

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
CƠ SỞ TẠI TP HỒ CHÍ MINH



BÁO CÁO TỔNG KẾT
ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN
NĂM 2024

***Đề tài:* Nghiên cứu các giải pháp thanh toán tiền dùng QR code**

05-SV-2024-TH2

Thuộc nhóm ngành khoa học: Công nghệ thông tin

TPHCM – 10/2024

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
CƠ SỞ TẠI TP. HỒ CHÍ MINH



BÁO CÁO TỔNG KẾT

ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN
NĂM 2024

***Đề tài:* Nghiên cứu các giải pháp thanh toán tiền dùng QR code**

05-SV-2024-TH2

Thuộc nhóm ngành khoa học: Công nghệ thông tin

Sinh viên thực hiện: Võ Quang Huy Nam, Nữ: Nam

Dân tộc: Kinh

Mã số sinh viên: N20DCCN022

Lớp: D20CQCNPM01-N

Khoa: Công nghệ thông tin 02 Năm thứ: 5/5

Ngành học: Công nghệ thông tin

Người hướng dẫn: Th.S Nguyễn Anh Hào

TPHCM – 10/2024

MỤC LỤC

Mở đầu

	Trang
CHƯƠNG I. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Tên đề tài: Nghiên cứu các giải pháp thanh toán tiền dùng QR Code.....	1
1.2 Thuộc lĩnh vực KHCN: Nghiên cứu ứng dụng công nghệ thông tin.	1
1.3 Mục tiêu, nội dung và kết quả:	1
CHƯƠNG II. TÌM HIỂU CÁCH PHÁT SINH QR CODE, VÀ ĐỌC QR CODE TRÊN ĐIỆN THOẠI DI ĐỘNG.....	2
2.1 Tìm hiểu về QR Code	2
2.1.1 Lịch sử	2
2.1.2 Mã QR là gì?.....	2
2.1.3 Đặc điểm kỹ thuật.....	2
2.1.4 Tổng quan về thủ tục mã hóa.....	9
2.1.5 Phân loại mã QR.....	10
2.1.6 Các tính năng của mã QR	11
2.1.7 Các lĩnh vực áp dụng mã QR.....	12
2.2 Ứng dụng trong thanh toán điện tử	13
2.2.1 Quy trình thanh toán	13
2.2.2 Đọc QR Code trên điện thoại di động	13
2.2.3 Ưu điểm của QR Code trong thanh toán	14
2.2.4 Thách thức và giải pháp.....	15
2.2.5 Xu hướng tương lai.....	16
2.3 Nhận xét và đề xuất	16
2.3.1 Nhận xét.....	16
2.3.2 Đề xuất.....	16
CHƯƠNG III. TÌM HIỂU THỦ TỤC THANH TOÁN TIỀN QUA API CỦA VNPAY, PAYON, PAYPAL VÀ MOMO.....	18
3.1 Giới thiệu	18
3.2 Cổng thanh toán VNPay	18
3.2.1 Quy trình công nghệ	18
3.2.2 Bảo mật.....	18
3.2.3 Tiện ích người dùng.....	18
3.2.4 Timeline.....	19

3.2.5 Các bước merchant cần xử lý tích hợp code cài đặt	19
3.2.6 Mô hình kết nối.....	20
3.2.7 Sơ đồ tuần tự.....	20
3.2.8 Thông tin cấu hình	21
3.2.9 Tích hợp vào hệ thống website thương mại điện tử	27
3.3 MoMo	36
3.3.1 Quy trình công nghệ	36
3.3.2 Bảo mật.....	36
3.3.3 Tiện ích người dùng.....	37
3.3.4 Mô hình kết nối.....	37
3.3.5 Thông tin cấu hình	39
3.3.6 Tích hợp vào hệ thống website thương mại điện tử	44
3.4 PayPal.....	52
3.4.1 Quy trình công nghệ	52
3.4.2 Bảo mật.....	52
3.4.3 Tiện ích người dùng.....	52
3.4.4 PayPal REST APIs	53
3.4.5 Tích hợp vào hệ thống website thương mại điện tử	55
3.5 PayOn	58
3.5.1 Quy trình công nghệ	58
3.5.2 Bảo mật.....	58
3.5.3 Tiện ích người dùng.....	58
3.5.4 Thông tin cấu hình	58
3.5.5 Mô hình kết nối.....	59
3.6 So sánh và kết luận.....	60
3.6.1 So sánh.....	60
3.6.2 Kết luận.....	61
CHƯƠNG VI: TÌM HIỂU TỔNG QUAN KỸ THUẬT CÔNG THANH TOÁN NAPAS.....	62
4.1. Tìm hiểu tổng quan	62
4.1.1 Giới thiệu	62
4.1.2 Quy trình giao dịch	62
4.1.3 Cơ chế hoạt động nội bộ	62
4.1.4 Công nghệ nền tảng	62
4.2. Đặc tả kỹ thuật.....	63
4.2.1. Kiểm soát truy cập bảo mật	63
4.2.2. Thông tin thẻ.....	63

4.3 Tích hợp thanh toán lưu trữ (Hosted Checkout).....	63
4.3.1 Giao dịch Data Key	63
4.3.2. Thanh toán / Thanh toán bằng Token đã trả về / Tạo Token – tích hợp với NAPAS Hosted Form	63
4.3.3 Thanh toán / Thanh toán bằng Token đã trả về / Tạo Token – tích hợp với NAPAS Payment Page.....	63
4.3.4 Nhận phản hồi từ NAPAS	64
4.4 Triển khai API trực tiếp	64
4.4.1. Thanh toán / Ủy quyền / Chụp / Hủy bỏ / Hoàn trả (đối với Xử lý Thẻ Quốc tế) / Hoàn trả giao dịch Nội địa (đối với Xử lý Thẻ Nội địa)	64
4.4.2. Thanh toán bằng giao dịch 3DSecure	64
4.4.3. Giao dịch OTP mua hàng – tích hợp với NAPAS Hosted Checkout	64
4.4.4. Giao dịch truy xuất đơn hàng	64
4.4.5. Giao dịch xóa Token.....	64
4.4.6 Tạo Giao dịch QR.....	65
4.4.7 Tham số lỗi	65
4.4.8 Mua hàng bằng giao dịch Token đã trả về - tích hợp với NAPAS Hosted Checkout	65
4.4.9 Giao dịch truy xuất Token	65
4.5 Mã phản hồi	65
4.5.1. Xử lý thẻ quốc tế.....	65
4.5.2. Xử lý thẻ nội địa	65
4.6 Phần kết luận	65
KẾT LUẬN.....	66
DANH MỤC TÀI LIỆU THAM KHẢO.....	67

DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH

Bảng 2. 1 Thống kê lưu trữ dữ liệu qua các phiên bản của QR code.....	6
Hình 2. 1 Cấu trúc một QR Code	3
Hình 2. 2 Các Version của QR Code	4
Hình 2. 3 Một mẫu thử cho trường hợp chọn Version	4
Hình 2. 4 Một Mã QR có kích thước có thể mã hóa 300 ký tự chữ và số.	5
Hình 2. 5 Hình ảnh minh họa đặc tính linh hoạt và Độ tin cậy của Mã QR	7
Hình 2. 6 Minh họa khả năng sửa lỗi của QR Code.....	7
Hình 2. 7 Các thành phần chính định vị căn chỉnh cho một mã QR.....	8
Hình 2. 8 Hình ảnh minh họa tính năng ghép nối của QR	8
Hình 2. 9 Các loại QR code phổ biến.....	10
Hình 2. 10 Các tính năng của QR code	11
Bảng 3. 1 Danh sách tham số - Thông tin gửi sang VNPAY (vnp_Command=pay)	23
Bảng 3. 2 Danh sách tham số - Thông tin nhận về từ VNPAY (vnp_Command=pay)	25
Bảng 3. 3 Bảng mã lỗi của hệ thống thanh toán VNPAY 1	27
Bảng 3. 4 Bảng mã lỗi của hệ thống thanh toán VNPAY 2.....	27
Bảng 3. 5 Tham số Instant Payment Notification	43
Bảng 3. 6 Thông tin cấu hình PayOn	58
Bảng 3. 7 So sánh tổng quan các công thanh toán	61
Sơ đồ 3. 1 Mô hình kết nối của VNPay.....	20
Sơ đồ 3. 2 Sơ đồ tuần tự của VNPay	21
Sơ đồ 3. 3 Luồng xử lý của MoMo	38
Hình 3. 1 Timeline tích hợp VNPay	19
Hình 3. 2 Cấu Hình IPN VNPay trả về	35
Hình 3. 3 Trang sit testing do VNPay hỗ trợ 1	35
Hình 3. 4 Trang sit testing do VNPay hỗ trợ 2	35
Hình 3. 5 Trang Return Url VNPay trả về với các tham số	36
Hình 3. 6 Trang thanh toán thành công của website	36
Hình 3. 7 Trang thanh toán của website với QR code MoMo	49
Hình 3. 8 Giao diện app sandbox của momo sau khi quét mã QR code	50
Hình 3. 9 Giao diện app sandbox của momo sau khi thanh toán thành công	51
Hình 3. 10 Hình 3. 12 Giao diện của website khi thanh toán momo thành công	52
Hình 3. 11 Đăng kí nhận Client ID và SecretKey của PayPal	53
Hình 3. 12 Luồng xử lý của PayOn với thẻ nội địa ATM và thẻ Credit/Visa/... ..	59
Hình 3. 13 Luồng xử lý của PayOn với QR code	60

MỞ ĐẦU

Trong thời đại công nghệ số hiện nay, các phương thức thanh toán điện tử đang trở thành xu hướng phổ biến và ngày càng được ưa chuộng. Một trong những phương pháp thanh toán tiện lợi và hiệu quả là việc sử dụng mã QR (Quick Response Code). Mã QR không chỉ giúp người dùng thực hiện các giao dịch một cách nhanh chóng mà còn mang lại sự tiện lợi trong việc thanh toán trực tuyến.

Đề tài "Nghiên cứu các giải pháp thanh toán tiền dùng QR Code" được thực hiện nhằm đáp ứng nhu cầu ngày càng cao về việc tích hợp thanh toán điện tử trong các hoạt động mua sắm trực tuyến. Mục tiêu chính của nghiên cứu là tìm hiểu và triển khai chức năng thanh toán tiền qua QR Code cho một website bán hàng, đồng thời tích hợp các cổng thanh toán điện tử phổ biến như VNPay, PayOn, PayPal, và MoMo.

Nghiên cứu này tập trung vào việc phân tích quy trình phát sinh và đọc QR Code trên thiết bị di động, đồng thời nghiên cứu các thủ tục thanh toán qua API của các cổng thanh toán nêu trên. Một phần quan trọng của đề tài là việc cài đặt các chức năng thanh toán, cho phép người dùng quét QR Code từ trình duyệt trên điện thoại và thực hiện thanh toán qua cổng thanh toán đã chọn, với việc đảm bảo tuân thủ chuẩn 3D Security để bảo vệ thông tin giao dịch.

Kết quả của nghiên cứu này bao gồm việc xây dựng một chương trình minh họa giải pháp thanh toán qua QR Code và hoàn thiện một quyển báo cáo tổng hợp các kết quả nghiên cứu. Qua đó, đề tài không chỉ đóng góp vào việc ứng dụng công nghệ QR Code trong thanh toán mà còn góp phần nâng cao hiệu quả và độ tin cậy của các phương thức thanh toán điện tử hiện đại.

CHƯƠNG I. GIỚI THIỆU ĐỀ TÀI

1.1 Tên đề tài: Nghiên cứu các giải pháp thanh toán tiền dùng QR Code.

1.2 Thuộc lĩnh vực KHCN: Nghiên cứu ứng dụng công nghệ thông tin.

1.3 Mục tiêu, nội dung và kết quả:

a. Mục tiêu:

- Tìm hiểu và cài đặt chức năng thanh toán tiền cho một website bán hàng bằng QR Code (đọc từ điện thoại di động) và các cổng thanh toán điện tử: VNPay, PayOn, PayPal, và MoMo.

b. Nội dung:

- Tìm hiểu cách phát sinh QR code và đọc QR code trên điện thoại di động.
- Tìm hiểu thủ tục thanh toán tiền qua API của VNPay, PayOn, PayPal và MoMo.
- Cài đặt các chức năng trả tiền mua hàng bằng cách dùng Edge/Chrome browser của điện thoại để đọc QR code và thanh toán tiền qua cổng thanh toán đã được khách chọn (tuân thủ chuẩn 3D Security).

c. Kết quả:

- Chương trình minh họa giải pháp thanh toán tiền bằng QR Code.
- Quyển báo cáo.

CHƯƠNG II. TÌM HIỂU CÁCH PHÁT SINH QR CODE, VÀ ĐỌC QR CODE TRÊN ĐIỆN THOẠI DI ĐỘNG

2.1 Tìm hiểu về QR Code

2.1.1 Lịch sử

- QR Code (Quick Response Code) là một loại mã vạch ma trận hai chiều, được phát triển bởi công ty Denso Wave, một công ty con của Tập đoàn Toyota, vào năm 1994. Masahiro Hara, kỹ sư trưởng của Denso Wave, đã dẫn đầu nhóm phát triển QR Code với mục tiêu tạo ra một mã vạch có thể được đọc dễ dàng và nhanh chóng bởi máy quét.
- Vì QR Code là mã mở mà bất kỳ ai cũng được phép sử dụng nên nó không chỉ được sử dụng ở Nhật Bản mà còn ở các quốc gia trên toàn thế giới. Khi các quy tắc sử dụng nó được quy định và mã đã được tiêu chuẩn hóa, việc sử dụng nó càng lan rộng hơn. Năm 1997, nó đã được phê duyệt làm tiêu chuẩn AIM để sử dụng trong ngành nhận dạng tự động. Năm 1999, nó đã được Tiêu chuẩn Công nghiệp Nhật Bản phê duyệt làm mã 2D tiêu chuẩn và tạo thành biểu tượng 2D tiêu chuẩn trên các biểu mẫu giao dịch tiêu chuẩn EDI của Hiệp hội các nhà sản xuất ô tô Nhật Bản. Hơn nữa, vào năm 2000, nó đã được ISO phê duyệt là một trong những tiêu chuẩn quốc tế. Hiện nay, việc sử dụng QR Code phổ biến đến mức không quá lời khi nói rằng nó được sử dụng ở mọi nơi trên thế giới.
- Ban đầu, QR Code được sử dụng trong ngành công nghiệp ô tô để theo dõi các bộ phận trong quá trình sản xuất. Tuy nhiên, do tính linh hoạt và khả năng lưu trữ thông tin lớn, nó nhanh chóng được áp dụng trong nhiều lĩnh vực khác như marketing, logistics, và gần đây nhất là trong lĩnh vực thanh toán điện tử.
- Mã QR được coi là phiên bản cải tiến của mã vạch truyền thống và đã được Tổ chức Tiêu chuẩn hóa Quốc tế (ISO) phê duyệt tiêu chuẩn quốc tế vào năm 2000

2.1.2 Mã QR là gì?

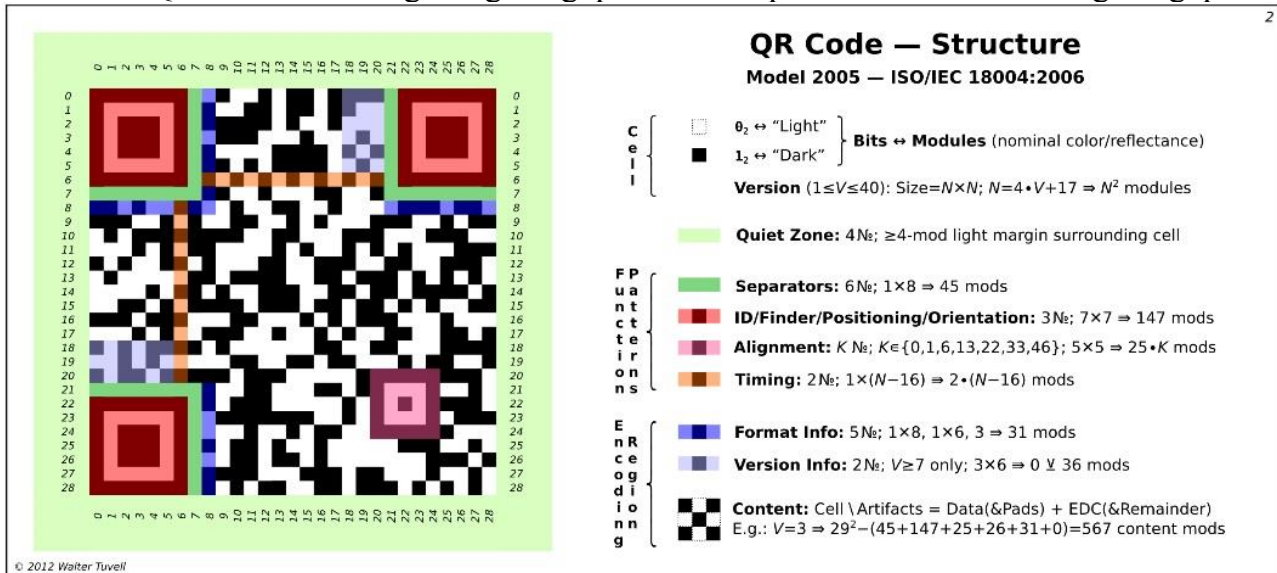
- Mã QR Code (hay còn gọi là mã QR) là viết tắt của Quick response code (Mã phản hồi nhanh), hoặc có thể gọi là Mã vạch ma trận (Matrix-barcode). Đây là một dạng thông tin được mã hóa để hiển thị sao cho máy có thể đọc được.
- Mã QR Code được phát triển vào những năm 1990 là phiên bản cải tiến của mã vạch truyền thống. Mã vạch truyền thống là một dãy các vạch được xếp liền kề nhau, chỉ chứa được tối đa 20 ký tự số, trong khi đó thì mã QR có khả năng chứa tối đa 7.089 ký tự số và 4.296 ký tự chữ số. Điều này cho phép lượng thông tin truyền tải sẽ nhiều hơn, hỗ trợ tốt hơn cho người dùng.
- Các đặc tính kỹ thuật của mã QR đã được 2 tổ chức tiêu chuẩn quốc tế là Tổ chức tiêu chuẩn hóa quốc tế (ISO) và Ủy ban điện quốc tế (IEC) công bố thành tiêu chuẩn quốc tế ISO/IEC 18004:2000 (hiện nay là ISO/IEC 18004:2015). Việt Nam có tiêu chuẩn quốc gia về mã QR là TCVN 7322:2009 (chấp nhận tiêu chuẩn ISO/IEC 18004:2006).

2.1.3 Đặc điểm kỹ thuật

a) Cấu trúc:

- QR Code chính là dạng mã vạch hai chiều (2D), bao gồm các ô vuông màu đen được sắp xếp theo dạng lưới (ma trận) trên nền trắng. QR Code có khả năng lưu giữ lượng thông tin lớn gấp hàng trăm lần với nhiều định dạng ký tự khác nhau.
- Mỗi mã QR sẽ được xây dựng từ các mô-đun hình vuông đặt trong một mảng hình vuông lớn, bao gồm vùng mã hoá (Encoding region) và các mẫu chức năng (Function Patterns), cụ thể là mẫu tìm kiếm (Finder Patterns), dải phân cách (Separator), mẫu thời gian (Timing Patterns), mẫu căn chỉnh (Alignment Patterns). Các mẫu chức năng này trong mã hoá dữ liệu. Mã sẽ được bao quanh bốn phía bằng vùng yên tĩnh (Quiet Zone) có đường viền kín.

- QR Code bao gồm các mô-đun đen được sắp xếp trong một lưới vuông trên nền trắng.
- Có ba "finder patterns" ở các góc để xác định vị trí và hướng của mã.
- "Alignment patterns" giúp điều chỉnh méo mó của hình ảnh.
- "Timing patterns" xác định kích thước của ma trận dữ liệu.
- "Quiet zone" là vùng trống xung quanh mã để phân biệt với môi trường xung quanh.



Hình 2. 1 Cấu trúc một QR Code

Hình ảnh này mô tả cấu trúc chi tiết của một QR Code theo tiêu chuẩn ISO/IEC 18004:2006.

Dưới đây là phân tích các thành phần chính:

- Cấu trúc tổng thể:

- QR Code được hiển thị là một ma trận vuông gồm các module đen và trắng.
- Kích thước: $N \times N$ modules, với $N = 4V + 17$, trong đó V là phiên bản ($1 \leq V \leq 40$).
- Các thành phần chính:
- Quiet Zone: Vùng trống xung quanh QR code, rộng ít nhất 4 modules.
- Function Patterns:

- Separators: Dải phân cách $6N_0$; $1 \times 8 = 45$ modules. Với 6 lần xuất hiện dải 1×8 nhưng trừ 3 bit mỗi vì mỗi dải đè lên nhau ($6 \cdot 8 - 3 = 45$)
- Finder Patterns (ID/finder/positioning/orientation): 3 hình vuông ở góc, mỗi cái $7 \times 7 = 147$ modules.
- Alignment Patterns: K hình vuông nhỏ, vị trí tùy theo phiên bản, mỗi cái 5×5 modules.
- Timing Patterns: 2 dải đen trắng xen kẽ, mỗi dải $2(N-16)$ modules.

- Encoding Regions:

- Format Info: Thông tin định dạng, tổng cộng 31 modules (5 lần xuất hiện với 2 dải 8 bit, 2 dải 6 bit và 1 dải 3 bit).
- Version Info: Thông tin phiên bản (chỉ có ở phiên bản ≥ 7), 18 bit, 36 modules.
- Data và Error Correction: Phần còn lại của QR code chứa dữ liệu và mã sửa lỗi.

- Ví dụ cụ thể:

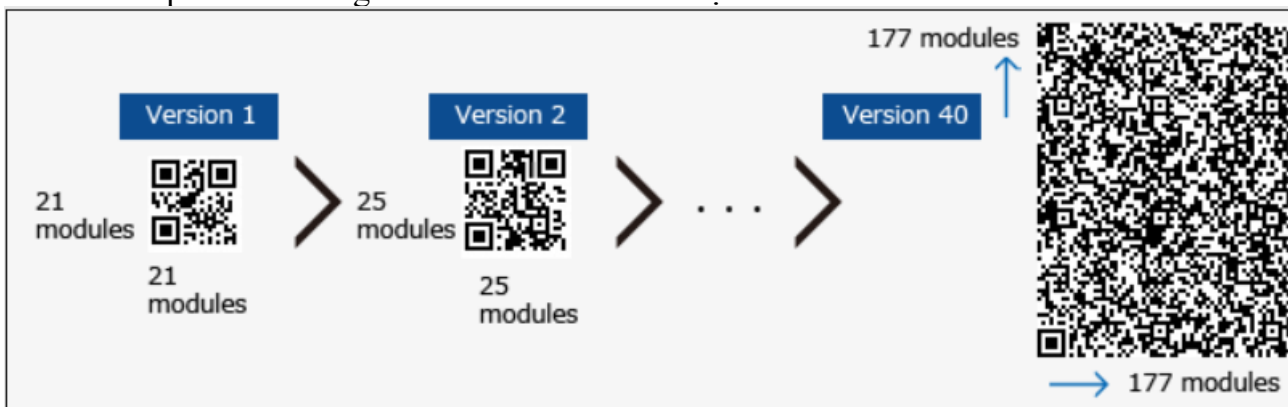
Với phiên bản 3 ($V=3$), tổng số module cho nội dung là 567, được tính như sau:

$$29^2 - (45 + 147 + 25 + 26 + 31 + 0) = 567 \text{ modules}$$

Mã hóa nội dung được trình bày theo cách chuyển thành mã nhị phân và lấp vào những ô còn trống theo thứ tự từ dưới lên trên từ phải qua trái.

b) Các version

- QR Code có 40 phiên bản, từ Version 1 (21x21 mô-đun) đến Version 40 (177x177 mô-đun).
- Mỗi phiên bản tăng lên 4 mô-đun cho mỗi cạnh.



