

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
LỚP CỬ NHÂN TÀI NĂNG**

NGUYỄN HỮU NHÂN-NGUYỄN CAO KỲ

**NGHIÊN CỨU VÀ CÀI ĐẶT
GIẢI PHÁP MOBILE PAYMENT
VỚI THỬ NGHIỆM TRÊN NỀN TẢNG
ANDROID**

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT

TP. HCM, 2014

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
LỚP CỬ NHÂN TÀI NĂNG**

NGUYỄN HỮU NHÂN – 1012270

NGUYỄN CAO KỲ – 1012205

**NGHIÊN CỨU VÀ CÀI ĐẶT
GIẢI PHÁP MOBILE PAYMENT
VỚI THỬ NGHIỆM TRÊN NỀN TẢNG
ANDROID**

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT

GIÁO VIÊN HƯỚNG DẪN

TS. ĐINH BÁ TIẾN

TS. TÔ BÁ LÂM

KHÓA 2010 - 2014

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TpHCM, ngày tháng năm

Giáo viên hướng dẫn
[Ký tên và ghi rõ họ tên]

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Khóa luận đáp ứng yêu cầu của Khóa luận cử nhân CNTT.

TpHCM, ngày tháng năm

Giáo viên phản biện

[Ký tên và ghi rõ họ tên]

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin chân thành cảm ơn Ban giám hiệu Trường Đại học Khoa Học Tự Nhiên, Đại học Quốc gia TP.HCM đã tạo điều kiện thuận lợi nhất để chúng em có thể học tập tại trường trong thời gian bốn năm Đại học vừa qua.

Chúng em xin phép được gửi lời cảm ơn sâu sắc đến các Thầy Cô tại Khoa Công Nghệ Thông Tin, vì Thầy Cô luôn tâm huyết trong việc dạy dỗ chúng em không chỉ về mặt kiến thức chuyên môn mà còn cả về cách những hiểu biết trong cuộc sống thực tế.

Chúng em xin trân trọng cảm ơn Thầy Đinh Bá Tiến và Thầy Tô Bá Lâm, vì 2 Thầy đã nhiệt tình hướng dẫn, giúp đỡ cũng như quan tâm và khích lệ chúng em mỗi khi chúng em gặp khó khăn đồng thời luôn tạo mọi điều kiện thuận lợi nhất có thể để chúng em có thể hoàn thành đề tài đúng thời gian.

Chúng em xin được cảm ơn các anh chị trong Công ty Cổ phần Dịch vụ Trực tuyến Cộng Đồng Việt (VietUnion) đã luôn quan tâm và nhiệt tình giúp đỡ chúng em trong suốt quá trình thực hiện đề tài.

Chúng em xin được cảm ơn các anh chị và các bạn trong Khoa Công Nghệ Thông Tin vì sự giúp đỡ, quan tâm và chia sẻ những kinh nghiệm quý báu cho chúng em trong thời gian chúng em thực hiện đề tài.

Cuối cùng, xin được gửi lời cảm ơn đến gia đình và bạn bè đã luôn quan tâm, chia sẻ cũng như động viên chúng em trong những lúc chúng em gặp khó khăn trong suốt khoảng thời gian vừa qua.

Mặc dù chúng em đã cố gắng hoàn thành luận văn trong phạm vi và khả năng cho phép, nhưng chắc chắn sẽ không tránh khỏi những thiếu sót. Kính mong sự cảm thông và tận tình chỉ bảo của quý Thầy Cô và các bạn.

Nhóm thực hiện

Nguyễn Cao Kỳ & Nguyễn Hữu Nhân

Khoa Công Nghệ Thông Tin

Lớp Cử Nhân Tài Năng

ĐỀ CƯƠNG CHI TIẾT

Tên Đề Tài: Nghiên Cứu Và Cài Đặt Giải Pháp Mobile Payment Với Thử Nghiệm Trên Nền Tảng Android
Giáo viên hướng dẫn: TS. Đinh Bá Tiến – TS. Tô Bá Lâm
Thời gian thực hiện: 12/10/2013 – 30/06/2014
Sinh viên thực hiện: Nguyễn Hữu Nhân (1012270) – Nguyễn Cao Kỳ (1012205)
Loại đề tài: Tìm hiểu công nghệ và Xây dựng ứng dụng

Nội Dung Đề Tài: <p>Phân tích, tìm hiểu những mô hình và giải pháp thanh toán di động đã và đang hoạt động, để tìm ra được những điểm mạnh, điểm yếu và hạn chế của mỗi giải pháp, từ đó đề xuất và lựa chọn giải pháp phù hợp nhất, có tính khả thi cao nhất đối với hệ thống thanh toán di động tại thị trường Việt Nam.</p>
Kế Hoạch Thực Hiện: <ul style="list-style-type: none">- 12/10/2013 – 01/11/2013: Thảo luận và thống nhất ý tưởng- 02/11/2013 – 23/11/2013: Nghiên cứu tình hình thanh toán di động trong và ngoài nước- 24/11/2013 – 07/12/2013: Đề xuất 3 mô hình thanh toán

<ul style="list-style-type: none"> - 08/12/2013 – 21/12/2013: Thiết kế kiến trúc hệ thống - 22/12/2013 – 11/01/2014: Thiết kế cơ sở dữ liệu - 12/01/2014 – 01/02/2014: Viết ứng dụng cho mô hình thanh toán 1 - 02/02/2014 – 22/02/2014: Tìm hiểu NFC và QRCode trên điện thoại Android - 23/02/2014 – 08/03/2014: Tìm hiểu Web Realtime trên ASP.NET MVC và Android - 09/03/2014 – 22/03/2014: Tìm hiểu Google Cloud Messaging và Google Maps trên Android - 23/03/2014 – 05/04/2014: Tìm hiểu cách tùy biến các Layout của Android - 06/04/2014 – 19/04/2014: Viết ứng dụng cho mô hình thanh toán 2 - 20/04/2014 – 03/05/2014: Tìm hiểu thực tại tăng cường và mã hóa trên Android, ASP.NET - 04/05/2014 – 17/05/2014: Viết ứng dụng cho mô hình thanh toán 3 - 18/05/2014 – 31/05/2014: Kiểm tra, hoàn thiện chức năng ứng dụng - 01/06/2014 – 30/06/2014: Khảo sát ý kiến người dùng và viết báo cáo 	
Xác nhận của GVHD	Ngày.....tháng.....năm..... SV Thực hiện



MỤC LỤC

LỜI CẢM ƠN	i
ĐỀ CƯƠNG CHI TIẾT	ii
MỤC LỤC.....	iv
DANH MỤC HÌNH ẢNH	x
DANH MỤC BẢNG.....	xiii
DANH MỤC CÁC THUẬT NGỮ, TỪ VIẾT TẮT.....	xiv
MỞ ĐẦU.....	1
Chương 1: THANH TOÁN DI ĐỘNG – MỘT XU HƯỚNG PHÁT TRIỂN CỦA PHƯƠNG THỨC THANH TOÁN.....	2
1.1. Thanh toán bằng tiền mặt.	2
1.2. Thanh toán bằng Thẻ Ghi Nợ Nội Địa	3
1.3. Thanh toán Online	5
1.4. Phát biểu bài toán.	7
Chương 2: CÁC MÔ HÌNH THANH TOÁN DI ĐỘNG.....	8
2.1. Phân tích các mô hình.....	8
2.1.1. Mô Hình MoPay	8
2.1.2. Mô Hình PayPass + PayPass Wallet + NFC Phone.....	10
2.1.3. Mô hình Google Wallet.	11
2.2. Đề xuất các mô hình.	12
2.2.1. Mô hình thanh toán 1	13
2.2.2. Mô hình thanh toán 2.	17
2.2.3. Mô hình thanh toán 3.	22

Chương 3: XÂY DỰNG CÁC MÔ HÌNH ĐỀ XUẤT.....	25
3.1. Thiết kế kiến trúc hệ thống.....	25
3.2. Thiết kế cơ sở dữ liệu	27
3.2.1. UserProfile.....	27
3.2.2. CustomersProfile.	28
3.2.3. CashierProfile.	28
3.2.4. Bank.....	29
3.2.5. BankInfo.	29
3.2.6. WalletDetail.....	30
3.2.7. Shop.....	31
3.2.8. Bill.....	32
3.3. Xây dựng ứng dụng.....	33
3.3.1. Mô Hình Thanh Toán 1.....	33
3.3.1.1. API được server cung cấp.....	33
3.3.1.2. Ứng dụng trên điện thoại di động.....	34
3.3.1.3. Web Client ở cửa hàng.....	35
3.3.2. Mô hình thanh toán 2.....	35
3.3.2.1. API được server cung cấp.....	35
3.3.2.2. Ứng dụng trên điện thoại di động.....	36
3.3.2.3. Web Client ở cửa hàng.....	37
3.3.3. Mô hình thanh toán 3.....	38
3.3.3.1. API được server cung cấp.....	38
3.3.3.2. Ứng dụng trên điện thoại di động.....	39
3.3.3.3. Web Client ở cửa hàng.....	40

3.3.4.	Các chức năng khác của ứng dụng trên điện thoại di động.	40
3.4.	Các vấn đề chính khi xây dựng Server cung cấp API – Web API	41
3.4.1.	Ngôn ngữ, nền tảng.....	41
3.4.2.	Lý do chọn ASP.NET MVC 4	41
3.4.3.	Thống nhất cấu trúc gói tin Web API gửi về Web Shop và App viPay	42
3.4.4.	Sử dụng thư viện ASP.Net SignalR thực hiện liên kết thời gian thực giữa Web API - ASP.Net với Web Client- ASP.Net, giữa Web API - ASP.Net với Mobile application – Android.	43
3.4.5.	Web API – Giải mã gói tin với thuật toán RSA	47
3.4.6.	Web API – Xử lý trường hợp trùng code	48
3.5.	Các vấn đề chính khi xây dựng Website cửa hàng – Web Shop (<i>ngoài một số vấn đề đã được nêu ở từng mô hình</i>).....	48
3.5.1.	Ngôn ngữ, nền tảng.....	48
3.5.2.	Xử dụng AJAX.	48
3.5.3.	QR code	49
3.5.4.	. SignalR (đã đề cập mục 3.4.4).....	49
3.5.5.	Xây dựng Windows Form hỗ trợ Web Shop – WinForm Shop	49
3.5.5.1.	Đọc QRCode sử dụng thư viện Zxing	49
3.5.5.2.	Sử dụng NFC trên Windows 8.....	50
3.5.5.3.	Gửi thông điệp từ trình duyệt web tới Windows Form.....	52
3.5.5.4.	Gửi thông điệp từ Windows Form tới trình duyệt web.....	52
3.6.	Các vấn đề chính khi xây dựng ứng dụng trên nền tảng Android – App viPay 53	
3.6.1.	Ngôn ngữ, nền tảng.....	53

3.6.2.	Lý do chọn hệ điều hành Android	53
3.6.3.	Truy vấn tới Web API.....	54
3.6.4.	Đọc và ghi dữ liệu JSON sử dụng thư viện Gson.....	55
3.6.5.	Push notification sử dụng Google Cloud Messaging	55
3.6.6.	Hiển thị bản đồ với Google Maps.....	57
3.6.7.	Chỉ đường với Google API.....	58
3.6.8.	Realtime giữa Web API và App viPay. (đã đề cập ở mục 3.4.4)	59
3.6.9.	Mã hóa gói tin với thuật toán RSA	59
3.6.10.	Kéo thả để cập nhật danh sách cho ListView	60
3.6.11.	Xây dựng Slider Menu.....	60
3.6.12.	Đọc và xuất QRCode sử dụng thư viện Zxing.....	61
3.6.13.	Tùy biến EditText với ClearableEditText	62
3.6.14.	NFC.....	63
3.6.15.	Xây dựng thực tại ảo tăng cường (Augmented Reality) để xem danh sách các cửa hàng	64
Chương 4: GIỚI THIỆU PHẦN MỀM VÀ KHẢO SÁT KHÁCH HÀNG SỬ DỤNG PHẦN MỀM.....		65
4.1.	Giới thiệu ứng dụng trên hệ điều hành Android cho Khách hàng.....	65
4.1.1.	Xác lập Mã pin ban đầu và nhập mã Pin để vào ứng dụng	65
4.1.2.	Đăng ký.....	66
4.1.3.	Đăng nhập	67
4.1.4.	Quên mật khẩu	67
4.1.5.	Màn hình chính.	68
4.1.5.1.	Mô hình thanh toán 1	68

4.1.5.2.	Mô hình thanh toán 2	69
4.1.5.3.	Mô hình thanh toán 3	69
4.1.6.	Xem lịch sử giao dịch	69
4.1.7.	Xem danh sách cửa hàng	70
4.1.8.	Hiển thị bản đồ cửa hàng xung quanh	71
4.1.9.	AR - Thực tại tăng cường	71
4.1.10.	Slider Menu chức năng	72
4.1.11.	Quản lý tài khoản	72
4.1.12.	Quản lý tài khoản ví điện tử	73
4.1.13.	Quản lý thẻ thanh toán	73
4.1.14.	Chức năng khác	74
4.2.	Giới thiệu ứng dụng web dành cho Nhân viên thanh toán	74
4.2.1.	Đăng nhập	74
4.2.2.	Thanh toán	75
4.2.2.1.	Mô hình thanh toán 1	75
4.2.2.2.	Mô hình thanh toán 2	76
4.2.2.3.	Mô hình thanh toán 3	77
4.2.3.	Xem lịch sử giao dịch	77
4.2.4.	Đăng tải thông tin quảng cáo	78
4.2.5.	Phần mềm trên Windows Form hỗ trợ cho Web – WinForm Shop	79
4.3.	Quy trình thực hiện thanh toán trong các mô hình	79
4.3.1.	Mô hình thanh toán 1	79
4.3.2.	Mô hình thanh toán 2	83
4.3.3.	Mô hình thanh toán 3	85

4.4.	Khảo sát khách hàng sử dụng phần mềm	87
4.4.1.	Khảo sát Mô hình thanh toán 1	88
4.4.2.	Khảo sát Mô hình thanh toán 2	89
4.4.3.	Khảo sát Mô hình thanh toán 3	90
4.4.4.	So sánh giữa 3 mô hình thanh toán.....	91
Chương 5:	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	92
5.1.	Kết luận.....	92
5.2.	Hướng phát triển.....	93
DANH MỤC	TÀI LIỆU THAM KHẢO	94
PHỤ LỤC A	96
A.1	Nội dung bảng khảo sát	96
A.2	Kết quả khảo sát.	99

DANH MỤC HÌNH ẢNH

Hình 2-1: Sơ đồ mô phỏng quy trình xử lý mô hình MoPay	9
Hình 2-2: Sơ đồ mô tả các dịch vụ do Google Wallet cung cấp.....	12
Hình 2-3: Sơ đồ xử lý mô hình thanh toán đề xuất 1.....	13
Hình 2-5: Sơ đồ xử lý mô hình thanh toán đề xuất 3	22
Hình 3-1: Sơ đồ kiến trúc hệ thống đề xuất	25
Hình 3-2: Sơ đồ kiến trúc cơ sở dữ liệu	27
Hình 3-3: Sơ đồ UseCase phân hệ khách hàng mô hình đề xuất 1	34
Hình 3-4: Sơ đồ UseCase phân hệ nhân viên thanh toán mô hình đề xuất 1	35
Hình 3-5: Sơ đồ UseCase phân hệ khách hàng mô hình đề xuất 2	36
Hình 3-6: Sơ đồ UseCase phân hệ nhân viên thanh toán mô hình đề xuất 2.....	37
Hình 3-7: Sơ đồ UseCase phân hệ khách hàng mô hình thanh toán đề xuất 3 khi có Internet	39
Hình 3-8: Sơ đồ UseCase phân hệ khách hàng mô hình thanh toán đề xuất 3 khi người dùng không có Internet.	39
Hình 3-9: Sơ đồ UseCase phân hệ nhân viên thanh toán mô hình thanh toán đề xuất 3	40
Hình 3-10: Mã nguồn lớp DataPackage	42
Hình 3-11: Mã nguồn lớp DataPackageObjectSuccess	42
Hình 3-12: Mã nguồn lớp DataPackageArraySuccess.....	42
Hình 3-13: Mã nguồn lớp DataPackageError	43
Hình 3-14: Mã nguồn lớp Startup (Server)	44
Hình 3-15: Mã nguồn lớp MyHub	44
Hình 3-16: Mã nguồn lớp Starup (Web-Client).....	45
Hình 3-17: Mã nguồn JavaScript trên Web-Client	45
Hình 3-18: Mã nguồn Android khai báo và sử dụng SiganlR	46
Hình 3-19: Mã nguồn Android khai báo hàm cho Server sử dụng trong SignalR....	47
Hình 3-20: Mã nguồn ASP.Net xử lý trùng mã thanh toán	48
Hình 3-21: Mã nguồn JavaScript sử dụng Ajax trên Web-Client.....	48
Hình 3-22: Mã nguồn ASP.Net khai báo hàm sử dụng Ajax.....	49
Hình 3-23: Mã nguồn JavaScript xử dụng dịch vụ tạo QR Code của Google.....	49
Hình 3-24: Mã nguồn Timer nhận hình từ Webcam, giải mã và gửi Mã thanh toán	50
Hình 3-25: Mã nguồn phát hiện 2 thiết bị NFC chạm vào nhau	51
Hình 3-26: Mã nguồn nhận thông điệp từ thiết bị NFC khác gửi tới.....	51
Hình 3-27: Mã nguồn gửi thông điệp tới thiết bị NFC khác.....	52
Hình 3-28: Cấu hình ứng dụng WinForm Shop trong Registry.....	52
Hình 3-29: Mã nguồn gửi thông điệp từ WinForm Shop tới trình duyệt web	53
Hình 3-30: Mã nguồn một số hàm trong interface IServer	54
Hình 3-31: Mã nguồn ví dụ sử dụng thư viện Gson	55

Hình 3-32: Mã nguồn khai báo GCM trong AndroidManifest.xml.....	56
Hình 3-33: Mã nguồn sử dụng thư viện GCM.....	57
Hình 3-34: Màn hình lấy API Key cho Google Maps API.....	57
Hình 3-35: Mã nguồn khai báo Google Maps API trong AndroidManifest.xml.....	58
Hình 3-36: Mã nguồn Thêm Fragment Google Maps trong layout XML file.....	58
Hình 3-37: Cú pháp gọi hàm chỉ đường của Google API.....	58
Hình 3-38: Mã nguồn lớp chỉ đường GMapV2Direction	59
Hình 3-39: Mã nguồn lớp CryptoHelper hỗ trợ cho việc mã hóa gói tin.....	59
Hình 3-40: Màn hình mô tả tính năng Pull To Refresh	60
Hình 3-41: Mã nguồn mô tả sự kiện setOnRefreshListener của lớp PullToRefreshListView.....	60
Hình 3-42: Mã nguồn ví dụ sử dụng thư viện SlidingMenu	61
Hình 3-44: Mã nguồn khai báo thư viện Zxing trong AndroidManifest.xml	61
Hình 3-45: Mã nguồn đọc, tạo mã QRCode sử dụng thư viện Zxing	62
Hình 3-43: Màn hình ClearableEditText	62
Hình 3-46: Mã nguồn khai báo sử dụng NFC trong AndroidManifest.xml	63
Hình 3-47: Mã nguồn kích hoạt sử dụng NFC và gói thông điệp gửi thông qua NFC	63
Hình 3-48: Mã nguồn nhận thông điệp từ thiết bị khác thông qua NFC	64
Hình 4-1: Màn hình xác lập mã pin ban đầu.....	66
Hình 4-2: Màn hình nhập mã pin	66
Hình 4-3: Màn hình đăng ký tài khoản	67
Hình 4-4: Màn hình đăng nhập trên ứng dụng di động.....	67
Hình 4-5: Màn hình quên mật khẩu	68
Hình 4-6: Màn hình chính của mô hình thanh toán 1	68
Hình 4-7: Màn hình chính của mô hình thanh toán 2	69
Hình 4-8: Màn hình chính của mô hình thanh toán 2	69
Hình 4-9: Màn hình xem lịch sử giao dịch và chi tiết lịch sử giao dịch	70
Hình 4-10: Màn hình xem danh sách cửa hàng.....	70
Hình 4-11: Màn hình bản đồ cửa hàng xung quanh và hướng dẫn chỉ đường.....	71
Hình 4-12: Màn hình thực tại tăng cường	71
Hình 4-13: Màn hình Slider Menu	72
Hình 4-14: Màn hình quản lý tài khoản và đổi mật khẩu	72
Hình 4-15: Màn hình quản lý ví điện tử và thêm ví điện tử.....	73
Hình 4-16: Màn hình quản lý thẻ thanh toán và thêm thẻ thanh toán.....	73
Hình 4-17: Màn hình cài đặt, trợ giúp, giới thiệu	74
Hình 4-18: Màn hình đăng nhập Web Shop	75
Hình 4-19: Màn hình chính của Web Shop trong mô hình thanh toán 1	76
Hình 4-20: Màn hình chính của Web Shop trong mô hình thanh toán 2	76
Hình 4-21: Màn hình chính của Web Shop trong mô hình thanh toán 3	77

Hình 4-22: Màn hình xem lịch sử giao dịch Web Shop.....	78
Hình 4-23: Màn hình đăng quảng cáo tới smartphone khách hàng - Web Shop	78
Hình 4-24: Màn hình chính WinForm Shop	79
Hình 4-25:Hình mô tả bước khách hàng đem áo sơ mi tới quầy tính tiền.....	79
Hình 4-26: Hình mô tả bước nhân viên thanh toán hỏi khách hàng loại hình thanh toán	80
Hình 4-27: Hình mô tả mô hình thanh toán 1, bước nhân viên thanh toán hỏi khách hàng mã thanh toán	80
Hình 4-28: Hình mô tả mô hình thanh toán 1, khách hàng lấy mã thanh toán và đọc cho nhân viên thanh toán.....	81
Hình 4-29: Hình mô tả mô hình thanh toán 1, nhân viên thanh toán nhập mã thanh toán và gửi lên Server.....	82
Hình 4-30: Hình mô tả mô hình thanh toán 1, khách hàng xác nhận hóa đơn thanh toán	82
Hình 4-31: Hình mô tả bước Server thông báo thanh toán thành công về nhân viên thanh toán và khách hàng	83
Hình 4-32: Hình mô tả bước nhân viên thanh toán thông báo cho khách hàng giao dịch đã hoàn tất	83
Hình 4-33: Hình mô tả mô hình thanh toán 2, nhân viên thanh toán đọc mã thanh toán cho khách hàng.....	84
Hình 4-34: Hình mô tả mô hình thanh toán 2, khách hàng nhập mã thanh toán và xác nhận hóa đơn thanh toán	85
Hình 4-35: Hình mô tả mô hình thanh toán 3, khách hàng nhập số tiền, xác nhận hóa đơn thanh toán và lấy mã thanh toán	86
Hình 4-36: Hình mô tả mô hình thanh toán 3, chương trình WinForm Shop tự động đọc mã QRCode hoặc thông qua NFC và ghi vào trong trình duyệt web	86
Hình 4-37:Kết quả khảo sát ý kiến người dùng mô hình thanh toán 1.	88
Hình 4-38:Kết quả khảo sát ý kiến người dùng mô hình thanh toán 2.	89
Hình 4-39:Kết quả khảo sát ý kiến người dùng mô hình thanh toán 3.	90
Hình 4-40: Kết quả khảo sát mức độ khả thi giữa các mô hình.....	91

DANH MỤC BẢNG

Bảng 3-1: Thông tin mô tả chi tiết cấu trúc bảng dữ liệu UserProfile	27
Bảng 3-2: Thông tin mô tả chi tiết cấu trúc bảng dữ liệu CustomersProfile	28
Bảng 3-3: Thông tin mô tả chi tiết cấu trúc bảng dữ liệu CashierProfile.....	29
Bảng 3-4: Thông tin mô tả chi tiết cấu trúc bảng dữ liệu Bank	29
Bảng 3-5: Thông tin mô tả chi tiết cấu trúc bảng dữ liệu BankInfo	30
Bảng 3-6: Thông tin mô tả chi tiết cấu trúc bảng dữ liệu WalletDetail	31
Bảng 3-7: Thông tin mô tả chi tiết cấu trúc bảng dữ liệu Shop	32
Bảng 3-8: Thông tin mô tả chi tiết cấu trúc bảng dữ liệu Bill	33
Bảng 3-9: Bảng chi tiết thị phần các nền tảng di động trên thị trường smartphone	54
Bảng 4-1: Tổng kết số liệu khảo sát 3 mô hình thanh toán.	91

DANH MỤC CÁC THUẬT NGỮ, TỪ VIẾT TẮT

Chữ cái viết tắt/thuật ngữ	Cụm từ đầy đủ/ý nghĩa.
API	Application Programming Interface - Giao diện lập trình ứng dụng
AR	Augmented Reality – Thực tại ảo tăng cường
ATM	Automatic Teller Machine – Máy giao dịch tự động
Cashier	Nhân viên thanh toán
Client	Máy khách hàng
Credit	Thẻ tín dụng
Customer	Khách hàng
Debit	Thẻ ghi nợ
Encrypt info	Gói thông tin đã được mã hóa
Framework	Một nền tảng lập trình và cũng là một nền tảng thực thi ứng dụng
GPS	Global Positioning System
Mobile Payment	Thanh toán thông qua thiết bị di động
NFC	Near Field Communication – Công nghệ giao tiếp tầm gần
NuGet	Một công cụ giúp bạn thêm các thư viện ngoài vào dự án.
OTP	One Time Password
PIN	Personal Identification Number: là mã số định danh cá nhân
POS	Point of Sale
QR code	Quick Response Code
Realtime	Thời gian thực

Server	Máy chủ
SMS	Short Message Services (Dịch vụ tin nhắn ngắn)
Store Procedure	Một nhóm câu lệnh Transact-SQL đã được compiled (biên dịch) và chứa trong SQL Server

TÓM TẮT KHÓA LUẬN

1. Mục tiêu đề tài:

Xây dựng một hệ thống thanh toán thông minh, hiệu quả, an toàn và nhanh chóng bằng cách sử dụng thiết bị di động như một công cụ thanh toán.

2. Tính mới và sáng tạo:

Thay đổi cách thức thanh toán truyền thống bằng một phương thức thanh toán mới với sự hiện đại và tính an toàn cao hơn.

Áp dụng được các công nghệ tiên tiến như QR Code, công nghệ giao tiếp tầm gần (Near Field Communication - NFC) nhằm đơn giản và bảo mật hóa quá trình thanh toán.

Áp dụng thực tại tăng cường vào việc tìm kiếm cửa hàng xung quanh người dùng.

Áp dụng Google Cloud Message để nhắn (push) trực tiếp tin khuyến mãi đến điện thoại người dùng một cách nhanh chóng nhất.

3. Kết quả nghiên cứu:

Nghiên cứu, phân tích ưu nhược điểm của nhiều mô hình thanh toán đã và đang được sử dụng trên thế giới có liên quan trực tiếp đến đề tài.

Xây dựng được 3 mô hình thanh toán khác nhau dựa trên kiến trúc hệ thống được thiết kế tối ưu.

