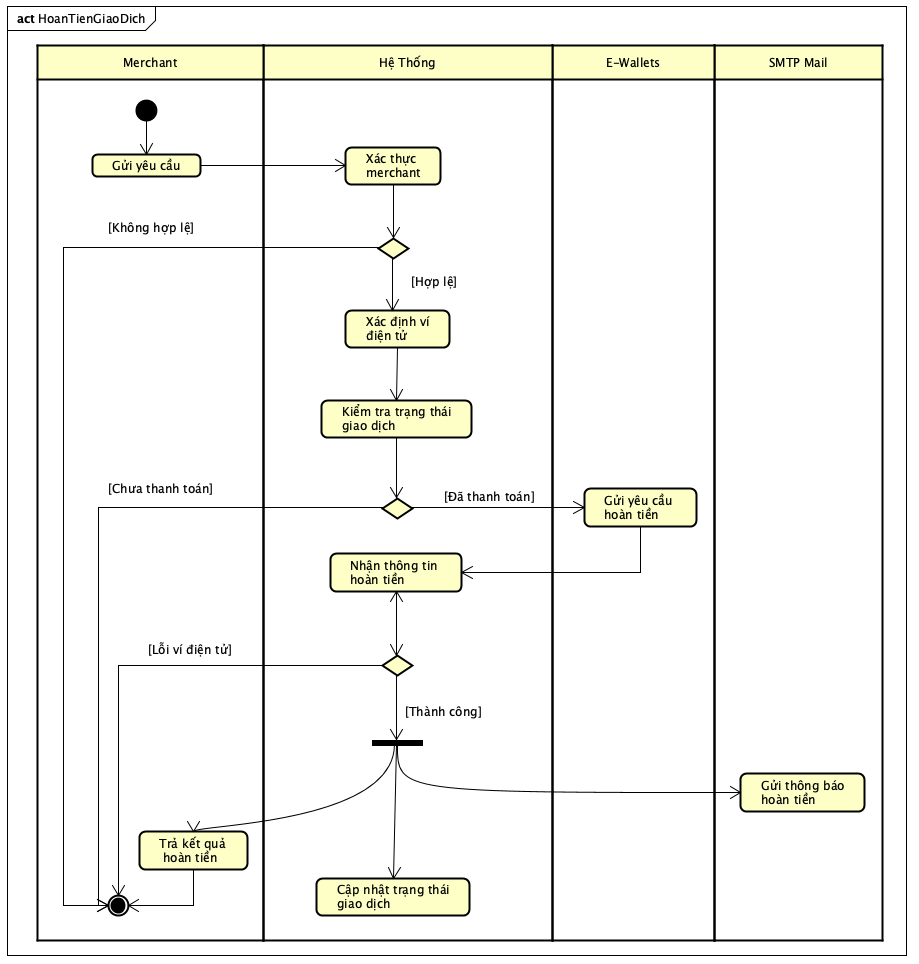
Hình ‑7. Sơ đồ trạng thái xác nhận giao dịch

#### Tra cứu trạng thái giao dịch



Hình ‑8. Sơ đồ hoạt động tra cứu trạng thái giao dịch

#### Hoàn tiền giao dịch



Hình ‑9. Sơ đồ hoạt động hoàn tiền giao dịch

### Thiết kế dữ liệu



Hình ‑. Lược đồ cơ sở dữ liệu

#### Đặc điểm

Hệ thống sẽ có 5 dịch vụ chính, tương đương với 5 cơ sở dữ liệu riêng biệt, tuân thủ đúng kiến trúc microservices.

* User service

Gồm 2 bảng

* User: Chứa thông tin người dùng
* Access Key: Chứa thông tin khóa truy cập của người dùng
* Transaction Service

Gồm 2 bảng

* Transaction: Chứa thông tin giao dịch tới các ví diện tử
* Payment Types: Chứa các phương thức thanh toán
* Notification Service

Gồm 2 bảng

* Notification: Chứa thông tin thông báo
* Notification Types: Chứa các loại thông báo
* Payment Gateway Service

Gồm 1 bảng

* Request log: Chứa lịch sử request tới các ví điện tử
* Billing Service

Gồm 3 bảng

* Billing: Chứa thông tin hóa đơn của người dùng
* Billing Details: Chứa thông tin chi tiết hóa đơn
* Fee Types: Chứa thông tin các loại phí

#### Mối quan hệ giữa các bảng

User - Access Key: (1-N) Một người dùng có thể có nhiều khóa truy cập, một khóa truy cập chỉ thuộc về một người dùng.

User – Transaction: (1-N) Một người dùng có thể thực hiện nhiều giao dịch, một giao dịch chỉ thuộc về một người dùng.

User – Billing: (1-N) Một người dùng có thể có nhiều hóa đơn, một hóa đơn chỉ thuộc một người dùng.

User – Notification: (1-N) Một người dùng có thể nhận nhiều thông báo, một thông báo chỉ thuộc về một người dùng.

Billing – Billing Details: (1-N) Một hóa đơn có thể có nhiều chi tiết hóa đơn, một chi tiết hóa đơn chỉ thuộc về một hóa đơn.

Fee Types – Billing Details: (1-N) Một loại phí có thể thuộc nhiều chi tiết hóa đơn, một chi tiết hóa đơn chỉ thuộc về một loại phí.

Transaction – Request Log: (1-N) Một giao dịch có thể có nhiều log request, một request log chỉ thuộc về một giao dịch.

Payment Types – Transaction: (1-N) Một loại thanh toán có thể thuộc nhiều giao dịch, một giao dịch chỉ thuộc về một loại thanh toán.

Notification Types – Notification: (1-N) Một loại thông báo có thể thuộc nhiều loại thông báo, một thông báo thuộc về một loại thông báo.

## Kiến trúc hệ thống

## Kết quả đề tài

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

## Hướng phát triển

## 

TÀI LIỆU THAM KHẢO

(Theo chuẩn IEEE – *bỏ dòng này khi viết báo cáo*)

PHỤ LỤC