Hình 2. 2 Các Version của QR Code

Version	Modules	ECC Level	Data bits (mixed)	Numeric	Alphanumeric	Binary	Kanji
1	21×21	L	152	41	25	17	10
		M	128	34	20	14	8
		Q	104	27	16	11	7
		H	72	17	10	7	4
2	25×25	L	272	77	47	32	20
		M	224	63	38	26	16
		Q	176	48	29	20	12
		H	128	34	20	14	8
3	29×29	L	440	127	77	53	32
		M	352	101	61	42	26
		Q	272	77	47	32	20
		H	208	58	35	24	15

Hình 2. 3 Một mẫu thử cho trường hợp chọn Version

- Cụ thể cho trường hợp dữ liệu gồm 100 chữ số:
 1. Chọn "numeral" (số) làm loại dữ liệu đầu vào.
 2. Chọn mức độ sửa lỗi từ L, M, Q, H. (Ví dụ này giả định chọn M)
 3. Trong bảng, tìm số gần nhất và lớn hơn hoặc bằng 100 ở cột "Numeric" (Số) và hàng có mức sửa lỗi M.
- Theo bảng:
 - Phiên bản 1: 34 chữ số (không đủ)
 - Phiên bản 2: 63 chữ số (không đủ)
 - Phiên bản 3: 101 chữ số (đủ và gần nhất với 100)
- Vậy phiên bản QR code phù hợp nhất là phiên bản 3, có kích thước 29x29 modules và có thể chứa 101 chữ số với mức sửa lỗi M.
- Bảng cung cấp thông tin về khả năng lưu trữ của các phiên bản QR code khác nhau cho nhiều loại dữ liệu (số, chữ và số, nhị phân, kanji) ở các mức độ sửa lỗi khác nhau.
- c) Khả năng lưu trữ:
 - Số: tối đa 7,089 ký tự
 - Chữ và số: tối đa 4,296 ký tự
 - Nhị phân: tối đa 2,953 byte
 - Kanji/Kana: tối đa 1,817 ký tự (Là một hệ thống ký hiệu được phát triển tại Nhật Bản, QR Code có khả năng mã hóa bộ ký tự kanji cấp 1 và cấp 2 theo tiêu chuẩn JIS. Trong trường hợp tiếng Nhật, một ký tự Kana hoặc Kanji có độ rộng đầy đủ được mã

hóa hiệu quả thành 13 bit, cho phép Mã QR chứa hơn 20% dữ liệu so với các ký hiệu 2D khác).

abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz
 klmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz
 tuvwx1234567890abcdefghijklmnopqrstuvwxyz1234567890
 34567890abcdefghijklmnopqrstuvwxyz1234567890
 abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz
 klmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz
 tuvwx1234567890abcdefghijklmnopqrstuvwxyz



Hình 2. 4 Một Mã QR có kích thước có thể mã hóa 300 ký tự chữ và số.

Cỡ hình	Số môđun/mặt (A)	Môđun mẫu chức năng (B)	Môđun thông tin định dạng và cỡ hình (C)	Môđun dữ liệu trừ (C) ($D = A^2 - B - C$)	Dung lượng dữ liệu [từ mã] ^a (E)	Các bit dư
M1	11	70	15	36	5	0
M2	13	74	15	80	10	0
M3	15	78	15	132	17	0
M4	17	82	15	192	24	0
1	21	202	31	208	26	0
2	25	235	31	359	44	7
3	29	243	31	567	70	7
4	33	251	31	807	100	7
5	37	259	31	1 079	134	7
6	41	267	31	1 383	172	7
7	45	390	67	1 568	196	0
8	49	398	67	1 936	242	0
9	53	406	67	2 336	292	0
10	57	414	67	2 768	346	0
11	61	422	67	3 232	404	0
12	65	430	67	3 728	466	0
13	69	438	67	4 256	532	0
14	73	611	67	4 651	581	3
15	77	619	67	5 243	655	3
16	81	627	67	5 867	733	3
17	85	635	67	6 523	815	3
18	89	643	67	7 211	901	3
19	93	651	67	7 931	991	3
20	97	659	67	8 683	1 085	3
21	101	882	67	9 252	1 156	4
22	105	890	67	10 068	1 258	4
23	109	898	67	10 916	1 364	4
24	113	906	67	11 796	1 474	4
25	117	914	67	12 708	1 588	4

26	121	922	67	13 652	1 706	4
27	125	930	67	14 628	1 828	4
28	129	1 203	67	15 371	1 921	3
29	133	1 211	67	16 411	2 051	3
30	137	1 219	67	17 483	2 185	3
31	141	1 227	67	18 587	2 323	3
32	145	1 235	67	19 723	2 465	3
33	149	1 243	67	20 891	2 611	3
34	153	1 251	67	22 091	2 761	3
35	157	1 574	67	23 008	2 876	0
36	161	1 582	67	24 272	3 034	0
37	165	1 590	67	25 568	3 196	0
38	169	1 598	67	26 896	3 362	0
39	173	1 606	67	28 256	3 532	0
40	177	1 614	67	29 648	3 706	0
^a Tất cả các từ mã đều có độ dài 8 bit, trừ các cờ hình M1 và M3 với từ mã dữ liệu cuối là có độ dài 4 bit.						

Bảng 2. 1 Thống kê lưu trữ dữ liệu qua các phiên bản của QR code [3]

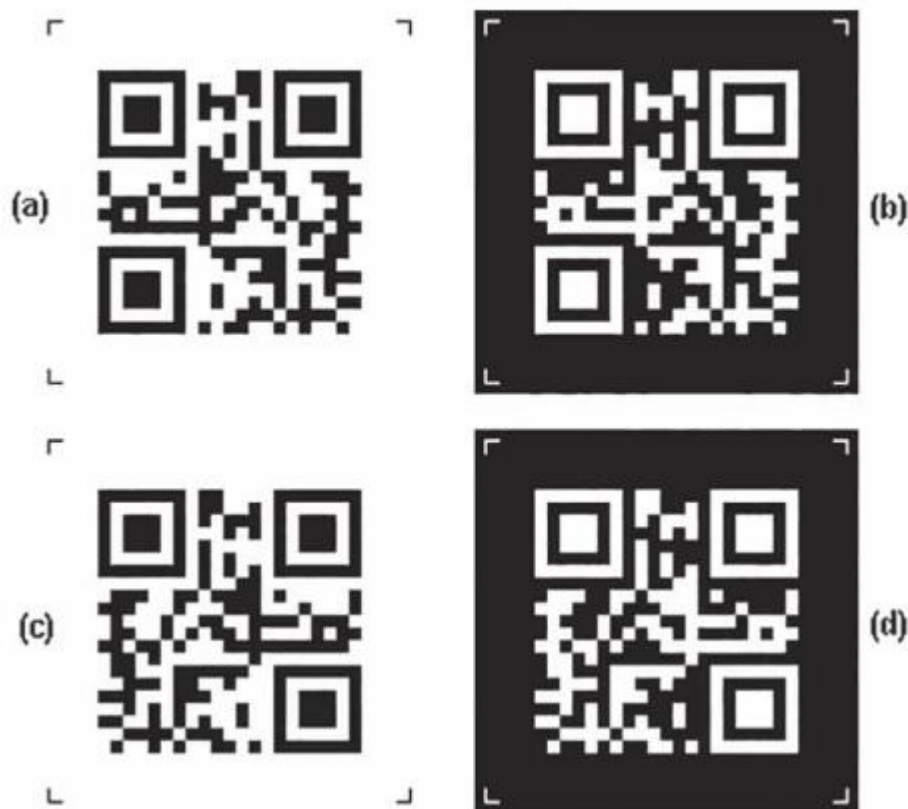
Trong khi mã vạch thông thường có khả năng lưu trữ tối đa khoảng 20 chữ số thì QR Code có khả năng xử lý lượng thông tin nhiều hơn vài chục đến vài trăm lần.

Mã QR có khả năng xử lý tất cả các loại dữ liệu, chẳng hạn như ký tự số và chữ cái, Kanji, Kana, Hiragana, ký hiệu, mã nhị phân và mã điều khiển. Có thể mã hóa tối đa 7.089 ký tự trong một ký hiệu.

d) Đặc tính linh hoạt và độ tin cậy của Mã QR

Mã QR có thể được thiết kế để có hệ số phản xạ đảo chiều, tức là mã có thể được tạo ra dưới dạng hình ảnh tối trên nền sáng hoặc ngược lại, hình ảnh sáng trên nền tối. Điều này giúp mã QR có thể linh hoạt trong thiết kế và in ấn trên các chất liệu khác nhau.

Các mã QR có thể được đọc dù bị phản chiếu theo chiều ngang hoặc chiều dọc. Điều này có nghĩa là mã QR vẫn có thể giải mã được ngay cả khi hình ảnh của mã bị đảo ngược. Đặc tính này giúp tăng tính linh hoạt và độ tin cậy khi quét mã trong các điều kiện khác nhau



Hình 2. 5 Hình ảnh minh họa đặc tính linh hoạt và Độ tin cậy của Mã QR

- (a) bình thường định hướng và bố trí phản xạ bình thường;
- (b) hướng bình thường và đảo ngược phản xạ;
- (c) hướng của ảnh phản chiếu và sự sắp xếp phản xạ chuẩn;
- (d) tấm gương hướng hình ảnh và phản xạ đảo ngược.

e) Khả năng sửa lỗi:

- Có 4 mức độ sửa lỗi: L (7%), M (15%), Q (25%), và H (30%)
- Mức độ sửa lỗi cao hơn cho phép khôi phục nhiều dữ liệu hơn nếu mã bị hỏng, nhưng cũng làm tăng kích thước của mã.
- QR Code có khả năng sửa lỗi. Dữ liệu có thể được khôi phục ngay cả khi biểu tượng bị bẩn một phần hoặc bị hỏng. Có thể khôi phục tối đa 30% từ mã*1*2.

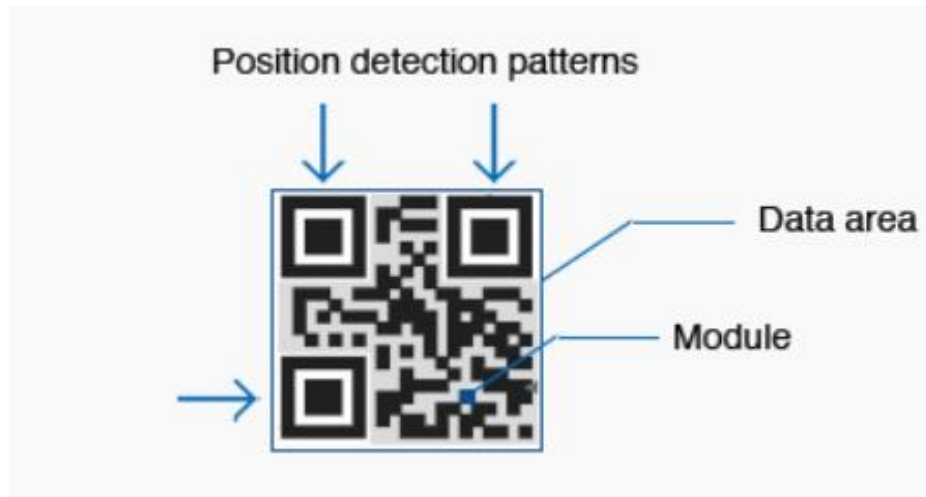


Hình 2. 6 Minh họa khả năng sửa lỗi của QR Code

*1: Từ mã là đơn vị cấu tạo nên vùng dữ liệu. Trong trường hợp Mã QR, một từ mã bằng 8 bit.

*2: Việc khôi phục dữ liệu có thể không được thực hiện đầy đủ tùy thuộc vào lượng bụi bẩn hoặc hư hỏng

f) Định vị căn chỉnh

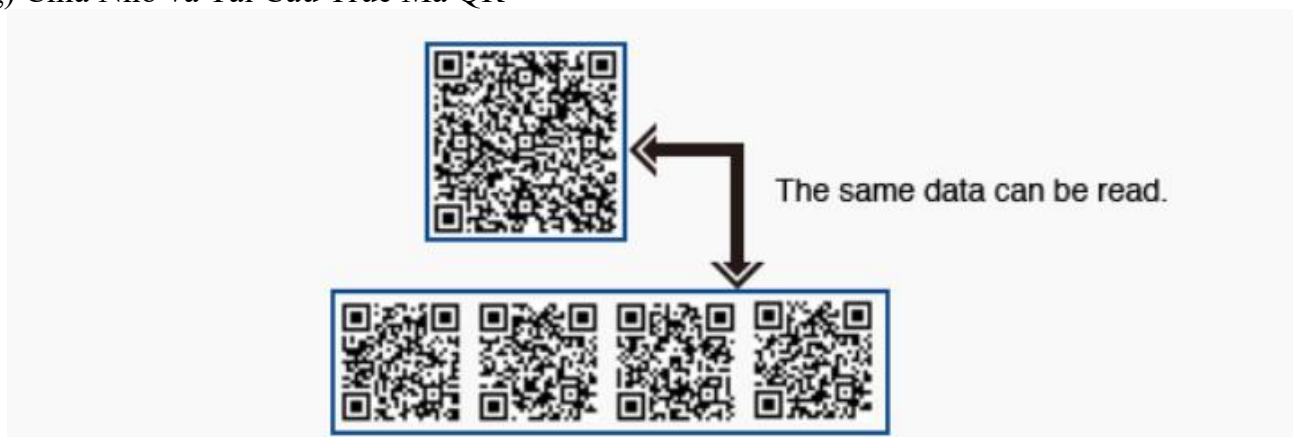


Hình 2. 7 Các thành phần chính định vị căn chỉnh cho một mã QR

- Khả năng đọc toàn diện:
 - Mã QR có thể được đọc ở tốc độ cao từ bất kỳ hướng nào trong 360° (đa hướng).
- Cơ chế hoạt động:
 - Khả năng này đạt được thông qua các mẫu phát hiện vị trí (position detection patterns) đặt ở ba góc của biểu tượng mã QR.
- Lợi ích của mẫu phát hiện vị trí:
 - Đảm bảo việc đọc ổn định ở tốc độ cao.
 - Khắc phục các ảnh hưởng tiêu cực từ nhiễu nền.
- Cấu trúc mã QR (như minh họa trong hình):
 - Position detection patterns: Các mẫu hình vuông lớn ở ba góc của mã QR.
 - Data area: Phần chính của mã QR chứa thông tin được mã hóa.
 - Module: Các ô vuông nhỏ nhất tạo nên mã QR, có thể là đen hoặc trắng.
- Ý nghĩa:
 - Thiết kế này cho phép máy quét nhận diện nhanh chóng hướng và vị trí của mã QR.
 - Người dùng có thể quét mã QR từ bất kỳ góc độ nào mà không cần điều chỉnh vị trí của mã hoặc thiết bị quét.

Tóm lại, tính năng này làm cho mã QR trở nên cực kỳ linh hoạt và dễ sử dụng, cho phép quét nhanh chóng và chính xác trong nhiều tình huống khác nhau, bất kể hướng hoặc góc độ của mã QR so với thiết bị quét.

g) Chia Nhỏ và Tái Cấu Trúc Mã QR



Hình 2. 8 Hình ảnh minh họa tính năng ghép nối của QR

- Chia nhỏ dữ liệu:
 - Một mã QR có thể được chia thành nhiều vùng dữ liệu nhỏ hơn.
 - Ngược lại, thông tin được lưu trữ trong nhiều biểu tượng mã QR có thể được tái cấu trúc thành một biểu tượng dữ liệu duy nhất.
 - Một biểu tượng dữ liệu có thể được chia thành tối đa 16 biểu tượng nhỏ hơn.
 - Ứng dụng:
 - Tính năng này cho phép in mã QR trong một khu vực hẹp hơn.
 - Minh họa trong hình:
 - Phía trên là một mã QR lớn chứa toàn bộ thông tin.
 - Phía dưới là bốn mã QR nhỏ hơn, mỗi mã chứa một phần của thông tin.
 - Mũi tên chỉ ra rằng cả hai cách biểu diễn đều chứa cùng một dữ liệu và có thể được đọc như nhau.
 - Lợi ích:
 - Tính linh hoạt trong việc trình bày thông tin.
 - Khả năng điều chỉnh kích thước và bố cục của mã QR để phù hợp với không gian có sẵn.
 - Duy trì khả năng đọc và tích hợp thông tin ngay cả khi chia nhỏ mã QR.
- Tóm lại, tính năng này cho phép linh hoạt trong việc trình bày và sử dụng mã QR, đặc biệt hữu ích khi cần in mã QR trong không gian hạn chế hoặc khi cần chia nhỏ thông tin thành nhiều phần. Các ký hiệu Mã QR lớn hơn.

2.1.4 Tổng quan về thủ tục mã hóa

- Phần này cung cấp tổng quan về các bước cần thiết để chuyển đổi dữ liệu đầu vào thành biểu tượng QR Code.
- Quy trình mã hóa QR Code bao gồm các bước sau đây:
 - **Bước 1: Phân tích dữ liệu**
 - **Mục đích:** Xác định các loại ký tự khác nhau trong luồng dữ liệu đầu vào để lựa chọn chế độ mã hóa phù hợp.
 - **Chi tiết:** QR Code hỗ trợ tính năng Giải thích Kênh Mở rộng để mã hóa dữ liệu khác với bộ ký tự mặc định. QR Code có nhiều chế độ mã hóa khác nhau để chuyển đổi các tập hợp ký tự thành chuỗi nhị phân hiệu quả nhất. Người dùng cần chọn mức Độ phát hiện và Sửa lỗi cần thiết. Nếu không có chỉ định về phiên bản biểu tượng, chọn phiên bản nhỏ nhất có thể chứa dữ liệu.
 - **Bước 2: Mã hóa dữ liệu**
 - **Mục đích:** Chuyển đổi dữ liệu thành chuỗi bit theo chế độ mã hóa đã chọn.
 - **Chi tiết:** Dữ liệu được chuyển đổi thành chuỗi bit theo các quy tắc của chế độ mã hóa. Các chỉ thị chế độ được chèn khi cần thiết để thay đổi chế độ. Cuối chuỗi dữ liệu sẽ có một ký tự kết thúc. Chuỗi bit sau đó được chia thành các từ mã 8 bit và thêm các ký tự đệm nếu cần để đạt đủ số lượng từ mã dữ liệu cho phiên bản.
 - **Bước 3: Mã hóa sửa lỗi**
 - **Mục đích:** Tạo mã sửa lỗi để bảo vệ dữ liệu.
 - **Chi tiết:** Chuỗi từ mã được chia thành các khối theo quy định. Mỗi khối sẽ được tạo các từ mã sửa lỗi và các từ mã này được thêm vào cuối chuỗi từ mã dữ liệu.
 - **Bước 4: Cấu trúc thông điệp cuối cùng**
 - **Mục đích:** Kết hợp dữ liệu và mã sửa lỗi thành một chuỗi duy nhất.
 - **Chi tiết:** Các từ mã dữ liệu và sửa lỗi từ mỗi khối được xen kẽ với nhau. Thêm các bit thừa nếu cần thiết để hoàn thành chuỗi.
 - **Bước 5: Đặt module vào ma trận**
 - **Mục đích:** Đặt các module mã hóa vào trong ma trận QR Code.

◦ **Chi tiết:** Các module từ mã, cùng với các mẫu tìm, bộ phân tách, mẫu thời gian và (nếu cần) các mẫu căn chỉnh, được đặt vào ma trận.

• **Bước 6: Mặt nạ dữ liệu**

◦ **Mục đích:** Áp dụng mặt nạ để tối ưu hóa độ cân bằng sáng/tối và giảm thiểu các mẫu không mong muốn.

◦ **Chi tiết:** Các mẫu mặt nạ dữ liệu được áp dụng lần lượt vào khu vực mã hóa của QR Code. Sau đó, đánh giá kết quả và chọn mẫu tối ưu nhất.

• **Bước 7: Thông tin định dạng và phiên bản**

◦ **Mục đích:** Hoàn thành QR Code với thông tin định dạng và phiên bản.

◦ **Chi tiết:** Tạo thông tin định dạng và (nếu cần) thông tin phiên bản, sau đó hoàn thành QR Code.

Quy trình này đảm bảo rằng dữ liệu được mã hóa vào QR Code một cách hiệu quả, bảo vệ được dữ liệu khỏi lỗi và tạo ra biểu tượng có thể đọc được dễ dàng bởi các máy quét.

2.1.5 Phân loại mã QR



Hình 2. 9 Các loại QR code phổ biến

Tùy thuộc vào nhu cầu sử dụng, loại sản phẩm và bề mặt dán mà mã QR được phân thành nhiều loại khác nhau, trong đó phổ biến với các loại mã:

Mã QR model 1

Mã QR gốc, mã có khả năng mã hóa 1.167 chữ số với phiên bản tối đa là 14 (mô-đun 73 x 73)

- **Mã QR model 2**

Mã QR được tạo bằng cách cải tiến mã QR Model 1 để mã này có thể được đọc trong ngay cả khi nó bị biến dạng theo một cách nào đó. Ngày nay thuật ngữ mã QR thường đề cập đến loại mã QR này.

Mã này có thể mã hóa tới 7.089 chữ số với phiên bản tối đa là 40 (mô-đun 177 x 177).

- **Mã QR micro**

Chỉ có một mẫu định hướng cho mã này, giúp có thể in nó trong một không gian nhỏ hơn so với trước đây.

Mã này có thể khả thi ngay cả khi độ rộng của lề của nó là 2 giá trị mô-đun (Mã QR yêu cầu lề ít nhất là 4 giá trị mô-đun xung quanh nó). Phiên bản lớn nhất của mã này là M4 (mô-đun 17 x 17), có thể lưu trữ tới 35 chữ số.

- **Mã rMQR**

Mã rMQR là mã hai chiều, dạng ma trận, dễ đọc và có thể lưu trữ một lượng lớn thông tin, đồng thời có dạng hình chữ nhật giúp dễ dàng in trong không gian hẹp.

Nó giải quyết nhu cầu in trong không gian hẹp, nơi không thể in Mã QR thông thường và để lưu trữ nhiều thông tin hơn có thể với Mã QR siêu nhỏ.

- **Mã SQRC**

Mã QR code có tính năng hạn chế đọc. Mã có thể lưu trữ được cả 2 loại dữ liệu công khai và riêng tư. Dữ liệu riêng tư chỉ có thể được đọc với một đầu đọc chuyên dụng có

khóa mật mã, giúp bảo vệ dữ liệu.

Hình thức của mã SQRC không khác so với mã QR thông thường.

- **Mã QR khung (Frame QR)**

Mã QR khung là mã QR có vùng hay khung để hiển thị hình ảnh. Vì hình dạng hay màu sắc của khung có thể thay đổi linh hoạt. Người dùng hoàn toàn có thể chèn thêm chữ hoặc hình ảnh vào khung nên mã này có thể được ứng dụng trong nhiều trường hợp khác nhau.

Mỗi phiên bản mã QR Code có dung lượng dữ liệu tối đa tùy theo lượng dữ liệu, loại ký tự và mức độ sửa lỗi. Nói cách khác, khi lượng dữ liệu tăng lên, cần có nhiều mô-đun hơn để tạo Mã QR, dẫn đến các ký hiệu Mã QR lớn hơn.

2.1.6 Các tính năng của mã QR



Hình 2. 10 Các tính năng của QR code

- **Khả năng mã hoá dữ liệu tốc độ cao**

Mã QR có khả năng xử lý thông tin gấp vài chục đến vài trăm lần.

Mã QR có khả năng xử lý tất cả các loại dữ liệu chẳng hạn như ký tự số và chữ cái, Kanji, Kana, Hiragana, ký hiệu, nhị phân và mã điều khiển. Có thể mã hoá tối đa 7,089 ký tự trong một mã.

- **Kích thước bản in nhỏ**

Vì mã QR mang thông tin theo cả chiều ngang và chiều dọc nên mã QR có khả năng mã hoá cùng dữ liệu trong 1/10 không gian của mã vạch truyền thông (Đối với kích thước bản in nhỏ hơn nữa, có thể sử dụng mã QR dạng micro)

- **Khả năng mã hoá các loại ký tự đặc biệt**

Mã QR có khả năng mã hoá bộ ký tự Kanji cấp 1 và cấp 2 của hệ thống tiêu chuẩn công nghiệp Nhật Bản (JIS). Một ký tự Kanji hay Kana có chiều rộng đầy đủ được mã hoá thành 13 bit, cho phép mã QR chứa hơn 20% dữ liệu so với các mã 2D khác.

- **Sửa lỗi, khôi phục dữ liệu khi mã bị bẩn, hỏng**

Mã QR có khả năng sửa lỗi hiệu quả. Dữ liệu có thể được khôi phục ngay cả khi mã bị bẩn hay hư hỏng một phần. Có thể khôi phục tối đa 30% từ mã (1 từ mã = 8 bit), tùy thuộc vào mức độ hư hỏng hay bẩn của mã.

- **Khả năng đọc dữ liệu 360 độ**

Mã QR có khả năng đọc dữ liệu 360 độ (đa hướng) với tốc độ đọc cao. Mã QR có khả năng nhận diện thông qua 3 ô vuông ở ba góc của mã. Các ô vuông này đảm bảo khả năng đọc ở tốc độ ổn định, tránh được các vấn đề nhiễu nền.

- **Khả năng mở rộng cấu trúc**

Mã QR có thể được chia thành nhiều vùng dữ liệu. Thông tin được lưu trữ trong nhiều mã QR có thể được gom vào trong 1 mã QR duy nhất. Một mã có thể được chia thành tối đa 16 mã và cho phép in trong không gian hẹp.