4. Đóng góp về mặt kinh tế - xã hội, giáo dục - đào tạo, an ninh, quốc phòng và khả năng áp dụng của đề tài:

Việc áp dụng công nghệ hiện đại cùng với quy trình thanh toán mới làm đơn giản hóa quá trình thanh toán, mọi quá trình giao dịch được diễn ra nhanh chóng. Điều này góp phần tạo động lực thúc đẩy phát triển nền kinh tế Nước nhà.

Sử dụng cách thức thanh toán mới, không dùng tiền mặt, góp phần hạn chế rủi ro đánh mất tiền, cướp giật, móc túi,... Nhằm góp phần gìn giữ trật tự và an toàn xã hội đồng thời xây dựng một xã hội văn minh.

Hiện tại, phương thức thanh toán không dùng tiền mặt đã được phát triển rất mạnh tại nhiều nước, được các chính phủ khuyến khích sử dụng và được rất nhiều tổ chức áp dụng. Ở Việt Nam, với điều kiện thị trường di động tăng trưởng mạnh như ở thời điểm này thì việc thanh toán dựa trên điện thoại di động sẽ là một kênh thanh toán không dùng tiền mặt có nhiều tiềm năng để phát triển, có thể là xu hướng trong thời gian sắp tới.

MỞ ĐẦU

Thanh toán của khách hàng trên thiết bị di động sẽ là một xu hướng trong tương lai. Nhiều giải pháp, thiết bị, hạ tầng cho lĩnh vực thanh toán dựa trên Mobile Payment được phát triển và giới thiệu, nhiều mô hình kinh doanh Mobile Payment được đưa ra, tạo ra sự đa dạng trong việc cung cấp dịch vụ. Phương thức thanh toán không dùng tiền mặt đã được phát triển rất mạnh tại nhiều nước, được các Chính phủ khuyến khích sử dụng và được rất nhiều tổ chức áp dụng, do đó việc thanh toán dựa trên điện thoại di động sẽ là một kênh thanh toán không dùng tiền mặt có nhiều tiềm năng để phát triển, có thể là xu hướng trong thời gian tới. Việt Nam có thể coi là thị trường có tiềm năng rất lớn để khai thác các dịch vụ thanh toán di động.

Về mặt pháp lý, Chính phủ đã ban hành Luật thanh toán điện tử và có những nghị định và văn bản khung hướng dẫn thực hiện dịch vụ thanh toán.

Từ những phân tích trên, tác giả thấy rằng dịch vụ thanh toán di động có rất nhiều tiềm năng và điều kiện thuận lợi để phát triển, có thể nói là rất khả thi tại Việt Nam trong thời điểm hiện tại và cần được nghiên cứu chuyên sâu hơn để có thể ứng dụng rộng rãi trong đời sống. Tuy nhiên, vẫn còn đó những vấn đề cần được nghiên cứu và giải quyết làm cho cho hệ thống thanh toán di động an toàn hơn, thông minh hơn, hiệu quả hơn và có thể thay thế các loại thanh toán khác. Tác giả thấy rằng việc chọn đề tài **“NGHIÊN CỨU VÀ CÀI ĐẶT GIẢI PHÁP MOBILE PAYMENT VỚI THỦ NGHIỆM TRÊN NỀN TẢNG ANDROID”** để nghiên cứu sẽ là hướng đi đúng đắn và thiết thực, có tính khả thi cao, có ý nghĩa về mặt lý thuyết và ứng dụng thực tế.

Trong khuôn khổ và giới hạn của đề tài, tác giả chỉ tập trung phân tích, tìm hiểu những mô hình và giải pháp Mobile Payment đã và đang hoạt động, để tìm ra được những điểm mạnh, điểm yếu và hạn chế của mỗi giải pháp, từ đó đề xuất và lựa chọn giải pháp phù hợp nhất, có tính khả thi cao nhất đối với hệ thống Mobile Payment tại thị trường Việt Nam.

Chương 1: THANH TOÁN DI ĐỘNG – MỘT XU HƯỚNG PHÁT TRIỂN CỦA PHƯƠNG THỨC THANH TOÁN

Cùng với sự phát triển về mặt kinh tế thì nhu cầu mua sắm của người dân ngày càng tăng cao. Tính riêng quý I năm 2014, tổng mức bán lẻ hàng hóa và doanh thu dịch vụ tiêu dùng ước tính đạt 701,4 nghìn tỷ đồng, tăng 10,2% so với cùng kỳ năm trước [14]. Điều đó cho thấy số lượng tiền tệ lưu thông thông qua kênh bán lẻ là rất lớn và ngày càng tăng. Vì vậy, một hình thức thanh toán giao dịch nhanh chóng, hiệu quả, thuận tiện và an toàn cho người sử dụng là cần thiết và cấp bách. Mặc dù các phương thức giao dịch hiện tại phần nào đáp ứng được nhu cầu nhưng còn tồn tại nhiều khuyết điểm khiến quá trình thanh toán gặp khó khăn.

Để làm rõ thêm vấn đề cần quan tâm, các phương thức thanh toán giao dịch chính đang tồn tại trên thị trường (ví dụ: thanh toán bằng tiền mặt, thanh toán bằng thẻ ghi nợ nội địa, thanh toán online, ...) sẽ được phân tích một cách chi tiết.

1.1. Thanh toán bằng tiền mặt.

Thanh toán bằng tiền mặt là việc tổ chức, cá nhân sử dụng tiền mặt để trực tiếp chi trả hoặc thực hiện các nghĩa vụ trả tiền khác trong các giao dịch thanh toán.

Ưu điểm:

- Thanh toán bằng tiền mặt là phương thức thanh toán đơn giản và tiện dụng nhất được sử dụng để mua bán hàng hoá một cách dễ dàng.
- Tuy nhiên, nó chỉ phù hợp với nền kinh tế có quy mô sản xuất nhỏ, sản xuất chưa phát triển hoặc việc trao đổi thanh toán hàng hoá diễn ra với số lượng nhỏ, trong phạm vi hẹp. Vì vậy, khi nền kinh tế ngày một phát triển với tốc độ cao cả về chất lượng và số lượng thì việc thanh toán bằng tiền mặt không còn đủ khả năng đáp ứng được những nhu cầu thanh toán của toàn bộ nền kinh tế. Việc thanh toán bằng tiền mặt đã bộc lộ một số hạn chế nhất định.

Hạn chế:

- Việc thanh toán bằng tiền mặt có độ an toàn không cao. Cụ thể hơn, khi thực hiện giao dịch cho một khối lượng hàng hóa hoặc dịch vụ lớn thì việc thanh toán trực tiếp bằng tiền mặt sẽ không an toàn và thuận tiện cho cả người chi trả và người thụ hưởng.
- Ngân hàng Nhà nước cần phải bỏ ra chi phí rất lớn để in ấn, vận chuyển và bảo quản tiền mặt.
- Một hạn chế quan trọng của việc thanh toán bằng tiền mặt là thanh toán bằng tiền mặt làm giảm khả năng tạo tiền của ngân hàng thương mại.

Tình hướng thanh toán phù hợp:

- Thanh toán nhỏ lẻ, giao dịch dân dụng trực tiếp.

Theo thống kê của Ngân hàng Nhà nước, tỷ lệ sử dụng tiền mặt trên tổng phương tiện thanh toán đang có xu hướng giảm dần từ 20,3% năm 2004 xuống 14% năm 2010 và hiện còn khoảng 12%. Có hơn 65% đơn vị thực hiện chi trả lương qua tài khoản cho đến năm 2013 [15]. Từ thực tế khách quan đó, sự ra đời của một phương thức thanh toán khác tiên tiến hơn, hiện đại hơn đó là phương thức thanh toán không dùng tiền mặt là rất phù hợp với xu thế.

1.2. Thanh toán bằng Thẻ Ghi Nợ Nội Địa

Thẻ ghi nợ nội địa thường được gọi là thẻ ATM là một loại thẻ theo chuẩn ISO 7810, bao gồm thẻ ghi nợ và thẻ tín dụng, dùng để thực hiện các giao dịch tự động như kiểm tra tài khoản, rút tiền hoặc chuyển khoản, thanh toán hóa đơn, mua thẻ điện thoại v.v. từ máy rút tiền tự động (ATM). Loại thẻ này cũng được chấp nhận như một phương thức thanh toán không dùng tiền mặt tại điểm thanh toán có chấp nhận thẻ.

Thẻ ngân hàng là phương tiện thanh toán đa dụng, tiện ích, được các tổ chức tính dụng chú trọng phát triển. Đến cuối năm 2013, số lượng thẻ phát hành đạt trên 66 triệu thẻ (tăng 22% so với cuối năm 2012 và tăng 60% so với cuối năm 2011); số lượng và giá trị giao dịch thẻ tăng lần lượt 25% và 43 % so với năm 2012 [15].

Ưu điểm:

- Tạo sự minh bạch trong các khoản chi tiêu của các đơn vị kinh doanh và cá nhân. Từ đó, dòng chảy tiền tệ được lưu thông rõ ràng và trơn tru hơn.
- Khác với hình thức thanh toán bằng tiền mặt, người dùng không cần mang theo tiền mặt khi giao dịch nên tránh được sự mất mát không mong muốn hoặc sự nhầm lẫn trong quá trình kiểm tra lượng tiền giao dịch. Việc thanh toán bằng thẻ có thể được thực hiện mọi lúc, mọi nơi.
- Số tiền duy trì trên tài khoản để thanh toán được hưởng lãi suất không kỳ hạn.

Bên cạnh nhiều lợi ích và thuận tiện mà thanh toán bằng thẻ mang lại thì nó cũng chứa đựng không ít khó khăn và bất tiện cho người dùng.

Hạn Chế:

- Một số loại thẻ cho phép thanh toán không cần xác nhận mã pin, do đó khi người dùng đánh mất thẻ vì lý do gì đó như không biết mình bị mất thẻ hay chưa thông báo đóng tài khoản tạm thời, kẻ xấu có thể thực hiện giao dịch thông qua thẻ đó mà chủ nhân thẻ không hề hay biết.
- Trường hợp người dùng lộ thông tin thẻ, kẻ xấu có thể lợi dụng sự khinh suất của chủ sở hữu thẻ để thực hiện các khoản thanh toán không mong muốn.
- Một người có thể cùng lúc sở hữu nhiều thẻ nên việc quản lý thông tin (mã PIN) của các thẻ thanh toán cũng là một vấn đề cho người dùng.

Tình huống thanh toán phù hợp:

- Thanh toán trực tiếp với điều kiện tại cơ sở thanh toán có hỗ trợ máy POS để thực hiện thanh toán.

Đến nay, cả nước có 52 tổ chức đăng ký phát hành thẻ, với khoảng 66 triệu thẻ đã được phát hành (trên 90% là thẻ ghi nợ nội địa); mạng lưới trên 15.200 ATM và trên 129.600 POS/EDC được lắp đặt. Tuy nhiên, theo đánh giá của một số chuyên gia, thị trường thẻ Việt Nam đang dần tiến tới bão hoà khi mà tốc độ tăng số lượng thẻ giảm rõ rệt. Nếu như giai đoạn 2010-2012, số lượng thẻ tăng trưởng trung bình

khoảng 30% thì đến năm 2013, chỉ tăng trên 4%, mặc dù các ngân hàng vẫn tiếp tục triển khai nhiều chương trình ưu đãi phát hành thẻ mới [15].

1.3. Thanh toán Online

Song song với sự phát triển nhanh chóng và mạnh mẽ của thương mại điện tử là sự phát triển của các cổng thanh toán trực tuyến. Thanh toán trực tuyến được đánh giá là một mô hình thanh toán hiện đại, mang lại nhiều lợi ích cho các bên.

Thanh toán trực tuyến là hành động thực hiện thanh toán cho các giao dịch thương mại điện tử, như giao dịch trên internet, điện thoại di động.

Mô hình thanh toán trực tuyến hoạt động theo hình thức:

- Người mua sẽ trả cho người bán 1 khoản tiền theo như thỏa thuận, thông qua các dịch vụ thanh toán trực tuyến (ví dụ: ví điện tử,...). Nhìn chung, các phương thức thanh toán trực tuyến khá tương đồng nhau nên việc thanh toán bằng ví điện tử sẽ được tập trung phân tích.
- Sau khi đã xác nhận thanh toán của người mua, người bán sẽ chuyển hàng đến cho người mua theo như thỏa thuận.

Dịch vụ Ví điện tử là dịch vụ cung cấp cho khách hàng một tài khoản điện tử định danh do các tổ chức cung ứng dịch vụ tạo lập trên một vật mang tin (như chip điện tử, sim điện thoại di động, máy chủ...), cho phép lưu trữ một giá trị tiền tệ được đảm bảo bằng giá trị tiền mặt tương đương và được sử dụng để thanh toán thay thế cho tiền mặt.

Đến nay, ngân hàng nhà nước đã cho phép 9 tổ chức không phải là ngân hàng thực hiện thí điểm dịch vụ ví điện tử thông qua 33 tổ chức tín dụng. Đến 31/12/2013, tổng số ví điện tử phát hành đạt trên 1,84 triệu; lượng giao dịch đạt 45,3 triệu với giá trị 23.350 tỷ đồng [15].

Ưu điểm:

- Không giới hạn phạm vi, không gian và thời gian.
- Trong khi chủ thẻ có khả năng mất tất cả tiền trong tài khoản nếu mất thẻ hay lộ thông tin thẻ thì với ví điện tử người dùng chỉ mất số tiền người dùng chủ động nạp vào ví từ trước.

- Bên cạnh đó, việc thanh toán bằng ví điện tử còn nhiều khâu xác nhận thanh toán nên độ an toàn cao.

Cũng như các hình thức thanh toán khác, bên cạnh những ưu điểm nổi bật thì thanh toán bằng ví điện tử cũng tồn tại những hạn chế.

Hạn Chế:

- Quy trình thanh toán bằng ví điện tử còn khá phức tạp. Trước tiên, người dùng cần phải tạo ví điện tử. Tiếp đến, ví điện tử cần được nạp tiền trước khi sử dụng. Khi thực hiện thanh toán người dùng cần trải qua nhiều bước để hoàn thành quá trình thanh toán.
- Tiếp theo phải kể đến là do cơ sở vật chất kỹ thuật ở nước ta còn hạn chế, những Hacker có thể xâm nhập để chống phá gây tổn thất cho doanh nghiệp và người tiêu dùng. Vì vậy, tâm lý nhiều người tiêu dùng còn chưa tin tưởng.
- Đây là mô hình mới ở Việt Nam, các tổ chức cung cấp ví điện tử còn gặp phải nhiều khó khăn khi phát triển cộng đồng cá nhân sở hữu ví.
- Người dùng cần có kiến thức nhất định về tin học và internet thì mới có thể sử dụng giao dịch bằng ví điện tử. Tuy nhiên, một phần không nhỏ người dân hiện nay còn thiếu kiến thức này do đó tập người sử dụng còn bị hạn chế.
- Không thể giúp người dùng thực hiện các giao dịch thường nhật như giao dịch ở các cửa hàng hay ở siêu thị...

Tình huống thanh toán phù hợp:

- Thanh toán trực tuyến giữa hai bên chấp nhận thanh toán bằng ví điện tử

Mặc dù cơ sở hạ tầng phục vụ cho việc thanh toán trực tuyến đã đáp ứng được những yêu cầu cần thiết nhưng việc thanh toán trực tuyến ở Việt Nam vẫn chưa thay thế được hình thức thanh toán truyền thống do thói quen của người tiêu dùng. Phần lớn khách hàng vẫn có thói quen sử dụng tiền mặt hơn là việc sử dụng thẻ/tài khoản ngân hàng mua sắm trực tuyến. Nhưng có thể nói rằng thanh toán qua thẻ tín dụng, thẻ ATM, ví điện tử sẽ là một xu hướng tất yếu. Việc xuất hiện hàng loạt những công thanh toán trực tuyến đã thể hiện rõ được xu hướng cũng như tiềm năng của phương thức thanh toán này.

1.4. Phát biểu bài toán.

Từ những phân tích trên cho chúng ta thấy, một mô hình thanh toán tận dụng được những ưu điểm của thanh toán bằng thẻ và ví điện tử và giảm thiểu những hạn chế của các hình thức thanh toán này là rất cần thiết. Mô hình thanh toán đó phải giúp người sở hữu thẻ không phải lo về việc mất hay lộ thông tin thẻ, người sở hữu ví điện tử có thể thanh toán các hóa đơn nhỏ lẻ, trực tiếp hằng ngày một cách linh hoạt hơn, không còn bó hẹp trong việc chỉ thanh toán các hóa đơn trực tuyến.

Trong thời gian gần đây, cùng với sự phát triển vượt bậc về mặt công nghệ thì việc sử dụng các thiết bị di động trở nên ngày càng phổ biến hơn. Chính việc sử dụng thiết bị di động một cách phổ biến đã mở ra xu hướng cho một phương thức thanh toán mới: thanh toán di động. Việc thực hiện một giao dịch thanh toán trong đó có ít nhất một bên liên quan sử dụng thiết bị di động được gọi là thanh toán di động [6].

Việc thanh toán di động mang lại rất nhiều lợi ích khác nhau. Đầu tiên, việc thanh toán bằng di động có thể giúp người dân thanh toán các giao dịch thương mại mà không cần mang theo tiền mặt hay thẻ thanh toán. Từ đó, các rủi ro về mất tiền hay mất thẻ có thể được giảm thiểu. Ngoài ra, quá trình thanh toán diễn ra nhanh chóng, chính xác nên chất lượng dịch vụ sẽ được nâng cao. Việc cung cấp một kênh thanh toán dễ dàng, nhanh chóng và thân thiện cho người dân sẽ kích thích nhu cầu mua sắm của người dân. Việc thúc đẩy nhu cầu mua sắm sẽ là đòn bẩy cho sự phát triển của nền kinh tế.

Theo đánh giá của các chuyên gia trong hội nghị Banking Vietnam 2011 thì thanh toán di động hiện vẫn chưa có nhiều “đất sống”. Và đến nay, hình thức này vẫn đang đứng bóng ở thị trường Việt Nam. Ý kiến của nhiều chuyên gia cũng cho rằng lý do khiến thanh toán di động khó phát triển là do khách hàng tại Việt Nam chưa tin tưởng vào tính an toàn và bảo mật của phương thức thanh toán di động.

Do đó, việc xây dựng mô hình thanh toán di động có thể đảm bảo được tính an toàn và bảo mật khi thực hiện giao dịch là cần thiết và có ý nghĩa thực tiễn để nâng cao chất lượng cuộc sống của người dân. Vì vậy nhóm xin được giới thiệu mô hình thanh toán thông qua điện thoại di động, được gọi là “**viPay**”.

Chương 2: CÁC MÔ HÌNH THANH TOÁN DI ĐỘNG

Nội dung chương này đề cập đến việc phân tích các mô hình hiện có ở Việt Nam và trên thế giới. Bên cạnh đó đề xuất mô hình thanh toán mới.

2.1. Phân tích các mô hình.

2.1.1. Mô Hình MoPay

Mopay là một công ty hàng đầu trong lĩnh vực cung cấp các giải pháp thanh toán cho các nhà cung ứng trực tuyến [8]. Nền tảng cốt lõi của Mopay cho phép các nhà cung ứng hàng hóa ảo, hàng hóa số hoặc hàng hóa vật lý có thể tính tiền hàng hóa trực tiếp vào hóa đơn điện thoại của khách hàng (điện thoại cố định, điện thoại di động). Mopay hiện đang được triển khai hơn 80 quốc gia khác nhau trên thế giới và lượng khách hàng đang sử dụng là hơn 4.3 tỉ khách hàng.

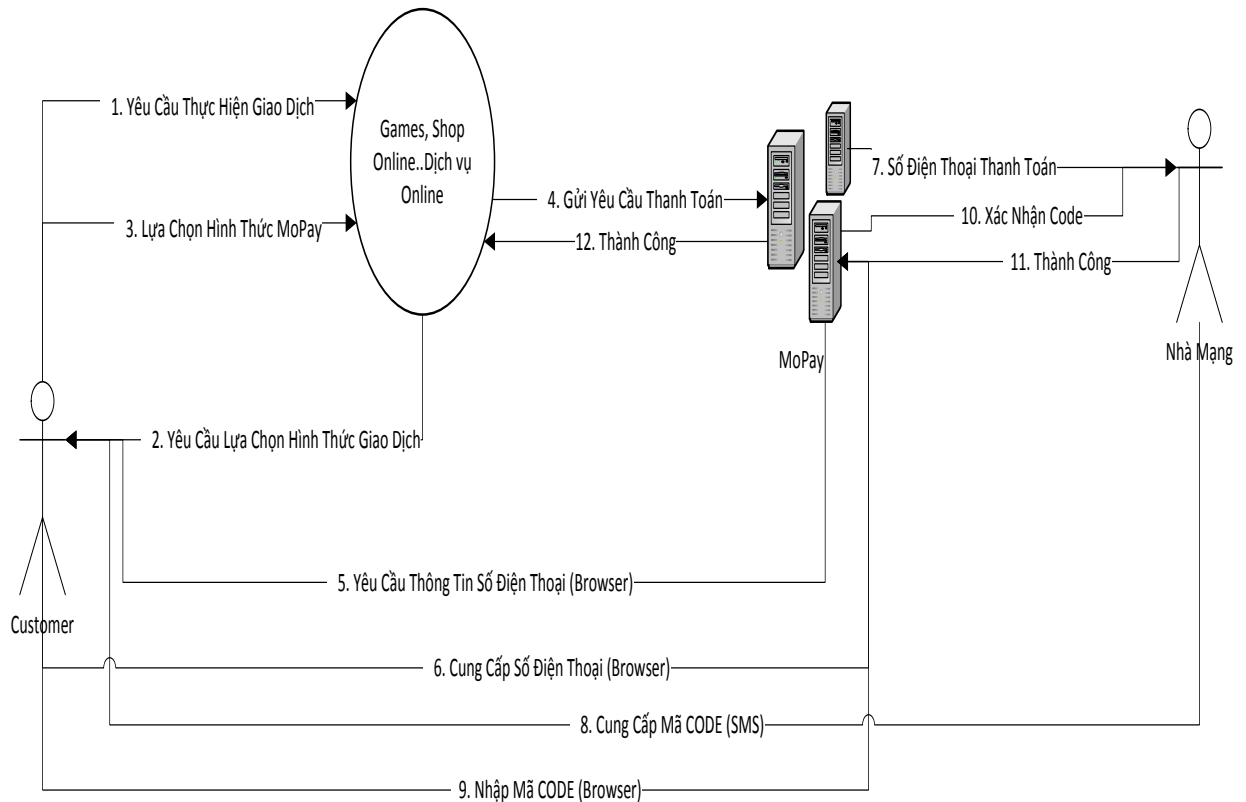
Mopay là một thành viên của tập đoàn MindMatics. Mopay hiện có hơn 100 nhân viên làm việc ở Mỹ, Đức, Áo, Tây Ban Nha, Vương quốc Anh, Hàn Quốc, Việt Nam, Indonesia, Trung Quốc, Rumani và Braxin

Vừa qua vào ngày 29/4/2014 Mopay cũng đã bước đầu bước chân vào thị trường Việt Nam thông qua việc ký hợp đồng hợp tác cùng Tập Đoàn Viễn Thông Quân Đội – Viettel.

Mô hình MoPay xây dựng kênh trung gian thực hiện giao dịch trực tuyến và tạo cầu nối để hai bên giao dịch thực hiện nhu cầu của mình một cách thuận tiện hơn. Cụ thể hơn, MoPay cung cấp một gói dịch vụ mà theo đó khách hàng chỉ cần dùng thuê bao di động của nhà mạng mà MoPay đã hợp tác (cụ thể ở Việt Nam là Viettel) là có thể tiến hành giao dịch. Cuối tháng, khách hàng chỉ cần thanh toán hóa đơn nhà mạng theo qui định.

Hiện tại MoPay có 3 hình thức thanh toán chính là: Direct Carrier Billing, Landline Billing, BroadBand Billing. Nhưng ở thị trường Việt Nam MoPay chỉ triển khai hình thức Carrier Billing cùng với Viettel.

Dựa vào thông tin từ website chính thức từ MoPay ta có thể mô tả mô hình thanh toán Carrier Billing của MoPay như sau:



Hình 2-1: Sơ đồ mô phỏng quy trình xử lý mô hình MoPay

Ưu Điểm:

- Xét về mặt tương tác, khách hàng chỉ thực hiện 2 bước căn bản là nhập số điện thoại, và nhập số Code là có thể thực hiện thanh toán lúc giao dịch nên có thể nói quá trình thanh toán về mặt bề nổi rất đơn giản, nhanh chóng.
- Bên cạnh đó đứng vai trò là trung gian thực hiện giao dịch, MoPay không cần quan tâm đến tài khoản của khách hàng, cách thức chuyển tiền như thế nào, mọi hoạt động liên quan đến tiền từ mặt giao dịch tiền với bên thứ ba cũng như là cách thức thu tiền đều do bên Nhà Mạng (Viettel) thực hiện.

Hạn Chế:

- Như đã nói đề cập, giao dịch diễn ra đơn giản, nhanh chóng. Nhưng bên cạnh đó để có thể thực hiện giao dịch như vậy khách hàng cần phải thực hiện một số yêu cầu bắt buộc như: thuê bao phải là thuê bao trả sau, hằng tháng phải tiến hành thanh toán tại trung tâm giao dịch của Nhà Mạng hay bằng một cách thức nào đó. Nên xét về toàn bộ quá trình thanh toán thì mô hình không đơn giản như bề nổi nó thể hiện.

- Để thực hiện giao dịch với MoPay bên cạnh việc người dùng cần sở hữu thuê bao của Nhà Mạng (Viettel) thì khách hàng phải thực hiện một bản hợp đồng sử dụng dịch vụ (đối với thuê bao trả trước) hay khách hàng buộc phải chuyển sang dùng thuê bao trả sau. Như vậy tồn tại hai vấn đề lớn ở đây: *i)* khách hàng phải bắt buộc phải dùng thuê bao của Nhà Mạng cố định; *ii)* phải thực hiện thủ tục đăng ký sử dụng dịch vụ. Do đó, tập người sử dụng dịch vụ bị thu hẹp khá nhiều.
- Cũng như mô hình thanh toán bằng Ví Điện Tử thì mô hình MoPay cũng tồn tại hai hạn chế là: khách hàng cần trang bị kiến thức nhất định về tin học và internet.

2.1.2. Mô Hình PayPass + PayPass Wallet + NFC Phone.

PayPass là một loại hình thanh toán bằng thẻ "không tiếp xúc" (contactless) của MasterCard dựa trên tiêu chuẩn ISO/IEC 14443 (Về các loại thẻ nhận dạng - thẻ tích hợp mạch không tiếp xúc - thẻ tiệm cận) và sử dụng tương thích với tiêu chuẩn EMV (Europay, MasterCard & VISA) [7]. Với PayPass, chủ thẻ có thể chi trả bằng cách chạm thẻ hoặc thiết bị chi trả như điện thoại vào một thiết bị đọc đầu cuối thay vì cào thẻ hay đút thẻ vào máy. Tuy nhiên, ngoài việc thiết lập hàng loạt các thiết bị đầu cuối với nhiều thương hiệu thanh toán tại các cửa hàng trên toàn lãnh thổ Hoa Kỳ, MasterCard còn muốn mở rộng dịch vụ của mình bằng cách biến những chiếc điện thoại thành thiết bị chi trả trong lòng hơn. Để làm được điều này, MasterCard cho ra đời dịch vụ PayPass Wallet.