2.1.7 Các lĩnh vực áp dụng mã QR

- **Sản xuất**

- Cập nhật nhật ký quá trình làm việc của các đơn vị sản xuất thông qua quét mã QR, thay thế cho việc ghi nhận bằng giấy tờ. Từ đó đảm bảo độ chính xác thông tin và thời gian thực tế làm việc.
- Quản lý quá trình lưu thông và chuyển giao của sản phẩm giữa các bộ phận theo thời gian thông qua dữ liệu trên mã QR, hạn chế lỗi khi sử dụng hồ sơ giấy tờ.
- Kiểm soát tình trạng hàng tồn kho, hàng quá hạn sử dụng và chỉ định chính xác vị trí của sản phẩm thông qua mã QR liên kết trên sản phẩm và kệ chứa khi cần thu hồi hay loại bỏ.
- Định danh từng lô sản phẩm/ sản phẩm riêng lẻ bằng mã QR trong quá trình đóng gói để dễ dàng kiểm soát số lượng và chất lượng sản phẩm được đóng vào thùng khi vận chuyển và truy xuất nguồn gốc khi cần.

- **Vận chuyển**

- Khách hàng tạo nhãn vận chuyển bằng Mã QR và dán nhãn đó lên bưu kiện. Sau đó, tài xế quét mã QR lúc lấy bưu kiện và in một bản khi cần thiết. Bằng cách này, có thể tránh được việc sử dụng nhãn vận chuyển nhiều phần, giúp giảm chi phí.
- Trạng thái giao bưu kiện của các tài xế có thể được quản lý theo thời gian thực thông qua dữ liệu lưu trữ trên hệ thống thông qua mã QR tại các vị trí giao – nhận. Hơn nữa, hướng dẫn giao hàng lại và các thông tin khác có thể được gửi từ trụ sở chính đến tài xế.
- Bằng cách quét mã QR trên mỗi bưu kiện và yêu cầu khách hàng ký chữ ký điện tử của họ trên màn hình của thiết bị đầu cuối di động, các thủ tục có thể trở nên không cần giấy tờ.

- **Bán lẻ**

- Danh sách hàng hóa sẽ nhận được lưu trữ trong thiết bị đầu cuối di động và hàng hóa thực nhận được đối chiếu với danh sách bằng cách quét mã vạch của chúng.
- Người tiêu dùng có thể quét mã QR để truy xuất nguồn gốc, xuất xứ của sản phẩm. Từ đó an tâm lựa chọn sản phẩm chính hãng và chất lượng cao.
- Bằng cách gửi phiếu giảm giá di động đến điện thoại thông minh của khách hàng và đọc chúng bằng máy quét POS, dịch vụ phiếu giảm giá có thể được áp dụng nhanh chóng.
- Hàng trong kho được kiểm soát bằng cách quét mã vạch phía trước các kệ để nhập số lượng hàng trong kho thực tế và các đơn đặt hàng được đặt tự động để có thể duy trì số dư kho phù hợp.
- Việc thanh toán có thể được thực hiện nhanh chóng bằng cách quét mã QR từ các ứng dụng ngân hàng, ví điện tử.

- **Y tế**
 - Nhiều bệnh viện, phòng khám sử dụng mã QR để Định danh bệnh nhân. Thông qua mã QR trên ứng dụng điện thoại của bệnh viện, nhân viên y tế dễ dàng xem hồ sơ và tiền sử bệnh của bệnh nhân.
 - Mã QR trên bao bì thuốc, đơn thuốc và nhãn được sử dụng để truy xuất nguồn gốc thuốc chính hãng, đạt tiêu chuẩn. Từ đó đảm bảo chống buôn lậu thuốc giả đảm bảo an toàn sức khỏe cho người bệnh.
 - Quét mã QR trong bệnh viện và phòng khám để truy cập thông tin chăm sóc sức khỏe. Bằng cách nhúng thông tin sức khỏe vào mã QR và sau đó đặt chúng ở những khu vực dễ nhìn thấy, mọi người có thể quét mã và truy cập thông tin về sức khỏe mà không cần tư vấn từ chuyên gia y tế.
- **Phương tiện công cộng**
 - Mã QR cho phương tiện giao thông công cộng thay thế vé bằng giấy hay thẻ xe buýt. Hành khách chỉ cần xuất trình mã QR để hoàn thành thủ tục lên xe.
 - Hành khách có thể thanh toán thông qua mã QR trên xe một cách dễ dàng thay vì thanh toán bằng thẻ ngân hàng hay tiền mặt.
 - Quét mã QR thực hiện khảo sát để phản hồi về chất lượng, dịch vụ của phương tiện di chuyển hay xem bản đồ/ lộ trình về các địa điểm du lịch có tuyến xe buýt.

2.2 Ứng dụng trong thanh toán điện tử

2.2.1 Quy trình thanh toán

- a) Tạo mã thanh toán:
 - Thông tin giao dịch (số tiền, ID người nhận, mô tả, v.v.) được mã hóa vào QR Code.
 - Mã có thể được tạo động cho mỗi giao dịch hoặc tĩnh cho người nhận cố định.
- b) Quét và xử lý:
 - Người dùng quét mã bằng camera của điện thoại thông minh.
 - Ứng dụng thanh toán giải mã thông tin và hiển thị chi tiết giao dịch.
 - Người dùng xác nhận và hoàn tất thanh toán.
- c) Tích hợp với các cổng thanh toán:
 - QR Code có thể chứa thông tin để kết nối trực tiếp với các cổng thanh toán như VNPAY, PayOn, PayPal, và MoMo.
 - Hỗ trợ các giao thức thanh toán tiêu chuẩn như EMV QR Code.
- d) Xác thực và bảo mật:
 - Sử dụng mã hóa để bảo vệ thông tin nhạy cảm trong QR Code.
 - Tích hợp với các phương thức xác thực như OTP hoặc sinh trắc học.

2.2.2 Đọc QR Code trên điện thoại di động

- a) Quy trình đọc QR Code:
 - Chụp ảnh: Sử dụng camera của điện thoại để chụp hình QR code.
 - Xử lý hình ảnh: Điều chỉnh độ sáng, độ tương phản và khử nhiễu để cải thiện chất lượng hình ảnh.
 - Nhận diện QR Code: Sử dụng thuật toán nhận dạng mẫu để xác định vị trí và hướng của QR code trong hình ảnh.
 - Giải mã: Chuyển đổi mẫu đen trắng thành dữ liệu nhị phân, sau đó giải mã thành thông tin gốc.
 - Xử lý dữ liệu: Phân tích và xử lý thông tin đã giải mã (ví dụ: mở URL, hiển thị thông tin thanh toán).
- b) Công nghệ và thư viện phổ biến:

- ZXing (Zebra Crossing): Thư viện mã nguồn mở phổ biến cho Android, có thể port sang các nền tảng khác.
- AVFoundation: Framework của Apple cho iOS để xử lý hình ảnh và video, bao gồm chức năng quét QR code.
- Google ML Kit: Bộ công cụ của Google cung cấp API để quét mã vạch và QR code trên cả Android và iOS.

c) Tối ưu hóa hiệu suất:

- Sử dụng xử lý hình ảnh real-time thay vì chụp ảnh tĩnh.
- Tối ưu hóa thuật toán nhận diện để giảm thời gian xử lý.
- Sử dụng đa luồng để tách biệt quá trình chụp ảnh và xử lý dữ liệu.

d) Xử lý các trường hợp đặc biệt:

- QR Code bị che khuất một phần: Sử dụng thuật toán sửa lỗi Reed-Solomon để khôi phục dữ liệu.
- Điều kiện ánh sáng kém: Tự động điều chỉnh độ sáng và độ tương phản của camera.
- QR Code có kích thước khác nhau: Sử dụng thuật toán scaling để xử lý QR code ở nhiều kích thước.

e) Trải nghiệm người dùng:

- Thiết kế giao diện trực quan và dễ sử dụng cho ứng dụng quét QR code.
- Cung cấp hướng dẫn cho người dùng về cách quét QR code hiệu quả.
- Tích hợp chức năng quét QR code vào các ứng dụng hiện có (ví dụ: ứng dụng ngân hàng, ví điện tử).

2.2.3 Ưu điểm của QR Code trong thanh toán .

- Vào năm 2017, Apple đã kích hoạt camera iPhone để quét mã QR, và chúng được hồi sinh. Android sau đó cũng làm theo, và đột nhiên mã QR trở nên dễ tiếp cận hơn với người tiêu dùng hàng ngày.
- Khi đại dịch xảy ra, các thanh toán bằng mã QR đã trở thành một phần của cuộc sống hàng ngày. Nó cho phép người tiêu dùng thực hiện các thanh toán an toàn, không tiếp xúc, khám phá menu và dịch vụ mà không cần phải nói chuyện trực tiếp với nhân viên và cho phép khách hàng kiểm tra vào các địa điểm một cách nhanh chóng và thuận tiện.
- Mã QR thực sự là một lối tắt đa năng để tiếp cận Internet. QR là viết tắt của "Phản hồi nhanh" và trong vài giây, người dùng có thể thực hiện thanh toán không tiếp xúc mà không tốn nhiều công sức, mang lại trải nghiệm thoải mái cho khách hàng.
- Một cách đơn giản để chấp nhận thanh toán - không giống như thẻ tín dụng và thẻ ghi nợ, yêu cầu đầu đọc thẻ đắt tiền, phí hợp đồng và các điều khoản và điều kiện, việc thiết lập và bắt đầu sử dụng mã QR để nhận thanh toán tương đối dễ dàng. Thông thường, tất cả những gì bạn cần làm là tải xuống một ứng dụng, làm cho việc triển khai trực tuyến hoặc trên nhiều địa điểm cực kỳ dễ dàng.
- Loại bỏ xích mích với hành trình khách hàng - thanh toán bằng mã QR nhanh chóng và thuận tiện, điều này thật tuyệt vời cho cả bạn và khách hàng của bạn. Điều đó có nghĩa là khách hàng hài lòng hơn, nhiều chuyển đổi hơn, và nhiều doanh thu hơn.
- Cung cấp khả năng kết hợp thanh toán và các chương trình khách hàng thân thiết - sử dụng mã QR là một cách tuyệt vời để xây dựng chương trình khách hàng thân thiết thành công. Thứ nhất, vì mọi thứ đều có thể được thực hiện trên các thiết bị thông minh, chúng loại bỏ sự cần thiết của thẻ và biên lai khách hàng thân thiết vật lý. Và thứ hai, vì khách hàng của bạn dễ dàng đăng ký và thu thập điểm, họ có nhiều khả năng làm như vậy hơn và gắn bó với chương trình của bạn.

- Cải thiện bảo mật - vì tất cả các thông tin thanh toán được mã hóa và không cần lưu trữ thông tin cá nhân từ khách hàng của bạn để hoàn thành giao dịch, mã QR mang lại sự an toàn và cách thanh toán tập trung.
- Thanh toán bằng mã QR là một trong những phương thức thanh toán an toàn nhất nhờ vào các yếu tố chính sau:
 - Kiểm tra thẻ tín dụng và thẻ ghi nợ: Các khoản thanh toán qua thẻ đi qua mạng lưới như Visa hoặc Mastercard, đảm bảo giao dịch ổn định và an toàn.
 - Phương thức thanh toán địa phương: Khách hàng có thể chọn phương thức thanh toán địa phương ưa thích của mình. Việc cung cấp dịch vụ an toàn và quen thuộc giúp khách hàng tự tin hơn khi tin tưởng doanh nghiệp của bạn thông qua thanh toán bằng mã QR.
 - Mã hóa và Tokenization: Mã QR được mã hóa và không cần lưu trữ dữ liệu khách hàng để thực hiện thanh toán. Với việc không có thông tin cá nhân để đánh cắp, thanh toán bằng mã QR có nguy cơ gian lận rất thấp.

Làm thế nào để các nhà bán hàng sử dụng liên kết thanh toán với mã QR?

- Nhà bán hàng có thể tạo liên kết thanh toán để sử dụng với mã QR. Liên kết thanh toán cho phép bạn chấp nhận thanh toán qua bất kỳ kênh hoặc thiết bị nào, đáp ứng sở thích của khách hàng. Bạn có thể tạo liên kết thanh toán trong API hoặc bảng điều khiển. Mỗi mã có thể được sử dụng để chấp nhận một khoản thanh toán duy nhất và có giá trị trong 24 giờ. Khách hàng chỉ cần quét mã QR để được đưa đến trang thanh toán và hoàn tất giao dịch.
- Trường hợp sử dụng mã QR cho các doanh nghiệp toàn cầu:
- Các khu vực khác nhau đã chấp nhận mã QR với tốc độ khác nhau. Ví dụ, người tiêu dùng ở Trung Quốc đã sử dụng mã QR với các nhà bán lẻ lớn, các nhà bán hàng đường phố và nhiều hơn nữa. Thị trường đã phát triển nhanh chóng và không có dấu hiệu chậm lại. WeChat Pay và Alipay cho phép người tiêu dùng thanh toán tại cửa hàng bằng cách quét mã hoặc hiển thị mã cá nhân của họ cho nhân viên thu ngân. Tiền sau đó được trừ từ ví di động, thường được liên kết với tài khoản ngân hàng.
- Trong tương lai, các địa chỉ ví tiền điện tử sẽ bao gồm một loạt các chữ cái và số — không dài quá 34 ký tự. Một ứng dụng tạo mã QR có thể biến số dài này thành mã QR để sử dụng, có thể dễ dàng quét nếu ai đó muốn thực hiện thanh toán ...
- Nhanh chóng và tiện lợi
- Đa nền tảng và linh hoạt
 - Hoạt động trên hầu hết các thiết bị di động hiện đại.
 - Có thể sử dụng cho cả giao dịch trực tuyến và tại cửa hàng.
 - Chi phí thấp
 - Không cần thiết bị đặc biệt để tạo hoặc đọc mã.
 - Giảm chi phí xử lý tiền mặt và các khoản phí liên quan đến thẻ tín dụng.
 - Tăng cường trải nghiệm khách hàng:
 - Quy trình thanh toán đơn giản và trực quan.
 - Tạo cảm giác hiện đại và tiên tiến cho doanh nghiệp.
 - Khả năng tích hợp cao
 - Dễ dàng tích hợp vào các hệ thống quản lý bán hàng hiện có.
 - Hỗ trợ các chương trình khuyến mãi và loyalty thông qua mã động.

2.2.4 Thách thức và giải pháp

- Việc áp dụng QR Code trong thanh toán đối mặt với một số thách thức đáng kể. Về bảo mật, QR Code có nguy cơ bị thay thế bởi mã độc hoặc dẫn đến trang web giả mạo. Để giải quyết vấn đề này, các giải pháp như sử dụng mã hóa end-to-end cho thông tin

nhạy cảm, tích hợp chữ ký số để xác thực nguồn gốc của mã, và áp dụng các biện pháp xác thực bổ sung như OTP đang được triển khai. Sự phụ thuộc vào Internet cũng là một rào cản, khi thanh toán có thể bị gián đoạn do mất kết nối. Giải pháp cho vấn đề này bao gồm phát triển các phương thức thanh toán offline sử dụng QR Code động và tối ưu hóa ứng dụng để hoạt động với băng thông thấp. Nhận thức của người dùng cũng là một thách thức, khi một số người có thể không quen thuộc hoặc tin tưởng vào thanh toán QR. Để khắc phục, cần triển khai các chiến dịch giáo dục người dùng về cách sử dụng an toàn QR Code và cung cấp hướng dẫn rõ ràng cùng hỗ trợ tại điểm bán hàng. Cuối cùng, việc tiêu chuẩn hóa cũng gặp khó khăn khi các định dạng QR khác nhau có thể gây nhầm lẫn. Giải pháp cho vấn đề này là thúc đẩy việc áp dụng các tiêu chuẩn như EMV QR Code cho thanh toán và phát triển các API và SDK chuẩn hóa cho việc tích hợp QR Code vào ứng dụng thanh toán.

2.2.5 Xu hướng tương lai

- Tương lai của QR Code trong thanh toán hứa hẹn nhiều đổi mới và cải tiến đáng kể. QR Code động, có khả năng thay đổi nội dung theo thời gian thực, sẽ hỗ trợ các giao dịch một lần và tăng cường bảo mật. Việc tích hợp AI và Machine Learning sẽ cải thiện khả năng nhận dạng và xử lý QR Code trong điều kiện khó khăn, đồng thời phát hiện mã độc và gian lận tự động. Kết hợp với công nghệ Blockchain sẽ tăng cường tính minh bạch và không thể thay đổi của giao dịch, đồng thời hỗ trợ thanh toán bằng tiền điện tử thông qua QR Code. Khả năng lưu trữ của QR Code cũng sẽ được mở rộng với các phiên bản mới có thể chứa nhiều thông tin hơn, thậm chí tích hợp dữ liệu đa phương tiện như hình ảnh và âm thanh. Trải nghiệm thực tế ảo tăng cường (AR) sẽ được tăng cường thông qua việc sử dụng QR Code như một điểm kích hoạt cho trải nghiệm AR trong thanh toán và mua sắm, cung cấp thông tin sản phẩm và dịch vụ tương tác. Cuối cùng, việc kết hợp sinh trắc học với QR Code sẽ tăng cường xác thực và phát triển các giải pháp thanh toán không chạm sử dụng kết hợp QR và nhận dạng khuôn mặt.

2.3 Nhận xét và đề xuất

2.3.1 Nhận xét

- QR Code đã trở thành công nghệ quan trọng trong lĩnh vực thanh toán điện tử, mang lại nhiều lợi ích như tốc độ, tiện lợi và chi phí thấp. Công nghệ này đáp ứng nhu cầu ngày càng tăng về thanh toán không tiếp xúc và được ứng dụng rộng rãi trong nhiều ngành nghề và lĩnh vực. Việc tích hợp QR Code với các công thanh toán đã mở ra nhiều cơ hội, cải thiện đáng kể trải nghiệm thanh toán cho khách hàng, tạo ra hệ sinh thái thanh toán đa dạng và linh hoạt, đồng thời hỗ trợ doanh nghiệp trong quá trình chuyển đổi số. Mặc dù còn tồn tại một số thách thức về bảo mật và tích hợp, các giải pháp kỹ thuật hiện đại như mã hóa end-to-end đang được áp dụng để giải quyết hiệu quả. Sự phát triển của các tiêu chuẩn chung cũng giúp đơn giản hóa quá trình tích hợp và nâng cao nhận thức của người dùng về bảo mật trong thanh toán QR Code. Xu hướng phát triển của QR Code trong tương lai rất hứa hẹn, với việc tích hợp AI và Machine Learning để nâng cao hiệu suất và bảo mật, kết hợp với blockchain để tăng tính minh bạch và không thể thay đổi của giao dịch, cũng như mở rộng ứng dụng trong thực tế ảo tăng cường (AR) để tạo ra trải nghiệm mua sắm mới.

2.3.2 Đề xuất

- Để tiếp tục phát triển và tối ưu hóa việc sử dụng QR Code trong thanh toán, cần đầu tư vào nghiên cứu và phát triển các giải pháp bảo mật tiên tiến, tập trung vào các công nghệ như mã hóa lượng tử và sinh trắc học, phát triển các thuật toán phát hiện gian lận thời gian thực, và tăng cường hợp tác giữa các tổ chức tài chính và công ty công nghệ. Việc phát triển một framework tích hợp đa nền tảng cũng rất quan trọng, bao gồm việc

tạo ra bộ công cụ phát triển (SDK) chuẩn hóa cho các nhà phát triển, hỗ trợ tích hợp liền mạch với nhiều cổng thanh toán khác nhau, và đảm bảo tính tương thích trên các hệ điều hành và thiết bị khác nhau. Đồng thời, cần tăng cường hợp tác giữa các bên liên quan để xây dựng và áp dụng các tiêu chuẩn chung cho thanh toán QR Code, đẩy mạnh giáo dục người dùng về cách sử dụng an toàn và hiệu quả QR Code trong thanh toán, tiếp tục nghiên cứu và thử nghiệm các ứng dụng mới của QR Code trong lĩnh vực thương mại điện tử và fintech. Cuối cùng, việc xem xét phát triển một hệ thống thanh toán QR Code thống nhất quốc gia có thể giúp tăng cường khả năng tương tác giữa các hệ thống thanh toán khác nhau, tạo điều kiện thuận lợi cho việc áp dụng rộng rãi công nghệ này trong nền kinh tế.

CHƯƠNG III. TÌM HIỂU THỦ TỤC THANH TOÁN TIỀN QUA API CỦA VNPAY, PAYON, PAYPAL VÀ MOMO

3.1 Giới thiệu

- Thanh toán qua QR code đã trở thành một xu hướng quan trọng trong ngành công nghệ tài chính nhờ vào sự tiện lợi và bảo mật. Báo cáo này sẽ nghiên cứu và so sánh các dịch vụ thanh toán QR code từ VNPay, PayPal, PayOn, và MoMo. Mỗi dịch vụ sẽ được phân tích về quy trình công nghệ, bảo mật, và tiện ích người dùng.
- VNPay, một trong những dịch vụ thanh toán QR code phổ biến tại Việt Nam, đã chứng minh khả năng tích hợp sâu vào hệ thống ngân hàng và dịch vụ tài chính của quốc gia. PayPal, với sự hiện diện toàn cầu, cung cấp một giải pháp thanh toán QR code mạnh mẽ và đáng tin cậy, đặc biệt cho các giao dịch quốc tế. PayOn, mặc dù không nổi bật bằng những tên tuổi khác, đang dần chứng tỏ mình với các dịch vụ thanh toán QR code chuyên biệt cho các thị trường mục tiêu. MoMo, nổi tiếng trong lĩnh vực ví điện tử tại Việt Nam, cung cấp một giải pháp thanh toán QR code tiện lợi với sự tích hợp vào hệ sinh thái tài chính rộng lớn.

3.2 Cổng thanh toán VNPay

- VNPay là một trong những dịch vụ thanh toán QR code hàng đầu tại Việt Nam, được tích hợp rộng rãi trong hệ thống ngân hàng và các ứng dụng ví điện tử. Được phát triển bởi công ty Cổ phần Giải pháp Thanh toán Việt Nam (VNPAY), dịch vụ này đã trở thành một phần quan trọng trong hệ sinh thái thanh toán điện tử của Việt Nam..

3.2.1 Quy trình công nghệ

- VNPay sử dụng mã QR tĩnh và tự động để thực hiện giao dịch. Mã QR tĩnh được sử dụng cho các giao dịch định kỳ như thanh toán hóa đơn, trong khi mã QR động được tạo ra cho từng giao dịch cụ thể.

3.2.2 Bảo mật

VNPay áp dụng nhiều lớp bảo mật để đảm bảo an toàn cho giao dịch:

- Mã hóa dữ liệu:
 - Sử dụng các thuật toán mã hóa tiên tiến để bảo vệ thông tin giao dịch.
 - Áp dụng chuẩn bảo mật SSL/TLS trong quá trình truyền tải dữ liệu.
- Xác thực hai yếu tố:
 - Yêu cầu người dùng xác nhận giao dịch thông qua mật khẩu và OTP.
 - Có thể tích hợp xác thực sinh trắc học (vân tay, nhận diện khuôn mặt) trên các thiết bị hỗ trợ.
- Giám sát giao dịch theo thời gian thực:
 - Hệ thống AI phát hiện các giao dịch bất thường.
 - Cảnh báo và tạm khóa tài khoản khi phát hiện hoạt động đáng ngờ.
- Tuân thủ các tiêu chuẩn bảo mật quốc tế:
 - Đạt chứng nhận PCI DSS (Payment Card Industry Data Security Standard).
 - Áp dụng các biện pháp bảo vệ thông tin cá nhân theo quy định GDPR.
- Hạn mức giao dịch:
 - Thiết lập hạn mức giao dịch hàng ngày để giảm thiểu rủi ro.
 - Cho phép người dùng tùy chỉnh hạn mức theo nhu cầu.

3.2.3 Tiện ích người dùng

- Thanh toán hóa đơn:
 - Hóa đơn điện, nước, internet, truyền hình cáp.
 - Hóa đơn điện thoại di động và cố định.

- Thanh toán phí bảo hiểm và học phí.
- Mua sắm và thanh toán:
- Thanh toán tại các cửa hàng offline có hỗ trợ QR VNPay.
- Tích hợp với các sàn thương mại điện tử lớn.
- Đặt vé máy bay, tàu xe, khách sạn.
- Chuyển tiền:
- Chuyển tiền nhanh chóng giữa các tài khoản ngân hàng.
- Chuyển tiền qua số điện thoại hoặc mã QR.
- Hỗ trợ chia sẻ hóa đơn giữa nhiều người.
- Dịch vụ tài chính:
- Mua bảo hiểm trực tuyến.
- Đầu tư vào các sản phẩm tài chính (chứng chỉ quỹ, trái phiếu).
- Tra cứu thông tin tín dụng.
- Tiện ích khác:
- Nạp tiền điện thoại và mua mã thẻ game.
- Đặt lịch và thanh toán dịch vụ y tế.
- Tích điểm và đổi ưu đãi từ các chương trình khuyến mãi.
- Quản lý tài chính cá nhân:
- Theo dõi chi tiêu và lập báo cáo tài chính.
- Thiết lập mục tiêu tiết kiệm.
- Nhắc nhở thanh toán hóa đơn định kỳ.

3.2.4 Timeline

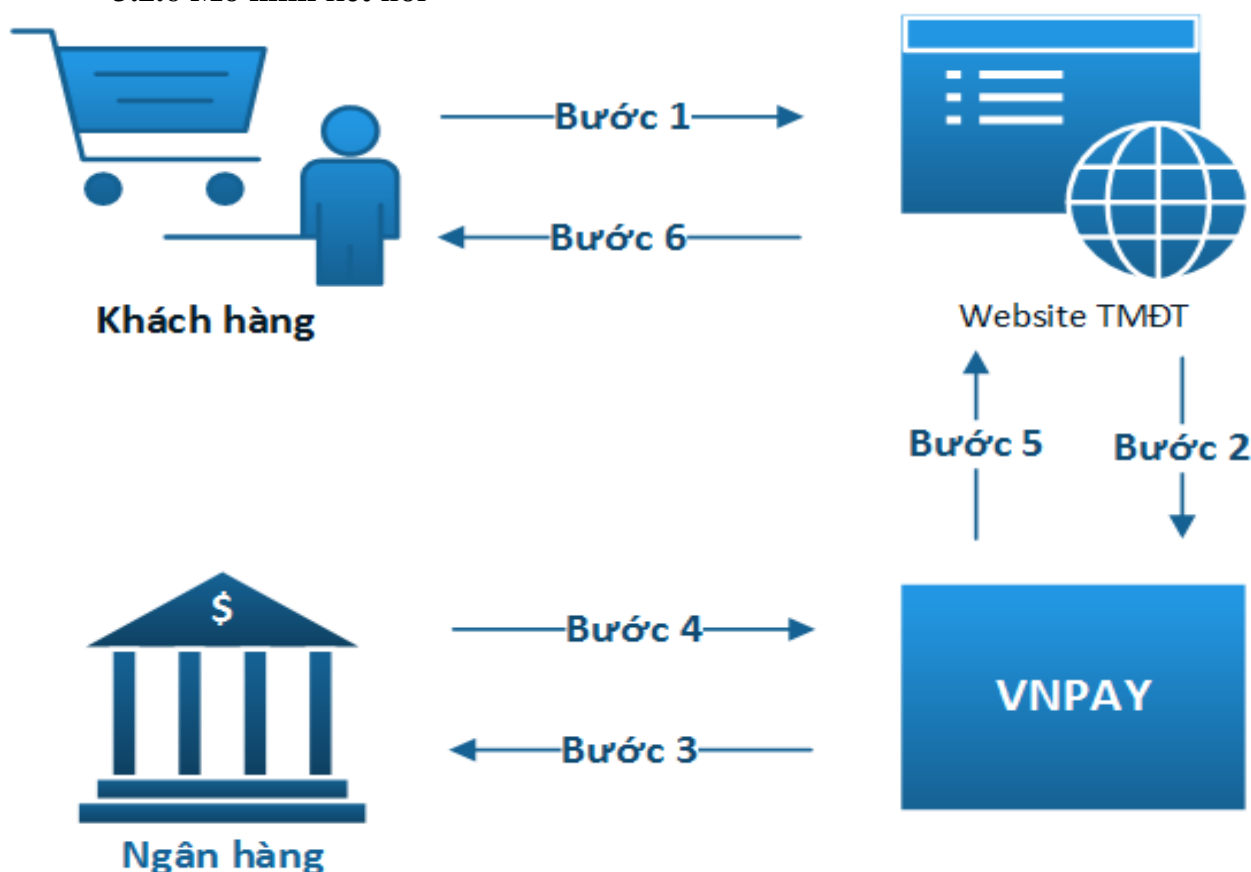


Hình 3. 1 Timeline tích hợp VNPay

3.2.5 Các bước merchant cần xử lý tích hợp code cài đặt

- (1) Cài đặt code build URL thanh toán chuyển hướng.
- (2) Cài đặt code vnp_ReturnUrl URL thông báo kết quả thanh toán.
- (3) Cài đặt code IPN URL cập nhật kết quả thanh toán. Gửi lại VNPAY URL này khi thiết lập xong.

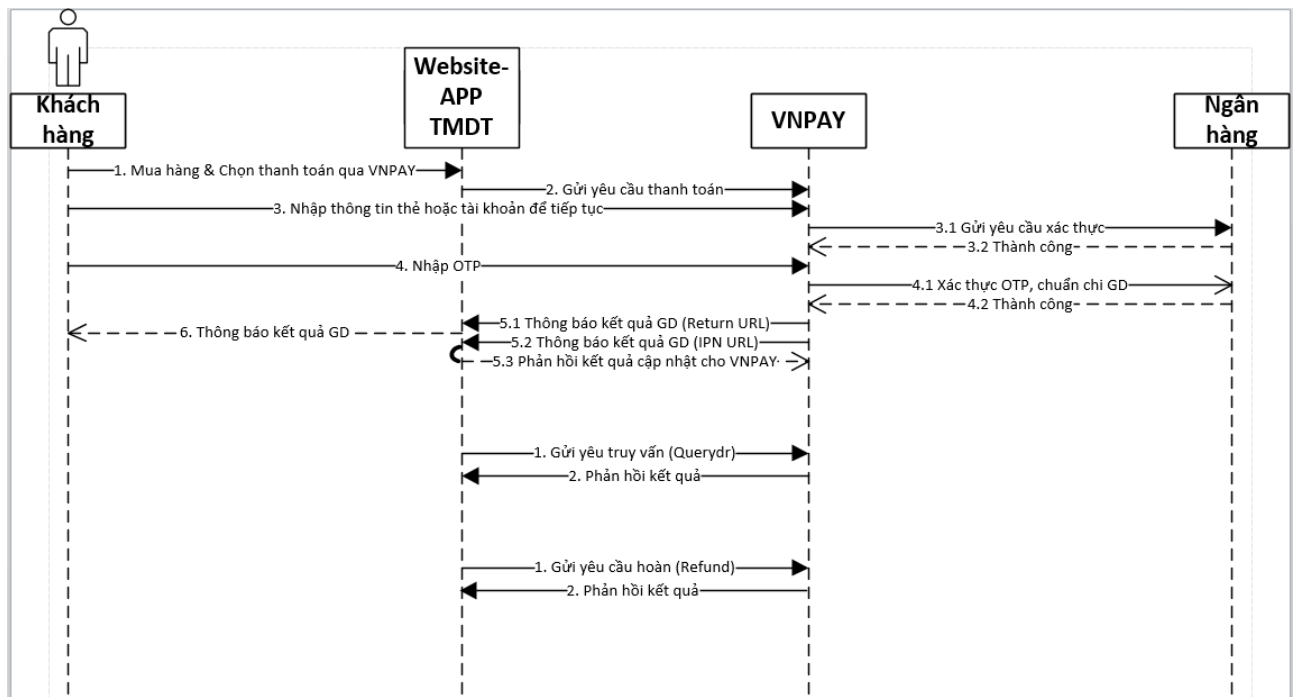
3.2.6 Mô hình kết nối



Sơ đồ 3. 1 Mô hình kết nối của VNPay

- **Bước 1:** Khách hàng thực hiện mua hàng trên Website - ứng dụng TMĐT và tiến hành thanh toán trực tuyến cho đơn hàng.
- **Bước 2:** Website - ứng dụng TMĐT thành lập yêu cầu thanh toán dưới dạng URL mang thông tin thanh toán và chuyển hướng khách hàng sang Cổng thanh toán VNPAY bằng URL đó.
Cổng thanh toán VNPAY xử lý yêu cầu thanh toán mà Website - ứng dụng TMĐT gửi sang. Khách hàng tiến hành nhập hoặc xử lý xác thực các thông tin được yêu cầu Thanh toán.
- **Bước 3,4:** Khách hàng nhập thông tin để xác minh tài khoản Ngân hàng của khách hàng và xác thực giao dịch (Nhập thông tin tài khoản, thẻ hoặc quét mã VNPAY-QR).
- **Bước 5:** Giao dịch thành công tại Ngân hàng, VNPAY tiến hành:
 - Chuyển hướng khách hàng về Website - ứng dụng TMĐT (vnp_ReturnUrl)
 - Thông báo cho Website - ứng dụng TMĐT kết quả thanh toán của khách hàng thông qua IPN URL. Merchant cập nhật kết quả thanh toán VNPAY gửi tại URL này.
- **Bước 6:** Merchant hiển thị kết quả giao dịch tới khách hàng (vnp_ReturnUrl).

3.2.7 Sơ đồ tuần tự



Sơ đồ 3. 2 Sơ đồ tuần tự của VNPay

3.2.8 Thông tin cấu hình

- Các thông tin cần thiết kết nối vào môi trường Sandbox Cổng thanh toán VNPAY:
- Mã TmnCode là mã định danh kết nối được khai báo tại hệ thống của VNPAY. Mã định danh tương ứng với tên miền website, ứng dụng, dịch vụ của merchant kết nối vào VNPAY. Mỗi đơn vị có thể có một hoặc nhiều mã TmnCode kết nối.
- URL thanh toán (Sandbox): <https://sandbox.vnpayment.vn/paymentv2/vpcpay.html>
- Secret Key vnp_HashSecret Chuỗi bí mật sử dụng để kiểm tra toàn vẹn dữ liệu khi hai hệ thống trao đổi thông tin (checksum).
- URL truy vấn kết quả giao dịch - hoàn tiền (Sandbox): https://sandbox.vnpayment.vn/merchant_webapi/api/transaction
- Nếu chưa có thông tin cấu hình tích hợp, bạn có thể đăng ký ngay tại đây <http://sandbox.vnpayment.vn/devreg/> Hệ thống sẽ gửi thông tin kết nối về email bạn đăng ký
- Tạo URL Thanh toán
- URL thanh toán (Sandbox): <https://sandbox.vnpayment.vn/paymentv2/vpcpay.html>
- Phương thức: GET
- URL Thanh toán là địa chỉ URL mang thông tin thanh toán. Website TMĐT gửi sang Cổng thanh toán VNPAY các thông tin này khi xử lý giao dịch thanh toán trực tuyến cho Khách mua hàng.
- URL có dạng
- https://sandbox.vnpayment.vn/paymentv2/vpcpay.html?vnp_Amount=1806000&vnp_Command=pay&vnp_CreateDate=20210801153333&vnp_CurrCode=VND&vnp_IpAddr=127.0.0.1&vnp_Locale=vn&vnp_OrderInfo=Thanh+toan+don+hang+%3A5&vnp_OrderType=other&vnp_ReturnUrl=https%3A%2F%2Fdomainmerchant.vn%2FReturnUrl&vnp_TmnCode=DEMOV210&vnp_TxnRef=5&vnp_Version=2.1.0&vnp_SecureHash=3e0d61a0c0534b2e36680b3f7277743e8784cc4e1d68fa7d276e79c23be7d6318d338b477910a27992f5057bb1582bd44bd82ae8009ffaf6d141219218625c42

- Danh sách tham số - Thông tin gửi sang VNPAY (vnp_Command=pay)

Tham số	Kiểu dữ liệu	Bắt buộc/Tùy chọn	Mô tả
<u>vnp_Version</u>	Alphanumeric[1,8]	Bắt buộc	Phiên bản api mà merchant kết nối. Phiên bản hiện tại là : 2.1.0
<u>vnp_Command</u>	Alpha[1,16]	Bắt buộc	Mã API sử dụng, mã cho giao dịch thanh toán là: pay
<u>vnp_TmnCode</u>	Alphanumeric[8]	Bắt buộc	Mã website của merchant trên hệ thống của VNPAY. Ví dụ: 2QXUI4J4
<u>vnp_Amount</u>	Numeric[1,12]	Bắt buộc	Số tiền thanh toán. Số tiền không mang các ký tự phân tách thập phân, phần nghìn, ký tự tiền tệ. Để gửi số tiền thanh toán là 10,000 VND (mười nghìn VNĐ) thì merchant cần nhân thêm 100 lần (khử phần thập phân), sau đó gửi sang VNPAY là: 1000000
<u>vnp_BankCode</u>	Alphanumeric[3,20]	Tùy chọn	Mã phương thức thanh toán, mã loại ngân hàng hoặc ví điện tử thanh toán. Nếu không gửi sang tham số này, chuyển hướng người dùng sang VNPAY chọn phương thức thanh toán. Lưu ý: Các mã loại hình thức thanh toán lựa chọn tại website-ứng dụng của merchant vnp_BankCode=VNPAYQRThanh toán quét mã QR vnp_BankCode=VNBANKThẻ ATM - Tài khoản ngân hàng nội địa vnp_BankCode=INTCARDThẻ thanh toán quốc tế
<u>vnp_CreateDate</u>	Numeric[14]	Bắt buộc	Là thời gian phát sinh giao dịch định dạng yyyyMMddHHmmss (Time zone GMT+7) Ví dụ: 20220101103111
<u>vnp_CurrCode</u>	Alpha[3]	Bắt buộc	Đơn vị tiền tệ sử dụng thanh toán. Hiện tại chỉ hỗ trợ VND
<u>vnp_IpAddr</u>	Alphanumeric[7,45]	Bắt buộc	Địa chỉ IP của khách hàng thực hiện giao dịch. Ví dụ: 13.160.92.202

<u>vnp_Locale</u>	Alpha[2,5]	Bắt buộc	Ngôn ngữ giao diện hiển thị. Hiện tại hỗ trợ Tiếng Việt (vn), Tiếng Anh (en)
<u>vnp_OrderInfo</u>	Alphanumeric[1,255]	Bắt buộc	Thông tin mô tả nội dung thanh toán quy định dữ liệu gửi sang VNPAY (Tiếng Việt không dấu và không bao gồm các ký tự đặc biệt) Ví dụ: Nạp tiền cho thuê bao 0123456789. Số tiền 100,000 VND
<u>vnp_OrderType</u>	Alpha[1,100]	Bắt buộc	Mã danh mục hàng hóa. Mỗi hàng hóa sẽ thuộc một nhóm danh mục do VNPAY quy định. Xem thêm bảng Danh mục hàng hóa
<u>vnp_ReturnUrl</u>	Alphanumeric[10,255]	Bắt buộc	URL thông báo kết quả giao dịch khi Khách hàng kết thúc thanh toán. Ví dụ: https://domain.vn/VnPayReturn
<u>vnp_ExpireDate</u>	Numeric[14]	Bắt buộc	Thời gian hết hạn thanh toán GMT+7, định dạng: yyyyMMddHHmmss
<u>vnp_TxnRef</u>	Alphanumeric[1,100]	Bắt buộc	Mã tham chiếu của giao dịch tại hệ thống của merchant. Mã này là duy nhất dùng để phân biệt các đơn hàng gửi sang VNPAY. Không được trùng lặp trong ngày. Ví dụ: 23554
<u>vnp_SecureHash</u>	Alphanumeric[32,256]	Bắt buộc	Mã kiểm tra (checksum) để đảm bảo dữ liệu của giao dịch không bị thay đổi trong quá trình chuyển từ merchant sang VNPAY. Việc tạo ra mã này phụ thuộc vào cấu hình của merchant và phiên bản api sử dụng. Phiên bản hiện tại hỗ trợ SHA256, HMACSHA512.

Bảng 3. 1 Danh sách tham số - Thông tin gửi sang VNPAY (vnp_Command=pay)

- Lưu ý
- Dữ liệu checksum được thành lập dựa trên việc sắp xếp tăng dần của tên tham số (QueryString)
- Số tiền cần thanh toán nhân với 100 để triệt tiêu phần thập phân trước khi gửi sang VNPAY
- vnp_BankCode: Giá trị này tùy chọn.
- Nếu loại bỏ tham số không gửi sang, khách hàng sẽ chọn phương thức thanh toán, ngân hàng thanh toán tại VNPAY.
- Nếu thiết lập giá trị (chọn Ngân hàng thanh toán tại Website-ứng dụng TMĐT),

Tham khảo bảng mã trả về tại API:

Endpoint: https://sandbox.vnpayment.vn/qrpayauth/api/merchant/get_bank_list

Http method: POST

Content-Type: application/x-www-form-urlencoded

key tmn_code

value Theo mã định danh kết nối (vnp_TmnCode) VNPAY cung cấp

- Trong URL thanh toán có tham số vnp_ReturnUrl là URL thông báo kết quả giao dịch khi Khách hàng kết thúc thanh toán

Cài đặt Code IPN URL

- Phương thức: GET

- Yêu cầu:

- IPN URL cần có SSL
- Nhận kết quả phản hồi từ Cổng thanh toán VNPAY, kiểm tra dữ liệu, cập nhật kết quả và phản hồi lại mã lỗi và mô tả mã lỗi (RspCode và Message) cho server VNPAY nhận biết

Đây là địa chỉ để hệ thống merchant nhận kết quả thanh toán trả về từ VNPAY. Trên URL VNPAY gọi về có mang thông tin thanh toán để căn cứ vào kết quả đó Website TMĐT xử lý các bước tiếp theo (ví dụ: cập nhật kết quả thanh toán vào Database ...) VNPAY trả về kết quả thanh toán URL có dạng:

https://{domain}/IPN?vnp_Amount=1000000&vnp_BankCode=NCB&vnp_BankTranNo=VNP14226112&vnp_CardType=ATM&vnp_OrderInfo=Thanh+toan+don+hang+thoi+gian%3A+2023-12-07+17%3A00%3A44&vnp_PayDate=20231207170112&vnp_ResponseCode=00&vnp_TmnCode=CTTVNP01&vnp_TransactionNo=14226112&vnp_TransactionStatus=00&vnp_TxnRef=166117&vnp_SecureHash=b6dababca5e07a2d8e32fdd3cf05c29cb426c721ae18e9589f7ad0e2db4b657c6e0e5cc8e271cf745162bcb100fdf2f64520554a6f5275bc4c5b5b3e57dc4b4b

- Danh sách tham số - Thông tin nhận về từ VNPAY (vnp_Command=pay)

Tham số	Kiểu dữ liệu	Bắt buộc/Tùy chọn	Mô tả
<u>vnp_TmnCode</u>	Alphanumeric[8]	Bắt buộc	Mã website của merchant trên hệ thống của VNPAY. Ví dụ: 2QXUI4J4
<u>vnp_Amount</u>	Numeric[1,12]	Bắt buộc	Số tiền thanh toán. VNPAY phản hồi số tiền nhân thêm 100 lần.
<u>vnp_BankCode</u>	Alphanumeric[3,20]	Bắt buộc	Mã Ngân hàng thanh toán. Ví dụ: NCB
<u>vnp_BankTranNo</u>	Alphanumeric[1,255]	Tùy chọn	Mã giao dịch tại Ngân hàng. Ví dụ: NCB20170829152730
<u>vnp_CardType</u>	Alpha[2,20]	Tùy chọn	Loại tài khoản/thẻ khách hàng sử dụng: ATM, QR CODE

<u>vnp_PayDate</u>	Numeric[14]	Tùy chọn	Thời gian thanh toán. Định dạng: yyyyMMddHHmmss
<u>vnp_OrderInfo</u>	Alphanumeric[1,255]	Bắt buộc	Thông tin mô tả nội dung thanh toán (Tiếng Việt, không dấu). Ví dụ: **Nạp tiền cho thuê bao 0123456789. Số tiền 100,000 VND**
<u>vnp_TransactionNo</u>	Numeric[1,15]	Bắt buộc	Mã giao dịch ghi nhận tại hệ thống VNPAY. Ví dụ: 20170829153052
<u>vnp_ResponseCode</u>	Numeric[2]	Bắt buộc	Mã phản hồi kết quả thanh toán. Quy định mã trả lời 00 ứng với kết quả Thành công cho tất cả các API. <u>Tham khảo thêm tại bảng mã lỗi</u>
<u>vnp_TransactionStatus</u>	Numeric[2]	Bắt buộc	Mã phản hồi kết quả thanh toán. Tình trạng của giao dịch tại Cổng thanh toán VNPAY. -00: Giao dịch thanh toán được thực hiện thành công tại VNPAY -Khác 00: Giao dịch không thành công tại VNPAY <u>Tham khảo thêm tại bảng mã lỗi</u>
<u>vnp_TxnRef</u>	Alphanumeric[1,100]	Bắt buộc	Giống mã gửi sang VNPAY khi gửi yêu cầu thanh toán. Ví dụ: 23554
<u>vnp_SecureHashType</u>	Alphanumeric[3,10]	Tùy chọn	Loại mã băm sử dụng: SHA256, HmacSHA512
<u>vnp_SecureHash</u>	Alphanumeric[32,256]	Bắt buộc	Mã kiểm tra (checksum) để đảm bảo dữ liệu của giao dịch không bị thay đổi trong quá trình chuyển từ VNPAY về Website TMĐT. Cần kiểm tra đúng checksum khi bắt đầu xử lý yêu cầu (trước khi thực hiện các yêu cầu khác)

Bảng 3. 2 Danh sách tham số - Thông tin nhận về từ VNPAY (vnp_Command=pay)

Lưu ý

- Merchant/website TMĐT thực hiện kiểm tra sự toàn vẹn của dữ liệu (checksum) trước khi thực hiện các thao tác khác

- Thao tác cập nhật/xử lý kết quả sau khi thanh toán được thực hiện tại URL này
- Đây là URL server - call - server (Máy chủ VNPAY gọi máy chủ Merchant/website TMĐT)
- Merchant trả dữ liệu lại cho VNPAY bằng mã RspCode và Message định dạng JSON:
Trong đó:
RspCode là mã lỗi tình trạng cập nhật trạng thái thanh toán của giao dịch tại đầu IPN của merchant.
Message là mô tả mã lỗi của RspCode
Merchant cần tuân thủ theo các trường hợp kiểm và phản hồi lại RspCode cho VNPAY. Vui lòng tham khảo thêm tại code demo IPN của VNPAY
- Cơ chế retry IPN:
Hệ thống VNPAY căn cứ theo RspCode phản hồi từ merchant để kết thúc luồng hay bật cơ chế retry
RspCode: 00, 02 là mã lỗi IPN của merchant phản hồi đã cập nhật được tình trạng giao dịch. VNPAY kết thúc luồng
RspCode: 01, 04, 97, 99 hoặc IPN timeout là mã lỗi IPN merchant không cập nhật được tình trạng giao dịch. VNPAY bật cơ chế retry IPN
Tổng số lần gọi tối đa: 10 lần
Khoảng cách giữa các lần gọi lại: 5 phút

Cài đặt Code Return URL

Dữ liệu VNPAY trả về bằng cách chuyển hướng trình duyệt web của khách hàng theo địa chỉ web mà Merchant cung cấp khi gửi yêu cầu thanh toán. Trên URL này mang thông tin kết quả thanh toán của khách hàng.

VNPAY trả về kết quả thanh toán URL có dạng:

`https://{domain}/ReturnUrl?vnp_Amount=1000000&vnp_BankCode=NCB&vnp_BankTranNo=VNP14226112&vnp_CardType=ATM&vnp_OrderInfo=Thanh+toan+d on+hang+thoi+gian%3A+2023-12-07+17%3A00%3A44&vnp_PayDate=20231207170112&vnp_ResponseCode=00&vnp_TmnCode=CTTVNP01&vnp_TransactionNo=14226112&vnp_TransactionStatus=00&vnp_TxnRef=166117&vnp_SecureHash=b6dababca5e07a2d8e32fdd3cf05c29cb426c721ae18e9589f7ad0e2db4b657c6e0e5cc8e271cf745162bcb100fdf2f64520554a6f5275bc4c5b5b3e57dc4b4b`

Trong đó `https://{domain}/ReturnUrl` là URL nhận kết quả hệ thống gửi sang VNPAY theo URL thanh toán qua tham số `vnp_ReturnUrl`

Danh sách tham số

Giống với tham số gửi về địa chỉ IPN URL

Lưu ý

- URL này chỉ kiểm tra toàn vẹn dữ liệu (checksum) và hiển thị thông báo tới khách hàng
- Không cập nhật kết quả giao dịch tại địa chỉ này

Bảng mã lỗi của hệ thống thanh toán PAY

Mã lỗi Mô tả

Bảng mã lỗi vnp_TransactionStatus

00	Giao dịch thành công
01	Giao dịch chưa hoàn tất
02	Giao dịch bị lỗi

Mã lỗi Mô tả

04	Giao dịch đảo (Khách hàng đã bị trừ tiền tại Ngân hàng nhưng GD chưa thành công ở VNPAY)
05	VNPAY đang xử lý giao dịch này (GD hoàn tiền)
06	VNPAY đã gửi yêu cầu hoàn tiền sang Ngân hàng (GD hoàn tiền)
07	Giao dịch bị nghi ngờ gian lận
09	GD Hoàn trả bị từ chối

Bảng 3. 3 Bảng mã lỗi của hệ thống thanh toán VNPAY 1

Mã lỗi	Mô tả
vnp_ResponseCode VNPAY phản hồi qua IPN và Return URL:	
00	Giao dịch thành công
07	Trừ tiền thành công. Giao dịch bị nghi ngờ (liên quan tới lừa đảo, giao dịch bất thường).
09	Giao dịch không thành công do: Thẻ/Tài khoản của khách hàng chưa đăng ký dịch vụ InternetBanking tại ngân hàng.
10	Giao dịch không thành công do: Khách hàng xác thực thông tin thẻ/tài khoản không đúng quá 3 lần
11	Giao dịch không thành công do: Đã hết hạn chờ thanh toán. Xin quý khách vui lòng thực hiện lại giao dịch.
12	Giao dịch không thành công do: Thẻ/Tài khoản của khách hàng bị khóa.
13	Giao dịch không thành công do Quý khách nhập sai mật khẩu xác thực giao dịch (OTP). Xin quý khách vui lòng thực hiện lại giao dịch.
24	Giao dịch không thành công do: Khách hàng hủy giao dịch
51	Giao dịch không thành công do: Tài khoản của quý khách không đủ số dư để thực hiện giao dịch.
65	Giao dịch không thành công do: Tài khoản của Quý khách đã vượt quá hạn mức giao dịch trong ngày.
75	Ngân hàng thanh toán đang bảo trì.
79	Giao dịch không thành công do: KH nhập sai mật khẩu thanh toán quá số lần quy định. Xin quý khách vui lòng thực hiện lại giao dịch
99	Các lỗi khác (lỗi còn lại, không có trong danh sách mã lỗi đã liệt kê)

Bảng 3. 4 Bảng mã lỗi của hệ thống thanh toán VNPAY 2

3.2.9 Tích hợp vào hệ thống website thương mại điện tử

- Khai báo VNPay config với các biến tĩnh lưu trữ thông tin cấu hình cần thiết cho việc kết nối và giao tiếp với dịch vụ VNPay
 - vnp_PayUrl: URL để thực hiện thanh toán.
 - vnp_ReturnUrl: URL để trả về sau khi thực hiện thanh toán.