PayPass Wallet là một hệ thống thanh toán trực tuyến với API cho phép lập trình viên tích hợp PayPass vào dịch vụ, tiện ích mà họ cung cấp. PayPass Wallet Services bao gồm 3 thành phần tách biệt là PayPass Acceptance Network (mạng lưới chấp nhận thanh toán PayPass quốc tế gồm PayPass Online và PayPass Contactless), PayPass Wallet và PayPass API.

Ưu điểm:

- Sau khi hoàn thành các bước đăng ký và chuẩn bị sẵn thiết bị, quá trình diễn ra rất nhanh chóng và đơn giản, cụ thể là khách hàng chỉ cần mở ứng dụng trên điện thoại di động, áp điện thoại vào thiết bị thanh toán và nghe thông báo bằng âm thanh là quá trình thanh toán hoàn tất.

- Sử dụng công nghệ NFC cùng với gói dịch vụ trọn gói (phần mềm trên thiết bị di động, thiết bị đọc và thanh toán NFC, API hỗ trợ) tạo nên tính bảo mật cao trong quá trình thanh toán.

Hạn chế:

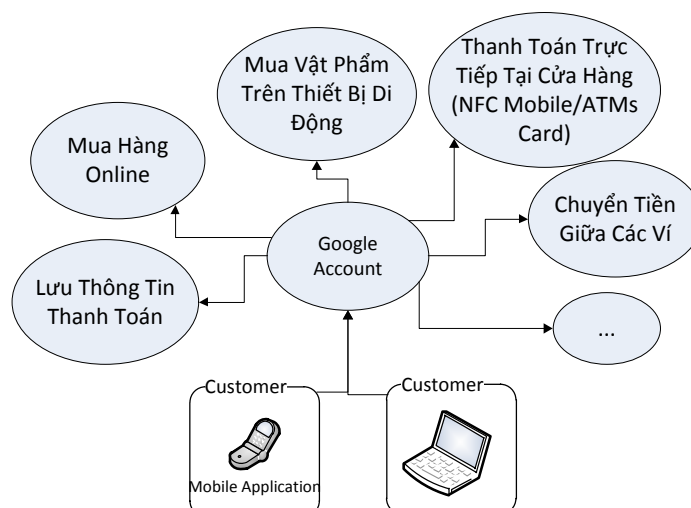
- Mô hình hiện tại chỉ có thể thanh toán với hóa đơn có mệnh giá nhỏ (dưới 20\$).
- Khi sử dụng mô hình, khách hàng phải sở hữu điện thoại di động có hỗ trợ công nghệ NFC, cửa hàng hỗ trợ dịch vụ phải sở hữu thiết bị đọc và thanh toán bằng NFC do MASTERCARD cung cấp.
- Để có thể tiến hành thanh toán, khách hàng phải trải qua nhiều bước: đăng ký trực tiếp với phía ngân hàng hỗ trợ, tải thông tin thẻ thanh toán về điện thoại và kích hoạt thẻ thanh toán.
- Mô hình giới hạn khách hàng trong việc buộc người sử dụng phải sở hữu tài khoản MASTERCARD.

2.1.3. Mô hình Google Wallet.

Google Wallet là dịch vụ do Google cung cấp, cho phép khách hàng lưu trữ thông tin hóa đơn thanh toán, lịch sử giao dịch, thông tin thẻ thanh toán, tài khoản ngân hàng...trong tài khoản Google [4].

Ngoài việc lưu trữ thông tin, Google Wallet còn cho phép khách hàng thanh toán Online tại các trang web mua sắm, tại các ứng dụng trên thiết bị di động, thực hiện chuyển tiền giữa các ví và thanh toán trực tiếp bằng công nghệ NFC hay thẻ ATM do Google cung cấp.

Google Wallet hỗ trợ khách hàng có thể sử dụng nhiều loại tài khoản khác nhau thực hiện giao dịch như: thẻ thanh toán (bao gồm credit hay debit), tài khoản ngân hàng hay ví điện tử do chính Google cung cấp.



Hình 2-2: Sơ đồ mô tả các dịch vụ do Google Wallet cung cấp.

Ưu điểm:

- Đến với Google Wallet khách hàng có thể tiến hành thanh toán bằng nhiều hình thức khác nhau như thanh toán trực tuyến bằng thẻ ATM do Google cung cấp, thanh toán bằng điện thoại hỗ trợ NFC, thanh toán các đơn hàng trực tuyến, thanh toán vật phẩm trên thiết bị di động...
- Với mỗi hình thức thanh toán, Google Wallet cho phép khách hàng lựa chọn tài khoản thanh toán một cách linh hoạt.

Hạn chế:

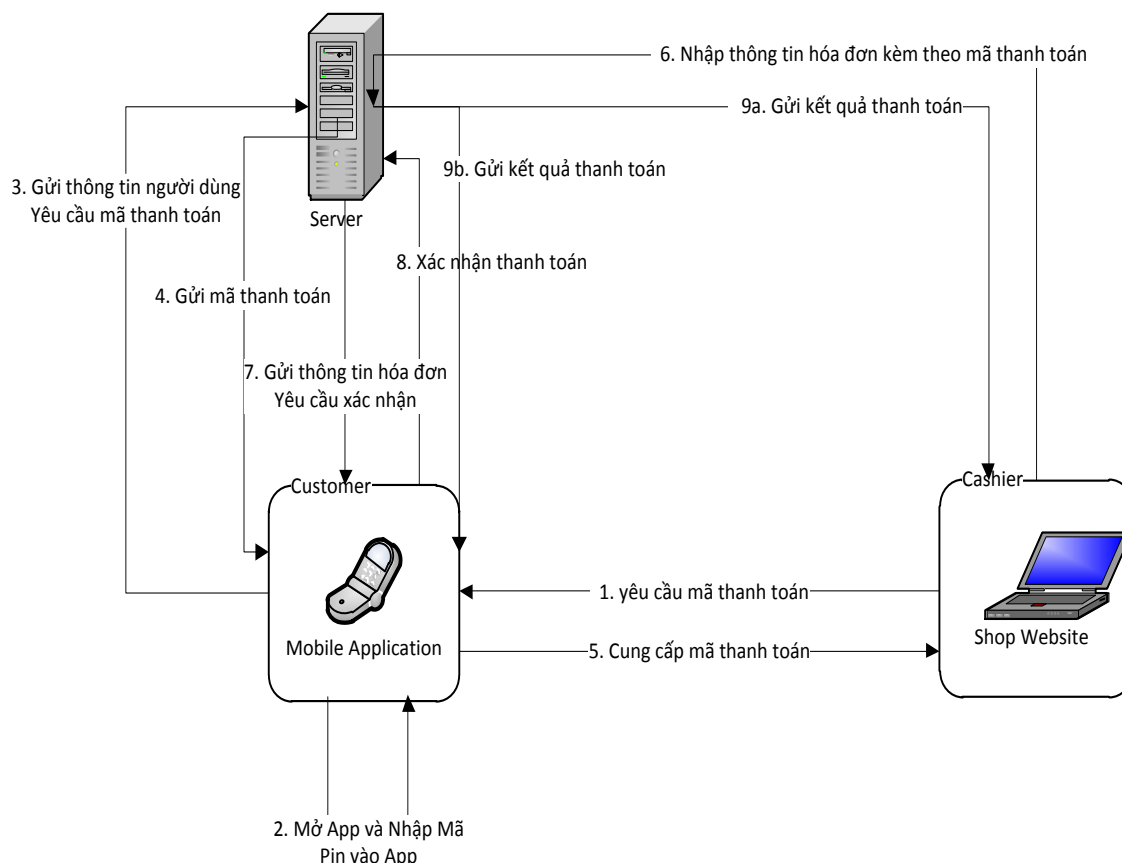
- Hiện tại ở Việt Nam vẫn chưa thể sử dụng ứng dụng Google Wallet.
- Việc quản lý tất cả các dịch vụ của Google mà người dùng sử dụng chỉ thông qua một tài khoản duy nhất, điều đó tuy tiện lợi nhưng mức độ nguy hại sẽ cao khi người dùng đánh mất hay bị lộ thông tin tài khoản.

2.2. Đề xuất các mô hình.

Từ những phân tích trên, nhóm đã rút ra được những ưu nhược điểm và đề xuất một số mô hình thanh toán mới với *mục tiêu* hạn chế tối đa nhược điểm, kế thừa các ưu điểm cùng với bổ sung các tiện ích mới cho người dùng. Từ phân tích cho thấy sự tiện lợi khi thực hiện thanh toán bằng thiết bị di động (điện thoại di động) cũng như là công nghệ NFC giúp cho quá trình thực hiện thanh toán được nhanh chóng, dễ

dàng và an toàn hơn. Việc tích hợp các dịch vụ được cung cấp sẵn từ phía ngân hàng hay dịch vụ ví điện tử cũng là một thuận lợi lớn khi xây dựng mô hình thanh toán...

2.2.1. Mô hình thanh toán 1



Hình 2-3: Sở xử lý mô hình thanh toán đề xuất 1.

Các bước của quá trình thanh toán:

- Bước 1: nhân viên thanh toán yêu cầu mã thanh toán từ Khách hàng.
 - Mã thanh toán: một số 5 chữ số, do Server cung cấp ngẫu nhiên khi khách hàng yêu cầu. Mã thanh toán được phát sinh duy nhất tại một thời điểm, là số định danh cho 1 khách hàng tại một khoảng thời gian nhất định để thực hiện thanh toán. Mã thanh toán này chỉ có hiệu lực trong vòng năm phút, sau năm phút nếu Khách hàng chưa hoàn tất việc thanh toán thì khách hàng phải yêu cầu mã thanh toán mới để thực hiện giao dịch, cùng lúc đó nhân viên thanh toán cũng không thể dùng mã thanh toán này đơn phương thanh toán giao dịch. Với mã thanh toán 5 chữ số, số lượng giao dịch tối đa có thể thực hiện cùng

một lúc là 100.000. Tuy nhiên, khi số giao dịch tăng cao, mã thanh toán vẫn có thể dễ dàng tăng số chữ số lên để đáp ứng nhu cầu.

- Bước 2: Khách hàng mở ứng dụng, và nhập mã PIN.
 - Mã PIN: dùng để đảm bảo độ bảo mật, tránh những thanh toán không mong muốn. Là một số gồm 4 chữ số.
- Bước 3: Khách hàng yêu cầu mã thanh toán.
 - Khi yêu cầu mã thanh toán, khách hàng gửi kèm thông tin người dùng, thông tin người dùng ở đây là tài khoản khách hàng đã đăng ký từ trước và đã đăng nhập ít nhất một lần ở điện thoại thực hiện thanh toán. Ứng dụng sẽ mã hóa thông tin đăng nhập này và lưu lại. Khi Khách hàng gửi yêu cầu mã thanh toán lên server, nếu server kiểm tra chưa thấy khách hàng đăng nhập ở phiên giao dịch này hay đã đăng nhập nhưng SESSION đã quá hạn thì server sẽ yêu cầu ứng dụng gửi lại thông tin người dùng đã được mã hóa để xác nhận. Thao tác gửi gói tin yêu cầu lên server là hoàn toàn tự động nên khách hàng chỉ đơn giản là ấn nút yêu cầu mã thanh toán. Khách hàng cũng có thể thanh toán bằng tài khoản khác bằng cách thay đổi tài khoản thanh toán.
- Bước 4: Server gửi mã thanh toán về cho khách hàng.
 - Sau khi tiếp nhận yêu cầu thanh toán từ phía khách hàng, server phát sinh mã thanh toán và gửi về cho khách hàng. Ngoài ra, server sẽ đính kèm thông tin khách hàng với mã thanh toán đó và lưu lại trên server.
- Bước 5: Khách hàng cung cấp mã thanh toán cho nhân viên thanh toán. Hiện tại có 3 cách:
 - Truyền miệng: trong điều kiện môi trường giao dịch có nhiều tiếng ồn thì cách thức truyền miệng tồn tại một số rủi ro như: khách hàng nghe không rõ mã thanh toán hoặc nghe nhầm mã thanh toán. Do đó, cách thức truyền miệng không được khuyến khích sử dụng.
 - Thông qua QR code: sau khi nhận thông tin mã thanh toán, ứng dụng sẽ tự xuất mã QR code ra màn hình ứng dụng, nhân viên thanh toán chỉ cần dùng camera quét QR code và nhận mã thanh toán (có thể sử dụng trong môi trường

âm thanh bị nhiễu). Điều kiện để sử dụng có thể truyền thông tin qua QR code là cửa hàng phải lắp đặt thiết bị đọc QR code (ví dụ: webcam..).

- Thông qua công nghệ NFC: để có thể sử dụng phương thức này, điện thoại khách hàng cần phải hỗ trợ công nghệ NFC. Khi nhận được mã thanh toán, ứng dụng tự chuyển dữ liệu thông qua kết nối NFC. Khách hàng chỉ cần áp điện thoại vào thiết bị đọc NFC của nhân viên thanh toán. Để có thể sử dụng cách thức truyền thông tin bằng công nghệ NFC, cửa hàng phải lắp đặt thiết bị đọc NFC.
- Bước 6: Nhân viên thanh toán nhập thông tin hóa đơn thanh toán và mã thanh toán.
 - Nhân viên thanh toán phải là người nhập thông tin hóa đơn và số tiền cần thanh toán.
 - Nhân viên thanh toán nhập mã thanh toán là để server nhận biết khách hàng nào là người thực hiện thanh toán qua đơn này, nhờ vào thông tin server đã lưu ở bước 4.
- Bước 7: Server gửi thông tin hóa đơn về cho khách hàng và yêu cầu xác nhận.
 - Sau khi nhận thông tin hóa đơn từ nhân viên thanh toán kèm theo mã thanh toán, Server dùng thông tin đã lưu ở bước 4 để gửi thông tin hóa đơn về cho khách hàng bằng cơ chế Realtime trên điện thoại di động.
- Bước 8: Gửi xác nhận thanh toán.
 - Khi nhận được thông tin hóa đơn, khách hàng có thể đồng ý thanh toán hóa đơn hoặc hủy hóa đơn do có sai sót hay vì một lý do nào đó.
 - Khi chấp nhận thanh toán, khách hàng có thể lựa chọn thanh toán bằng tài khoản thẻ thanh toán (ATM/Credit/Debit) hay ví điện tử.
- Bước 9: Thông báo kết quả giao dịch cho khách hàng (SMS) và nhân viên thanh toán (internet).

Ưu điểm:

- Mô hình tận dụng được những công cụ thanh toán (ví điện tử, tài khoản thẻ ngân hàng) và những công nghệ mới (NFC, QR code) để xây dựng nên một mô hình

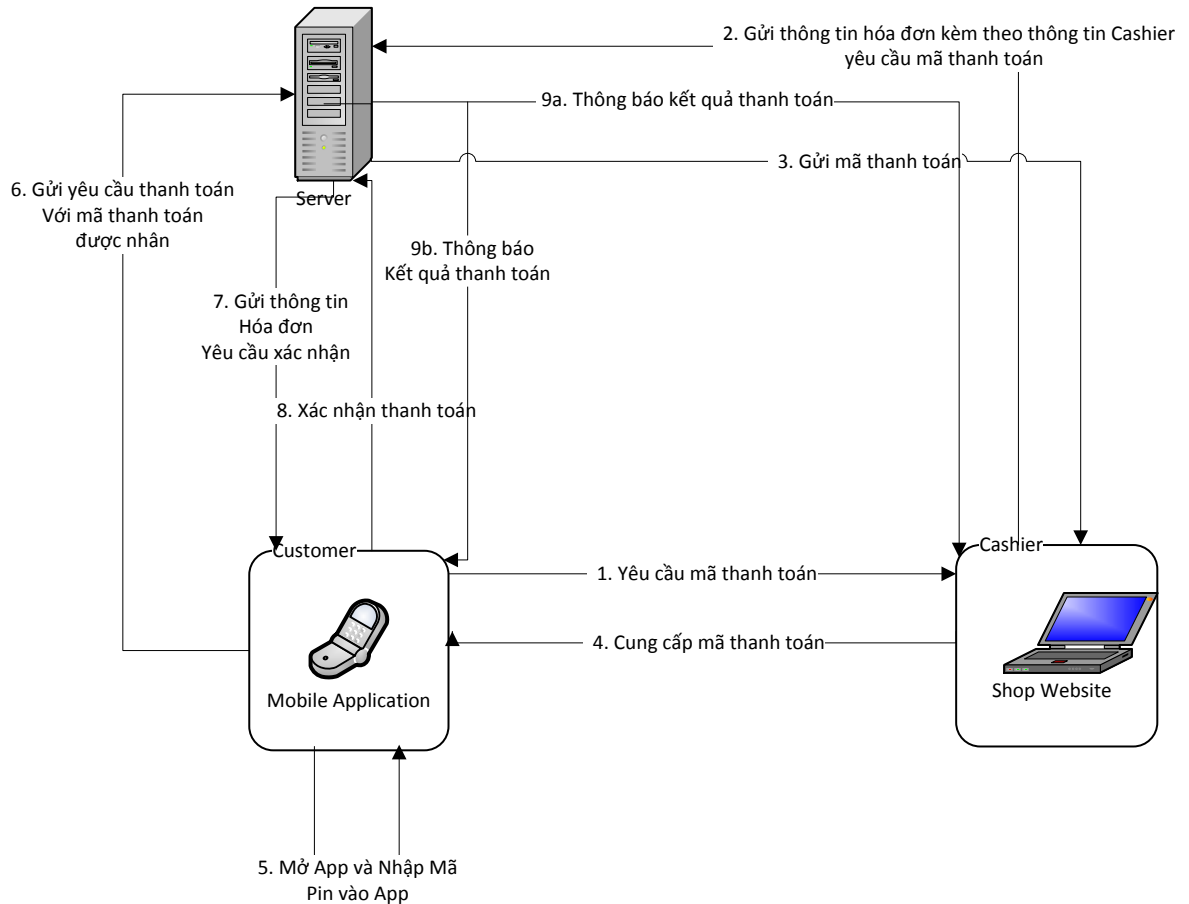
thanh toán tiện lợi, nhanh chóng và an toàn. Người dùng không cần quan tâm mình có mang thẻ thanh toán hay tiền mặt không. Khách hàng chỉ cần mang theo điện thoại và tiến hành thanh toán.

- Bằng việc sử dụng mã thanh toán làm đại diện thì khách hàng không cần tiết lộ bất cứ thông tin nào của mình với nhân viên thanh toán khiến khách hàng được an tâm hơn.
- Nếu xem xét số lượng thao tác của khách hàng thì ở mô hình thanh toán này khách hàng chỉ cần thực hiện một vài thao tác đơn giản nên mô hình này rất thuận tiện cho khách hàng.
- Quá trình thực hiện diễn ra nhanh chóng nhưng vẫn đảm bảo được sự minh bạch trong quá trình thanh toán. Cụ thể hơn, để khách hàng cảm thấy dễ dàng hơn thì nhân viên thanh toán là người nhập thông tin hóa đơn nhưng sau đó thông tin nhân viên thanh toán nhập được hiển thị trực tiếp trên màn hình khách hàng có thể xem rõ thông tin hóa đơn mình thanh toán.

Hạn chế:

- Do quá trình thanh toán dựa trên dịch vụ từ phía các ngân hàng hay nhà cung cấp dịch vụ ví điện tử nên mô hình còn lệ thuộc ở bước thanh toán sau cùng là chuyển tiền. Nếu bên cung cấp dịch vụ quá tải thì mô hình cũng không thực hiện được thanh toán tuy quá trình vẫn hoàn tất với thông báo trả về các bên thanh toán.
- Cả nhân viên thanh toán và khách hàng cần phải có kết nối internet.

2.2.2. Mô hình thanh toán 2.



Hình 2-4: Sơ đồ xử lý mô hình thanh toán đề xuất 2

Các bước của quá trình thanh toán:

- Bước 1: Khách hàng yêu cầu mã thanh toán từ nhân viên thanh toán .
 - Mã thanh toán ở mô hình thanh toán 2 khác ở mô hình thanh toán 1. Trong mô hình thanh toán 1, mã thanh toán đại diện cho phiên giao dịch, đồng thời cũng là đại diện cho thông tin khách hàng. Trong mô hình thanh toán 2, mã thanh toán đại diện cho hóa đơn mà khách hàng cần thanh toán.
- Bước 2: nhân viên thanh toán yêu cầu mã thanh toán.
 - Khi yêu cầu mã thanh toán, nhân viên thanh toán gửi kèm theo thông tin của nhân viên thanh toán. Thông tin nhân viên thanh toán ở đây là thông tin tài khoản của nhân viên trực tiếp tiến hành thanh toán hóa đơn này, thông tin của

hàng nhân viên thanh toán đang thực hiện thanh toán. Đồng thời nhân viên thanh toán cũng gửi thông tin hóa đơn cần khách hàng thanh toán.

- Bước 3: Server gửi mã thanh toán về cho nhân viên thanh toán.
 - Sau khi tiếp nhận yêu cầu thanh toán từ phía nhân viên thanh toán, server phát sinh mã thanh toán gửi về cho nhân viên thanh toán, đồng thời server sẽ đính kèm thông tin hóa đơn thanh toán với mã thanh toán đó và lưu lại trên server.
- Bước 4: nhân viên thanh toán cung cấp mã thanh toán cho khách hàng. Hiện tại khách hàng có 3 cách.
 - Truyền miệng: trong điều kiện môi trường giao dịch có nhiều tiếng ồn thì cách thức truyền miệng tồn tại một số rủi ro như: khách hàng nghe không rõ mã thanh toán hoặc nghe nhầm mã thanh toán. Do đó, cách thức truyền miệng không được khuyến khích sử dụng.
 - Thông qua QR code: sau khi nhận thông tin mã thanh toán, nhân viên thanh toán có thể in ra QR code đó và gửi cho khách hàng. Điều kiện để sử dụng có thể truyền thông tin qua QR code là cửa hàng phải lắp đặt thiết bị đọc QR code (ví dụ: webcam..).
 - Thông qua công nghệ NFC: điện thoại khách hàng phải có hỗ trợ công nghệ NFC. Khi nhận được mã thanh toán, Web Client truyền mã thanh toán sang điện thoại của Khách hàng thông qua tín hiệu NFC. Khách hàng chỉ cần áp điện thoại vào và nhận mã thanh toán. Điều kiện để sử dụng có thể truyền thông tin qua NFC là cửa hàng phải lắp đặt thiết bị ghi NFC.
- Bước 5: Khách hàng mở ứng dụng, và nhập mã PIN.
 - Mã PIN: dùng để đảm bảo độ bảo mật, tránh những thanh toán không mong muốn. Mã PIN là một số gồm 4 chữ số.
- Bước 6: Khách hàng yêu cầu thanh toán từ mã thanh toán được nhận.
 - Chỉ khi khách hàng nhập mã thanh toán vào bằng tay thì khách hàng mới cần phải chọn nút thanh toán, còn lại nếu khách hàng chọn chức năng nhận code bằng QR code hay NFC thì sau bước 5, ứng dụng sẽ tự động thực hiện bước này cho khách hàng.

- Khi gửi yêu cầu thanh toán với mã số thanh toán thì server sẽ hiệu chỉnh lại thông tin người thanh toán được lưu trong thông tin hóa đơn đã được đính kèm với mã thanh toán ở bước 2. Việc hiệu chỉnh như vậy nhằm tránh các trường hợp nhập trùng mã thanh toán của khách hàng.
- Bước 7: Server gửi thông tin hóa đơn về cho khách hàng và yêu cầu xác nhận.
 - Sau khi nhận thông tin hóa đơn từ nhân viên thanh toán kèm theo mã thanh toán, server dùng thông tin đã lưu ở bước 4 để gửi thông tin hóa đơn về cho khách hàng bằng cơ chế Realtime trên điện thoại di động.
- Bước 8: Gửi xác nhận thanh toán.
 - Khi nhận được thông tin hóa đơn khách hàng có thể đồng ý thanh toán hóa đơn hoặc hủy hóa đơn do có sai sót hay vì một lý do nào đó.
 - Khi chấp nhận thanh toán, khách hàng có thể lựa chọn thanh toán bằng tài khoản thẻ thanh toán hay ví điện tử.
- Bước 9: thông báo kết quả giao dịch cho khách hàng (SMS) và nhân viên thanh toán (Internet).

Ưu điểm:

- Ở mô hình thanh toán 2 này khách hàng chủ động hơn trong việc thanh toán và xác nhận giao dịch. Cụ thể hơn, khách hàng là người trực tiếp yêu cầu hóa đơn khi nhập mã thanh toán. Server sẽ trả về gần như ngay lập tức cho Khách hàng thay vì phải thụ động chờ hóa đơn từ phía nhân viên thanh toán.

Nhược điểm:

- Do cách xử lý thì đều như nhau nên mô hình 2 thanh toán có cùng nhược điểm với mô hình thanh toán 1 là còn lệ thuộc vào bên cung cấp dịch vụ giao dịch. Nhân viên thanh toán và Khách hàng đều cần phải có kết nối internet.

Bên cạnh những vấn đề cần phải xử lý riêng cho từng mô hình thì mô hình thanh toán 1 và mô hình thanh toán 2 còn đối mặt với một số vấn đề chung như sau:

- ❖ Giới hạn số lượng mã giao dịch tại một thời điểm.

Mã giao dịch được cấp cho người dùng có thời gian hiệu lực nhất định, nó chỉ được tái sử dụng khi khách hàng thực hiện hủy giao dịch hay hết thời gian hiệu lực.

Vậy tồn tại trường hợp số lượng mã thanh toán không đủ để đáp ứng nhu cầu thanh toán do việc yêu cầu hàng loạt mã thanh toán trong một khoảng thời gian ngắn nhưng không thực hiện thanh toán hay chưa hoàn tất thanh toán.

Nguy cơ người dùng vô tình hay cố tình yêu cầu hàng loạt mã thanh toán trong một thời gian ngắn là rất lớn, do đó cần có giải pháp để tránh trường hợp này xảy ra.

Giải pháp:

- Server: sử dụng thư viện có sẵn của ASP.NET là Web Api Throttling Handler.
 - Web Api Throttling Handler là thư viện giúp kiểm soát tần suất yêu cầu từ client dựa trên địa chỉ IP, API Key (khóa được Web API cung cấp) đến toàn bộ các hàm hay một số hàm nhất định của Web API.
 - Bằng cách tùy chỉnh một số thông số ban đầu như số lượng yêu cầu tối đa trong một giây, trong một phút, trong một giờ, trong một ngày... Web Api Throttling Handler sẽ giới hạn số lượng yêu cầu từ client một cách tuyệt đối theo ý người dùng mong muốn.
 - Ngoài tránh những yêu cầu mã thanh toán vô tình từ người dùng, Web Api Throttling Handler còn tránh được những yêu cầu với ý đồ không tốt từ một số cá nhân.
- Mobile App, Web Client:
 - Khi người dùng thực hiện thao tác yêu cầu mã xác nhận, ứng dụng sẽ thực hiện thao tác kiểm tra người dùng đã yêu cầu và nhận được mã thanh toán nhưng chưa hoàn thành giao dịch hay chưa. Nếu chưa hay đã nhận mã thanh toán nhưng đã hoàn thành giao dịch, ứng dụng sẽ tiếp tục gửi yêu cầu mã thanh toán về server. Nếu đã nhận mã thanh toán nhưng chưa hoàn thành giao dịch, ứng dụng sẽ yêu cầu người dùng xác nhận lại việc yêu cầu mã thanh toán mới có cần thiết không.
 - Thao tác yêu cầu người dùng xác nhận lại việc yêu cầu mã thanh toán có tác dụng nhắc nhở khi vô ý yêu cầu mã thanh toán và góp phần hạn chế yêu cầu có chủ ý từ người dùng.
- ❖ Đảm bảo sự tương tác giữa các thành phần theo thời gian thực.