- vnp_TmnCode: Mã Terminal (mã định danh của điểm bán hàng).
- secretKey: Khóa bí mật để mã hóa dữ liệu.
- vnp_ApiUrl: URL API của VNPay để thực hiện các giao dịch.

```
public static String vnp_PayUrl =
"https://sandbox.vnpayment.vn/paymentv2/vpcpay.html";
    public static String vnp_ReturnUrl = "/vnPayPayment";
    public static String vnp_TmnCode = "T9N6VB3M";
    public static String secretKey =
"XQ8B9COYDS8H5H8WVNL8RTWYF73OW57P";
    public static String vnp_ApiUrl =
"https://sandbox.vnpayment.vn/merchant_webapi/api/transaction";
```

khai báo các hàm mã hóa:

```
public static String md5(String message) {
    String digest = null;
    try {
        MessageDigest md = MessageDigest.getInstance("MD5");
        byte[] hash = md.digest(message.getBytes("UTF-8"));
        StringBuilder sb = new StringBuilder(2 * hash.length);
        for (byte b : hash) {
            sb.append(String.format("%02x", b & 0xff));
        }
        digest = sb.toString();
    } catch (UnsupportedEncodingException ex) {
        digest = "";
    } catch (NoSuchAlgorithmException ex) {
        digest = "";
    }
    return digest;
}
```

```
public static String Sha256(String message) {
    String digest = null;
    try {
        MessageDigest md = MessageDigest.getInstance("SHA-256");
        byte[] hash = md.digest(message.getBytes("UTF-8"));
        StringBuilder sb = new StringBuilder(2 * hash.length);
        for (byte b : hash) {
            sb.append(String.format("%02x", b & 0xff));
        }
        digest = sb.toString();
    } catch (UnsupportedEncodingException ex) {
        digest = "";
    } catch (NoSuchAlgorithmException ex) {
        digest = "";
    }
    return digest;
}
```

```

    }
    public static String hmacSHA512(final String key, final String data) {
        try {

            if (key == null || data == null) {
                throw new NullPointerException();
            }
            final Mac hmac512 = Mac.getInstance("HmacSHA512");
            byte[] hmacKeyBytes = key.getBytes();
            final SecretKeySpec secretKey = new SecretKeySpec(hmacKeyBytes,
"HmacSHA512");
            hmac512.init(secretKey);
            byte[] dataBytes = data.getBytes(StandardCharsets.UTF_8);
            byte[] result = hmac512.doFinal(dataBytes);
            StringBuilder sb = new StringBuilder(2 * result.length);
            for (byte b : result) {
                sb.append(String.format("%02x", b & 0xff));
            }
            return sb.toString();

        } catch (Exception ex) {
            return "";
        }
    }
}

```

```

public static String hashAllFields(Map fields) {
    List fieldNames = new ArrayList(fields.keySet());
    Collections.sort(fieldNames);
    StringBuilder sb = new StringBuilder();
    Iterator itr = fieldNames.iterator();
    while (itr.hasNext()) {
        String fieldName = (String) itr.next();
        String fieldValue = (String) fields.get(fieldName);
        if ((fieldValue != null) && (fieldValue.length() > 0)) {
            sb.append(fieldName);
            sb.append("=");
            sb.append(fieldValue);
        }
        if (itr.hasNext()) {
            sb.append("&");
        }
    }
    return hmacSHA512(secretKey,sb.toString());
}

```

createOrder trong Service :

```
@Override
```

```

    public String createOrder(int total, String orderInfor, String urlReturn,String
vnp_TxnRef) {
        String vnp_Version = "2.1.0";
        String vnp_Command = "pay";
        String vnp_IpAddr = "127.0.0.1";
        String vnp_TmnCode = VNPayConfig.vnp_TmnCode;
        String orderType = "order-type";
        Map<String, String> vnp_Params = new HashMap<>();
        vnp_Params.put("vnp_Version", vnp_Version);
        vnp_Params.put("vnp_Command", vnp_Command);
        vnp_Params.put("vnp_TmnCode", vnp_TmnCode);
        vnp_Params.put("vnp_Amount", String.valueOf(total*100));
        vnp_Params.put("vnp_CurrCode", "VND");
        vnp_Params.put("vnp_TxnRef", vnp_TxnRef);
        vnp_Params.put("vnp_OrderInfo", orderInfor);
        vnp_Params.put("vnp_OrderType", orderType);

        String locate = "vn";
        vnp_Params.put("vnp_Locale", locate);

        urlReturn += VNPayConfig.vnp_ReturnUrl;
        vnp_Params.put("vnp_ReturnUrl", urlReturn);
        vnp_Params.put("vnp_IpAddr", vnp_IpAddr);

        Calendar cld = Calendar.getInstance(TimeZone.getTimeZone("Etc/GMT+7"));
        SimpleDateFormat formatter = new
SimpleDateFormat("yyyyMMddHHmmss");
        String vnp_CreateDate = formatter.format(cld.getTime());
        vnp_Params.put("vnp_CreateDate", vnp_CreateDate);

        cld.add(Calendar.MINUTE, 15);
        String vnp_ExpireDate = formatter.format(cld.getTime());
        vnp_Params.put("vnp_ExpireDate", vnp_ExpireDate);

        List fieldNames = new ArrayList(vnp_Params.keySet());
        Collections.sort(fieldNames);
        StringBuilder hashData = new StringBuilder();
        StringBuilder query = new StringBuilder();
        Iterator itr = fieldNames.iterator();
        while (itr.hasNext()) {
            String fieldName = (String) itr.next();
            String fieldValue = (String) vnp_Params.get(fieldName);
            if ((fieldValue != null) && (fieldValue.length() > 0)) {
                hashData.append(fieldName);
                hashData.append('=');
                try {
                    hashData.append(URLEncoder.encode(fieldValue,
StandardCharsets.US_ASCII.toString()));

```

```

        query.append(URLEncoder.encode(fieldName,
StandardCharsets.US_ASCII.toString()));
        query.append('=');
        query.append(URLEncoder.encode(fieldValue,
StandardCharsets.US_ASCII.toString()));
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    if (itr.hasNext()) {
        query.append('&');
        hashData.append('&');
    }
}
String queryUrl = query.toString();
String vnp_SecureHash =
VNPayConfig.hmacSHA512(VNPayConfig.secretKey, hashData.toString());
vnp_Params.put("vnp_SecureHash", vnp_SecureHash);
queryUrl += "&vnp_SecureHash=" + vnp_SecureHash;
String paymentUrl = VNPayConfig.vnp_PayUrl + "?" + queryUrl;
return paymentUrl;
}

```

Order return:

```

@Override
public int orderReturn(HttpServletRequest request){
    Map fields = new HashMap();
    for (Enumeration params = request.getParameterNames());
params.hasMoreElements();) {
    String fieldName = null;
    String fieldValue = null;
    try {
        fieldName = URLEncoder.encode((String) params.nextElement(),
StandardCharsets.US_ASCII.toString());
        fieldValue = URLEncoder.encode(request.getParameter(fieldName),
StandardCharsets.US_ASCII.toString());
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    if ((fieldValue != null) && (fieldValue.length() > 0)) {
        fields.put(fieldName, fieldValue);
    }
}

String vnp_SecureHash = request.getParameter("vnp_SecureHash");
if (fields.containsKey("vnp_SecureHashType")) {
    fields.remove("vnp_SecureHashType");
}
if (fields.containsKey("vnp_SecureHash")) {

```

```

        fields.remove("vnp_SecureHash");
    }
    String signValue = VNPayConfig.hashAllFields(fields);
    if (signValue.equals(vnp_SecureHash)) {
        if ("00".equals(request.getParameter("vnp_TransactionStatus"))) {
            return 1;
        } else {
            return 0;
        }
    } else {
        return -1;
    }
}

```

Trong Controller checkout:

```

else if(paymentMethod.equals("vnpay")) {
    bookings.setQrCode(orderId);
    System.out.println(orderId);
    bookingService.save(bookings);
    String baseUrl = request.getScheme() + "://" + request.getServerName() + ":" +
"4000";
    String vnpayUrl = vnPayService.createOrder((int)
bookings.getSchedule().getPrice()*bookings.getBookingDetails().size(), "Thanh toán EVI
BUS.", baseUrl,orderId);
    Map<String, String> responseMap = new HashMap<>();
    System.out.println(vnpayUrl);
    responseMap.put("code", vnpayUrl);
    return ResponseEntity.ok(responseMap);
}

```

Xử lý IPN từ VNPay trả về:

```

@GetMapping("/IPN")
@ResponseBody
public ResponseEntity<String> PayWithVNPayIPN(HttpServletRequest request,Model model) {
    int paymentStatus = vnPayService.orderReturn(request);
    String vnp_Amount = request.getParameter("vnp_Amount");
    String vnp_BankCode = request.getParameter("vnp_BankCode");
    String vnp_BankTranNo = request.getParameter("vnp_BankTranNo");
    String vnp_CardType = request.getParameter("vnp_CardType");
    String vnp_OrderInfo = request.getParameter("vnp_OrderInfo");
    String vnp_PayDate = request.getParameter("vnp_PayDate");
    String vnp_ResponseCode = request.getParameter("vnp_ResponseCode");
    String vnp_TmnCode = request.getParameter("vnp_TmnCode");
    String vnp_TransactionNo = request.getParameter("vnp_TransactionNo");
    String vnp_TransactionStatus = request.getParameter("vnp_TransactionStatus");
    String vnp_TxnRef = request.getParameter("vnp_TxnRef");
    String vnp_SecureHash = request.getParameter("vnp_SecureHash");
    JSONObject response = new JSONObject();
}

```



```

System.out.println(request);
boolean checkOrderId = false;
boolean checkAmount = false;
boolean checkOrderStatus = false;
if(!vnp_ResponseCode.equals("00")) {
    System.out.println("Thanh toán thất bại");
    response.put("RspCode", "00");
    response.put("Message", "Confirm Success");
    return ResponseEntity.ok()
        .contentType(MediaType.APPLICATION_JSON)
        .body(response.toString());
} else {
    System.out.println(vnp_TxnRef);
    System.out.println("Thanh toán thành công"+request);
    Bookings bookings = bookingService.findByQrCode(vnp_TxnRef);
    String ticketCode = generateTicketCode(bookings.getBookingId());
    bookings.setStatus(Bookings.STATUS_PENDING);
    bookings.setPaymentMethod("VNPay");
    bookings.setPaid(true);
    bookings.setQrCode(ticketCode);
    try {
        bookings = bookingService.save(bookings);
        if (bookings != null ){
            checkOrderId = true;
        }
        int expectedAmount = (int) Math.round(bookings.getSchedule().getPrice() *
bookings.getBookingDetails().size());
        int vnpAmountInSmallestUnit = Integer.parseInt(vnp_Amount) / 100;

        if (expectedAmount == vnpAmountInSmallestUnit) {
            checkAmount = true;
        }
        if(bookings.getStatus().equals(Bookings.STATUS_PENDING)) {
            checkOrderStatus= true;
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    try {
        sendEmailWithQRCode(ticketCode,bookings);
    } catch (MessagingException | IOException | WriterException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    try {
        Map<String, String> fields = new HashMap<>();
        Enumeration<String> params = request.getParameterNames();

```



```

while (params.hasMoreElements()) {
    String fieldName = params.nextElement();
    String fieldValue = request.getParameter(fieldName);
    if (fieldValue != null && !fieldValue.isEmpty()) {
        fields.put(URLEncoder.encode(fieldName,
StandardCharsets.US_ASCII.toString()), URLEncoder.encode(fieldValue,
StandardCharsets.US_ASCII.toString()));
    }
}
fields.remove("vnp_SecureHashType");
fields.remove("vnp_SecureHash");
System.out.println(checkOrderId + " "+checkAmount+" "+checkOrderStatus);
// Check checksum
String signValue = VNPayConfig.hashAllFields(fields); // Implement
Config.hashAllFields(fields)
if (signValue.equals(vnp_SecureHash)) {

    if (checkOrderId) {
        if (checkAmount) {
            if (checkOrderStatus) {
                if ("00".equals(request.getParameter("vnp_ResponseCode"))) {
                    response.put("RspCode", "00");
                    response.put("Message", "Confirm Success");
                } else {
                    response.put("RspCode", "00");
                    response.put("Message", "Confirm Success");
                }
            } else {
                response.put("RspCode", "02");
                response.put("Message", "Order already confirmed");
            }
        } else {
            response.put("RspCode", "04");
            response.put("Message", "Invalid Amount");
        }
    } else {
        response.put("RspCode", "01");
        response.put("Message", "Order not Found");
    }
} else {
    response.put("RspCode", "97");
    response.put("Message", "Invalid Checksum");
}
} catch (Exception e) {
    response.put("RspCode", "99");
    response.put("Message", "Unknown error");
}

```

```

return ResponseEntity.ok()
    .contentType(MediaType.APPLICATION_JSON)
    .body(response.toString());
}

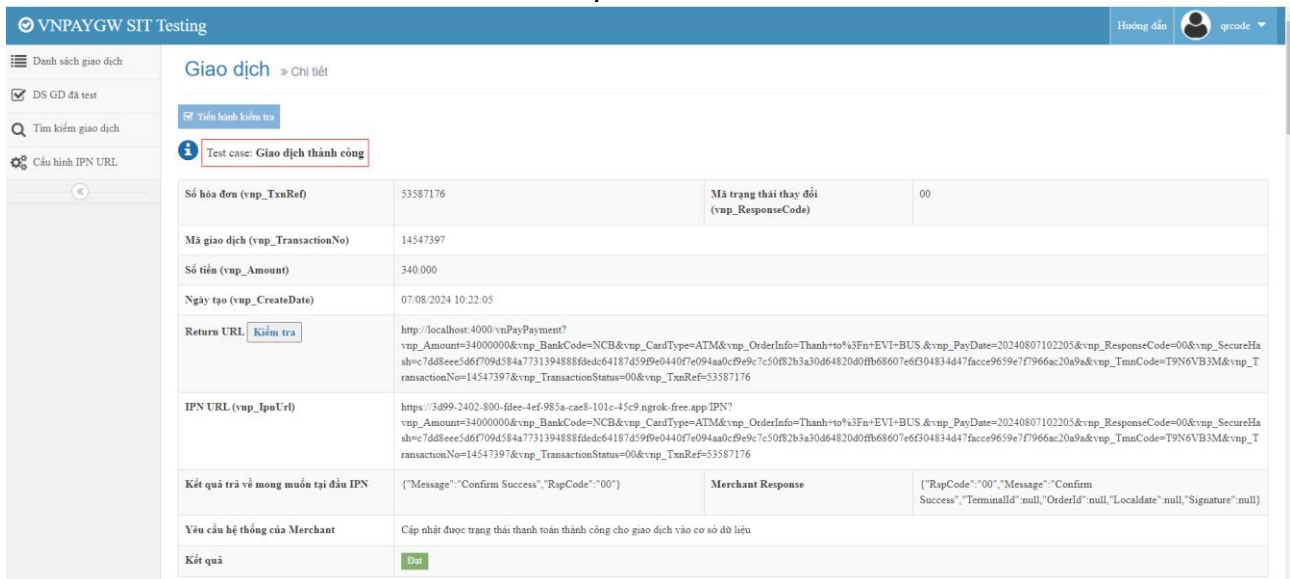
```

- Cấu hình IPN URL để VNPay trả về:

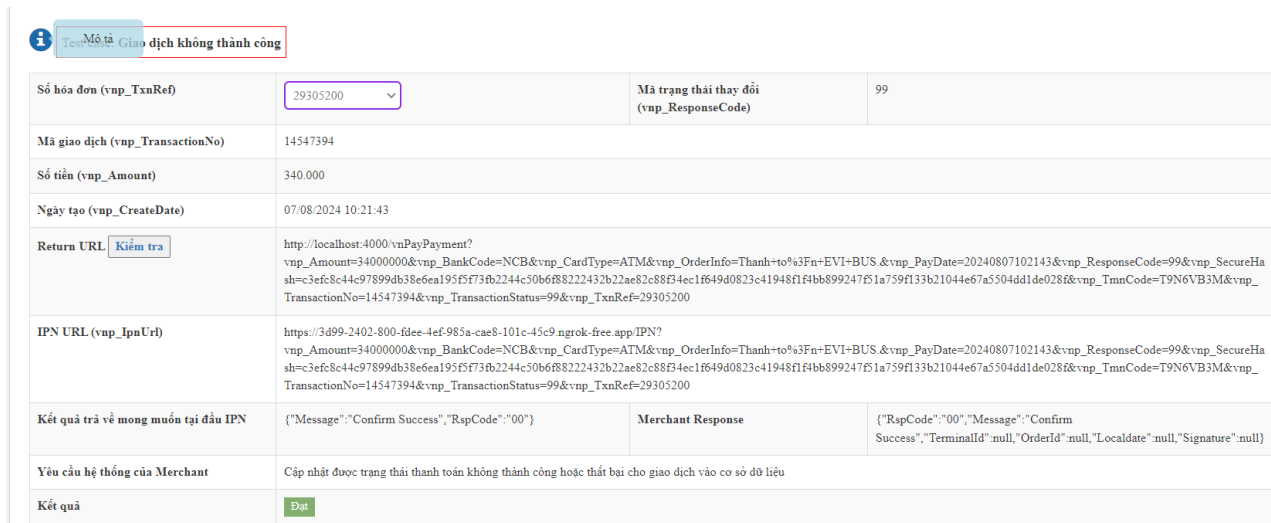


Hình 3. 2 Cấu Hình IPN VNPay trả về

Lưu ý: URL IPN cần public ra internet có ssl và không có port lạ server VNPAY mới có thể call api

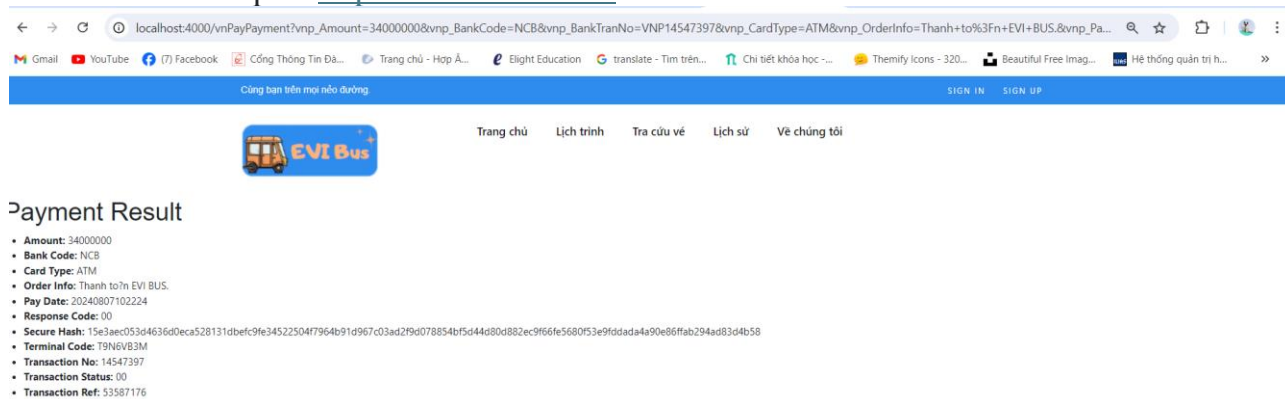


Hình 3. 3 Trang sit testing do VNPay hỗ trợ 1

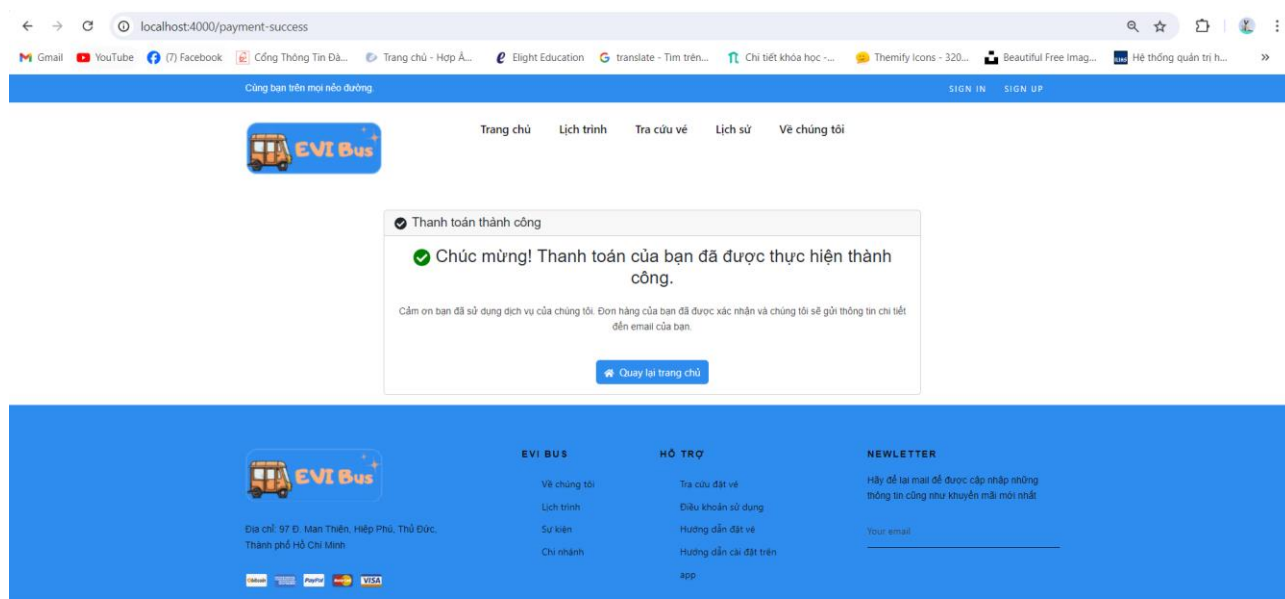


Hình 3. 4 Trang sit testing do VNPay hỗ trợ 2

Return Url trả về port <http://localhost:4000/>



Hình 3. 5 Trang Return Url VNPay trả về với các tham số



Hình 3. 6 Trang thanh toán thành công của website

3.3 MoMo

MoMo là ví điện tử hàng đầu tại Việt Nam, cung cấp dịch vụ thanh toán QR code rộng rãi.

3.3.1 Quy trình công nghệ

MoMo sử dụng mã QR động cho mỗi giao dịch, tạo ra từ ứng dụng di động để đảm bảo tính bảo mật và chính xác.

3.3.2 Bảo mật

MoMo áp dụng nhiều biện pháp bảo mật tiên tiến:

- Mã hóa end-to-end:
- Sử dụng các thuật toán mã hóa mạnh để bảo vệ dữ liệu người dùng.
 - Xác thực đa yếu tố:
- Yêu cầu mật khẩu, mã PIN, và OTP cho các giao dịch quan trọng.
- Tích hợp xác thực sinh trắc học như vân tay hoặc nhận diện khuôn mặt.
 - Tokenization:
- Thay thế thông tin nhạy cảm bằng các token để tăng cường bảo mật.
 - Giám sát giao dịch:

- Sử dụng AI để phát hiện giao dịch bất thường và ngăn chặn gian lận.
 - Chứng nhận bảo mật:
- Đạt chứng nhận PCI DSS level 1, đảm bảo tuân thủ các tiêu chuẩn bảo mật quốc tế.
 - Bảo hiểm giao dịch:
- Cung cấp bảo hiểm cho các giao dịch, bảo vệ người dùng khỏi các rủi ro tài chính.

3.3.3 Tiện ích người dùng

- Thanh toán hóa đơn:
- Hóa đơn điện, nước, internet, truyền hình cáp.
- Thanh toán khoản vay, bảo hiểm.
 - Chuyển tiền:
- Chuyển tiền nhanh chóng giữa các tài khoản MoMo.
- Chuyển tiền đến tài khoản ngân hàng.
- Rút tiền tại ATM không cần thẻ.
 - Mua sắm và giải trí:
- Mua sắm trực tuyến trên các sàn thương mại điện tử.
- Đặt vé xem phim, vé máy bay, vé xe khách.
- Đặt đồ ăn và dịch vụ giao hàng.
 - Dịch vụ tài chính:
- Tiết kiệm online với lãi suất hấp dẫn.
- Mua bảo hiểm.
- Đầu tư vào quỹ mở.
 - Tiện ích khác:
- Nạp tiền điện thoại và mua mã thẻ game.
- Quyên góp từ thiện.
- Tích điểm và đổi ưu đãi.
 - MoMo Pay Later:
- Dịch vụ mua trước trả sau với hạn mức tín dụng.
 - Quản lý tài chính:
- Theo dõi chi tiêu và lập báo cáo.
- Thiết lập mục tiêu tiết kiệm.

3.3.4 Mô hình kết nối

Luồng xử lý

Thanh toán thường dùng bằng ví MoMo dành cho đối tác sử dụng các kênh bán hàng sau:

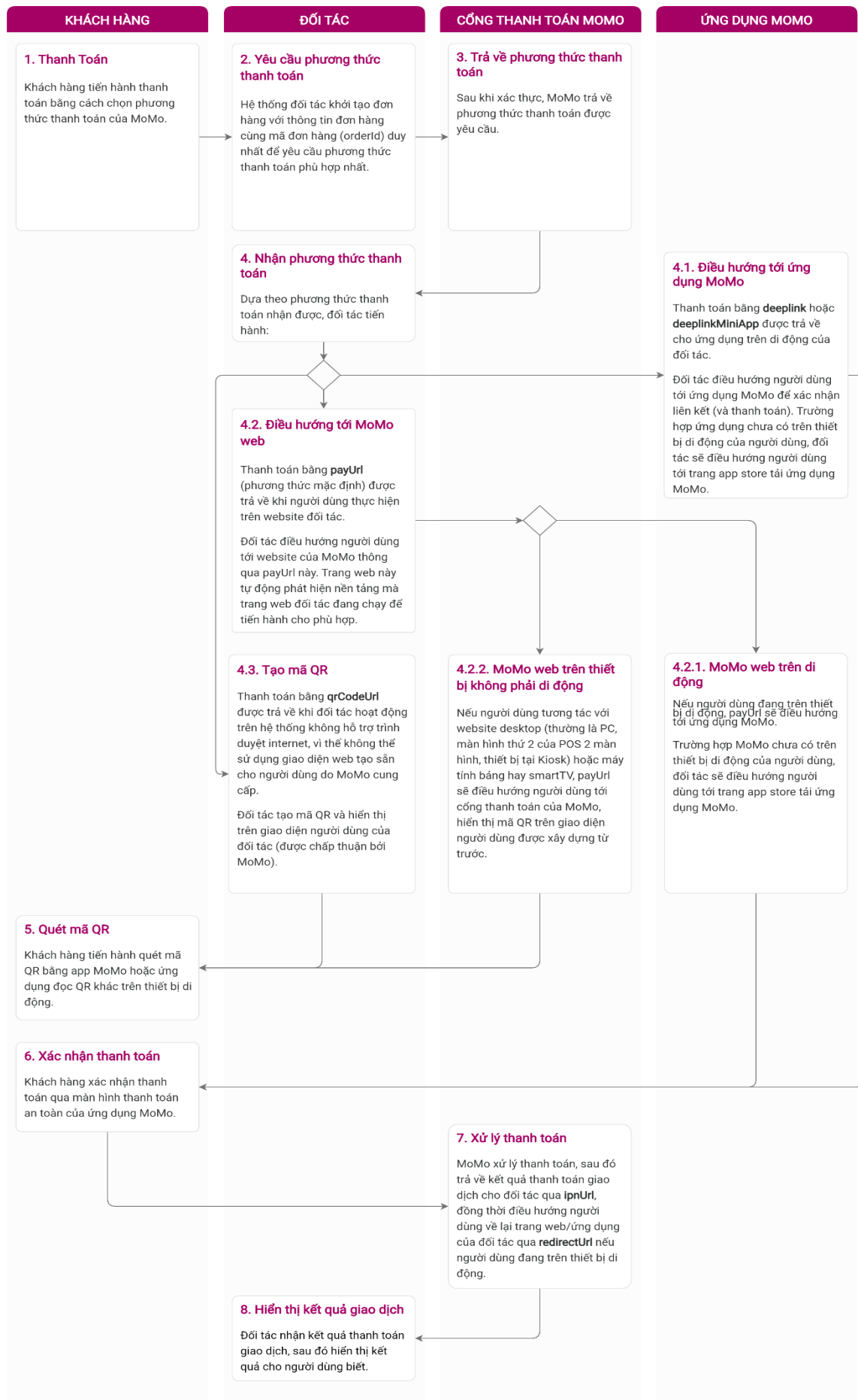
- Website trên máy tính
- Website trên di động
- Ứng dụng di động
- Thiết bị không hỗ trợ trình duyệt internet

Thanh toán thường dùng bằng ứng dụng Ngân hàng dành cho đối tác sử dụng các kênh bán hàng sau:

- Website trên máy tính
- Thiết bị không hỗ trợ trình duyệt internet

Hướng dẫn bên dưới cho bạn biết cách tích hợp phù hợp với từng nền tảng và cách hoạt động của API.

Luồng xử lý:



Sơ đồ 3. 3 Luồng xử lý của MoMo

3.3.5 Thông tin cấu hình

Danh sách tham số HTTP Request

Attribute	Type	Required	Description
partnerCode	String(50)		Thông tin tích hợp
subPartnerCode	String(50)		Định danh duy nhất của tài khoản M4B của bạn
storeName	String		Tên đối tác
storeId	String(50)		Mã cửa hàng
requestId	String(50)		Định danh duy nhất cho mỗi yêu cầu Đối tác sử dụng requestId để xử lý idempotency
amount	Long		Số tiền cần thanh toán Nhỏ Nhất: 1.000 VND Tối đa: 50.000.000 VND Tiền tệ: VND Kiểu dữ liệu: Long
orderId	String(200)		Mã đơn hàng của đối tác
orderInfo	String(255)		Thông tin đơn hàng
orderGroupId	Long		orderGroupId được MoMo cung cấp để phân nhóm đơn hàng cho các hoạt động vận hành sau này. Vui lòng liên hệ với MoMo để biết chi tiết cách sử dụng
redirectUrl	String		Một URL của đối tác. URL này được sử dụng để chuyển trang (redirect) từ MoMo về trang mua hàng của đối tác sau khi khách hàng thanh toán. Hỗ trợ: AppLink and WebLink
ipnUrl	String		API của đối tác. Được MoMo sử dụng để gửi kết quả thanh toán theo phương thức <u>IPN</u> (server-to-server)
requestType	String		captureWallet
extraData	String(1000)		Giá trị mặc định là rỗng"" Encode base64 theo định dạng Json: {"key": "value"} Ví dụ với dữ liệu: {"username": "momo", "skus": "value1,value2"} thì data extraData: eyJ1c2VybmFtZSI6Im1vbW8iLCJza3VzIjoidmFsdWUxLHZhbHVlMiJ9
items	List		Danh sách các sản phẩm hiển thị trên trang thanh toán. Tối đa: 50 loại sản phẩm
deliveryInfo	Object		Thông tin giao hàng của đơn hàng
userInfo	Object		Thông tin người dùng
referenceId	String(255)		Mã tham chiếu phụ của đối tác Ví dụ dùng trong các trường hợp như mã khách hàng, mã hộ gia đình, mã hóa đơn, mã thuê bao v.v

autoCapture	Boolean		Nếu giá trị false, giao dịch sẽ không tự động capture. Mặc định là true
lang	String		Ngôn ngữ của message được trả về (vi hoặc en)
signature	String		Chữ ký để xác nhận giao dịch. Sử dụng thuật toán Hmac_SHA256 với data theo định dạng được sort từ a-z: accessKey=\$accessKey&amount=\$amount&extraData=\$extraData &ipnUrl=\$ipnUrl&orderId=\$orderId&orderInfo=\$orderInfo &partnerCode=\$partnerCode&redirectUrl=\$redirectUrl &requestId=\$requestId&requestType=\$requestType

Chi tiết nội dung của items

Attribute	Type	Required	Description
id	String		SKU number
name	String		Tên sản phẩm
description	String		Miêu tả sản phẩm
category	String		Phân loại ngành hàng của sản phẩm
imageUrl	String		Link hình ảnh của sản phẩm
manufacturer	String		Tên nhà sản xuất
price	Long		Đơn giá
currency	String		VND
quantity	Integer		Số lượng của sản phẩm. Cần là một số lớn hơn 0
unit	String		Đơn vị đo lường của sản phẩm này
totalPrice	Long		Tổng giá = Đơn giá x Số lượng
taxAmount	Long		Tổng thuế

Mẫu Item:

```
{
  "id": "204727",
  "name": "YOMOST Bac Ha&Viet Quat 170ml",
  "description": "YOMOST Sua Chua Uong Bac Ha&Viet Quat 170ml/1 Hop",
  "category": "beverage",
  "imageUrl": "https://momo.vn/uploads/product1.jpg",
  "manufacturer": "Vinamilk",
  "price": 11000,
  "quantity": 5,
  "unit": "hộp",
  "totalPrice": 55000,
  "taxAmount": "200"
}
```

Chi tiết nội dung của deliveryInfo

Attribute	Type	Required	Description
deliveryAddress	String		Địa chỉ giao hàng
deliveryFee	String		Phí giao hàng
quantity	String		Số lượng sản phẩm

Mẫu deliveryInfo

```
{
  "deliveryAddress": "Phu My Hung Tower",
  "deliveryFee": "30000",
  "quantity": "2"
}
```

Chi tiết nội dung của userInfo

Attribute	Type	Required	Description
name	String		Tên của người dùng
phoneNumber	String		Số điện thoại của người dùng
email	String		Email của người dùng

Mẫu userInfo

```
{
  "name": "Nguyen Van A",
  "phoneNumber": "0999888999",
  "email": "email_add@domain.com"
}
```

Danh sách tham số HTTP Response

Attribute	Type	Required	Description
partnerCode	String		<u>Thông tin tích hợp</u>
requestId	String		Giống với yêu cầu ban đầu
orderId	String		<u>Mã đơn hàng của đối tác</u>
amount	Long		Giống với số tiền yêu cầu ban đầu
responseTime	Long		Thời gian trả kết quả thanh toán về đối tác Định dạng: timestamp
message	String		Mô tả lỗi, ngôn ngữ dựa trên lang
resultCode	int		<u>Result Code</u>
payUrl	String		URL để chuyển từ trang mua hàng của đối tác sang trang thanh toán của MoMo
deeplink	String		URL để mở ứng dụng trực tiếp MoMo (Khách hàng phải cài đặt ứng dụng MoMo trước) và trang xác nhận thanh toán. Nếu bạn gặp sự cố khi mở deeplink ở android 11, vui lòng xem hướng dẫn
qrCodeUrl	String		Dữ liệu để tạo mã QR nếu bạn muốn khách hàng quét mã QR trực tiếp trên trang mua hàng hoặc in mã lên hoá đơn.

			Note: Đây không phải URL chứa hình ảnh của mã QR, bạn cần sử dụng thư viện ngoài để tạo mã QR.
deeplinkMiniApp	String		URL mở màn hình xác nhận thanh toán của ứng dụng MoMo. Áp dụng khi đối tác sử dụng mini app nhúng vào trong ứng dụng MoMo.
signature	String		Chữ ký (/docs/payment/api/other/signature) để xác nhận giao dịch. Sử dụng thuật toán Hmac_SHA256 theo định dạng : accessKey=\$accessKey&amount=\$amount&orderId=\$orderId &partnerCode=\$partnerCode&payUrl= &payUrl&requestId= \$requestId&responseTime=\$responseTime&resultCode=\$resultCode
userFee	Long		User Fee

Mẫu Response

```
{
  "partnerCode": "MOMOT5BZ20231213_TEST",
  "orderId": "Partner_Transaction_ID_1721720620078",
  "requestId": "Request_ID_1721720620078",
  "amount": 1000,
  "responseTime": 1721720619912,
  "message": "Successful.",
  "resultCode": 0,
  "payUrl": "https://test-payment.momo.vn/v2/gateway/pay?t=TU9NT1Q1QloyMDIzMTIxM19URVNUfFBhcnRuZXJfVHJhbnNhY3Rpb25fSURfMTcyMTcyMDYyMDA3OA&s=6c14385cd4355e0abe0e0563a2da20705bceca9fac79746b2bf6a4c380374b44",
  "deeplink": "momo://app?action=payWithApp&isScanQR=false&serviceType=app&sid=TU9NT1Q1QloyMDIzMTIxM19URVNUfFBhcnRuZXJfVHJhbnNhY3Rpb25fSURfMTcyMTcyMDYyMDA3OA&v=3.0",
  "qrCodeUrl": "00020101021226110007vn.momo38260010A0000007270208QRIBFTTA5303704540410005802VN62480515MMTCfKWQmuH5nQR0825Thank you for your purcha6304B293"
}
```

Thanh toán

Bạn cần cài đặt ứng dụng MoMo Test và sử dụng MoMo TEST Account để thực hiện giao dịch.

Hướng dẫn thanh toán: Đăng nhập ứng dụng MoMo > Màn hình chính > Quét mã.

Xử lý kết quả thanh toán

Khi việc xử lý thanh toán hoàn tất, MoMo sẽ thông báo cho đối tác ngay cả khi giao dịch này có thành công hay không.

Giao diện (Redirect)

Sau khi luồng thanh toán hoàn tất, khách hàng được điều hướng đến redirectUrl mà bên đối tác đã cung cấp trong create request. Một vài thông số sẽ được thêm vào URL theo dạng sau:

Method: **GET**

redirectUrl ?{parameters}

IPN - Instant Payment Notification

Hệ thống của MoMo sử dụng API được khai báo trong ipnUrl để gửi HTTP request với cấu hình bên dưới đến hệ thống đối tác.

Method: **POST**

Attribute	Value	Description
URL	ipnUrl	URL
Method	POST	Phương thức của HTTP request
Headers	Content-type: application/json	HTTP Headers
Payload	Result Transaction	Nội dung của HTTP body

Bảng 3. 5 Tham số Instant Payment Notification

Mẫu Request IPN

```
curl --location 'https://example.com/momo_ip' \
--header 'Content-Type: application/json' \
--data '{
  "orderType": "momo_wallet",
  "amount": 1000,
  "partnerCode": "MOMOT5BZ20231213_TEST",
  "orderId": "Partner_Transaction_ID_1721720620078",
  "extraData": "eyJza3VzIjoiIn0=",
  "signature":
"7b9f4ca728076c32f16041cbc917ebf5e6e7359f0bde343dde3add69a518cf0d",
  "transId": 4088878653,
  "responseTime": 1721720663942,
  "resultCode": 0,
  "message": "Successful.",
  "payType": "qr",
  "requestId": "Request_ID_1721720620078",
  "orderInfo": "Thank you for your purchase at MoMo_test"
}'
```

Reponse của đối tác sẽ đến MoMo sau khi MoMo gửi HTTP request đến địa chỉ ipnUrl.

Bên đối tác cần phản hồi với HTTP code 204 (không cần gửi thêm nội dung)!

Lưu ý: Đối tác cần phản hồi lại trong vòng 15 giây.

Đối tác cần kiểm tra tính hợp lệ của chữ ký trong IPN để đảm bảo kết quả của giao dịch. PartnerCode, OrderId & Amount,... trong notification phải khớp với PartnerCode, OrderId & Amount,... đối tác lưu trong database bên đó!

Field resultCode và message do đối tác xử lý. Tham chiếu đến result code mà MoMo trả về. MoMo sẽ dùng những thông tin này để phản hồi lại khách hàng nếu có bất cứ lỗi nào xảy ra trong quá trình xử lý thanh toán bên nhà đối tác.

Trạng Thái Giao Dịch Sử dụng field resultCode để xác định trạng thái của giao dịch:

- resultCode = 0: giao dịch thành công.

- resultCode = 9000: giao dịch được cấp quyền (authorization) thành công .
- resultCode <> 0: giao dịch thất bại.

Tham khảo [Result code](#) để xác định chi tiết lỗi của giao dịch.

3.3.6 Tích hợp vào hệ thống website thương mại điện tử

Cài đặt thư viện trong java spring boot

```
<dependency>
    <groupId>io.github.momo-wallet</groupId>
    <artifactId>momopayment</artifactId>
    <version>1.0</version>
</dependency>
```

```
public class Constant {
    public static final String accessKey = "M8brj9K6E22vXoDB";
    public static final String serectkey =
"nqQiVSgDMY809JoPF6OzP5OdBUB550Y4";
    public static final String orderType = "momo_wallet";
    public static final String requestType="captureWallet";
    public static final String payType="webApp";
    public static final String redirectUrl="";
    public static final String ipnUrl="https://d0aa-2401-d800-b01-99de-ecb6-41e-abaf-
c49d.ngrok-free.app/momo_ipn";
    public static final String Url="https://test-
payment.momo.vn/v2/gateway/api/create";
    public static final String IDMOMO="MOMO5RGX20191128";
}
```

```
public class Decode {

    public static String encode(String key, String data) {
        try {
            byte[] keySect = "nqQiVSgDMY809JoPF6OzP5OdBUB550Y4".getBytes();
            HmacUtils hm256 = new HmacUtils(HmacAlgorithms.HMAC_SHA_256,
keySect);
            //hm256 object can be used again and again
            String hmac = hm256.hmacHex(data);
            return hmac;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }
}
```

```

public class MomoModel {
    public String partnerCode;
    public String orderId ;
    public String storeId ;
    public String redirectUrl ;
    public String ipnUrl ;
    public String requestId ;
    public String requestType ;
    public String amount;
    public String partnerUserId;
    public String orderType;
    public String transId;
    public String resultCode;
    public String message;
    public String payType;
    public String responseTime;
    public String orderInfo;
    public String extraData;
    public String signature;
}

```

```

public class ResultMoMo {
    public String t;
    public String partnerCode;
    public String requestId;
    public String deeplink;
    public String deeplinkMiniApp;
    public String orderId;
    public String amount;
    public String responseTime;
    public String message;
    public String resultCode;
    public String payUrl;
    public String qrCodeUrl;
}

```

```

    if (paymentMethod.equals("momo")) {
        bookings.setQrCode(orderId);
        bookingService.save(bookings);
        MomoModel jsonRequest = new MomoModel();
        jsonRequest.setPartnerCode(Constant.IDMOMO);
        jsonRequest.setOrderId(orderId);
        jsonRequest.setStoreId(orderId);
        jsonRequest.setRedirectUrl(Constant.redirectUrl);
        jsonRequest.setIpnUrl(Constant.ipnUrl);
        double price =
        bookings.getSchedule().getPrice()*bookings.getBookingDetails().size();
    }

```

```

int intPrice = (int) price;
String amountString = String.valueOf(intPrice);
jsonRequest.setAmount(amountString);
jsonRequest.setOrderInfo("Thanh toán EVI BUS.");
jsonRequest.setRequestId(orderId);
jsonRequest.setOrderType(Constant.orderType);
jsonRequest.setRequestType(Constant.requestType);
jsonRequest.setTransId("1");
jsonRequest.setResultCode("200");
jsonRequest.setMessage("");
jsonRequest.setPayType(Constant.payType);
jsonRequest.setResponseTime("300000");
jsonRequest.setExtraData("");

String decode = "accessKey=" + Constant.accessKey + "&amount=" +
jsonRequest.getAmount() + "&extraData=" +
    jsonRequest.getExtraData() + "&ipnUrl=" + Constant.ipnUrl + "&orderId=" +
orderId + "&orderInfo=" +
    jsonRequest.getOrderInfo() + "&partnerCode=" + jsonRequest.getPartnerCode()
+ "&redirectUrl=" +
    Constant.redirectUrl + "&requestId=" + jsonRequest.getRequestId() +
"&requestType=" +
    Constant.requestType;
String signature = Decode.encode(Constant.sereckey, decode);
jsonRequest.setSignature(signature);

// Convert the request to JSON
ObjectMapper mapper = new ObjectMapper();
String json;
try {
    json = mapper.writeValueAsString(jsonRequest);
} catch (Exception e) {
    e.printStackTrace();
    return ResponseEntity.status(500).body(Map.of("error", "Failed to create payment
request JSON"));
}

HttpClient client = HttpClient.newHttpClient();
ResultMoMo res;
try {
    HttpRequest requestMomo = HttpRequest.newBuilder()
        .uri(new URI(Constant.Url))
        .POST(HttpRequest.BodyPublishers.ofString(json))
        .headers("Content-Type", "application/json")
        .build();
    HttpResponse<String> response = client.send(requestMomo,
        HttpResponse.BodyHandlers.ofString());
    res = mapper.readValue(response.body(), ResultMoMo.class);
}

```

```

    } catch (Exception e) {
        e.printStackTrace();
        return ResponseEntity.status(500).body(Map.of("error", "Failed to communicate
with Momo"));
    }
    if (res == null || res.getPayUrl() == null) {
        return ResponseEntity.status(500).body(Map.of("error", "Payment failed"));
    }
    Map<String, String> responseMap = new HashMap<>();
    responseMap.put("code", res.getDeeplink());
    return ResponseEntity.ok(responseMap);
}

```

```

@RestController
@RequestMapping("/momo_ipn")
public class MoMoIpnController {

    @Autowired
    MailService mailService;
    @Autowired
    private BookingService bookingService;
    @PostMapping()
    public ResponseEntity<String> handleMoMoIpn(@RequestBody MoMoIPNRequest
request) {
        if (request.getResultCode() == 0) {
            System.out.println("Thanh toán thành công"+request);
            Bookings bookings = bookingService.findByQrCode(request.getOrderId());
            String ticketCode = generateTicketCode(bookings.getBookingId());
            bookings.setStatus(Bookings.STATUS_PENDING);
            bookings.setPaymentMethod("momo");
            bookings.setPaid(true);
            bookings.setQrCode(ticketCode);
            bookings = bookingService.save(bookings);
            try {
                sendEmailWithQRCode(ticketCode, bookings);
            } catch (MessagingException | IOException | WriterException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            return ResponseEntity.ok("Payment processed successfully");
        } else {
            System.out.println("Thanh toán thất bại");
            return
ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body("Payment
processing failed");
        }
    }
}

```

Tại sao sử dụng IPN

Instant Payment Notification (IPN) được sử dụng để thông báo (notify) kết quả giao dịch ngay lập tức đến PSP:

- Thanh Toán: Web, SmartTv.
- Thanh Toán Trực Tuyến
- Thanh Toán Định Kỳ

Đối tác có thể sử dụng hệ thống của mình để xử lý thông tin nhận được từ PSP:

- Cập nhật trạng thái giao dịch (khuyến dùng)
- Cập nhật số dư tài khoản (nợ tiền), xuất sản phẩm, thông báo kết quả giao dịch đến ứng dụng di động,...
- Gửi hóa đơn điện tử

Sử dụng IPN để khắc phục trường hợp thanh toán thành công nhưng người dùng không nhận được sản phẩm.

Lý do:

- Người dùng đóng trình duyệt
- Không thể điều hướng đến trang ban đầu của đối tác: do đường truyền, hệ thống quá tải,...

Cách thực thi IPN


Đối tác cần tạo API phía backend để 'lắng nghe' kết quả từ MoMo và cung cấp URL của API đó trong field ipnUrl.

Sau khi người dùng sử dụng thanh toán trên MoMo (web hoặc app), MoMo sẽ thông báo kết quả giao dịch ngay lập tức đến URL này. API của đối tác nhận kết quả thanh toán và tiếp tục xử lý trong hệ thống bên họ.