Từ quy trình thanh toán đã đặt ra, có thể nhận thấy khách hàng hay nhân viên thanh toán không thể đơn phương thực hiện giao dịch, việc thanh toán phải dựa trên sự xác thực từ hai phía thông qua việc giao tiếp với server (Web API) làm trung gian. Như vậy đòi hỏi server phải đảm bảo được hai yêu cầu then chốt như sau:

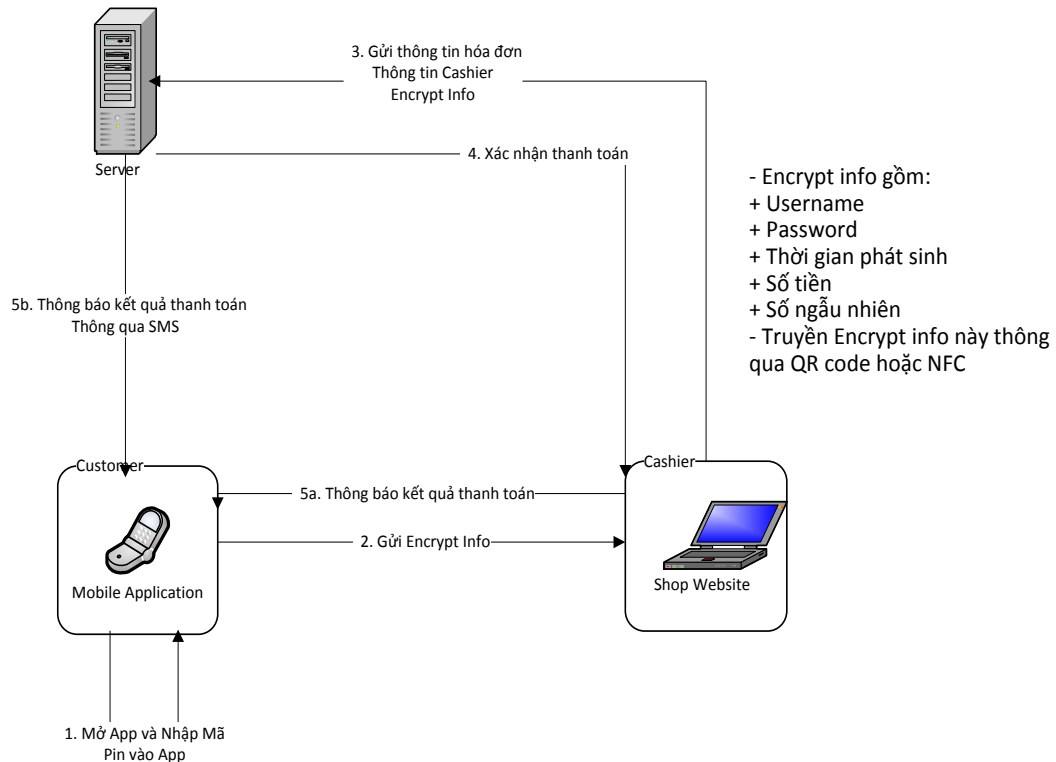
- Xác định được khách hàng nào đang thực hiện giao dịch với nhân viên thanh toán nào và ngược lại. Phải đảm bảo được quan hệ “một – một”, nghĩa là tại một thời điểm một khách hàng chỉ có thể thực hiện thanh toán với một nhân viên thanh toán và ngược lại.
- Đảm bảo kênh kết nối liên tục đến ứng dụng trên điện thoại di động được khách hàng sử dụng và Web Client do nhân viên thanh toán sử dụng. Trong quy trình thanh toán ở cả 2 mô hình, tồn tại những bước “nghỉ”. Bước “nghỉ” chính là giai đoạn một bên giao dịch chờ xác nhận từ bên giao dịch kia (bước 5 và bước 6 ở mô hình thanh toán 1, bước 4 ở mô hình thanh toán 2). Việc tồn tại của những bước “nghỉ” khiến việc mở kết nối liên tục là bắt buộc, vì sau khi xử lý thông tin xác nhận từ một bên giao dịch Server cần gửi thông tin về cho bên giao dịch còn lại một cách chủ động để quy trình giao dịch được khép kín.

Giải pháp:

- Mỗi quan hệ “một-một”: tận dụng yếu tố có sẵn trong mô hình là mã thanh toán. Tại một thời điểm mã thanh toán chỉ phát sinh cho một người dùng duy nhất (cho khách hàng ở mô hình thanh toán 1 hoặc cho nhân viên thanh toán ở mô hình thanh toán 2) đồng nghĩa với việc mã giao dịch cũng đại diện phiên giao dịch giữa 2 bên giao dịch tại một thời điểm, vì có sự trao đổi mã giao dịch giữa khách hàng và nhân viên thanh toán. Từ đó server dễ dàng thiết lập mối quan hệ “một-một” một cách chính xác.
- Mở kênh kết nối liên tục: Sử dụng thư viện có sẵn của ASP.NET là SignalR [13].
 - SignalR: SignalR là 1 thư viện viết cho .NET giúp xây dựng các ứng dụng web (hiện tại đã có hỗ trợ trên thiết bị di động) sử dụng tương tác thời gian thực giữa máy khách với máy chủ, máy khách có thể gửi yêu cầu về máy chủ và máy chủ có thể “push” trực tiếp dữ liệu về máy khách.

- Bằng cách vận dụng sự tương tác thời gian thực giữa Server-Web Client (ASP.NET – ASP.NET) và Server - Ứng dụng trên di động (ASP.NET - Android) đã giải quyết được vấn đề tại các bước “nghỉ”.

2.2.3. Mô hình thanh toán 3.



Hình 2-5: Sơ đồ xử lý mô hình thanh toán đề xuất 3

Các bước của quá trình thanh toán.

- Bước 1: Khách hàng mở ứng dụng, và nhập mã PIN.
 - Mã PIN: dùng để đảm bảo độ bảo mật, tránh những thanh toán không mong muốn. Mã PIN là một số gồm 4 chữ số.
- Bước 2: Khách hàng gửi về cho nhân viên thanh toán 1 Encrypt info (bằng QR code hoặc NFC)
 - Gói tin Encrypt info được mã hóa bằng thuật toán RSA. Ứng dụng cài đặt trên điện thoại của khách hàng dùng public key để mã hóa gói tin.
- Bước 3: Nhân viên thanh toán sẽ gửi thông tin hóa đơn thanh toán và Encrypt info tới server để thực hiện thanh toán sau khi nhận Encrypt info từ khách hàng.

- Bước 4: Server xác nhận thanh toán và gửi kết quả về cho nhân viên thanh toán và khách hàng. Các thao tác sau sẽ được thực hiện:
 - Dùng Private Key để giải mã gói tin Encrypt info và trích rút được thông tin của khách hàng
 - Xét tính hợp lệ của gói tin Encrypt Info: dựa vào nhiều thông tin để xét tính hợp lệ.
 - Độ sai lệch về thời gian: Thời gian từ lúc gói dữ liệu được tạo ra đến lúc server xử lý không được quá một khoản thời gian nhất định (2 phút).
 - Thông tin tài khoản khách hàng: Thông tin tài khoản khách hàng thanh toán không tồn tại đồng nghĩa với gói tin không hợp lệ.
 - Gói tin xử lý sau cùng: Mỗi khách hàng khi thanh toán bằng hình thức này, server sẽ lưu trữ thông tin thời gian gói tin sau cùng được thanh toán (dù thanh công hay thất bại). Mọi gói tin được gửi lên nếu có thời gian phát sinh sớm hơn so với thông tin thời gian gói tin sau cùng được lưu trữ thì gói tin ấy không hợp lệ.
 - Tổng tiền: Tổng tiền do nhân viên thanh toán không được vượt quá số tiền khách hàng chấp nhận thanh toán.
- Bước 5: Thông báo kết quả giao dịch cho khách hàng (SMS) và nhân viên thanh toán (internet).

Ưu điểm:

- Một cách khái quát nhất thì mô hình thanh toán 3 mang lại sự tiện lợi, nhanh chóng nhất cho khách hàng khi thực hiện giao dịch. Cụ thể là khách hàng chỉ cần nhập số tiền cần thanh toán vào ứng dụng sau đó truyền thông tin cho nhân viên thanh toán và chờ nhận thông tin kết quả giao dịch thông qua SMS là xem như hoàn thành giao dịch. Hiện tại cả quá trình giao dịch chỉ diễn ra trong 5 bước.
- Ở mô hình thanh toán 3, khách hàng không cần phải có kết nối internet tại thời điểm thanh toán. Khi có kết nối internet, khách hàng có thể dễ dàng xem lại lịch sử giao dịch nếu muốn.

Nhược điểm:

- Vấn đề lớn nhất gặp phải là tính bảo mật, tâm lý khách hàng còn e dè khi gửi thông tin của mình qua gói Encrypt info.
- Khi thực hiện thanh toán theo mô hình giao dịch này thì nhân viên thanh toán bắt buộc phải sử dụng Camera để đọc QR code hoặc khách hàng sử dụng NFC thì nhân viên thanh toán phải sử dụng thiết bị đọc NFC.

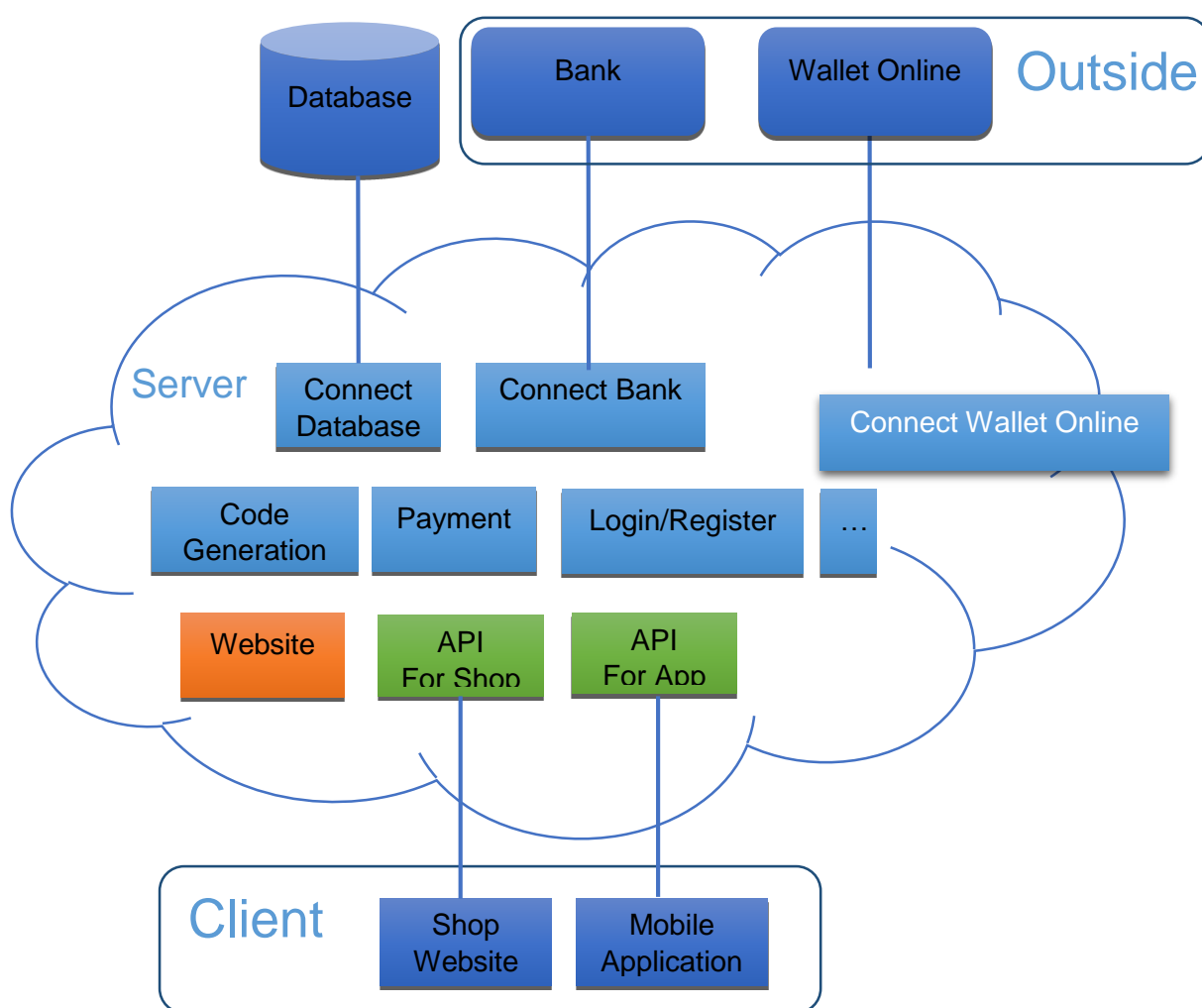
Sau khi đề xuất và phân tích ưu nhược điểm cũng như phân tích quy trình xử lý của các mô hình thanh toán, kiến trúc hệ thống sau được xây dựng nhằm mục tiêu đáp ứng tất cả các yêu cầu cần thiết.

Chương 3: XÂY DỰNG CÁC MÔ HÌNH ĐỀ XUẤT

✍ Nội dung chương này đề cập đến thiết kế kiến trúc hệ thống và các vấn đề chính trong quá trình xây dựng ứng dụng.

3.1. Thiết kế kiến trúc hệ thống

Từ những đề xuất và phân tích ưu nhược điểm của các mô hình thanh toán và tham khảo quy trình xử lý của các mô hình thanh toán khác, kiến trúc hệ thống sau được xây dựng nhằm mục tiêu đáp ứng tất cả các yêu cầu cần thiết đó.



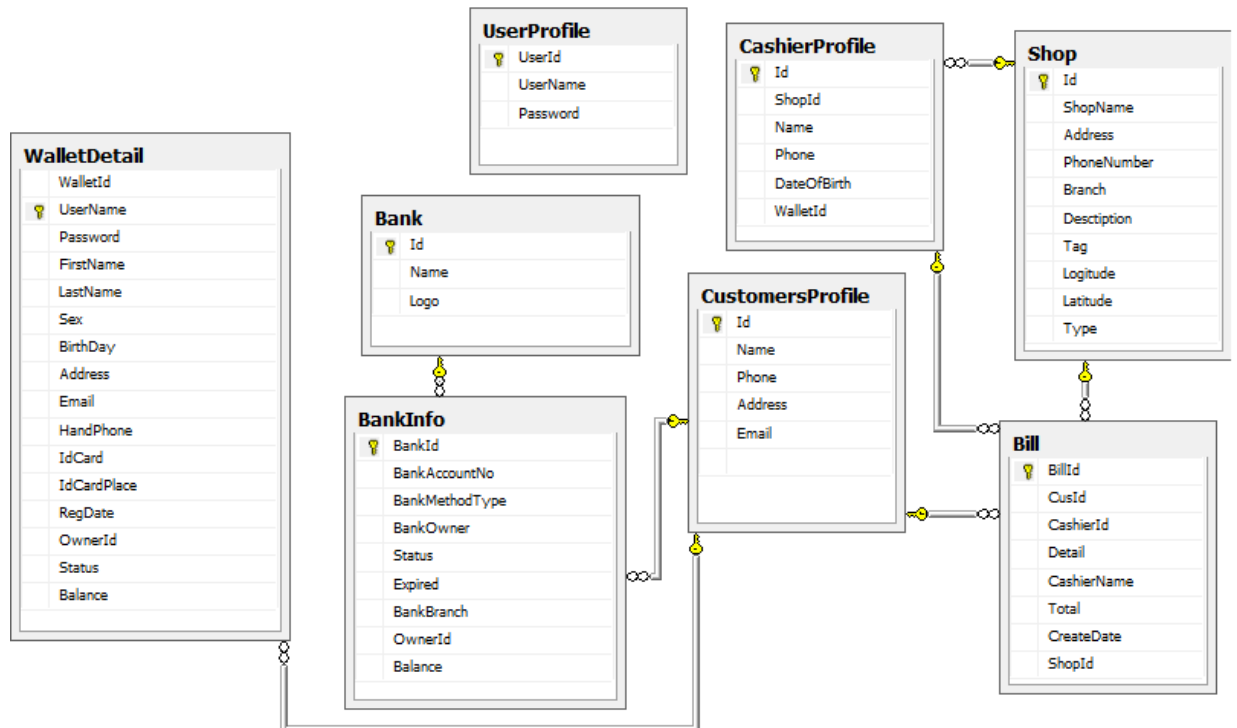
Hình 3-1: Sơ đồ kiến trúc hệ thống đề xuất

Hệ thống bao gồm 4 thành phần chính là: Server, Client, Database-side, Outside.

- Server: Server cung cấp RestfulAPI được dựng trên nền ASP.NET MVC4.

- API For Shop: Cung cấp các hàm API cho website cửa hàng (cửa hàng, nơi thanh toán chấp nhận mô hình thanh toán này).
- API For App: Cung cấp các hàm API cho ứng dụng trên smartphone (khách hàng thanh toán thông qua mô hình này)
- Website: Trang web giới thiệu về mô hình cũng như hướng dẫn sử dụng dịch vụ.
- Code Generation, Payment, Login/Register,...: Các dịch vụ bên dưới API For Shop, API For App.
- Connect Database: Thực hiện thao tác gọi các Store Procedure được khai báo ở Database-side.
- Connect Bank: Thực hiện nhiệm vụ giao tiếp với dịch vụ bên ngân hàng (Bank) cung cấp để thực hiện các giao dịch.
- Connect Wallet Online: Thực hiện nhiệm vụ giao tiếp với dịch vụ ví điện tử (Wallet Online) cung cấp thực hiện các giao dịch.
- OutSide:
 - Bank: Dịch vụ được cung cấp bởi các ngân hàng đã thỏa thuận.
 - Wallet Online: Dịch vụ được cung cấp bởi các nhà cung cấp dịch vụ ví điện tử Online.
- Database:
 - Lưu trữ thông tin.
 - Chứa các Store Procedure thao tác trên cơ sở dữ liệu.
- Client:
 - Shop Website: Trang web sử dụng API cho Shop, dành cho cửa hàng, nơi thanh toán thực hiện giao dịch.
 - Mobile Application: Ứng dụng trên thiết bị di động giúp khách hàng thực hiện giao dịch.

3.2. Thiết kế cơ sở dữ liệu



Hình 3-2: Sơ đồ kiến trúc cơ sở dữ liệu

Mô tả chi tiết cấu trúc cơ sở dữ liệu.

3.2.1. **UserProfile.**

Lưu thông tin căn bản của người dùng (tên đăng nhập/số điện thoại đăng nhập, mật khẩu...)

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	UserId	int	Khóa Chính	Id tài khoản người dùng
2	UserName	nvarchar (MAX)	Không rỗng	Tên đăng nhập/số điện thoại đăng nhập
3	Password	nvarchar (MAX)	Không rỗng	Mật Khẩu

Bảng 3-1: Thông tin mô tả chi tiết cấu trúc bảng dữ liệu UserProfile

3.2.2. **CustomersProfile.**

Lưu thông tin khách hàng sử dụng dịch vụ Mobile Payment đã đăng ký.

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	Id	int	Khóa Chính	Id của khách hàng (cũng là Id ở bản UserProfile)
2	Name	nvarchar (50)	Không rỗng	Tên
3	Phone	nvarchar (50)	Không rỗng	Số điện thoại
4	Address	nvarchar (MAX)	Không rỗng	Địa chỉ
5	Emai	nvarchar (100)	Không rỗng	Email của khách hàng để thông báo lịch sử giao dịch, cấp lại mật khẩu.

Bảng 3-2: Thông tin mô tả chi tiết cấu trúc bảng dữ liệu CustomersProfile

3.2.3. **CashierProfile.**

Lưu thông tin nhân viên các cửa hàng sử dụng dịch vụ Mobile Payment (người trực tiếp thực hiện)

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	Id	int	Khóa Chính	Id nhân viên thanh toán (cũng là Id ở bản UserProfile)
2	ShopId	int	Khóa ngoại, không rỗng (liên kết với trường Id bảng Shop)	Id cửa hàng nhân viên thực hiện thanh toán
3	Name	nvarchar	Không rỗng	Tên nhân viên

		(50)		
4	Phone	nvarchar (50)	Không rỗng	Số điện thoại nhân viên
5	DateOfBirth	date	Không rỗng	Ngày sinh nhân viên
6	WalletId	int	Không rỗng	Khi đăng ký sử dụng dịch vụ, nhân viên sẽ được cấp thông tin ví điện tử để thực hiện giao dịch. WalletId là Id ví nhân viên này sở hữu.

Bảng 3-3: Thông tin mô tả chi tiết cấu trúc bảng dữ liệu CashierProfile

3.2.4. **Bank.**

Lưu trữ thông tin các ngân hàng đã hợp tác, cung cấp dịch vụ thanh toán.

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	Id	int	Khóa Chính	Id nhân viên thanh toán (cũng là Id ở bản UserProfile)
2	Name	nvarchar (100)	Không rỗng	Tên ngân hàng
3	Logo	nvarchar (MAX)	Không rỗng	URL Logo của ngân hàng.

Bảng 3-4: Thông tin mô tả chi tiết cấu trúc bảng dữ liệu Bank

3.2.5. **BankInfo.**

Lưu thông tin chi tiết tài khoản thẻ thanh toán người dùng đăng ký để giao dịch trên hệ thống (xây dựng nhằm mô tả luồng xử lý, khi đưa vào thực tế thông tin này do các ngân hàng hợp tác lưu giữ).

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	BankId	int	Khóa Chính	Id thẻ thanh toán.

2	BankAccountNo	nvarchar(50)	Không rỗng	Mã thẻ được in trên thẻ
3	BankMethodType	nvarchar(50)	Không rỗng	Credit Card/Debit Card
4	BankOwner	nvarchar(50)	Không rỗng	Tên chủ thẻ được in trên thẻ
5	Status	nvarchar(50)	Không rỗng	Tình trạng thẻ
6	Expired	date	Không rỗng	Ngày hết hạn.
7	BankBranch	int	Khóa ngoại, không rỗng (liên kết đến trường Id của bảng Bank)	Thẻ thuộc hệ thống ngân hàng nào.
8	OwnerId	int	Khóa ngoại (liên kết đến trường Id bảng Khách hàngsProfile) Có thể rỗng	Khi đã được kích hoạt bởi một khách hàng, trường OwnerId sẽ lưu Id người dùng đó để quản lý.
9	Balance	float	Không rỗng	Số dư khả dụng.

Bảng 3-5: Thông tin mô tả chi tiết cấu trúc bảng dữ liệu BankInfo

3.2.6. *WalletDetail.*

Lưu thông tin ví online (xây dựng nhằm mô tả luồng xử lý, khi đưa vào thực tế thông tin này do bên cung cấp dịch vụ ví điện tử hợp tác lưu giữ)

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	WalletId	int	Không rỗng	Id ví, số tự tăng.
2	UserName	nvarchar(50)	Khóa chính	Tên ví điện tử
3	Password	nvarchar(50)	Không rỗng	Mật khẩu ví điện tử.
4	FirstName	nvarchar(50)	Không rỗng	Tên chủ ví
5	LastName	nvarchar(50)	Không rỗng	Họ chủ ví

6	Sex	nvarchar(50)	Không rỗng	Giới tính chủ ví
7	BirthDay	date	Không rỗng	Ngày sinh chủ ví
8	Address	nvarchar(MAX)	Không rỗng	Địa chỉ chủ ví
9	Email	nvarchar(MAX)	Không rỗng	Địa chỉ mail chủ ví dùng để thông báo thông tin giao dịch.
10	HandPhone	nvarchar(50)	Không rỗng	Số điện thoại chủ ví.
11	IdCard	nvarchar(50)	Không rỗng	Số chứng minh chủ thẻ.
12	IdCardPlace	nvarchar(50)	Không rỗng	Nơi cấp giấy chứng minh.
13	RegDate	date	Không rỗng	Ngày đăng ký.
14	OwnerId	int	Khóa ngoại (liên kết đến trường Id bảng Khách hàngsProfile) Có thể rỗng	Khi đã được kích hoạt bởi một khách hàng, trường OwnerId sẽ lưu Id người dùng đó để quản lý.
15	Status	nvarchar(50)	Không rỗng	Tình trạng ví.
16	Balance	float	Không rỗng	Số dư khả dụng.

Bảng 3-6: Thông tin mô tả chi tiết cấu trúc bảng dữ liệu WalletDetail

3.2.7. **Shop.**

Lưu thông tin shop đăng ký sử dụng dịch vụ.

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	ShopId	int	Không rỗng	Id cửa hàng đăng ký sử dụng dịch vụ
2	ShopName	nvarchar(50)	Không rỗng	Tên cửa hàng
3	Address	nvarchar(MAX)	Không rỗng	Địa chỉ cửa hàng

4	PhoneNumber	nvarchar(50)	Không rỗng	Số điện thoại của hàng.
5	Type	nvarchar(50)	Không rỗng	Loại mặt hàng của hàng bán
6	Branch	nvarchar(50)	Không rỗng	Tên hệ thống của hàng (vd: KFC...)
7	Description	nvarchar(MAX)	Không rỗng	Mô tả của hàng.
8	Tag	nvarchar(MAX)	Có thể rỗng	Thông tin dùng để tìm kiếm của hàng.
9	Longitude	float	Có thể rỗng	Kinh độ
10	Latitude	float	Có thể rỗng	Vĩ độ

Bảng 3-7: Thông tin mô tả chi tiết cấu trúc bảng dữ liệu Shop

3.2.8. **Bill.**

Lưu thông tin mỗi khi giao dịch thành công.