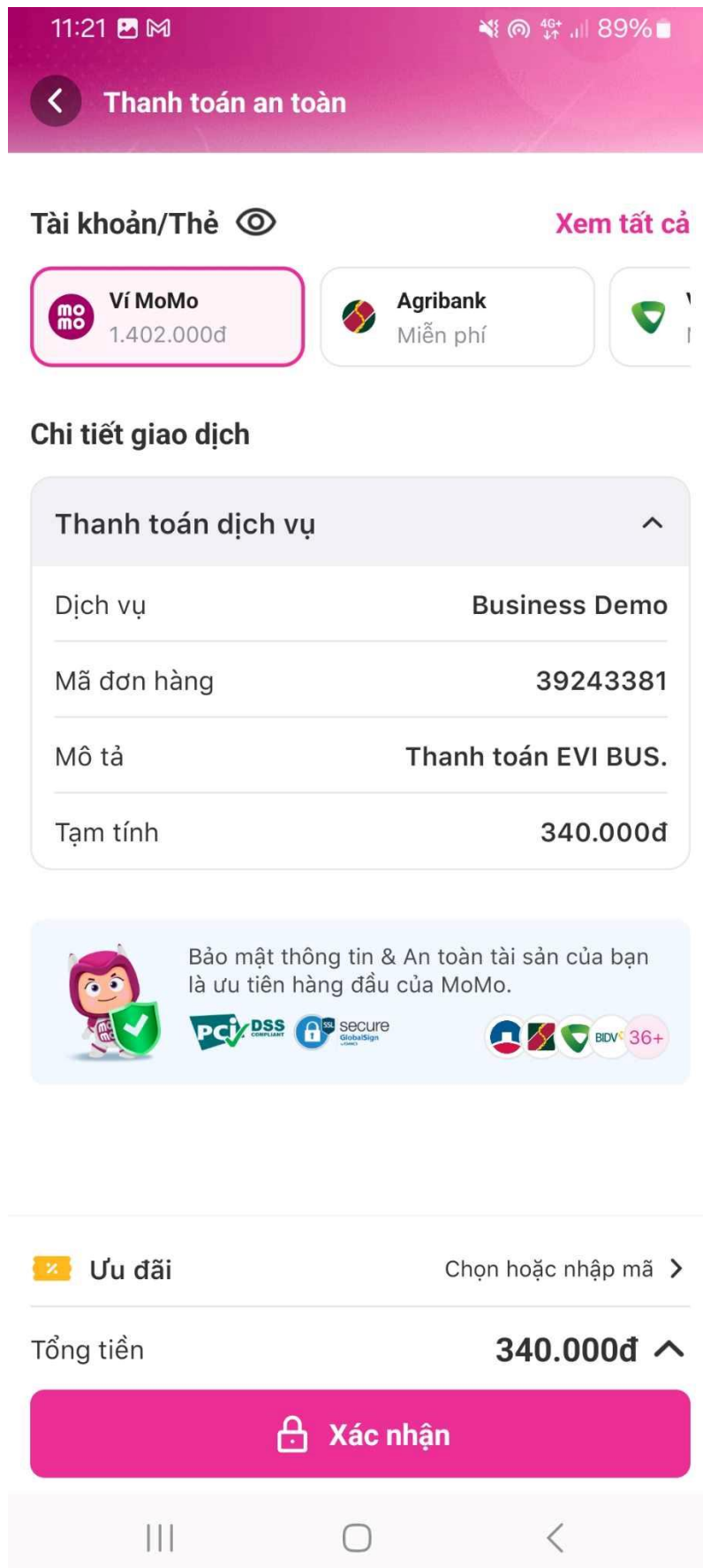
Xây dựng API

- HTTP Produces Header: Sử dụng để MoMo gửi HTTP Request Content-Type: application/json
- HTTP Consumers Header: Đối tác phản hồi lại MoMo: HTTP code 204 (không cần gửi thêm nội dung)
- Endpoint: URL chỉ nên bao gồm đường dẫn, tham số và không chứa ký tự đặc biệt, unicode hoặc khoảng cách.

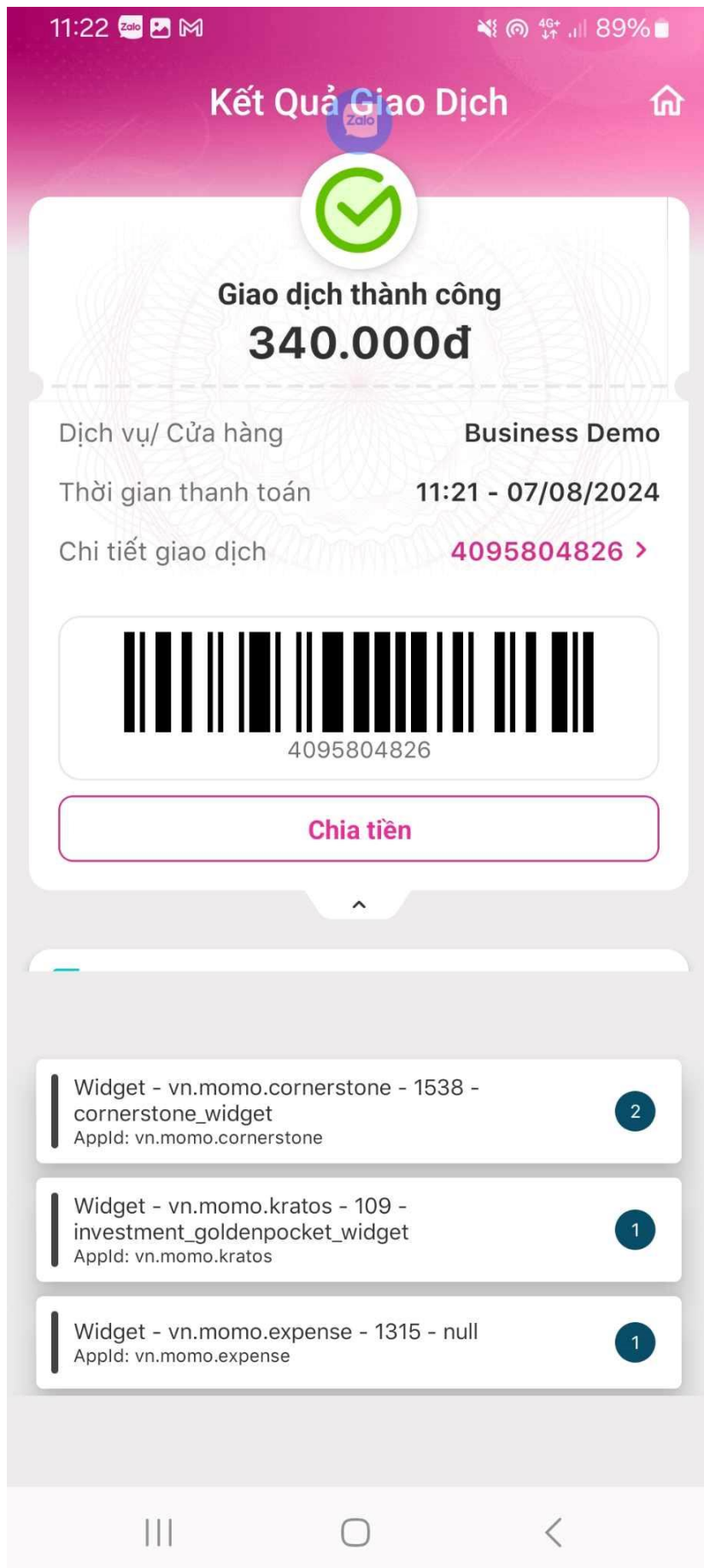
Kết quả

<div>Thông tin lượt đi</div> <div><p>Tuyến xe: Bến xe Quảng Ngãi → Bến xe An Sương</p><p>Ngày đi: 15/8/2024</p><p>Giờ đi: 08:08</p><p>Điểm đón: Bến xe Quảng Ngãi</p><p>Điểm trả: BX Quy Nhơn</p><p>Số lượng ghế: 1</p><p>Bạn đã chọn các ghế: a04</p><p>Chi tiết giá</p><p>Giá vé lượt đi: 340.000 đ Phí thanh toán: 0 đ</p><p>Tổng tiền: 340.000 đ</p></div>	<div>Chọn phương thức thanh toán</div> <div><p><input checked="" type="radio"/> Momo</p><p><input type="radio"/> Vnpay</p><p><input type="radio"/> Paypal</p><p><input type="radio"/> Thanh toán Khi lên xe</p><p>Thanh toán ngay</p></div>
<div>Thông tin khách hàng</div> <div><p>Huy Vo</p><p>0375250833</p><p>voquanghuy08102000@gmail.com</p></div>	<div>Thời gian thanh toán</div> <div><p>Thời gian giữ chỗ còn lại: 9m 56s</p><div></div><p>Hướng dẫn thanh toán bằng Momo</p><ol style="list-style-type: none">Mở ứng dụng Momo trên điện thoạiDùng biểu tượng để quét mã QRQuét mã ở trang này và thanh toán</div>
<div>Lưu ý quan trọng</div>	

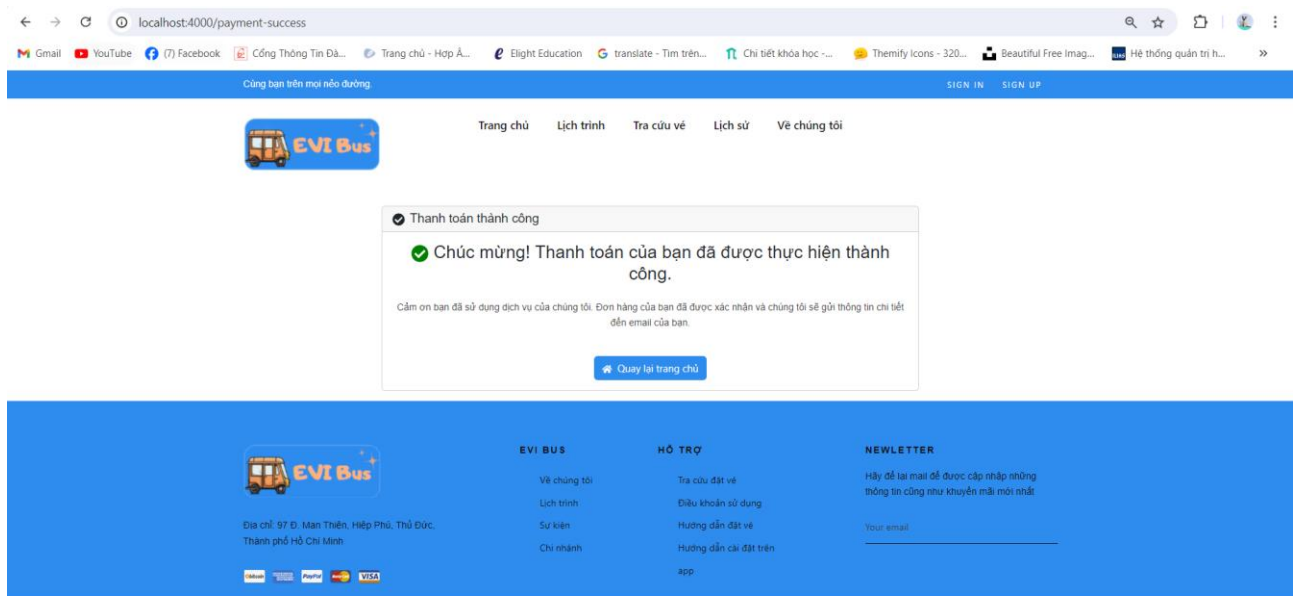
Hình 3. 7 Trang thanh toán của website với QR code MoMo



Hình 3. 8 Giao diện app sandbox của momo sau khi quét mã QR code



Hình 3. 9 Giao diện app sandbox của momo sau khi thanh toán thành công



Hình 3. 10 Hình 3. 12 Giao diện của website khi thanh toán momo thành công

3.4 PayPal

PayPal là dịch vụ thanh toán quốc tế phổ biến, cung cấp cả phương thức thanh toán qua QR code.

3.4.1 Quy trình công nghệ

PayPal tích hợp mã QR trong ứng dụng di động, cho phép người dùng tạo mã QR cá nhân hoặc doanh nghiệp để nhận thanh toán.

3.4.2 Bảo mật

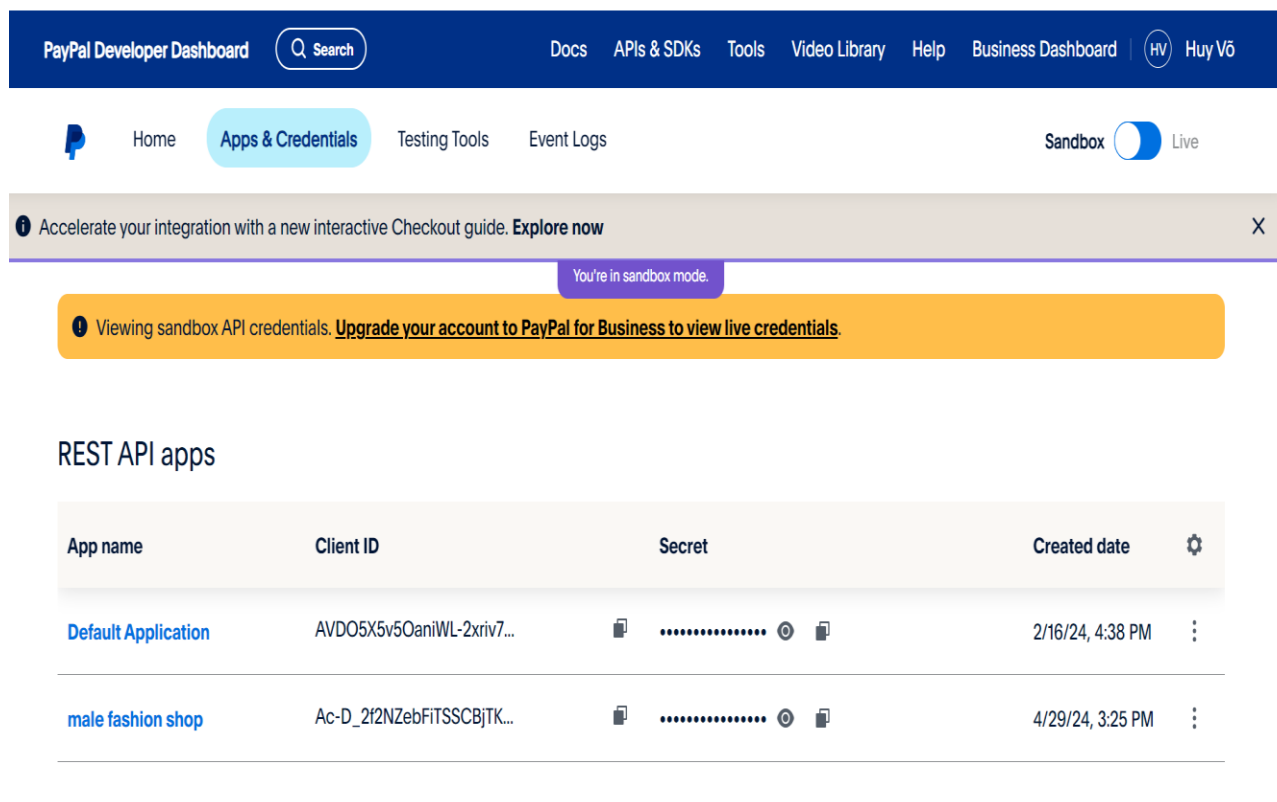
PayPal sử dụng các giao thức bảo mật cao cấp như mã hóa SSL, xác thực hai yếu tố và hệ thống phát hiện gian lận tiên tiến.

3.4.3 Tiện ích người dùng

- Thanh toán trực tuyến và tại cửa hàng.
- Chuyển tiền quốc tế dễ dàng.
- Bảo vệ người mua và người bán trong giao dịch.

3.4.4 PayPal REST APIs

API PayPal sử dụng REST, xác thực bằng mã thông báo truy cập OAuth 2.0 và trả về mã phản hồi HTTP cũng như phản hồi được mã hóa bằng JSON. Bạn có thể kiểm tra khả năng tích hợp của Hoa Kỳ bằng tài khoản Nhà phát triển PayPal. Lấy client ID và client secret khi đăng kí tài khoản sandbox của Paypal



Hình 3. 11 Đăng kí nhận Client ID và SecretKey của PayPal

Nhận access token

```
curl -v -X POST "https://api-m.sandbox.paypal.com/v1/oauth2/token"\n-u "CLIENT_ID:CLIENT_SECRET"\n-H "Content-Type: application/x-www-form-urlencoded"\n-d "grant_type=client_credentials"
```

Danh sách tham số HTTP Request

Parameter	Type	Description
count	integer	Số lượng mục cần liệt kê trong phản hồi.
end_time	integer	Ngày và giờ kết thúc để phạm vi hiển thị trong phản hồi, ở định dạng ngày và giờ trên Internet. Ví dụ: end_time=2016-03-06T11:00:00Z
page	integer	Số trang cho biết tập hợp mục nào sẽ được trả về trong phản hồi. Vì vậy, sự kết hợp của page=1 và page_size=20 trả về
page_size	integer	Số lượng mục cần trả về trong phản hồi.
total_count_required	boolean	Cho biết có hiển thị tổng số trong phản hồi hay không.

<code>sort_by</code>	string	Sắp xếp các khoản thanh toán trong phản hồi theo một giá trị được chỉ định, chẳng hạn như thời gian tạo hoặc thời gian cập nhật.
<code>sort_order</code>	string	Sắp xếp các mục trong phản hồi theo thứ tự tăng dần hoặc giảm dần.
<code>start_id</code>	string	ID của tài nguyên bắt đầu trong phản hồi. Khi kết quả được phân trang, bạn có thể sử dụng giá trị <code>next_id</code> làm <code>start_id</code> để tiếp tục với nhóm kết quả tiếp theo.
<code>start_index</code>	integer	Chỉ số bắt đầu của các khoản thanh toán được liệt kê. Thông thường, bạn sử dụng <code>start_index</code> để chuyển đến một vị trí cụ thể trong lịch sử tài nguyên dựa trên giỏ hàng của nó. Ví dụ: để bắt đầu ở mục thứ hai trong danh sách kết quả, hãy chỉ định <code>?start_index=2</code> .
<code>start_time</code>	string	Ngày và giờ bắt đầu để phạm vi hiển thị trong phản hồi, ở định dạng ngày và giờ trên Internet. Ví dụ: <code>start_time=2016-03-06T11:00:00Z</code> .

Bảng 3. 6 Danh sách tham số API requests

Mẫu response

Đối với tất cả các lỗi ngoại trừ lỗi Nhận dạng, PayPal trả về nội dung phản hồi lỗi bao gồm các chi tiết lỗi bổ sung ở định dạng này.

```
{
  "name": "ERROR_NAME",
  "message": "Error message.",
  "debug_id": "debug_ID",
  "details": [
    {
      "field": "field_name",
      "value": "value_passed",
      "location": "field_location",
      "issue": "problem_with_field",
      "description": "Error description."
    }
  ],
  "links": [
    {
      "https://error_documentation_link",
      "rel": "information_link",
      "encType": "application/json"
    }
  ]
}
```

Nội dung phản hồi cho các lỗi Nhận dạng bao gồm các chi tiết lỗi bổ sung ở định dạng sau:

```
{
  "error": "ERROR_NAME",
  "error_description": "ERROR_DESCRIPTION"
}
```

3.4.5 Tích hợp vào hệ thống website thương mại điện tử PayPalConfig

```
@Configuration
public class PaypalConfig {
    @Value("${paypal.client.app}")
    private String clientId;
    @Value("${paypal.client.secret}")
    private String clientSecret;
    @Value("${paypal.mode}")
    private String mode;

    @Bean
    public Map<String, String> paypalSdkConfig() {
        Map<String, String> sdkConfig = new
        HashMap<>();
    }
```

```

        sdkConfig.put("mode", mode);
        return sdkConfig;
    }

    @Bean
    public OAuthTokenCredential authTokenCredential()
    {
        return new OAuthTokenCredential(clientId,
        clientSecret, paypalSdkConfig());
    }

    @Bean
    public APIContext apiContext() throws
    PayPalRESTException {
        APIContext apiContext = new
        APIContext(authTokenCredential().getAccessToken());

        apiContext.setConfigurationMap(paypalSdkConfig());
        return apiContext;
    }
}

PaypalPaymentIntent
public enum PaypalPaymentIntent {
    sale, authorize, order
}

PaypalPaymentIntent
public enum PaypalPaymentMethod {
    credit_card, paypal
}

Utils
public class Utils {
    public static String getBaseUrl(HttpServletRequest
    request) {
        String scheme = request.getScheme();
        String serverName = request.getServerName();
        int serverPort = request.getServerPort();
        String contextPath = request.getContextPath();
        StringBuffer url = new StringBuffer();

        url.append(scheme).append("://").append(serverName);
        if ((serverPort != 80) && (serverPort != 443))
        {
            url.append(":").append(serverPort);
        }
        url.append(contextPath);
        if (url.toString().endsWith("/")) {
            url.append("/");
        }
        return url.toString();
    }
}

```

```

    }
}

PaypalService
@Service
public class PaypalService {
    @Autowired
    private APIContext apiContext;

    public Payment createPayment(
        Double total,
        String currency,
        PaypalPaymentMethod method,
        PaypalPaymentIntent intent,
        String description,
        String cancelUrl,
        String successUrl) throws
PayPalRESTException {
        Amount amount = new Amount();
        amount.setCurrency(currency);
        amount.setTotal(String.format("%.2f", total));

        Transaction transaction = new Transaction();
        transaction.setDescription(description);
        transaction.setAmount(amount);

        List<Transaction> transactions = new
ArrayList<>();
        transactions.add(transaction);

        Payer payer = new Payer();
        payer.setPaymentMethod(method.toString());

        Payment payment = new Payment();
        payment.setIntent(intent.toString());
        payment.setPayer(payer);
        payment.setTransactions(transactions);
        RedirectUrls redirectUrls = new
RedirectUrls();
        redirectUrls.setCancelUrl(cancelUrl);
        redirectUrls.setReturnUrl(successUrl);
        payment.setRedirectUrls(redirectUrls);
        apiContext.setMaskRequestId(true);
        return payment.create(apiContext);
    }

    public Payment executePayment(String paymentId,
String payerId) throws PayPalRESTException {
        Payment payment = new Payment();
        payment.setId(paymentId);

```



```

        PaymentExecution paymentExecute = new
PaymentExecution();
        paymentExecute.setPayerId(payerId);
        return payment.execute(apiContext,
paymentExecute);
    }
}

```

3.5 PayOn

PayOn là một nền tảng thanh toán tích hợp, cung cấp giải pháp thanh toán QR code cho các doanh nghiệp.

3.5.1 Quy trình công nghệ

PayOn sử dụng API để tích hợp mã QR vào các hệ thống POS và ứng dụng di động của các doanh nghiệp.

3.5.2 Bảo mật

PayOn đảm bảo an toàn bằng cách sử dụng mã hóa AES, xác thực giao dịch và giám sát hoạt động gian lận.

3.5.3 Tiện ích người dùng

- Hỗ trợ thanh toán đa kênh.
- Tích hợp dễ dàng với các hệ thống kinh doanh.
- Báo cáo và phân tích giao dịch chi tiết.

3.5.4 Thông tin cấu hình

Parameter	Description
URL_API	Đường dẫn API
MC_AUTH_USER	Tên Auth basic
MC_AUTH_PASS	Mật khẩu Http Auth basic
MC_SECRET_KEY	Khóa để thực hiện mã hóa tham số data trong các hàm nghiệp vụ
MC_ID	ID Merchant để định danh khách hàng trên PayOn
APP_ID	ID ứng dụng để định danh ứng dụng tích hợp

Bảng 3. 7 Thông tin cấu hình PayOn

Các bước lấy thông tin kết nối:

Bước 1: Đăng nhập PayOn bằng tài khoản bạn đã đăng ký.

Bước 2: Chọn Tích hợp thanh toán trên Menu màn hình

Bước 3: Nhấn nút Thông tin kết nối

Bước 4: Tại đây copy các thông tin tương ứng với các key:

Url Api: URL_API - Đường dẫn API.

Http Auth User: MC_AUTH_USER - Tên Auth basic.

Http Auth Pass: MC_AUTH_PASS - Mật khẩu Http Auth basic.

Merchant ID: MC_ID - ID Merchant.

Application ID: APP_ID - ID - Ứng dụng để định danh ứng dụng tích hợp.
Secret Key: MC_SECRET_KEY - Khóa để thực hiện mã hóa tham số data trong các hàm nghiệp vụ.

(Thư viện mã hoá data : <https://github.com/payon-sdk/aes-everywhere>.)

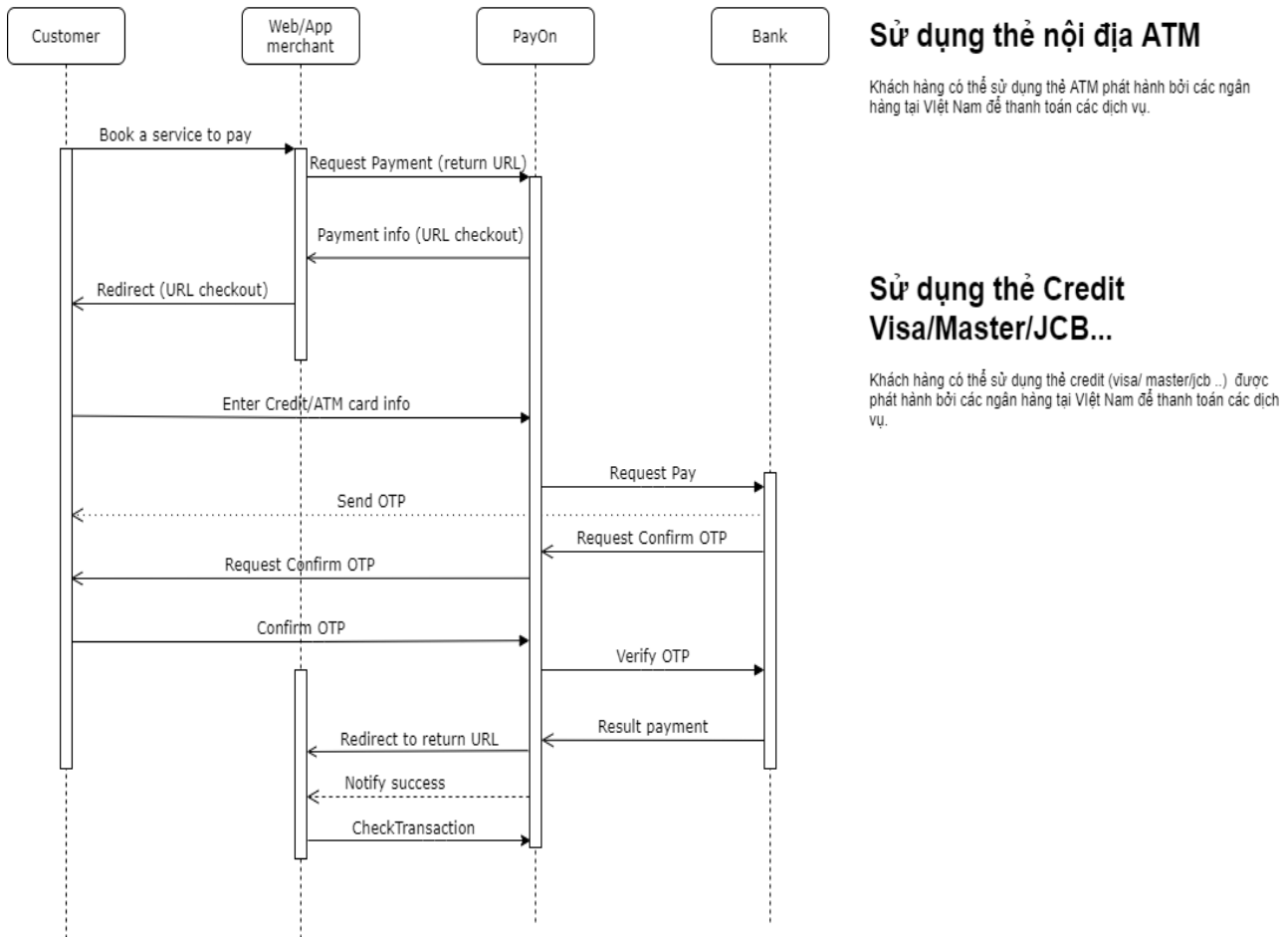
Bước 5: Paste các thông tin vừa Copy vào phần tích hợp thanh toán trên ứng dụng của bạn

Phương thức giao tiếp:

API_URL Môi trường Sandbox: <https://dev-api-merchant.payon.vn/v1/merchant> (Test).

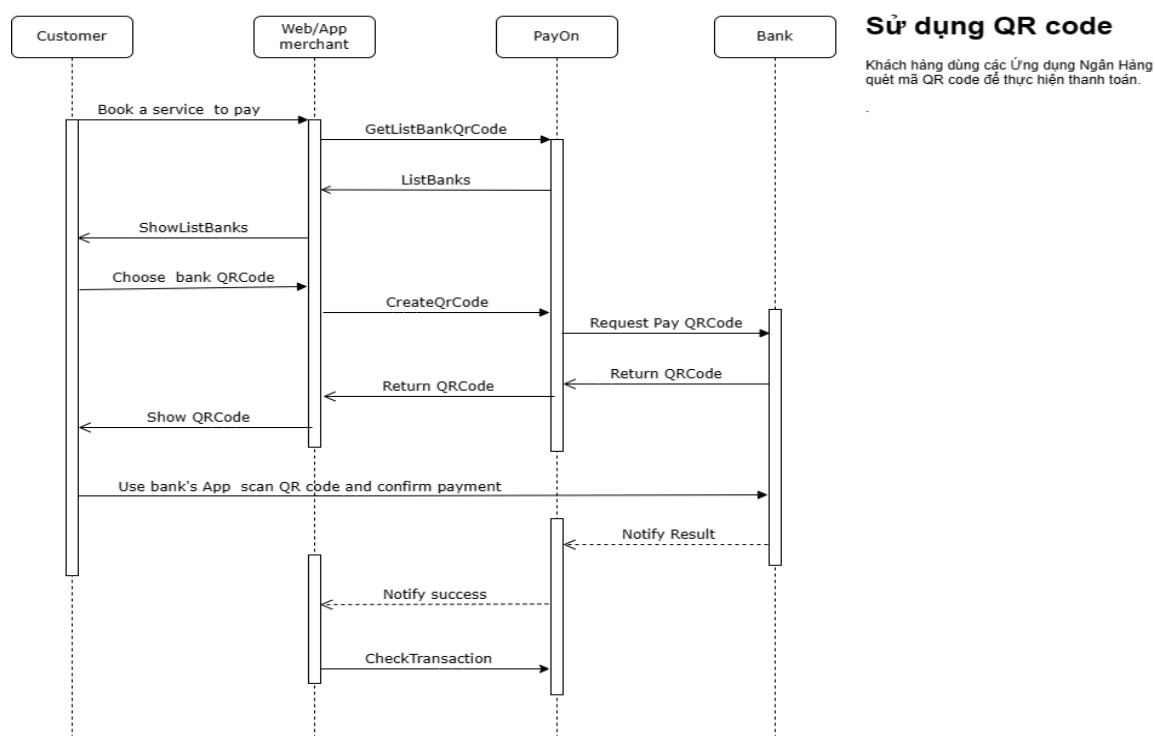
API_URL Môi trường Product: <https://sdk.payon.vn/v1/merchant> (Live).

3.5.5 Mô hình kết nối



Hình 3. 12 Luồng xử lý của PayOn với thẻ nội địa ATM và thẻ Credit/Visa/...

Thanh toán qua QR code



Hình 3. 13 Luồng xử lý của PayOn với QR code

3.6 So sánh và kết luận

Các dịch vụ thanh toán QR code của VNPay, PayPal, PayOn và MoMo đều mang lại tiện ích lớn cho người dùng và doanh nghiệp.

3.6.1 So sánh

Tiêu chí	VNPay	PayPal	PayOn (NextPay)	MoMo
Phạm vi hoạt động				
Hỗ trợ mã QR	Mã QR tĩnh và động	Mã QR tĩnh và động	Mã QR tĩnh và động	Mã QR tĩnh và động
Hỗ trợ giao dịch quốc tế	Không	Có	Không	Không
Tích hợp API	Hỗ trợ API dễ tích hợp	Hỗ trợ API đa nền tảng nhưng phức tạp	API đơn giản, dễ triển khai	API thân thiện, dễ tích hợp
Tiêu chuẩn bảo mật	PCI DSS	PCI DSS, bảo mật hai lớp	PCI DSS	PCI DSS
Hỗ trợ nhiều loại tiền tệ	Không	Có	Không	Không
Tốc độ xử lý giao dịch	Nhanh và ổn định	Tương đối nhanh	Nhanh	Nhanh
Phí giao dịch	Thấp, phù hợp cho các doanh nghiệp nhỏ	Tương đối cao, tùy vào loại giao dịch	Thấp, phù hợp cho doanh nghiệp nhỏ	Thấp, phổ biến với người dùng cá nhân

Hệ sinh thái liên kết	Nhiều ngân hàng và ứng dụng tại Việt Nam	Toàn cầu, nhiều nền tảng và dịch vụ	Các đối tác nội địa	Nhiều đối tác và dịch vụ trong nước
Phù hợp với doanh nghiệp	Doanh nghiệp vừa và nhỏ tại Việt Nam	Doanh nghiệp có nhu cầu giao dịch quốc tế	Doanh nghiệp vừa và nhỏ tại Việt Nam	Doanh nghiệp vừa và nhỏ, cửa hàng bán lẻ
Chi phí triển khai	Thấp	Cao	Thấp	Thấp
Khả năng mở rộng	Tốt trong nước	Tốt toàn cầu	Hạn chế quốc tế	Hạn chế quốc tế

Bảng 3. 8 So sánh tổng quan các công thanh toán

3.6.2 Kết luận

Mỗi dịch vụ thanh toán QR code có những ưu điểm và đặc điểm riêng, phù hợp với nhu cầu và mục tiêu sử dụng khác nhau. Sự phát triển và áp dụng rộng rãi của thanh toán QR code đang mang lại nhiều lợi ích thiết thực cho cả người dùng và doanh nghiệp, đồng thời thúc đẩy sự phát triển của ngành công nghệ tài chính.

CHƯƠNG VI: TÌM HIỂU TỔNG QUAN KỸ THUẬT CÔNG THANH TOÁN NAPAS

4.1. Tìm hiểu tổng quan

4.1.1 Giới thiệu

- Công ty Cổ phần Thanh toán Quốc gia Việt Nam (NAPAS) là đơn vị cung cấp dịch vụ thanh toán liên ngân hàng, kết nối các ngân hàng thương mại và các tổ chức tài chính tại Việt Nam. NAPAS cung cấp một loạt các dịch vụ thanh toán điện tử, bao gồm chuyển khoản, thanh toán hóa đơn, và thanh toán thương mại điện tử. Nghiên cứu này tập trung vào phân tích công nghệ của công thanh toán NAPAS.

4.1.2 Quy trình giao dịch

- Quy trình giao dịch
 - Quy trình giao dịch của NAPAS bắt đầu từ việc khách hàng khởi tạo giao dịch tại một đơn vị chấp nhận thanh toán. Hệ thống NAPAS sau đó sẽ xử lý giao dịch này và liên kết với các ngân hàng liên quan để hoàn tất thanh toán.
 - Tạo tài khoản và liên kết ngân hàng
 - Người dùng tạo tài khoản trên hệ thống NAPAS và liên kết tài khoản này với tài khoản ngân hàng của họ. Quá trình này bao gồm việc xác thực thông tin cá nhân và thông tin tài khoản ngân hàng.
 - Khởi tạo giao dịch
 - Khi khách hàng thực hiện mua hàng hoặc thanh toán dịch vụ, họ chọn NAPAS làm phương thức thanh toán và nhập thông tin cần thiết. Hệ thống NAPAS sẽ xác thực thông tin và khởi tạo giao dịch.
 - Xử lý giao dịch
 - Hệ thống NAPAS xử lý giao dịch bằng cách liên kết với ngân hàng của người mua để xác nhận và chuyển tiền. Sau khi ngân hàng của người mua xác nhận, NAPAS sẽ chuyển tiền đến tài khoản của người bán.

4.1.3 Cơ chế hoạt động nội bộ

- Xử lý thanh toán
 - Hệ thống NAPAS sử dụng một loạt các bước xử lý để đảm bảo giao dịch được thực hiện nhanh chóng và an toàn. Quy trình bao gồm kiểm tra thông tin giao dịch, xác thực người dùng, và chuyển tiền giữa các tài khoản.
- Quản lý rủi ro và bảo mật
 - Bảo mật là một trong những yếu tố quan trọng nhất của hệ thống NAPAS. Hệ thống sử dụng các phương pháp mã hóa tiên tiến để bảo vệ thông tin giao dịch. Ngoài ra, NAPAS còn áp dụng các biện pháp xác thực mạnh mẽ và hệ thống giám sát gian lận để đảm bảo an toàn cho người dùng.

4.1.4 Công nghệ nền tảng

- NAPAS sử dụng nhiều công nghệ tiên tiến để duy trì và phát triển hệ thống thanh toán của mình.

API RESTful

- Hệ thống NAPAS cung cấp các API RESTful để tích hợp với các dịch vụ thanh toán và ngân hàng. API này cho phép các nhà phát triển tích hợp dịch vụ NAPAS vào các ứng dụng và trang web của họ một cách dễ dàng.

Cơ sở dữ liệu phân tán

- NAPAS sử dụng hệ thống cơ sở dữ liệu phân tán để đảm bảo tính sẵn sàng và hiệu suất cao cho hệ thống. Cơ sở dữ liệu này cho phép NAPAS xử lý một lượng lớn giao dịch mà không gặp sự cố.

Hệ thống máy học

- Công nghệ máy học được áp dụng để phân tích và dự đoán các mẫu gian lận, giúp bảo vệ hệ thống và người dùng khỏi các hoạt động gian lận.

4.2. Đặc tả kỹ thuật

4.2.1. Kiểm soát truy cập bảo mật

- Hệ thống NAPAS sử dụng giao thức OAuth 2.0 để xác thực và cấp quyền truy cập API. Đối tác cần lấy mã thông báo truy cập từ NAPAS để thực hiện các giao dịch.
"Authentication endpoint: [https:// dps-staging.napas .com.vn/api/oauth/token](https://dps-staging.napas.com.vn/api/oauth/token)"

4.2.2. Thông tin thẻ

Liệt kê thông tin chi tiết về các loại thẻ được hỗ trợ, bao gồm:

- Thẻ quốc tế: MasterCard (3DS và không 3DS), Visa, American Express, JCB.
- Thẻ nội địa.

a) Thông tin xác thực

Cung cấp thông tin về các tham số xác thực bao gồm: grant_type, client_id, client_secret, username và password.

b) Thông tin thẻ

Cung cấp ví dụ về thông tin thẻ cho các loại thẻ khác nhau bao gồm:

- MasterCard (3DS và không 3DS)
- Visa
- American Express
- JCB
- Thẻ nội địa

4.3 Tích hợp thanh toán lưu trữ (Hosted Checkout)

4.3.1 Giao dịch Data Key

- Đối tác sử dụng giao dịch này để khởi tạo đơn hàng và lấy các khóa bảo mật cần thiết để tích hợp NAPAS Hosted Checkout.
- "Partner uses this transaction to initialize the order and obtain the secure keys needed for integrating Napas Hosted Checkout."
- URL: <https://dpsstaging.napas.com.vn/api/rest/version/32/merchant/{merchantId}/datakey>

4.3.2. Thanh toán / Thanh toán bằng Token đã trả về / Tạo Token – tích hợp với NAPAS Hosted Form

- **Thanh toán:** Khách hàng chỉ thanh toán cho hàng hóa / dịch vụ.
- **Thanh toán bằng Token đã trả về:** Khách hàng thanh toán cho hàng hóa / dịch vụ và cũng nhận được Token có thể được sử dụng để thanh toán sau này.
- **Tạo Token:** Khách hàng nhập thông tin thẻ, sau đó nhận được Token có thể được sử dụng để thanh toán sau này (lưu lượng này chỉ hỗ trợ Thẻ có thương hiệu).

4.3.3 Thanh toán / Thanh toán bằng Token đã trả về / Tạo Token – tích hợp với NAPAS Payment Page

- **Thanh toán:** Khách hàng chỉ thanh toán cho hàng hóa / dịch vụ.
- **Thanh toán bằng Token đã trả về:** Khách hàng thanh toán cho hàng hóa / dịch vụ và cũng nhận được Token có thể được sử dụng để thanh toán sau này.
- **Tạo Token:** Khách hàng nhập thông tin thẻ, sau đó nhận được Token có thể được sử dụng để thanh toán sau này (lưu lượng này chỉ hỗ trợ thẻ có thương hiệu quốc tế).

4.3.4 Nhận phản hồi từ NAPAS

Đối tác sẽ nhận được phản hồi theo 2 cách riêng biệt:

- Gửi biểu mẫu: kết quả sẽ được gửi đến URL được xác định trong tham số action. Kết quả sẽ được bao gồm trong tham số napasResult.
- IPN: kết quả sẽ được POST đến URL IPN mà Đối tác đã đăng ký với NAPAS. Kết quả sẽ được bao gồm trong requestBody.

4.4 Triển khai API trực tiếp

4.4.1. Thanh toán / Ủy quyền / Chụp / Hủy bỏ / Hoàn trả (đối với Xử lý Thẻ Quốc tế) / Hoàn trả giao dịch Nội địa (đối với Xử lý Thẻ Nội địa)

- Đối tác sử dụng thông tin sau để tích hợp với hệ thống NAPAS:
- URL: <https://dpsstaging.napas.com.vn/api/rest/version/1/merchant/{merchantId}/order/{orderId}/transaction/{transactionId}>
- Phương thức HTTP: PUT
- Kiểu nội dung: Application/json
- Xác thực: Thao tác này yêu cầu xác thực thông qua phương pháp sau: OAuth2

4.4.2. Thanh toán bằng giao dịch 3DSecure

- Đối tác sử dụng thông tin sau để tích hợp với hệ thống NAPAS:
- URL: <https://dpsstaging.napas.com.vn/api/rest/version/1/merchant/{merchantId}/order/{orderId}/transaction/{transactionId}>
- Phương thức HTTP: PUT
- Kiểu nội dung: Application/json
- Xác thực: Thao tác này yêu cầu xác thực thông qua phương pháp sau: OAuth2.

4.4.3. Giao dịch OTP mua hàng – tích hợp với NAPAS Hosted Checkout

- Đối tác sử dụng thông tin sau để tích hợp với hệ thống NAPAS:
- URL: <https://dpsstaging.napas.com.vn/api/rest/version/32/merchant/{merchantId}/purchaseOtp>
- Phương thức HTTP: POST
- Kiểu nội dung: Application/json
- Xác thực: Thao tác này yêu cầu xác thực thông qua phương pháp sau: OAuth2

4.4.4. Giao dịch truy xuất đơn hàng

- Đối tác sử dụng thông tin sau để tích hợp với hệ thống NAPAS:
- URL (đối với truy xuất thẻ có thương hiệu):
<https://dpsstaging.napas.com.vn/api/rest/version/1/merchant/{merchantId}/order/{orderId}/international>
- URL (đối với truy xuất thẻ nội địa):
<https://dpsstaging.napas.com.vn/api/rest/version/1/merchant/{merchantId}/order/{orderId}/domestic>

- Phương thức HTTP: GET
- Kiểu nội dung: Application/json
- Xác thực: Thao tác này yêu cầu xác thực thông qua phương pháp sau: OAuth2

4.4.5. Giao dịch xóa Token

- Đối tác sử dụng thông tin sau để tích hợp với hệ thống NAPAS:
- URL: <https://dpsstaging.napas.com.vn/api/rest/version/1/merchant/{merchantId}/token/{token}>
- Phương thức HTTP: DELETE
- Kiểu nội dung: Application/json
- Xác thực: Thao tác này yêu cầu xác thực thông qua phương pháp sau: OAuth2.

4.4.6 Tạo Giao dịch QR

- Đối tác sử dụng thông tin sau để tích hợp với hệ thống NAPAS:
- URL: <https://dpsstaging.napas.com.vn/api/rest/version/1/merchant/{merchantId}/qrCode>
- Phương thức HTTP: POST
- Kiểu nội dung: Application/json
- Xác thực: Thao tác này yêu cầu xác thực thông qua phương pháp sau: OAuth2.

4.4.7 Tham số lỗi

- Trong trường hợp xảy ra lỗi trong quá trình xác thực yêu cầu, các trường sau sẽ được gửi lại cho đối tác:
 - error.cause
 - error.explanation
 - result

4.4.8 Mua hàng bằng giao dịch Token đã trả về - tích hợp với NAPAS Hosted Checkout

- Đối tác sử dụng thông tin sau để tích hợp với hệ thống NAPAS:
- URL: <https://dpsstaging.napas.com.vn/api/rest/version/32/merchant/{merchantId}/purchaseWithReturnedToken>
- Phương thức HTTP: POST
- Kiểu nội dung: Application/json
- Xác thực: Thao tác này yêu cầu xác thực thông qua phương pháp sau: OAuth2

4.4.9 Giao dịch truy xuất Token

- Đối tác sử dụng thông tin sau để tích hợp với hệ thống NAPAS:
- URL: <https://dpsstaging.napas.com.vn/api/rest/version/1/merchant/{merchantId}/orderId/{orderId}/getToken>
- Phương thức HTTP: GET
- Kiểu nội dung: Application/json
- Xác thực: Thao tác này yêu cầu xác thực thông qua phương pháp sau: OAuth2.

4.5 Mã phản hồi

- Tài liệu cung cấp danh sách đầy đủ các mã phản hồi cho cả quy trình thẻ quốc tế và thẻ nội địa.

4.5.1. Xử lý thẻ quốc tế

- Mã phản hồi được chia thành 3 loại:
 - **Thành công (SUCCESS):** Giao dịch được chấp thuận.
 - **Thất bại (FAILURE):** Giao dịch bị từ chối vì nhiều lý do khác nhau.
 - **Lỗi (ERROR):** Xảy ra lỗi trong quá trình xử lý giao dịch.

4.5.2. Xử lý thẻ nội địa

- Tương tự như thẻ quốc tế, mã phản hồi cho thẻ nội địa cũng được chia thành 3 loại:
 - **Thành công (SUCCESS):** Giao dịch được chấp thuận.
 - **Thất bại (FAILURE):** Giao dịch bị từ chối vì nhiều lý do khác nhau.
 - **Lỗi (ERROR):** Xảy ra lỗi trong quá trình xử lý giao dịch.

4.6 Phần kết luận

- Đặc tả kỹ thuật NAPAS cung cấp thông tin chi tiết về cách tích hợp với hệ thống cổng thanh toán NAPAS. bao gồm các thông tin về bảo mật, thông tin thẻ, tích hợp thanh toán, triển khai API, mã phản hồi, v.v. Việc hiểu rõ là điều kiện tiên quyết để các đối tác có thể tích hợp thành công và khai thác tối đa các tính năng của hệ thống NAPAS.

KẾT LUẬN

- Nghiên cứu "Nghiên cứu các giải pháp thanh toán tiền dùng QR Code" đã đạt được các mục tiêu đề ra, cung cấp một cái nhìn tổng quan và chi tiết về việc ứng dụng công nghệ QR Code trong lĩnh vực thanh toán điện tử. Qua quá trình nghiên cứu, đã đạt được những kết quả chính sau:
 1. Đã tìm hiểu sâu về cơ chế hoạt động, quy trình phát sinh và đọc QR Code trên các thiết bị di động, làm rõ được tiềm năng ứng dụng của công nghệ này trong lĩnh vực thanh toán.
 2. Nghiên cứu chi tiết về thủ tục thanh toán qua API của các cổng thanh toán phổ biến như VNPay, PayOn, PayPal và MoMo, cung cấp cái nhìn toàn diện về quy trình công nghệ, bảo mật và tiện ích người dùng của từng nền tảng.
 3. Thực hiện so sánh và đánh giá các cổng thanh toán, giúp người dùng và nhà phát triển có cơ sở để lựa chọn giải pháp phù hợp nhất cho nhu cầu của mình.
 4. Tìm hiểu tổng quan về kỹ thuật công thanh toán NAPAS, một hệ thống quan trọng trong hạ tầng thanh toán quốc gia, góp phần làm rõ cơ chế hoạt động của hệ thống thanh toán điện tử tại Việt Nam.
 5. Xây dựng được chương trình minh họa giải pháp thanh toán tiền bằng QR Code, tích hợp các cổng thanh toán đã nghiên cứu và tuân thủ chuẩn bảo mật 3D Secure.
- Kết quả nghiên cứu này không chỉ có ý nghĩa học thuật mà còn có giá trị ứng dụng thực tiễn cao. Việc triển khai thanh toán qua QR Code góp phần đẩy mạnh xu hướng thanh toán không tiền mặt, nâng cao trải nghiệm người dùng và tăng cường an ninh trong giao dịch tài chính.
- Tuy nhiên, nghiên cứu cũng chỉ ra một số thách thức cần được giải quyết trong tương lai như vấn đề bảo mật thông tin, tính tương thích giữa các hệ thống, và nâng cao nhận thức của người dùng về thanh toán điện tử.
- Hướng phát triển tiếp theo của nghiên cứu có thể tập trung vào việc tối ưu hóa quy trình thanh toán, tăng cường tính năng bảo mật, và mở rộng tích hợp với các nền tảng thanh toán mới nổi trên thị trường. Ngoài ra, việc nghiên cứu ứng dụng các công nghệ mới như blockchain trong thanh toán QR Code cũng là một hướng đi tiềm năng.

DANH MỤC TÀI LIỆU THAM KHẢO

Tiếng Việt:

1. Công ty cổ phần Postech, *Mã QR code là gì? Cấu tạo của mã QR code!*, Hà Nội, 2020.
2. Công ty Cổ phần Thanh toán Quốc gia Việt Nam (NAPAS), *NAPAS System Technical Specification*, phiên bản 3.4, Hà Nội, 2019.

Tiếng Anh:

1. International Organization for Standardization (ISO), *ISO/IEC 18004:2015 Information technology – Automatic identification and data capture techniques – QR Code bar code symbology specification*, Geneva, Switzerland, 2015.

Danh mục các Website tham khảo:

1. *A guide to QR code*. <https://www.checkout.com/>
2. *MoMo Developer, Tài liệu kỹ thuật*, <https://developers.momo.vn/v3/vi/docs/payment/api/wallet/onetime/>
3. *PayOn API docs*, <https://docs.nextpay.vn/#gi-i-thi-u>
4. *Paypal Developer*, <https://developer.paypal.com/tools/sandbox/>
5. *QR code.com*: <https://www.qrcode.com/en/about/>