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	BillId	int	Khóa chính	Id hóa đơn
2	CusId	int	Khóa ngoại, không rỗng (liên kết đến trường Id bảng Khách hàngsProfile)	Id khách hàng thanh toán
3	Id	int	Khóa ngoại, không rỗng (liên kết đến trường Id bảng nhân viên thanh toánProfile)	Id nhân viên thanh toán
4	Detail	nvarchar (MAX)	Không rỗng	Thông tin chi tiết các món hàng thanh toán
5	Total	Float	Không rỗng	Tổng số tiền thanh toán

6	CreateDate	datetime	Không rỗng	Thời gian thực hiện giao dịch
7	ShopId	Int	Khóa ngoại, không rỗng (liên kết đến trường Id bảng Shop)	Id tại cửa hàng thực hiện thanh toán hóa đơn

Bảng 3-8: Thông tin mô tả chi tiết cấu trúc bảng dữ liệu Bill

3.3. Xây dựng ứng dụng.

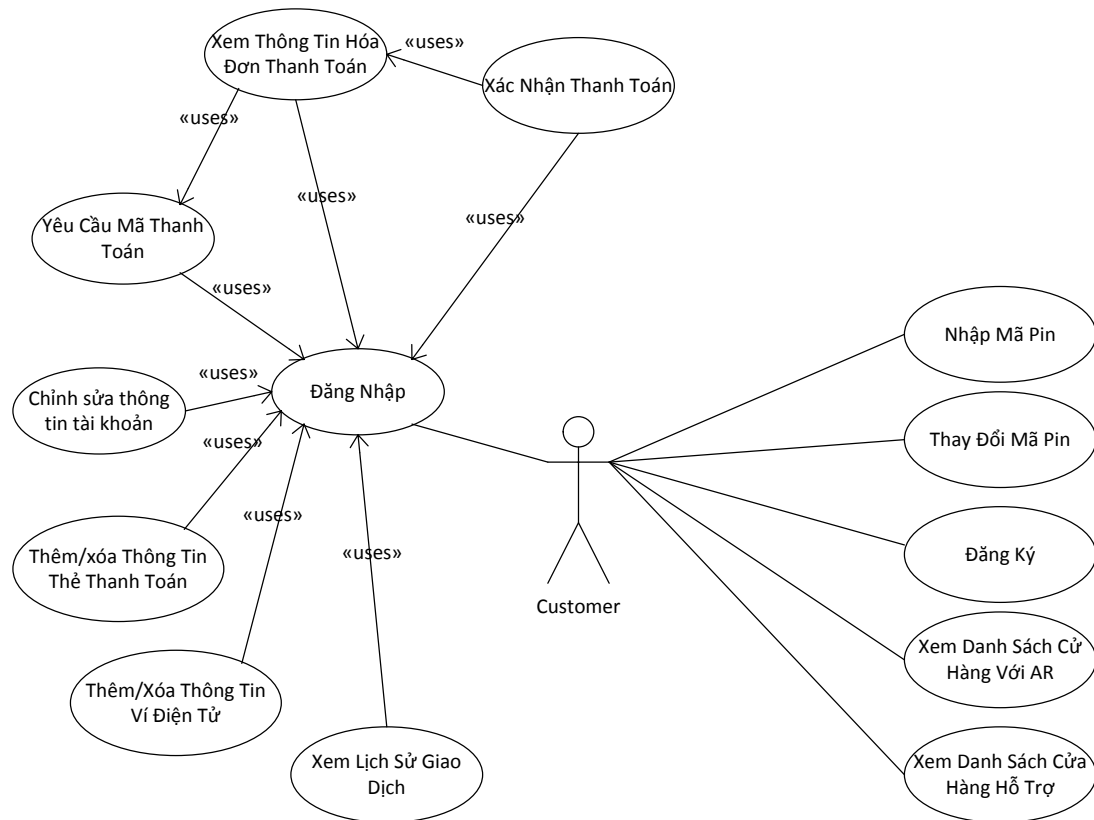
3.3.1. Mô Hình Thanh Toán 1.

3.3.1.1. API được server cung cấp

Các dịch vụ chính được API cung cấp:

- API for App:
 - Đăng nhập/ đăng xuất.
 - Đăng ký tài khoản mới.
 - Chỉnh sửa thông tin cá nhân.
 - Thêm/xóa thông tin ngân hàng cá nhân.
 - Thêm/xóa thông tin ví điện tử cá nhân.
 - Cấp mã thanh toán.
 - Xác nhận thông tin hóa đơn.
 - Thông báo kết quả thanh toán.
 - Cung cấp lịch sử giao dịch.
 - Trả về thông tin mã lỗi, chi tiết lỗi trong quá trình giao dịch.
- API for Shop:
 - Đăng nhập/đăng xuất.
 - Tiếp nhận yêu cầu thanh toán hóa đơn có kèm theo mã thanh toán.
 - Thông báo kết quả giao dịch.
 - Cung cấp lịch sử giao dịch.
 - Trả về thông tin mã lỗi, chi tiết lỗi trong quá trình giao dịch.
- Giao tiếp với các bên cung cấp dịch vụ.
 - Gửi và tiếp nhận thông tin thanh toán với các dịch vụ ngân hàng.
 - Gửi và tiếp nhận thông tin thanh toán với các dịch vụ ví điện tử.
 - Giao tiếp với bên cung cấp dịch vụ SMS.
 - Gửi yêu cầu hiệu chỉnh, trích xuất dữ liệu với Database.

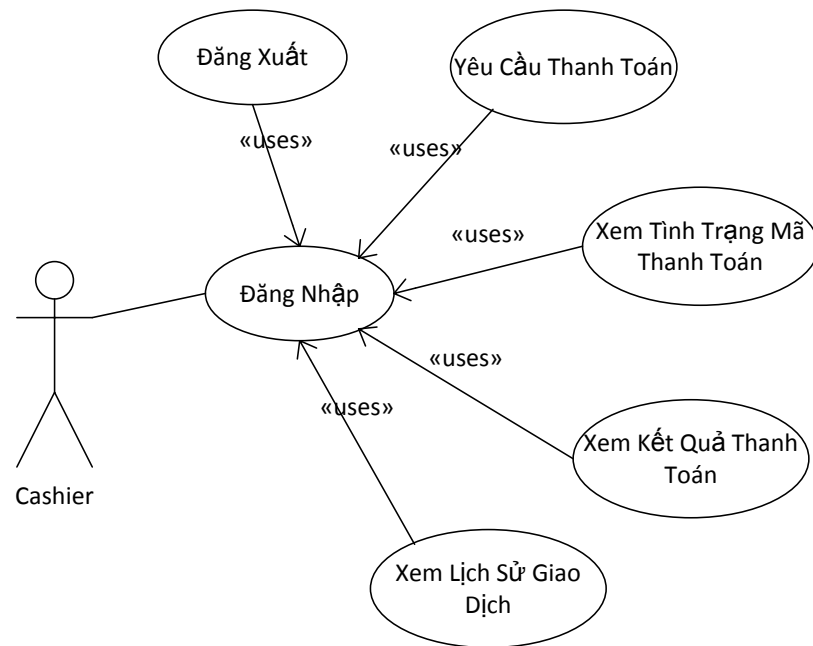
3.3.1.2. Ứng dụng trên điện thoại di động.



Hình 3-3: Sơ đồ UseCase phân hệ khách hàng mô hình đề xuất 1

- Nhập mã PIN.
- Thay đổi mã PIN.
- Đăng Nhập/ Đăng Xuất.
- Đăng ký mới tài khoản.
- Quản lý thông tin người dùng
- Quản lý thông tin thẻ ngân hàng (thêm/xóa).
- Quản lý thông tin ví điện tử (thêm/xóa).
- Hiển thị thông tin hóa đơn thanh toán.
- Xác nhận thông tin thanh toán.
- Xem lịch sử giao dịch.
- Xem danh sách cửa hàng có hỗ trợ thanh toán bằng mô hình này.
- Xem danh sách cửa hàng quanh vị trí người đứng với công nghệ AR.
- Cài đặt, thay đổi ngôn ngữ hiển thị...
- Trợ giúp: hướng dẫn, trợ giúp người dùng.
- Yêu cầu mã thanh toán.

3.3.1.3. Web Client ở cửa hàng.



Hình 3-4: Sơ đồ UseCase phân hệ nhân viên thanh toán mô hình đề xuất 1

Các chức năng chính:

- Đăng nhập/Đăng xuất.
- Yêu cầu thanh toán có kèm theo thông tin hóa đơn và mã thanh toán.
- Hiện thị tình trạng mã thanh toán.
- Thông báo kết quả thanh toán.
- Xem lịch sử giao dịch.

3.3.2. Mô hình thanh toán 2

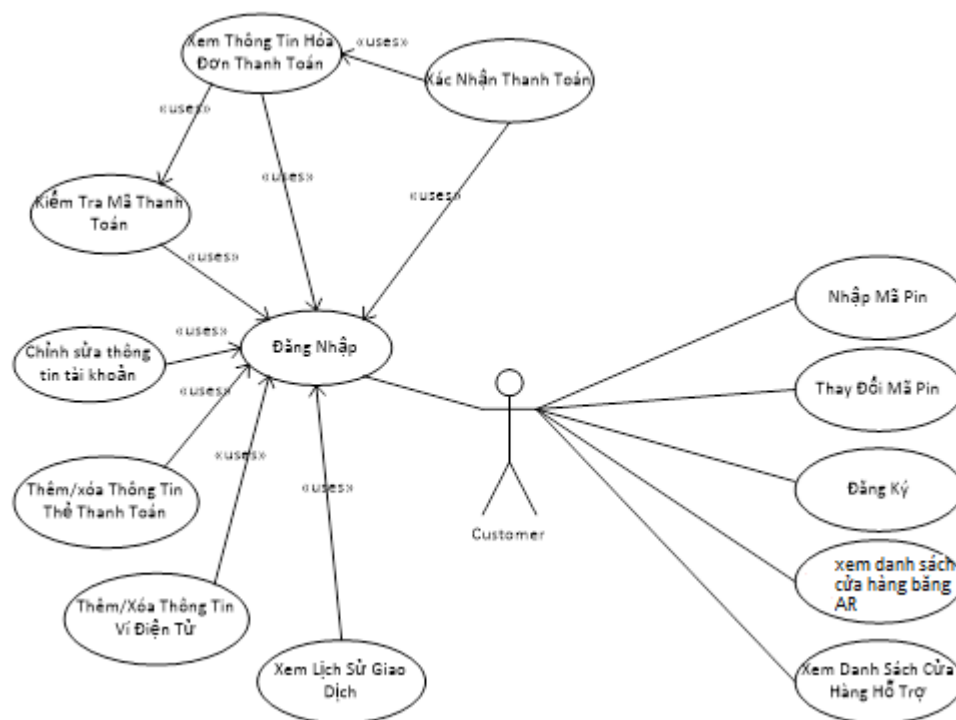
3.3.2.1. API được server cung cấp.

Các dịch vụ chính được API cung cấp:

- API for App:
 - Đăng nhập/ đăng xuất.
 - Đăng ký tài khoản mới.
 - Chỉnh sửa thông tin cá nhân.
 - Thêm/xóa thông tin ngân hàng cá nhân.
 - Kiểm tra mã thanh toán.
 - Xác nhận thông tin hóa đơn.
 - Thông báo kết quả thanh toán.
 - Cung cấp lịch sử giao dịch.
 - Trả về thông tin mã lỗi, chi tiết

- Thêm/xóa thông tin ví điện tử cá nhân.
- Thêm/xóa thông tin mã lỗi trong quá trình giao dịch.
- API for Shop:
 - Đăng nhập/đăng xuất.
 - Cung cấp mã thanh toán.
 - Thông báo kết quả giao dịch.
 - Cung cấp lịch sử giao dịch.
 - Trả về thông tin mã lỗi, chi tiết lỗi trong quá trình giao dịch.
- Giao tiếp với các bên cung cấp dịch vụ.
 - Gửi và tiếp nhận thông tin thanh toán với các dịch vụ ngân hàng.
 - Gửi và tiếp nhận thông tin thanh toán với các dịch vụ ví điện tử.
 - Giao tiếp với bên cung cấp dịch vụ SMS.
 - Gửi yêu cầu hiệu chỉnh, trích xuất dữ liệu với Database.

3.3.2.2. Ứng dụng trên điện thoại di động.

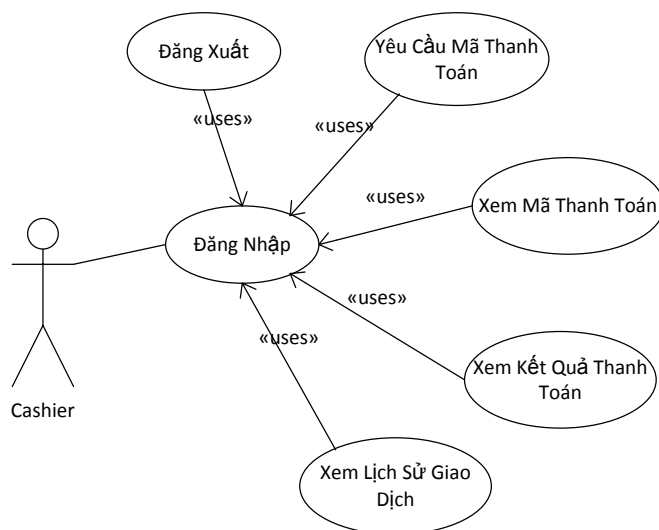


Hình 3-5: Sơ đồ UseCase phân hệ khách hàng mô hình đề xuất 2

Các chức năng chính (những chức năng đã mô tả ở Mô hình thanh toán 1 sẽ không trình bày lại):

- Nhập mã PIN.
- Thay đổi mã PIN.
- Đăng Nhập/ Đăng Xuất.
- Đăng ký mới tài khoản.
- Quản lý thông tin người dùng
- Quản lý thông tin thẻ ngân hàng (thêm/xóa).
- Kiểm tra mã thanh toán.
- Hiện thị thông tin hóa đơn thanh toán.
- Xác nhận thông tin thanh toán.
- Xem lịch sử giao dịch.
- Xem danh sách cửa hàng có hỗ trợ thanh toán bằng mô hình này.
- Xem danh sách cửa hàng quanh vị trí người đứng với công nghệ AR.
- Cài đặt, thay đổi ngôn ngữ hiển thị...
- Trợ giúp: hướng dẫn, trợ giúp người dùng.

3.3.2.3. Web Client ở cửa hàng.



Hình 3-6: Sơ đồ UseCase phân hệ nhân viên thanh toán mô hình đề xuất 2

- Đăng nhập/Đăng xuất.
- Yêu cầu mã thanh toán.
- Hiện thị thông tin mã thanh toán.
- Thông báo kết quả thanh toán.
- Xem lịch sử giao dịch.

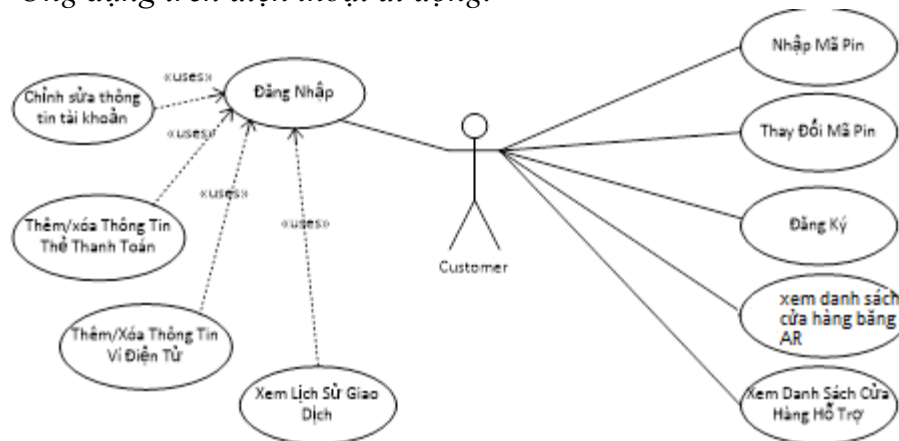
3.3.3. **Mô hình thanh toán 3**

3.3.3.1. *API được server cung cấp*

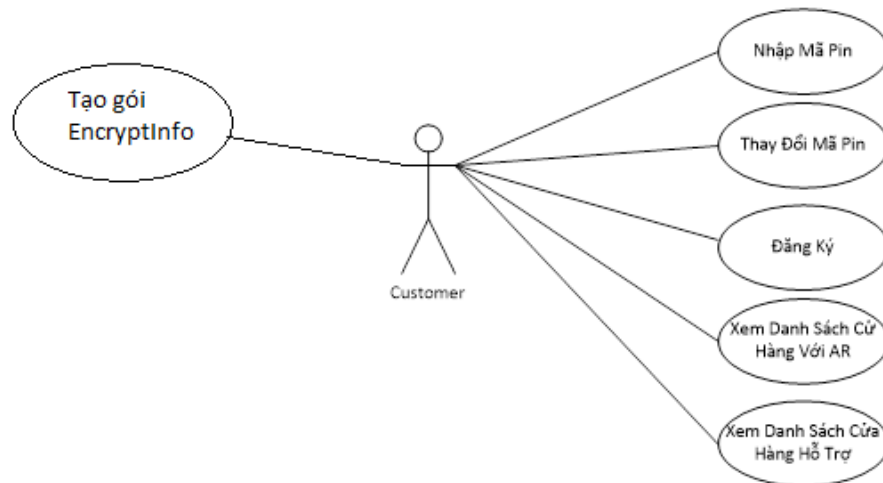
Các dịch vụ chính được API cung cấp:

- API for App:
 - Đăng nhập/ đăng xuất (có Internet).
 - Đăng ký tài khoản mới (có Internet).
 - Chỉnh sửa thông tin cá nhân (có Internet).
 - Thêm/xóa thông tin ngân hàng cá nhân (khi có Internet).
 - Thêm/xóa thông tin ví điện tử cá nhân (khi có Internet).
 - Thông báo kết quả thanh toán thông qua SMS.
 - Cung cấp lịch sử giao dịch (khi có Internet).
 - Trả về thông tin mã lỗi, chi tiết lỗi trong quá trình giao dịch (khi có Internet).
- API for Shop:
 - Đăng nhập/ đăng xuất.
 - Tiếp nhận yêu cầu thanh toán có kèm theo gói Encrypt Info.
 - Thông báo kết quả giao dịch.
 - Cung cấp lịch sử giao dịch.
 - Trả về thông tin mã lỗi, chi tiết lỗi trong quá trình giao dịch.
- Giao tiếp với các bên cung cấp dịch vụ.
 - Gửi và tiếp nhận thông tin thanh toán với các dịch vụ ngân hàng.
 - Gửi và tiếp nhận thông tin thanh toán với các dịch vụ ví điện tử.
 - Giao tiếp với bên cung cấp dịch vụ SMS.
 - Gửi yêu cầu hiệu chỉnh, trích xuất dữ liệu với Database.

3.3.3.2. Ứng dụng trên điện thoại di động.



Hình 3-7: Sơ đồ UseCase phân hệ khách hàng mô hình thanh toán đề xuất 3 khi có Internet



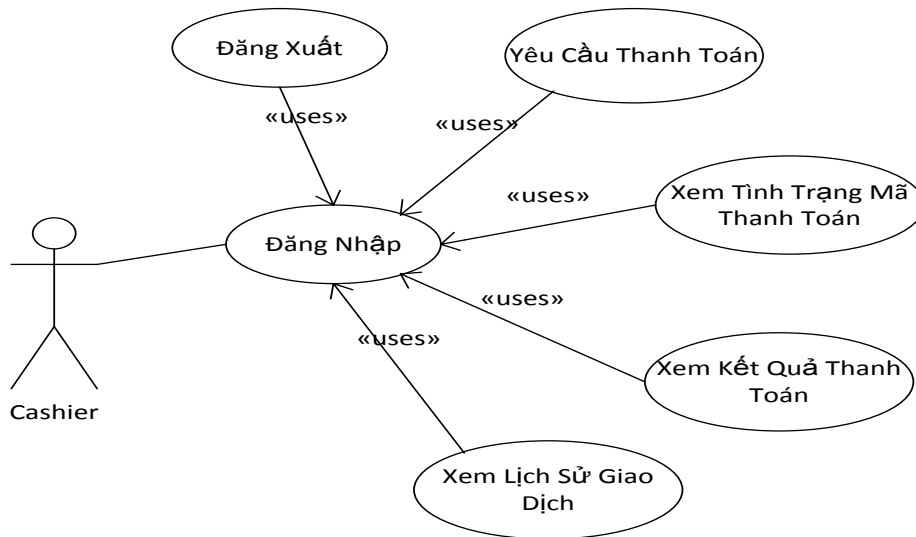
Hình 3-8: Sơ đồ UseCase phân hệ khách hàng mô hình thanh toán đề xuất 3 khi người dùng không có Internet.

Các chức năng chính (những chức năng đã mô tả ở Mô hình thanh toán 1 sẽ không trình bày lại):

- Nhập mã PIN
- Thay đổi mã PIN
- Đăng Nhập/ Đăng Xuất (có Internet)
- Đăng ký mới tài khoản (có Internet)
- Quản lý thông tin người dùng (có Internet)
- Tiến hành thanh toán
- Xem lịch sử giao dịch
- Xem danh sách cửa hàng có hỗ trợ thanh toán bằng mô hình này
- Xem danh sách cửa hàng quanh vị trí người đứng với công nghệ AR

- Quản lý thông tin thẻ ngân hàng (thêm/xóa) (có Internet)
- Quản lý thông tin ví điện tử (thêm/xóa) (có Internet)
- Cài đặt, thay đổi ngôn ngữ hiển thị...
- Trợ giúp: hướng dẫn, trợ giúp người dùng

3.3.3.3. Web Client ở cửa hàng.



Hình 3-9: Sơ đồ UseCase phân hệ nhân viên thanh toán mô hình thanh toán đề xuất 3

Các chức năng chính:

- Đăng nhập/Đăng xuất.
- Nhận gói dữ liệu Encrypt info, nhập thông tin hóa đơn và tiến hành thanh toán.
- Thông báo kết quả thanh toán.
- Xem lịch sử giao dịch.

3.3.4. Các chức năng khác của ứng dụng trên điện thoại di động.

Ngoài một số chức năng chính phục vụ cho quy trình thanh toán, ứng dụng trên điện thoại còn tích hợp một số chức năng tiện ích khác như:

- | | |
|---------------------------------------|--------------------------------|
| ○ Xác lập Mã pin | ○ AR - Thực tại tăng cường |
| ○ Đăng ký | ○ Slider Menu chức năng |
| ○ Đăng nhập, Đăng xuất | ○ Cài đặt tài khoản |
| ○ Quên mật khẩu | ○ Quản lý tài khoản ví điện tử |
| ○ Hiện thị bản đồ cửa hàng xung quanh | ○ Quản lý thẻ thanh toán |

3.4. Các vấn đề chính khi xây dựng Server cung cấp API – Web API

3.4.1. *Ngôn ngữ, nền tảng*

Web Services:

- Nền tảng: ASP.NET 4.5
- Kiến trúc: MVC Web, Web API
- Công cụ phát triển: Microsoft Visual Studio 2012
- Cơ sở dữ liệu: Microsoft SQL 2008

3.4.2. *Lý do chọn ASP.NET MVC 4*

ASP.NET MVC 4 là một bộ khung (framework) cho phép xây dựng những ứng dụng web chuẩn, có khả năng mở rộng, sử dụng các mẫu thiết kế (design patterns) tốt và sức mạnh của ASP.NET cũng như .NET Framework.

ASP.NET MVC được thiết kế một cách gọn nhẹ, giảm thiểu sự phức tạp của việc xây dựng ứng dụng web bằng cách chia 1 ứng dụng thành 3 tầng (layer): Model, View và Controller. Sự chia nhỏ này giúp lập trình viên dễ dàng kiểm soát các thành phần khi phát triển, cũng như lợi ích lâu dài trong việc kiểm tra, bảo trì và nâng cấp.

Nền tảng ASP.NET MVC mang lại những lợi ích sau:

- Dễ dàng quản lý sự phức tạp của ứng dụng bằng cách chia ứng dụng thành ba thành phần model, view, controller
- Nó không sử dụng view state hoặc server-based form. Điều này tốt cho những lập trình viên muốn quản lý hết các khía cạnh của một ứng dụng.
- Nó sử dụng mẫu Front Controller, mẫu này giúp quản lý các requests (yêu cầu) chỉ thông qua một Controller. Nhờ đó bạn có thể thiết kế một hạ tầng quản lý định tuyến
- Hỗ trợ tốt hơn cho mô hình phát triển ứng dụng hướng kiểm thử
- Nó hỗ trợ tốt cho các ứng dụng được xây dựng bởi những đội có nhiều lập trình viên và thiết kế mà vẫn quản lý được tính năng của ứng dụng

3.4.3. **Thống nhất cấu trúc gói tin Web API gửi về Web Shop và App viPay**

Khi client nhận gói tin trả về từ server bằng JSON gặp phải khó khăn là client không biết được đối tượng thuộc lớp nào, đối tượng trả về là thông điệp thành công hay thất bại và mỗi hàm sẽ phát sinh ra một gói tin với lớp khác nhau dẫn đến khó thống nhất sau này, do đó để dễ dàng thống nhất trong việc truyền các gói tin theo nhiều kiểu khác nhau có thể sử dụng Design Pattern mẫu Template Method [2] với các lớp như sau:

- Lớp `DataPackage`, server trả về để cho biết là hàm đó thực hiện thành công hay thất bại

```
// Lớp DataPackage tổng quát
public abstract class DataPackage {
    public bool Success { get; set; } // Gói tin thành công hay thất bại
}
```

Hình 3-10: Mã nguồn lớp `DataPackage`

- Lớp `DataPackageObjectSuccess`, server trả về gói tin thành công với dữ liệu là về 1 đối tượng `Data` có kiểu là `T`

```
// Lớp DataPackageObjectSuccess trả về 1 đối tượng Data có kiểu là T
public class DataPackageObjectSuccess<T>: DataPackage {
    public T Data { get; set; } // Đối tượng trả về
}
```

Hình 3-11: Mã nguồn lớp `DataPackageObjectSuccess`

- Lớp `DataPackageArraySuccess`, server trả về gói tin thành công với dữ liệu là danh sách các đối tượng trong `Data` có kiểu là `T`

```
// Lớp DataPackageArraySuccess trả về 1 danh sách đối tượng trong Data có kiểu là T
public class DataPackageArraySuccess<T> : DataPackage {
    public List<T> Data { get; set; } // Danh sách đối tượng trả về
    public int Count { get; set; } // Số lượng đối tượng trả về
}
```

Hình 3-12: Mã nguồn lớp `DataPackageArraySuccess`

- Lớp `DataPackageError`, server trả về gói tin thất bại với mã lỗi và chuỗi mô tả lỗi đó

```
// Lớp DataPackageError trả về mã lỗi và chuỗi mô tả lỗi đó
public class DataPackageError : DataPackage {
    public int Error { get; set; } // Mã lỗi
    public string Message { get; set; } // Chuỗi mô tả lỗi
}
```

Hình 3-13: Mã nguồn lớp `DataPackageError`

3.4.4. Sử dụng thư viện *ASP.Net SignalR* thực hiện liên kết thời gian thực giữa Web API - ASP.Net với Web Client- ASP.Net, giữa Web API - ASP.Net với Mobile application – Android.

Để server thông báo tức thì kết quả giao dịch tới Web Shop cũng như *viPay App*, cần tới thư viện hỗ trợ Real-time web, ở đây sử dụng thư viện *ASP.NET SignalR*.

SignalR là một thư viện lập trình do 2 thành viên của đội phát triển *ASP.net* Microsoft phát triển. *SignalR* cho phép lập trình viên tạo kênh liên lạc giữa server và client theo cơ chế notification [13]. Trong môi trường *ASP.net* chỉ cần bổ xung thư viện *SignalR* Server lập trình bằng *C#*, còn tại browser (hay mobile) dùng thư viện *javascript SignalR Client*.

Mô hình bình thường client gửi request đến server. Nhưng với *SignalR*, Server có thể “push” trực tiếp dữ liệu về client, theo thời gian thực. Ngoài ra, mô hình này cũng giúp ta xây dựng ứng dụng bất đồng bộ lên đến hàng nghìn kết nối 1 lúc với máy chủ.

Vận dụng *SignalR* vào mô hình.

- Server: cài đặt gói thư viện *SignalR* từ NuGet với dòng lệnh `Install-Package Microsoft.AspNet.SignalR`
 - Khai báo, cấu hình *SignalR*.


```
//Khai báo, cấu hình SignalR tại lớp Startup.
public class Startup {
    public void Configuration(IApplicationBuilder app) {
        app.Map("/signalr", map => {
            map.UseCors(CorsOptions.AllowAll);
            var hubConfiguration = new HubConfiguration {
            };
            map.RunSignalR(hubConfiguration);
        });
    }
}
```

Hình 3-14: Mã nguồn lớp Startup (Server)

- Khai báo lớp MyHub (kế thừa từ lớp Hub) chứa các hàm, Client và Server sử dụng các hàm này để có thể giao tiếp với nhau:

```
// Lớp MyHub khai báo các function, Server và Client sử dụng các function này để giao
tiếp với nhau.
public class MyHub : Hub {
    public void registerConnectionId(int id) {
        Groups.Add(Context.ConnectionId, id.ToString());
    }
    public void BroadcastMessageToAll(string message) {
        Clients.All.newMessageReceived(message);
    }
    public void RemoveFromAGroup(string group) {
        Groups.Remove(Context.ConnectionId, group);
    }
    public void BroadcastToGroup(string message, string group) {
        Clients.Group(group).newMessageReceived(message);
    }
}
```

Hình 3-15: Mã nguồn lớp MyHub

- Client (Web-Browser): cài đặt gói thư viện SignalR từ NuGet với dòng lệnh `Install-Package Microsoft.AspNet.SignalR`
 - Khai báo, cấu hình SignalR.

```
//Khai báo, cấu hình SignalR tại lớp Startup.
public class Startup {
    public void Configuration(IApplicationBuilder app) {
        app.Map("/signalr", map => {
            map.UseCors(CorsOptions.AllowAll);
            var hubConfiguration = new HubConfiguration {
            };
            map.RunSignalR(hubConfiguration);
        });
    }
}
```

Hình 3-16: Mã nguồn lớp Startup (Web-Client)

- Tiến hành kết nối với Server có cung cấp SignalR, sử dụng các hàm có sẵn ở Server hay khai báo hàm mới để Server có thể sử dụng, nhằm mục đích giao tiếp giữa Server và Client.

```
//Đoạn javascript được viết trực tiếp tại View (trang html).
<script>
    $(function () {
        var myChatHub = $.connection.myChatHub;
        //Khai báo hàm để Server sử dụng
        myChatHub.client.message = function (condition, message) {
        }
        $.connection.hub.url = ["host"]/signalr";
        $.connection.hub.start().done(function () {
            //gọi hàm đã khai báo ở Server
            myChatHub.server.registerConnecttionId(data);
        }).fail(function (error) {
        });
    });
</script>
```

Hình 3-17: Mã nguồn JavaScript trên Web-Client

- Mobile Application (Android):

```

protected static HubConnection conn = null;
protected IHubProxy hub = null;
@Override
public void SignalRConnect(HubOnDataCallback callback) {
    if (conn != null)
        return;
    // Khai báo Uri đến Server
    final Uri address = Uri.parse(strServerURL);
    conn = new HubConnection(address.toString(), activity, new
LongPollingTransport()) {
        @Override
        public void OnStateChanged(StateBase oldState, StateBase
newState) {
            switch (newState.getState()) {
            case Connected:
                HubInvokeCallback callback = new
                HubInvokeCallback() {
                    @Override
                    public void OnResult(boolean success,
                    String resp) {}
                };
                List<Integer> arggg = new
                    ArrayList<Integer>(1);
                arggg.add(userID);
                // Gọi hàm "registerConnecttionId" đã khai
                    báo ở Server
                hub.Invoke("registerConnecttionId", arggg
                    , callback);
                break;
            default:
                break;
            }
        }
    };
    try {
        hub = conn.CreateHubProxy("MyChatHub");
    } catch (OperationApplicationException e) {
    }

    hub.On("showBill", callback);
    conn.Start();
}

```

Hình 3-18: Mã nguồn Android khai báo và sử dụng SignalR

```
//Khai báo hàm cho Server sử dụng
private HubOnDataCallback showBill = new HubOnDataCallback() {
    @Override
    public void OnReceived(JSONArray arg0) {
        // Xử lý dữ liệu arg0 từ Server gửi về
    };
}
```

Hình 3-19: Mã nguồn Android khai báo hàm cho Server sử dụng trong SignalR

3.4.5. *Web API – Giải mã gói tin với thuật toán RSA*

Trong mô hình thanh toán 3 để đảm bảo tính bảo mật khi truyền và nhận Mã thanh toán cần sử dụng thuật toán RSA để mã hóa Mã thanh toán. Khi đó, viPay App dùng Public Key để mã hóa Mã thanh toán và Web API dùng Private Key để giải mã, nhận được các thông tin cần thiết

RSA là một thuật toán mật mã hóa khóa công khai. Đây là thuật toán đầu tiên phù hợp với việc tạo ra chữ ký điện tử đồng thời với việc mã hóa. Nó đánh dấu một sự tiến bộ vượt bậc của lĩnh vực mật mã học trong việc sử dụng khóa công cộng. RSA đang được sử dụng phổ biến trong thương mại điện tử và được cho là đảm bảo an toàn với điều kiện độ dài khóa đủ lớn [12]

Trong ASP.NET sử dụng thư viện System.Security.Cryptography để thực hiện việc mã hóa RSA, ngoài ra nhóm tác giả còn sử dụng thư viện được cung cấp tại: <http://www.codeproject.com/Articles/421656/RSA-Library-with-Private-Key-Encryption-in-Csharp>

3.4.6. Web API – Xử lý trường hợp trùng code

```
//Kiểm tra mã thanh toán đã tồn tại chưa, nếu chưa tồn tại sẽ cấp mã thanh toán
này cho người dùng.
while (DataStatic.virtualDetail.ContainsKey(newcode)) {
    temp = DataStatic.virtualDetail[newcode];
    // Trường hợp mã thanh toán đã tồn tại, kiểm tra mã thanh toán đã hết hạn
    chưa, nếu hết hạn thì hủy mã thanh toán .
    if (temp.code.Create.AddMinutes(temp.code.Duration) > DateTime.Now){
        DataStatic.virtualDetailV2.Remove(newcode);
        break;
    }
    else {
        newcode = newCode();
    }
}
// Sau khi đã kiểm tra và bảo đảm mã thanh toán chưa tồn tại, mã thanh toán được
trả về cho người sử dụng, và lưu lại thông tin tồn tại của mã thanh toán.
DataStatic.virtualDetail.Add(newcode);
```

Hình 3-20: Mã nguồn ASP.Net xử lý trùng mã thanh toán

3.5. Các vấn đề chính khi xây dựng Website cửa hàng – Web Shop (ngoài một số vấn đề đã được nêu ở từng mô hình)

3.5.1. Ngôn ngữ, nền tảng

- Nền tảng: ASP.NET 4.5
- Kiến trúc: MVC Web
- Công cụ phát triển: Microsoft Visual Studio 2012

3.5.2. Xử dụng AJAX.

- Web view:

```
function CheckCode() {
    $.ajax({
        url: 'Home/CheckCode',
        type: 'GET',
        data: { amount: amount, detail: detail, code:code},
        dataType: 'json',
        cache: false,
        success: function (data) {
            // Xử lý khi kết quả trả về thành công
        }
    });
}
```

Hình 3-21: Mã nguồn JavaScript sử dụng Ajax trên Web-Client

- Controller:

```

public ActionResult CheckCode(int amount, string detail,int code)    {
    Try{
        //Thao tác, xử lý dữ liệu từ Web View
    }
    catch    {
        return Json("logout", JsonRequestBehavior.AllowGet);
    }
}

```

Hình 3-22: Mã nguồn ASP.Net khai báo hàm sử dụng Ajax

3.5.3. **QR code**

```

// JavaScript sử dụng dịch vụ cung cấp bởi google tạo ra QR Code
<script>
var src =
"http://chart.apis.google.com/chart?chs=80x80&cht=qr&chld=|1&chl="
+ data;

$("#code").append("");
</script>

```

Hình 3-23: Mã nguồn JavaScript xử dụng dịch vụ tạo QR Code của Google

3.5.4. . **SignalR (đã đề cập mục 3.4.4)**

3.5.5. **Xây dựng Windows Form hỗ trợ Web Shop – WinForm Shop**

Hiện tại, trình duyệt web không hỗ trợ công nghệ NFC cũng như không có thư viện đọc QRCode miễn phí nên để dễ dàng cho việc áp dụng công nghệ NFC và QRCode cần có ứng dụng trung gian để chuyển Mã Thanh toán từ trình duyệt web tới viPay App hoặc ngược lại, do đó ứng dụng WinForm Shop được viết để giải quyết vấn đề này.

3.5.5.1. *Đọc QRCode sử dụng thư viện Zxing*

Zxing là một thư viện hỗ trợ giải mã và tạo mã vạch (ví dụ QR Code, PDF 417, EAN, UPC, Aztec, Data Matrix, Codabar) bằng hình ảnh [16]

Bộ giải mã hỗ trợ các mã vạch sau:

UPC-A, UPC-E, EAN-8, EAN-13, Code 39, Code 93, Code 128, ITF, Codabar, MSI, RSS-14 (all variants), QR Code, Data Matrix, Aztec and PDF-417.

Bộ mã hóa hỗ trợ các định dạng sau:

UPC-A, EAN-8, EAN-13, Code 39, Code 128, ITF, Codabar, Plessey, MSI, QR Code, PDF-417, Aztec, Data Matrix

```
// Timer có nhiệm vụ nhận hình từ Webcam, giải mã và gửi Mã thanh toán
void webCamTimer_Tick(object sender, EventArgs e){
    var bitmap = wCam.GetCurrentImage(); // Lấy hình từ webcam
    if (bitmap == null)
        return;
    var reader = new BarcodeReader();
    var result = reader.Decode(bitmap); // Giải mã
    if (result != null)
    {
        webCamTimer.Stop();
        this.Invoke((MethodInvoker)delegate
        {
            SendText(result.Text); // Gửi Mã thanh toán
        });
    }
}
```

Hình 3-24: Mã nguồn Timer nhận hình từ Webcam, giải mã và gửi Mã thanh toán

3.5.5.2. *Sử dụng NFC trên Windows 8*

NFC (Near-Field Communications) là công nghệ kết nối không dây phạm vi tầm ngắn trong khoảng cách < 2 cm, sử dụng cảm ứng từ trường để thực hiện kết nối và truyền dữ liệu giữa các thiết bị khi có sự tiếp xúc trực tiếp hay để gần nhau [9]

Hệ điều hành Windows 8 có hỗ trợ NFC trong thư viện Windows.Networking.Proximity thuộc lớp Windows Runtime hỗ trợ các kết nối giữa các thiết bị có trong phạm vi gần với nhau. Bằng cách sử dụng API này, ứng dụng có thể thiết lập một kết nối và trao đổi dữ liệu với nhau

```

void _proximityDevice_DeviceArrived(ProximityDevice sender){
    NfcDetected = true;
    if (Environment.GetCommandLineArgs().Length <= 0)
        return;
    string Arg = Environment.GetCommandLineArgs()[1];
    string Pattern = "code=\\d{5}$";
    if (!System.Text.RegularExpressions.Regex.IsMatch(Arg, Pattern))
        return;
    Code = Regex.Match(Regex.Match(Arg, Pattern).Value, "\\d{5}$").Value;
    ThreadStart work = delegate {
        while (NfcDetected) {
            DoWriteTagBackground();
        }
    };
    new Thread(work).Start();
}

```

Hình 3-25: Mã nguồn phát hiện 2 thiết bị NFC chạm vào nhau

```

private void MessageReceivedHandler(ProximityDevice sender, ProximityMessage
message){
    string receivedString="";
    try{
        using (var reader = DataReader.FromBuffer(message.Data)) {
            reader.UnicodeEncoding =
Windows.Storage.Streams.UnicodeEncoding.Utf16LE;
            receivedString = reader.ReadString(reader.UnconsumedBufferLength / 2
- 1);
        }
    }
    catch (Exception e){ }
    this.Invoke((MethodInvoker)delegate {
        SendText(receivedString);
    });
}

```

Hình 3-26: Mã nguồn nhận thông điệp từ thiết bị NFC khác gửi tới


```
private void DoWriteTagBackground(){
    try{
        using (var writer = new DataWriter { UnicodeEncoding =
Windows.Storage.Streams.UnicodeEncoding.Utf16LE })
        {
            writer.WriteString(Code);
            long id = _proximityDevice.PublishBinaryMessage("WindowsUri",
writer.DetachBuffer());
            _proximityDevice.StopPublishingMessage(id);
        }
        this.Invoke((MethodInvoker)delegate {
            Console.Beep();
            Application.Exit();
        });
        NfcDetected = false;
    }
    catch (Exception e){}
}
```

Hình 3-27: Mã nguồn gửi thông điệp tới thiết bị NFC khác

3.5.5.3. Gửi thông điệp từ trình duyệt web tới Windows Form

Để có thể kích hoạt ứng dụng WinForm Shop cần phải cấu hình trong Registry như hình bên dưới và trong Web cần gọi theo liên kết “viPay:sendcode?message=” sau từ “message=” có thể là Mã thanh toán để gửi tới viPay App

```
Windows Registry Editor Version 5.00
[HKEY_CLASSES_ROOT\viPay]
@="URL: viPay Protocol"
"URL Protocol"=""
[HKEY_CLASSES_ROOT\viPay\shell]
[HKEY_CLASSES_ROOT\viPay\shell\open]
[HKEY_CLASSES_ROOT\viPay\shell\open\command]
@="\"E:\ShopWinForm.exe\" %1"
```

Hình 3-28: Cấu hình ứng dụng WinForm Shop trong Registry

3.5.5.4. Gửi thông điệp từ Windows Form tới trình duyệt web

Để gửi thông điệp từ WinForm Shop tới trình duyệt web cần phải sao chép chuỗi thông điệp đó vào Clipboard và sử dụng tổ hợp phím Ctrl+V để dán vào ô trong trình duyệt web như đoạn mã mô tả trong hình...

```

void SendText(string text){
    Clipboard.SetText(text); // Sao chép và Clipboard
    SendKeys.Send("^{v}"); // Tổ hợp phím Ctrl+V
    Console.Beep();
    Application.Exit();
}

```

Hình 3-29: Mã nguồn gửi thông điệp từ WinForm Shop tới trình duyệt web

3.6. Các vấn đề chính khi xây dựng ứng dụng trên nền tảng Android – App viPay

3.6.1. Ngôn ngữ, nền tảng

- Nền tảng: Android
- Phiên bản sử dụng: >= Android 4.0
- Công cụ phát triển: Eclipse

3.6.2. Lý do chọn hệ điều hành Android

Hãng nghiên cứu thị trường International Data Corporation (IDC) vừa công bố tình hình thị trường smartphone trong 3 tháng của quý III/2013, tính đến hết tháng 9 vừa qua. Đáng chú ý nhất trong đó nền tảng Android của Google đã lần đầu tiên đạt đến mức 81% thị phần smartphone trên toàn cầu, cao nhất từ trước đến nay.

Theo IDC, sở dĩ Android có thể chiếm dụng một lượng thị trường lớn như vậy nhờ vào sự đang dạng trong các hãng sản xuất, trong đó Samsung vẫn tiếp tục là hãng đi tiên phong trong sản xuất smartphone chạy Android khi chiếm đến 39,9% số lượng Android được bán ra trên thị trường, trong khi các hãng sản xuất smartphone Android khác cũng đang có những chỗ đứng nhất định trên thị trường.

So sánh tổng thể thì nhóm tác giả thấy smartphone sử dụng hệ điều hành Android là sự lựa chọn tối ưu cho các chức năng và hoạt động của ứng dụng viPay.

Nền tảng	Quý III/2013 (triệu máy)	Thị phần quý III/2013	Quý III/2012 (triệu máy)	Thị phần quý III/2012	Tỷ lệ thay đổi
Android	211,6	81,0%	139,9	74,9%	51,3%
iOS	33,8	12,9%	26,9	14,4%	25,6%
WP	9,5	3,6%	3,7	2%	156,0%

BlackBerry	4,5	1,7%	7,7	4,1%	-41,6%
Khác	1,7	0,6%	8,4	4,5%	-80,1%
Tổng cộng	261,1	100%	186,7	100%	39,9%

Bảng 3-9: Bảng chi tiết thị phần các nền tảng di động trên thị trường smartphone

(Nguồn: <http://dantri.com.vn/suc-manh-so/android-thong-linh-thi-truong-smartphone-voi-81-thi-phan-toan-cau-802376.htm>)

3.6.3. Truy vấn tới Web API

Để truy vấn tới Web API trong Android có hỗ trợ lớp DefaultHttpClient để thực hiện đóng gói đối tượng và thực thi các request lên Server với các loại request thuộc các lớp HttpGet, HttpPost, HttpPut, HttpDelete

Nhóm đã giả lập các hàm trong lớp IServer giống như trên Web API để tiện và dễ dàng cho việc sử dụng. Cài đặt cụ thể các hàm được cài đặt trong lớp Server, trong lớp này sẽ kết nối và thực hiện gọi các hàm ở Web API thông qua lớp RestClient.

```
public abstract interface IServer {
    public DataPackage getBill(int code);
    public DataPackage getShop(double Latitude, double Longitude);
    public DataPackage getHistory();
    public DataPackage Logout(Activity activity);
    public DataPackage Login(String mobileNumber, String password);
    public DataPackage Register(User user);
    public DataPackage ConfirmSMS(int Code);
    public DataPackage ReCodeSMS();
    public DataPackage getListBank();
    public DataPackage RegisterBankCard(BankInfo bank);
    public DataPackage getUserListBank();
    public DataPackage DeleteUserBank(String token);
    public DataPackage RegisterWallet(WalletAccount wallet);
    public DataPackage getUserListWallet();
    public DataPackage DeleteUserWallet(String token);
    public DataPackage Update(User user);
    public DataPackage RegisterGCM(String idGCM);
    ...
}
```

Hình 3-30: Mã nguồn một số hàm trong interface IServer

3.6.4. **Đọc và ghi dữ liệu JSON sử dụng thư viện Gson**

JSON (JavaScript Object Notation) là cú pháp để lưu trữ và trao đổi thông tin văn bản, được sử dụng rất rộng rãi trong việc truyền và nhận thông tin từ server hiện nay. Do đó, trong ứng dụng nhóm sử dụng JSON để thực hiện việc gửi và nhận dữ liệu giữa client và server

Google Gson là một java library hỗ trợ việc chuyển đổi Java Objects sang định dạng JSON tương ứng, và ngược lại, nó cũng có thể sử dụng để chuyển đổi từ JSON sang Java Objects. [5]

```
class BagOfPrimitives {  
    private int value1 = 1;  
    private String value2 = "abc";  
    private transient int value3 = 3;  
    BagOfPrimitives() {  
        // no-args constructor  
    }  
}
```

```
(Serialization)  
BagOfPrimitives obj = new BagOfPrimitives();  
Gson gson = new Gson();  
String json = gson.toJson(obj);  
==> Chuỗi json sẽ là {"value1":1,"value2":"abc"}
```

Lưu ý là bạn không thể serialize các đối tượng tham chiếu vòng vì sẽ dẫn đến đệ quy vô hạn.

```
(Deserialization)  
BagOfPrimitives obj2 = gson.fromJson(json, BagOfPrimitives.class);  
==> obj2 giống như obj
```

Hình 3-31: Mã nguồn ví dụ sử dụng thư viện Gson

3.6.5. **Push notification sử dụng Google Cloud Messaging**

Google Cloud Messaging (GCM) là dịch vụ cho phép gửi dữ liệu từ server đến các thiết bị Android, và ngược lại. GCM hoàn toàn miễn phí và không giới hạn băng thông. Dịch vụ hoạt động trên các gói dữ liệu có dung lượng nhỏ hơn 4kb và tin nhắn tới thiết bị Android là tức thời (Push-notification). GCM phù hợp cho việc xây dựng các ứng dụng nhắn tin tức thời hoặc tương tác giữa người dùng và nhà phát triển ứng dụng. [3]

Để sử dụng GCM làm theo các bước cơ bản sau:

Bước 1: Tạo một Google API project trên Google Developers Console, lưu lại **Project Number**; Kích hoạt Google Cloud Messaging for Android lên ON và **lấy API Key**

Bước 2: Tại server gửi request tới <https://android.googleapis.com/gcm/send> với:

- API Key vừa lấy được ở Bước 1
- Danh sách các thiết bị đăng ký registration_ids
- Dữ liệu gửi tới client

Bước 3: Tại client sử dụng thư viện GCM để nhận thông điệp

- Cần phải khai báo các thông tin trong AndroidManifest như hình sau:

```
<service android:name="com.mobile_payment2.GCMIntentService" />
    <receiver
        android:name="com.google.android.gcm.GCMBroadcastReceiver"
        android:permission="com.google.android.c2dm.permission.SEND" >
        <intent-filter>
            <action android:name="com.google.android.c2dm.intent.RECEIVE"
        />
            <action
        android:name="com.google.android.c2dm.intent.REGISTRATION" />
            <category android:name="com.mobile_payment2" />
        </intent-filter>
    </receiver>
```

Hình 3-32: Mã nguồn khai báo GCM trong AndroidManifest.xml

- Để nhận được thông điệp cần tạo ra 1 lớp GCMIntentService kế thừa từ lớp GCMBroadcastReceiver như trong hình sau:

```

public class GCMIntentService extends GCMBaseIntentService {
    // SENDER_ID chính là Project Number
    public static final String SENDER_ID = "552171066374";
    public GCMIntentService() {
        super(SENDER_ID);
    }
    @Override
    protected void onRegistered(Context context, String registrationId) { //
registrationId đăng ký với Server
    }

    @Override
    protected void onUnregistered(Context context, String registrationId) {
    }

    @Override
    protected void onMessage(Context context, Intent data) {
        // Nhận thông tin được push từ Server
    }
}

```

Hình 3-33: Mã nguồn sử dụng thư viện GCM

3.6.6. *Hiện thị bản đồ với Google Maps*

Để sử dụng Google Maps cần làm theo các bước sau: (chi tiết có thể xem tại: <https://developers.google.com/maps/documentation/android/start>)

Bước 1: Trong Google Developers Console tại trang Service, kích hoạt "Google Maps Android API v2" sau đó lấy API Key

Key for Android applications

API KEY	AlzaSyCYfof49zXyWUqJOxEu8Recf-IOTS_k6a8
ANDROID APPLICATIONS	7C:2C:D9:5D:93:8D:12:EC:0C:CB:65:0B:37:E6:FC:98:2E:03:04:46;com.mobile_payment
ACTIVATION DATE	Jun 25, 2014 2:51 AM
ACTIVATED BY	kycomputer555@gmail.com (you)

Edit allowed Android applications

Regenerate key

Delete

Hình 3-34: Màn hình lấy API Key cho Google Maps API

Bước 2: Khai báo các thông tin trong AndroidManifest như sau:

```

<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission
android:name="com.mobile_payment2.permission.MAPS_RECEIVE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE"/>
<permission
android:name="com.newway.mobile_payment2.permission.MAPS_RECEIVE"
android:protectionLevel="signature" />
<application >
<meta-data
android:name="com.google.android.maps.v2.API_KEY"
android:value="Key API" />
<meta-data
android:name="com.google.android.gms.version"
android:value="@integer/google_play_services_version" />
</application />

```

Hình 3-35: Mã nguồn khai báo Google Maps API trong AndroidManifest.xml

Bước 3: Thêm Fragment trong layout XML file như hình sau:

```

<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/map"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:name="com.google.android.gms.maps.MapFragment"/>

```

Hình 3-36: Mã nguồn Thêm Fragment Google Maps trong layout XML file

3.6.7. **Chỉ đường với Google API**

Google có cung cấp API miễn phí cho việc chỉ dẫn đường đi ngắn nhất từ vị trí này tới vị trí khác thông qua việc gọi tới đường dẫn theo cú pháp như hình sau:

```

// start: Tọa độ vị trí bắt đầu
// end: Tọa độ vị trí đích
String url = "http://maps.googleapis.com/maps/api/directions/xml?"
+ "origin=" + start.latitude + "," + start.longitude
+ "&destination=" + end.latitude + "," + end.longitude
+ "&sensor=false&units=metric&mode=driving&language=vi";

```

Hình 3-37: Cú pháp gọi hàm chỉ đường của Google API
Cấu trúc lớp GMapV2Direction được mô tả như trong hình sau:

```

public class GMapV2Direction {
// Hàm lấy danh sách chỉ đường
    public static Document getDocument(LatLng start, LatLng end, String mode);
    public static String getDurationText (Document doc) ;
    public static int getDurationValue (Document doc);
    public static String getDistanceText (Document doc);
    public static int getDistanceValue (Document doc);
    public static String getStartAddress (Document doc);
    public static String getEndAddress (Document doc);
    public static String getCopyRights (Document doc);
    public static ArrayList<LatLng> getDirection (Document doc);
    private static int getNodeIndex(NodeList nl, String nodename);
// Hàm lấy về danh sách tọa độ chỉ đường
    private static ArrayList<LatLng> decodePoly(String encoded);
}

```

Hình 3-38: Mã nguồn lớp chỉ đường GMapV2Direction

3.6.8. *Realtime giữa Web API và App viPay. (đã đề cập ở mục 3.4.4)*

3.6.9. *Mã hóa gói tin với thuật toán RSA*

Để thuận tiện và dễ dàng cho việc mã hóa gói tin, nhóm xây dựng lớp CryptoHelper chứa các hàm encodeDataToByteArray và encodeDataToString. Sử dụng lớp Cipher của Android để mã hóa gói tin theo thuật toán RSA với Public Key của Server

```

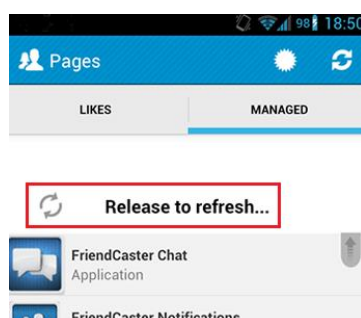
public class CryptoHelper {
    private static String mod = "";
    private static String ex = "";
    public static byte[] encodeDataToByteArray(String data) {
        BigInteger modulus = new BigInteger(Base64.decode(mod, Base64.DEFAULT));
        BigInteger exponent = new BigInteger(Base64.decode(ex, Base64.DEFAULT));
        PublicKey pubKey = null;
        byte[] encodedBytes = null;
        try {
            pubKey = KeyFactory.getInstance("RSA").generatePublic(new
RSAPublicKeySpec(modulus, exponent));
            Cipher c =
Cipher.getInstance("RSA/NONE/OAEPWithSHA1AndMGF1Padding");
            c.init(Cipher.ENCRYPT_MODE, pubKey);
            encodedBytes = c.doFinal(data.getBytes("UTF-8"));
        }
        return encodedBytes;
    }
}

```

Hình 3-39: Mã nguồn lớp CryptoHelper hỗ trợ cho việc mã hóa gói tin

3.6.10. **Kéo thả để cập nhật danh sách cho ListView**

Kéo và thả để cập nhật trong ListView là một tính năng rất hữu ích cho người sử dụng, làm tăng tính thân thiện với người dùng do đó nhóm sử dụng thư viện trên trang <https://github.com/chrisbanes/Android-PullToRefresh> để cho người dùng dễ dàng hơn cho việc sử dụng. Mô tả tính năng Pull To Refresh như trong hình sau:



Hình 3-40: Màn hình mô tả tính năng Pull To Refresh

Sử dụng thư viện bằng cách PullToRefreshListView vào trong Layout thay cho ListView như bình thường và bắt sự kiện setOnRefreshListener như hình sau:

```
lstResult.setOnRefreshListener(new OnRefreshListener() {  
    @Override  
    public void onRefresh(PullToRefreshBase refreshView) {  
        // Thực hiện hàm nào đó khi nhận được sự kiện kéo thả  
    }  
});
```

Hình 3-41: Mã nguồn mô tả sự kiện setOnRefreshListener của lớp PullToRefreshListView

3.6.11. **Xây dựng Slider Menu**

SlidingMenu là một thư viện mã nguồn mở Android cho phép các nhà phát triển dễ dàng tạo ra các ứng dụng với menu trượt tương tự những ứng dụng phổ biến như Google+, YouTube, Facebook,... Thư viện được tự do để thêm vào trong các ứng dụng Android với điều kiện trích dẫn dự án này và bao gồm các giấy phép trong ứng dụng. Tham khảo thêm thư viện tại: <https://github.com/jfeinstein10/SlidingMenu>

Để sử dụng thư viện: Trong Eclipse, chỉ cần thêm thư viện giống như là một dự án thư viện Android. Project > Clean để tạo ra những mã cơ bản trong Project, như R.java,... Sau đó, thêm SlidingMenu vào trong Project như trong hình sau:

```

public class SlidingExample extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setTitle(R.string.attach);
        // set the content view
        setContentView(R.layout.content);
        // configure the SlidingMenu
        SlidingMenu menu = new SlidingMenu(this);
        menu.setMode(SlidingMenu.LEFT);
        menu.setTouchModeAbove(SlidingMenu.TOUCHMODE_FULLSCREEN);
        menu.setShadowWidthRes(R.dimen.shadow_width);
        menu.setShadowDrawable(R.drawable.shadow);
        menu.setBehindOffsetRes(R.dimen.slidingmenu_offset);
        menu.setFadeDegree(0.35f);
        menu.attachToActivity(this, SlidingMenu.SLIDING_CONTENT);
        menu.setMenu(R.layout.menu);
    }
}

```

Hình 3-42: Mã nguồn ví dụ sử dụng thư viện SlidingMenu

3.6.12. **Đọc và xuất QRCode sử dụng thư viện Zxing**

Thư viện Zxing đã được mô tả trong mục 3.5.5.1, thư viện này hỗ trợ rất tốt cho nền tảng Android. Để sử dụng thư viện cần khai báo khai báo thư viện Zxing trong AndroidManifest như hình sau:

```

<activity
    android:name="com.google.zxing.client.android.CaptureActivity" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
    <intent-filter>
        <action android:name="eu.michalu.SCAN" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
    <intent-filter>
        <action android:name="com.google.zxing.client.android.SCAN" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>

```

Hình 3-43: Mã nguồn khai báo thư viện Zxing trong AndroidManifest.xml

Sau đó, để đọc hoặc tạo mã QRCode thì viết mã nguồn như hình sau:

```
// Khởi động Activity của thư viện Zxing để đọc QRCode
Intent intent = new Intent("eu.michalu.SCAN");
intent.putExtra("SCAN_MODE", "ONE_D_MODE,QR_CODE_MODE");
getActivity().startActivityForResult(intent, Constants.DIALOG_SCANNER_RESULT);

// Override lại phương thức onActivityResult trong Activity để nhận kết quả trả về
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == Constants.DIALOG_SCANNER_RESULT) {
        if (resultCode == Activity.RESULT_OK) {
            String contents = data.getStringExtra("SCAN_RESULT");
            // Contents là nội dung đọc được từ QRCode
        }
    }
}

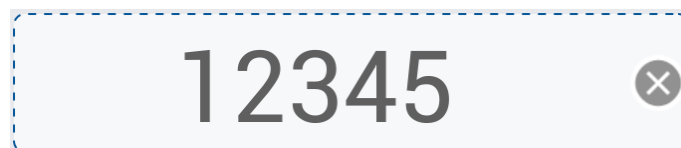
// Xuất QRCode ra Bitmap
public static Bitmap Create2DCode(String str, int w, int h) throws WriterException,
    UnsupportedEncodingException {
    BitMatrix matrix = new MultiFormatWriter().encode(
        new String(str.getBytes("GBK"), "ISO-8859-1"),
        BarcodeFormat.QR_CODE, w, h);
    int[] pixels = new int[width * height];
    int width = matrix.getWidth();
    int height = matrix.getHeight();
    for (int y = 0; y < height; y++) {
        for (int x = 0; x < width; x++) {
            if (matrix.get(x, y)) {
                pixels[y * width + x] = 0xff000000;
            }
        }
    }
    Bitmap bmp = Bitmap.createBitmap(width, height, Bitmap.Config.ARGB_8888);
    bmp.setPixels(pixels, 0, width, 0, 0, width, height);
    return bmp;
}
```

Hình 3-44: Mã nguồn đọc, tạo mã QRCode sử dụng thư viện Zxing

3.6.13. *Tùy biến EditText với ClearableEditText*

ClearableEditText là lớp tùy biến từ lớp EditText của Android, lớp này sử dụng thư viện của Alex Yanchenko, tham khảo tại địa chỉ:

<https://github.com/yanchenko/droidparts/blob/master/droidparts/src/org/droidparts/widget/ClearableEditText.java>



Hình 3-45: Màn hình ClearableEditText

3.6.14. **NFC**

Như đã giới thiệu NFC ở mục 3.5.5.2, trong Android có thư viện hỗ trợ công nghệ NFC thông qua lớp NfcAdapter. Để sử dụng công nghệ NFC đầu tiên trong ứng dụng cần khai báo trong AndroidManifest như trong Hình 3-46 sau đó để gửi và nhận thông qua NFC cần khai báo mã nguồn như trong Hình 3-47 và Hình 3-48

```
<uses-permission android:name="android.permission.NFC" />
```

Hình 3-46: Mã nguồn khai báo sử dụng NFC trong AndroidManifest.xml

```
// Tạo gói thông điệp gửi qua thiết bị NFC khác
mMessage = new NdefMessage(new NdefRecord[] {
    NdefRecord.createUri("Thông điệp") });

// Cấu hình NFC khi ứng dụng được kích hoạt
@Override
public void onResume() {
    super.onResume();
    if (adapterNFC != null)
        adapterNFC.enableForegroundNdefPush(this, mMessage);
}

// Ngừng NFC khi ứng dụng bị tạm dừng
@Override
public void onPause() {
    super.onPause();
    if (adapterNFC != null)
        adapterNFC.disableForegroundNdefPush(this);
}
```

Hình 3-47: Mã nguồn kích hoạt sử dụng NFC và gói thông điệp gửi thông qua NFC

```

@Override
protected void onNewIntent(Intent intent) {
    if (NfcAdapter.ACTION_TAG_DISCOVERED.equals(intent.getAction())) {
        mytag = intent.getParcelableExtra(NfcAdapter.EXTRA_TAG);
        Tag tag = mytag;
        Ndef ndef = Ndef.get(tag);
        if (ndef == null) {
            // NDEF is not supported by this Tag.
            return;
        }
        NdefMessage ndefMessage = ndef.getCachedNdefMessage();
        NdefRecord[] records = ndefMessage.getRecords();
        for (NdefRecord ndefRecord : records) {
            try {
                if (isReady) {
                    int code = Integer.parseInt(readText(ndefRecord));
                    GetBillTask task = new GetBillTask(code);
                    task.execute();
                }
            } catch (UnsupportedEncodingException e) {
            }
        }
    }
}

```

Hình 3-48: Mã nguồn nhận thông điệp từ thiết bị khác thông qua NFC

3.6.15. *Xây dựng thực tại ảo tăng cường (Augmented Reality) để xem danh sách các cửa hàng*

Khái niệm Augmented Reality (AR) có thể được dịch là "thực tại ảo tăng cường" hay "công nghệ tương tác thời gian thực", là công nghệ sử dụng các tín hiệu từ hình ảnh thực tế thông qua camera, định vị GPS, micro,... rồi cung cấp cho người dùng thông tin cụ thể từ những tín hiệu đó, tạo cho người dùng cảm giác như đang tương tác thật sự với những sự vật hiện tượng trước mắt thông qua thiết bị điện tử [11]

Nhóm sử dụng AR để xây dựng việc tìm kiếm các cửa hàng có hỗ trợ mô hình thanh toán cho khách hàng dễ dàng tìm kiếm tới. Nhóm tham khảo và kế thừa mã nguồn của Luận văn khóa trước đây [10]

Chương 4: GIỚI THIỆU PHẦN MỀM VÀ KHẢO SÁT KHÁCH HÀNG SỬ DỤNG PHẦN MỀM

✍ Chương này sẽ giới thiệu các chức năng của ứng dụng cũng như quy trình hoạt động của các mô hình. Bên cạnh đó, khảo sát ý kiến khách hàng về quy trình hoạt động của các mô hình và đánh giá tính năng, giao diện của ứng dụng

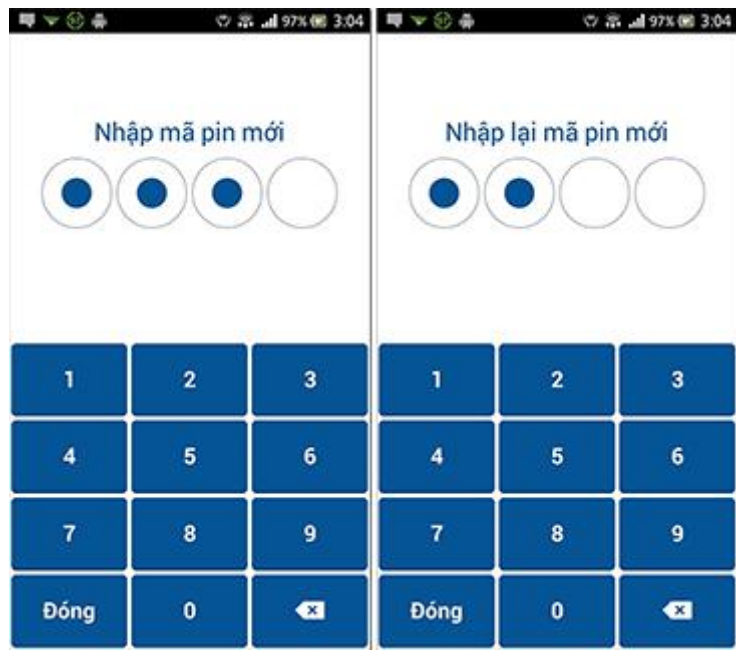
4.1. Giới thiệu ứng dụng trên hệ điều hành Android cho Khách hàng.

Ứng dụng đã được nhóm đưa lên chợ ứng dụng của Google theo địa chỉ sau:

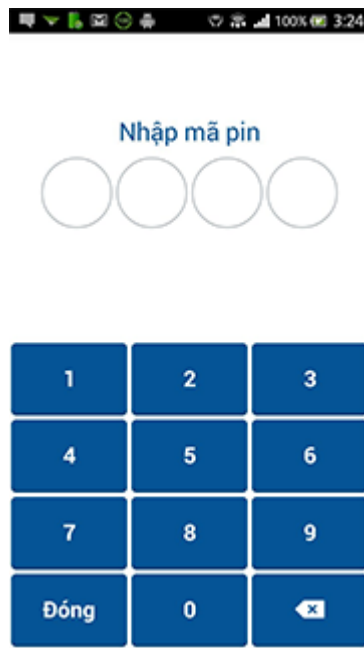
- Mô hình thanh toán 1:
https://play.google.com/store/apps/details?id=com.mobile_payment
- Mô hình thanh toán 2:
https://play.google.com/store/apps/details?id=com.mobile_payment2
- Mô hình thanh toán 3:
https://play.google.com/store/apps/details?id=com.mobile_payment3

4.1.1. **Xác lập Mã pin ban đầu và nhập mã Pin để vào ứng dụng**

Khi cài đặt xong, ứng dụng sẽ bắt người dùng phải cài đặt mã pin cho ứng dụng như hình 4.1. Từ lần sau, khi mở ứng dụng người dùng phải nhập mã pin đã cài đặt trước đó để vào ứng dụng như hình 4.2



Hình 4-1: Màn hình xác lập mã pin ban đầu

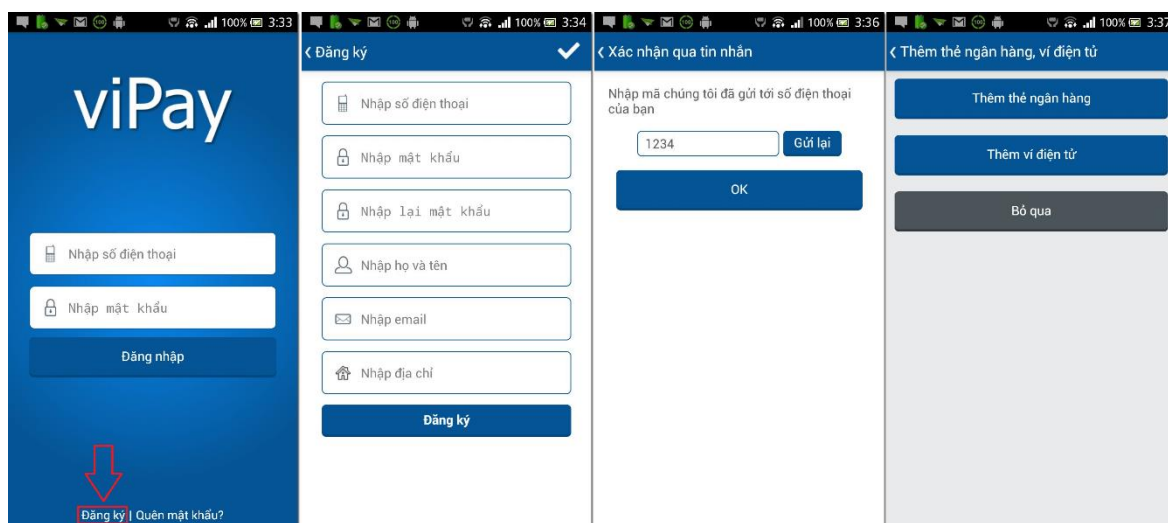


Hình 4-2: Màn hình nhập mã pin

4.1.2. **Đăng ký**

Để đăng ký mới tài khoản, tại màn hình đăng nhập người dùng nhấn chọn “Đăng ký” → Tại màn hình đăng ký, người dùng nhập các thông tin yêu cầu → Nhập mã OTP được gửi tới điện thoại để xác thực → Thêm thẻ ngân hàng hoặc ví điện tử,

người dùng có thể bỏ qua bước này bằng cách chọn nút “Bỏ qua” → Tới đây là đã đăng ký tài khoản mới thành công



Hình 4-3: Màn hình đăng ký tài khoản

4.1.3. **Đăng nhập**

Người dùng có thể đăng nhập bằng số điện thoại và mật khẩu đã đăng ký tài khoản trước đó

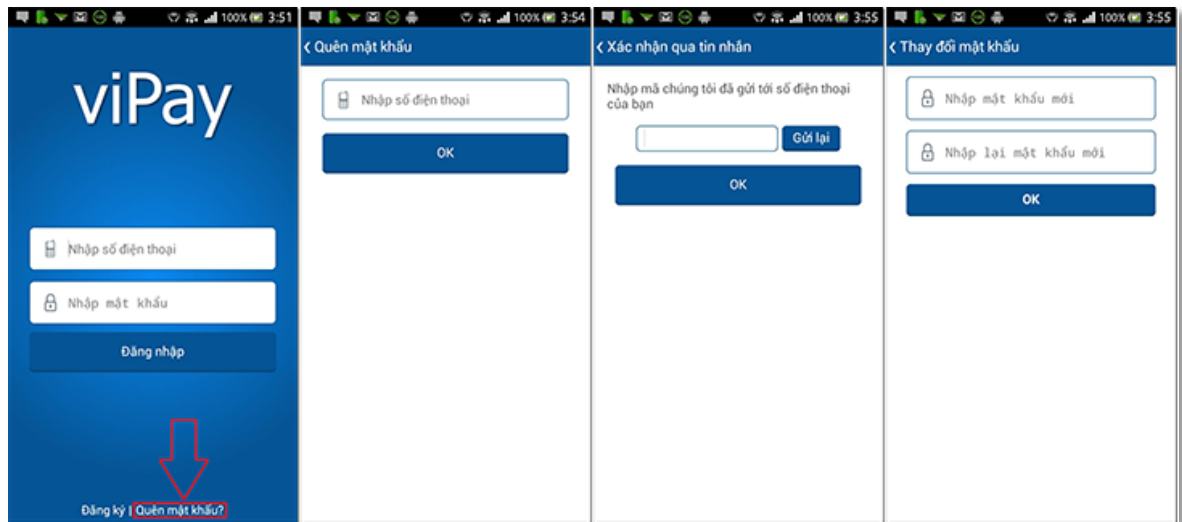


Hình 4-4: Màn hình đăng nhập trên ứng dụng di động.

4.1.4. **Quên mật khẩu**

Khi người dùng quên mật khẩu, để đăng nhập vào tài khoản, người dùng nhấn vào “Quên mật khẩu?” tại màn hình đăng nhập → nhập số điện thoại → Nhập mã

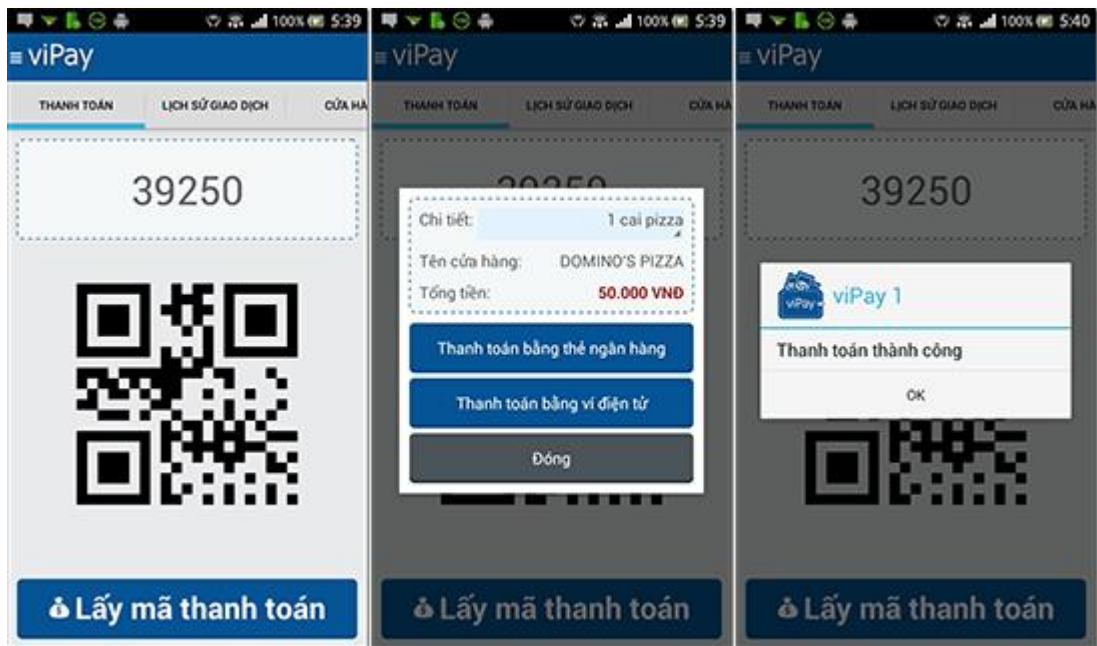
OTP đã được gửi tới số điện thoại → Nhập lại mật khẩu mới → Đăng nhập với mật khẩu mới thành công



Hình 4-5: Màn hình quên mật khẩu

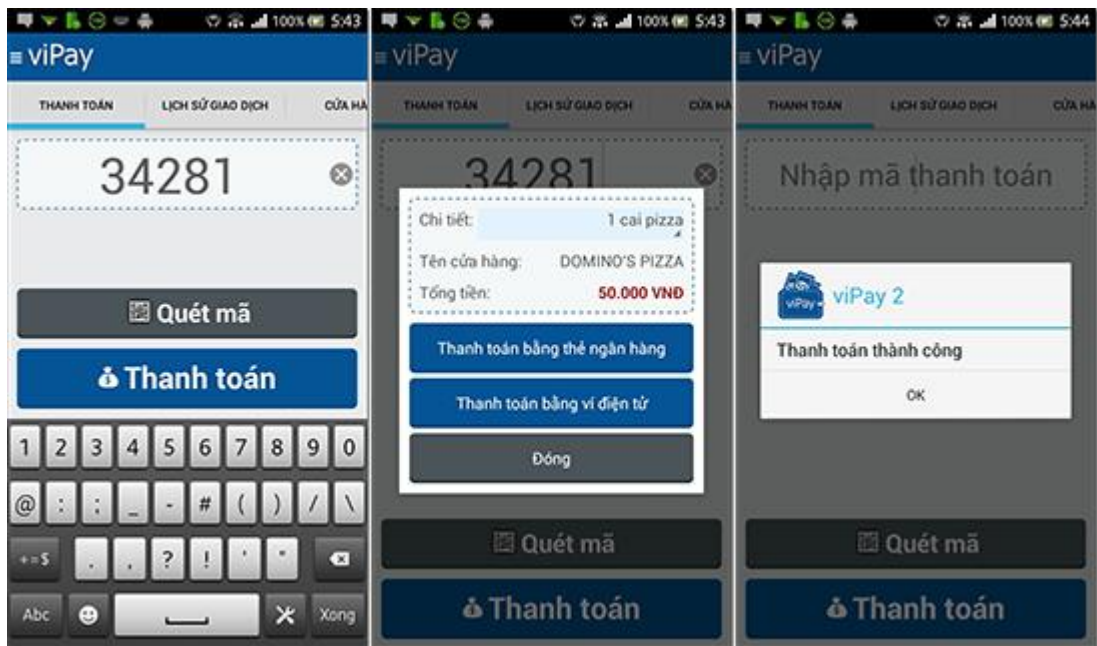
4.1.5. Màn hình chính.

4.1.5.1. Mô hình thanh toán 1



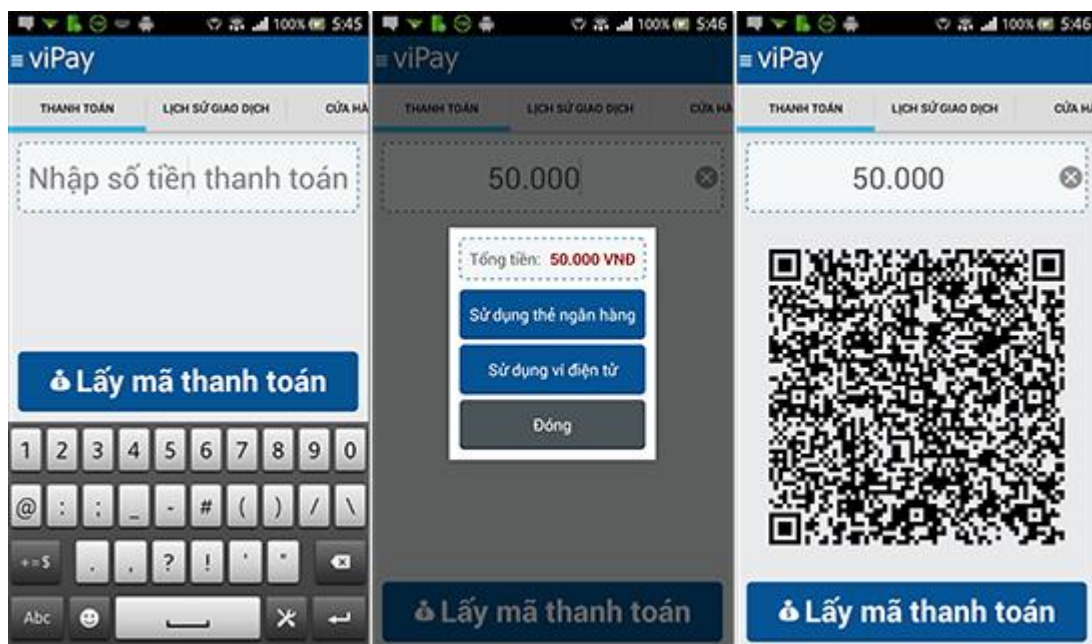
Hình 4-6: Màn hình chính của mô hình thanh toán 1

4.1.5.2. Mô hình thanh toán 2.



Hình 4-7: Màn hình chính của mô hình thanh toán 2

4.1.5.3. Mô hình thanh toán 3

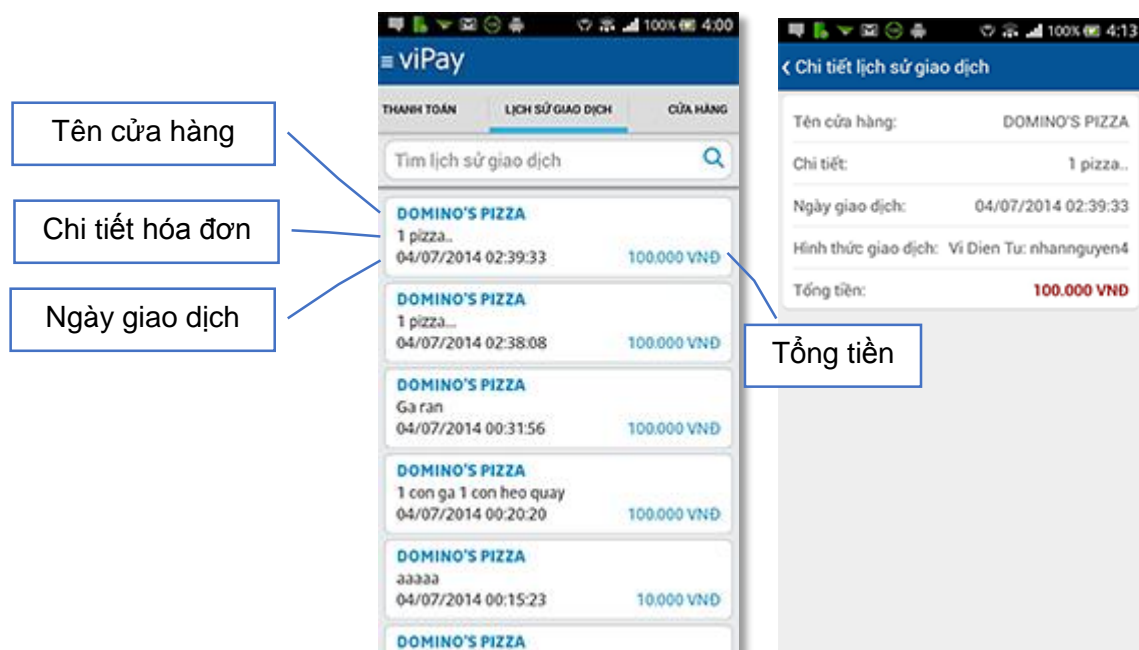


Hình 4-8: Màn hình chính của mô hình thanh toán 2

4.1.6. Xem lịch sử giao dịch

Để xem lịch sử giao dịch, người dùng vào thẻ thứ 2 của ứng dụng tại màn hình chính, nhấn vào từng mục trong danh sách để xem chi tiết

Ngoài ra có thể tìm lịch sử giao dịch theo tên , số tiền, chi tiết và ngày giao dịch

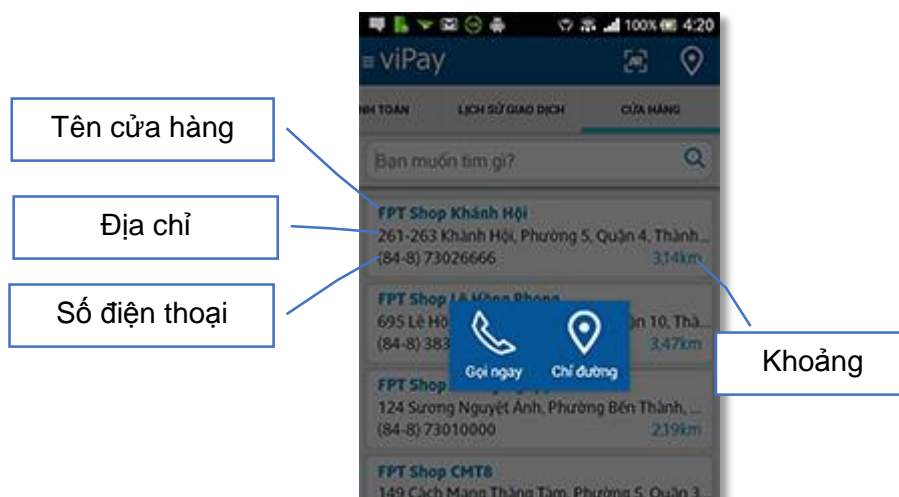


Hình 4-9: Màn hình xem lịch sử giao dịch và chi tiết lịch sử giao dịch

4.1.7. Xem danh sách cửa hàng


Để xem danh sách cửa hàng, người dùng vào thẻ thứ 3 của ứng dụng tại màn hình chính, nhấn vào từng mục trong danh sách để gọi tới cửa hàng đó hoặc chỉ đường tới cửa hàng đó

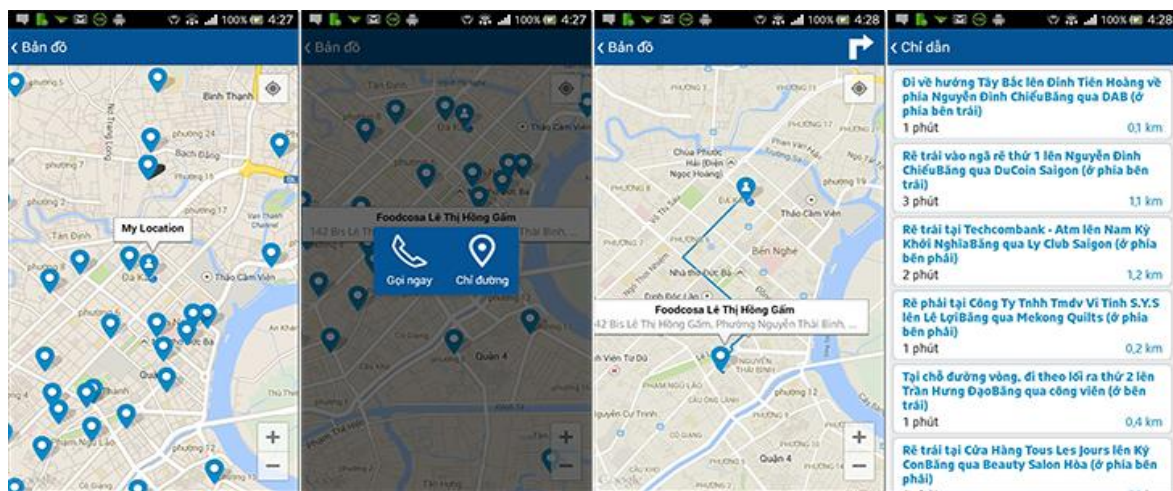
Ngoài ra có thể tìm cửa hàng theo tên, địa chỉ, số điện thoại



Hình 4-10: Màn hình xem danh sách cửa hàng


4.1.8. **Hiển thị bản đồ cửa hàng xung quanh**

Tại màn hình xem danh sách cửa hàng, nhấn vào nút  (Bản đồ) để xem các cửa hàng xung quanh gần nhất. Khi nhấn chọn 1 cửa hàng ứng dụng có thể chỉ dẫn đường tới cửa hàng đó và hiện danh sách hướng dẫn cách đi như hình sau:



Hình 4-11: Màn hình bản đồ cửa hàng xung quanh và hướng dẫn chỉ đường

4.1.9. **AR - Thực tại tăng cường**

Tại màn hình xem danh sách cửa hàng, nhấn vào nút  (AR) để xem danh sách các cửa hàng với thực tại ảo tăng cường



Hình 4-12: Màn hình thực tại tăng cường

4.1.10. *Slider Menu chức năng*

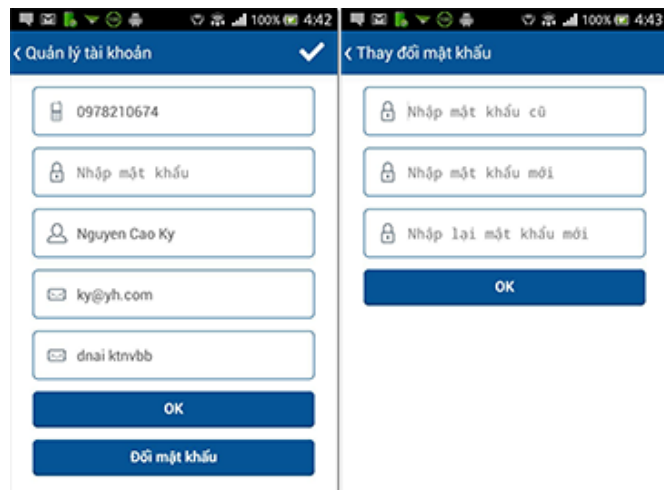
Trượt màn hình từ trái qua phải để hiện Slider Menu chức năng. Màn hình hiển thị thông tin người dùng, các chức năng của ứng dụng và có thể đăng xuất tài khoản



Hình 4-13: Màn hình Slider Menu

4.1.11. *Quản lý tài khoản*

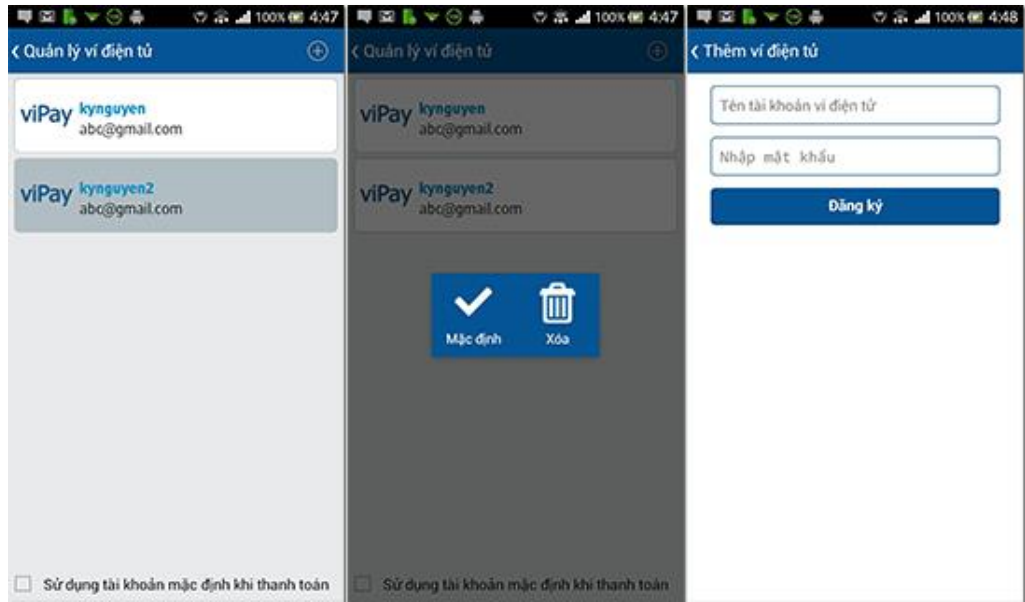
Tại màn hình Slider Menu nhấn chọn chức năng “Quản lý tài khoản”. Tại đây người dùng có thể thay đổi thông tin tài khoản hoặc thay đổi mật khẩu



Hình 4-14: Màn hình quản lý tài khoản và đổi mật khẩu

4.1.12. Quản lý tài khoản ví điện tử

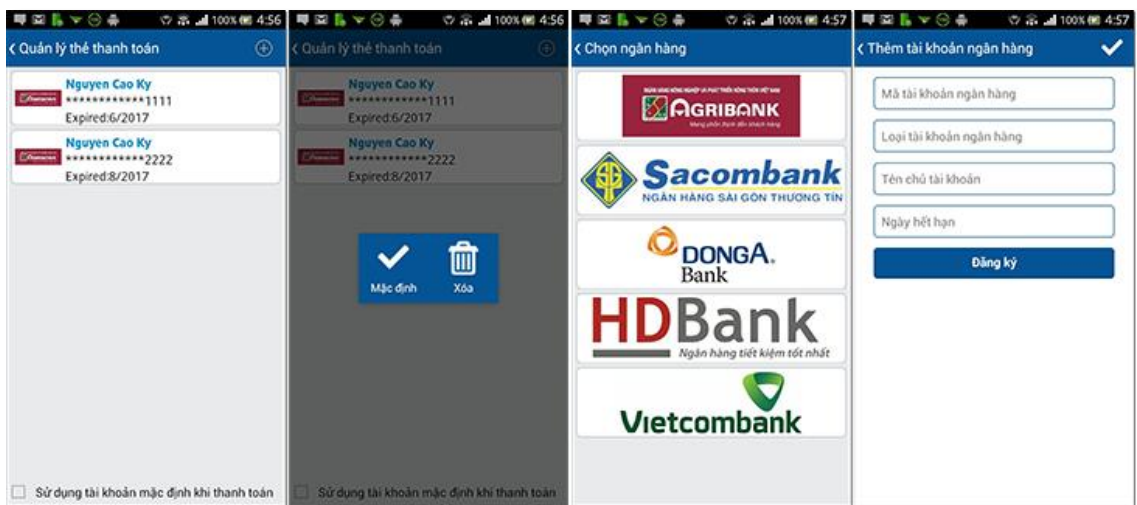
Tại màn hình Slider Menu nhấn chọn chức năng “QL tài khoản ví điện tử”. Tại đây người dùng có thể xem danh sách tài khoản ví điện tử, thêm ví điện tử và chọn tài khoản ví điện tử làm mặc định



Hình 4-15: Màn hình quản lý ví điện tử và thêm ví điện tử

4.1.13. Quản lý thẻ thanh toán

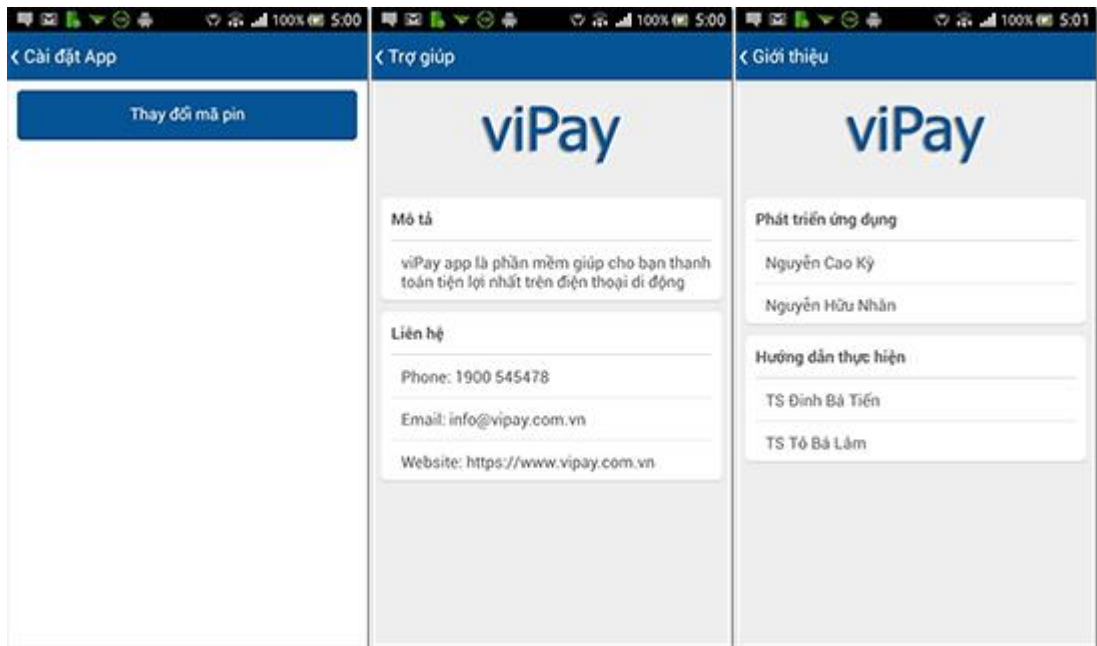
Tại màn hình Slider Menu nhấn chọn chức năng “QL thẻ thanh toán”. Tại đây người dùng có thể xem danh sách thẻ thanh toán, thêm thẻ thanh toán và chọn thẻ thanh toán làm mặc định



Hình 4-16: Màn hình quản lý thẻ thanh toán và thêm thẻ thanh toán

4.1.14. Chức năng khác

Các chức năng khác: cài đặt, trợ giúp, giới thiệu



Hình 4-17: Màn hình cài đặt, trợ giúp, giới thiệu

4.2. Giới thiệu ứng dụng web dành cho Nhân viên thanh toán

Tương tự như ứng dụng trên SmartPhone thì 3 mô hình thanh toán nên có 3 Web Shop khác nhau nhưng nhìn chung các chức năng cơ bản giống nhau, vì vậy mỗi chức năng nhóm chỉ giới thiệu một lần và sẽ được trích dẫn ở các phần sau.

Ứng dụng web dành cho nhân viên thanh toán truy cập theo địa chỉ sau:

- Mô hình thanh toán 1: <http://nguyencaokyvn.com/ShopWeb1>
- Mô hình thanh toán 2: <http://nguyencaokyvn.com/ShopWeb2>
- Mô hình thanh toán 3: <http://nguyencaokyvn.com/ShopWeb3>

4.2.1. Đăng nhập

Nhân viên thanh toán có thể đăng nhập với User name và Password mà hệ thống cấp.

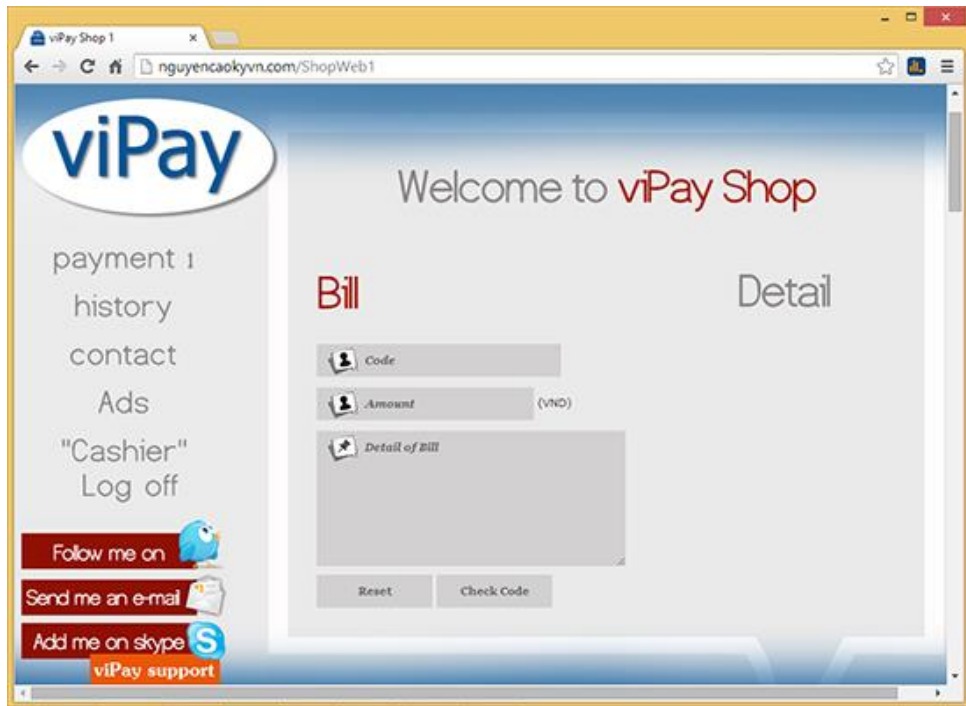


Hình 4-18: Màn hình đăng nhập Web Shop

4.2.2. Thanh toán

4.2.2.1. Mô hình thanh toán 1

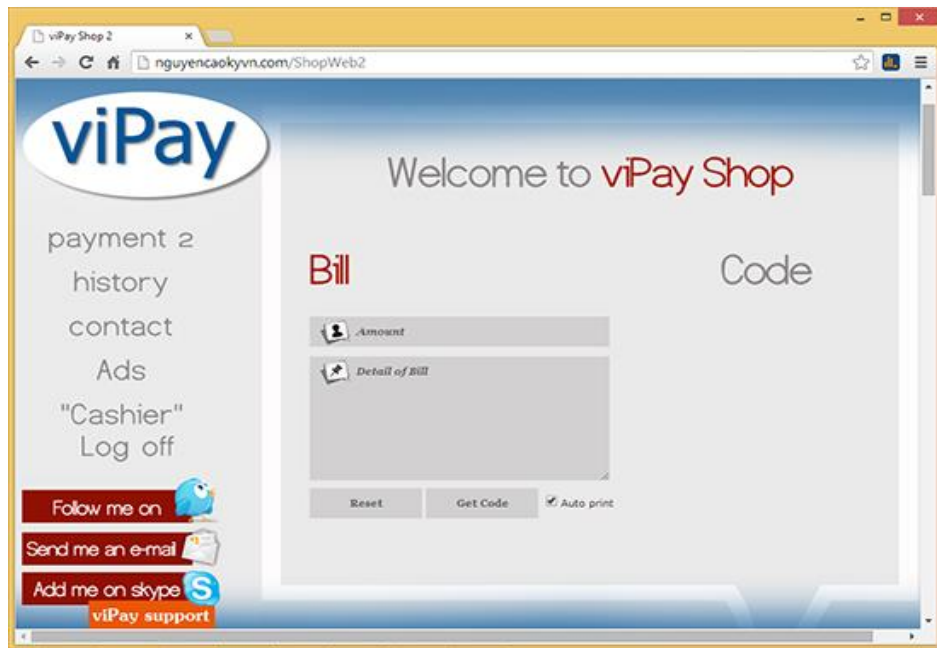
Nhân viên thanh toán nhập tổng tiền vào ô “Amount”, chi tiết hóa đơn vào ô “Detail of Bill” và hỏi khách mã thanh toán rồi nhập vào ô “Code”, sau đó nhấn vào nút “Check Code” để thực hiện đưa hóa đơn và mã thanh toán lên Server rồi đợi người dùng xác nhận hóa đơn thanh toán



Hình 4-19: Màn hình chính của Web Shop trong mô hình thanh toán 1

4.2.2.2. Mô hình thanh toán 2

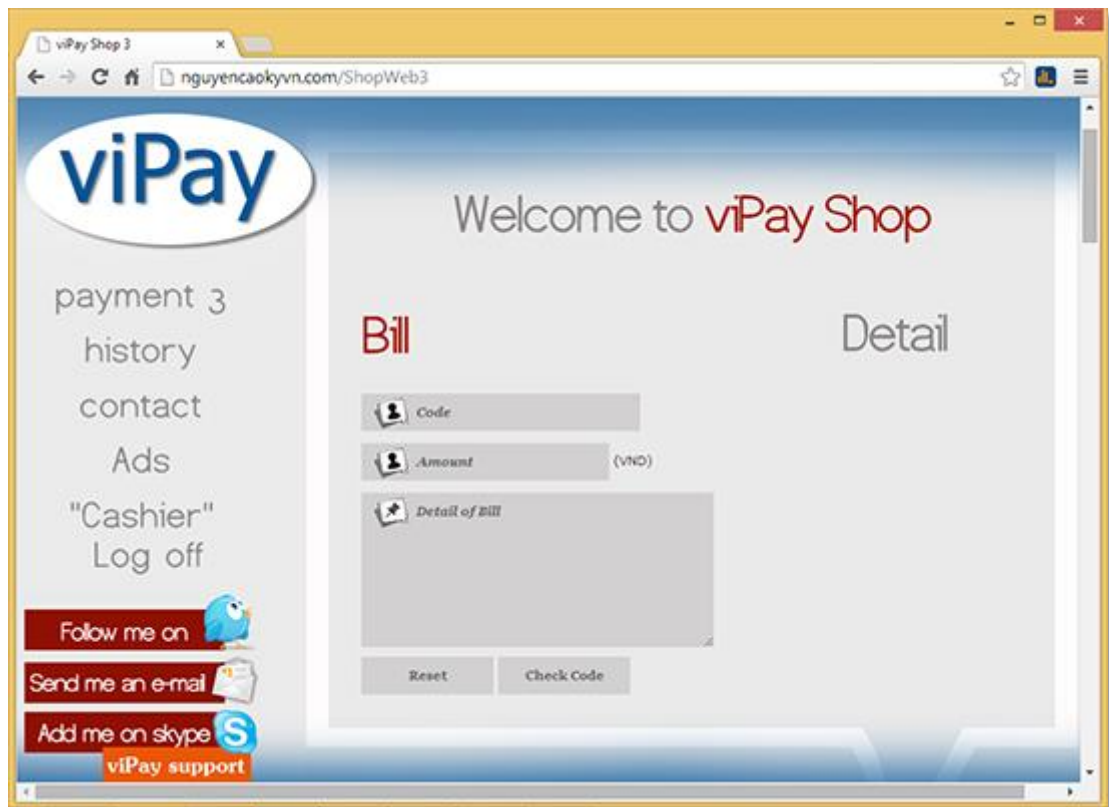
Nhân viên thanh toán nhập tổng tiền vào ô “Amount”, chi tiết hóa đơn vào ô “Detail of Bill” rồi nhấn vào nút “Get Code” để thực hiện đưa hóa đơn lên Server và nhận về mã thanh toán



Hình 4-20: Màn hình chính của Web Shop trong mô hình thanh toán 2

4.2.2.3. Mô hình thanh toán 3

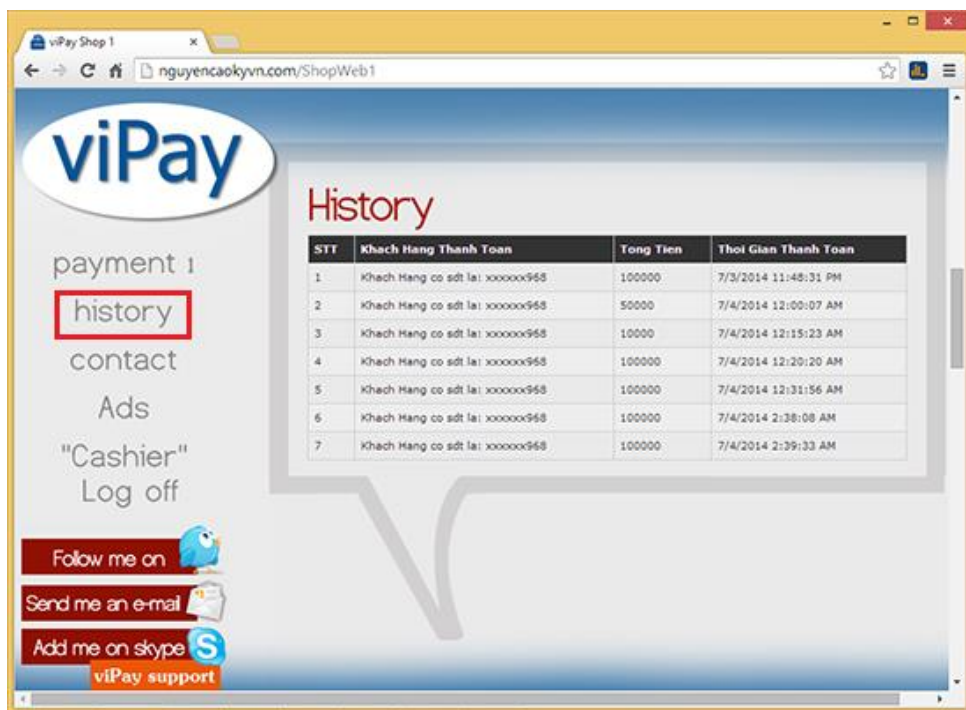
Nhân viên thanh toán nhập tổng tiền vào ô “Amount”, chi tiết hóa đơn vào ô “Detail of Bill” và hỏi khách mã thanh toán rồi sử dụng đầu đọc QRCode hoặc NFC để nhập Code, sau đó nhấn vào nút “Check Code” để thực hiện đưa hóa đơn và mã thanh toán lên Server rồi đợi thông báo kết quả thực hiện giao dịch từ Server



Hình 4-21: Màn hình chính của Web Shop trong mô hình thanh toán 3

4.2.3. Xem lịch sử giao dịch

Nhân viên thanh toán có thể dễ dàng xem lại lịch sử giao dịch bằng cách nhấn vào mục “history” ở phần Menu bên trái



Hình 4-22: Màn hình xem lịch sử giao dịch Web Shop

4.2.4. Đăng tải thông tin quảng cáo

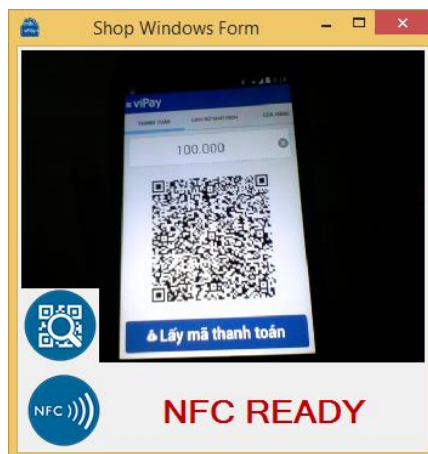
Sau khi đăng nhập, nhân viên thanh toán có thể dùng chức năng “Ads” để đăng quảng cáo trực tiếp tới khách hàng như hình sau:



Hình 4-23: Màn hình đăng quảng cáo tới smartphone khách hàng - Web Shop

4.2.5. Phần mềm trên Windows Form hỗ trợ cho Web – WinForm Shop

Phần mềm WinForm Shop có thể đọc QRCode từ webcam, kết nối với thiết bị NFC và hỗ trợ trình duyệt web việc trao đổi thông điệp với viPay App



Hình 4-24: Màn hình chính WinForm Shop

4.3. Quy trình thực hiện thanh toán trong các mô hình

Để thuận tiện giải thích quy trình, nhóm đưa ra bối cảnh như sau:

- Địa điểm: tại cửa hàng bán quần áo ABC
- Số lượng người: 1 khách hàng, 1 nhân viên thanh toán
- Vật mua bán: 1 cái áo sơ mi
- Giá: 100.000 đ
- Nhân viên thanh toán tính tiền thông qua ứng dụng web của viPay
- Khách hàng sử dụng smartphone Android chạy ứng dụng viPay

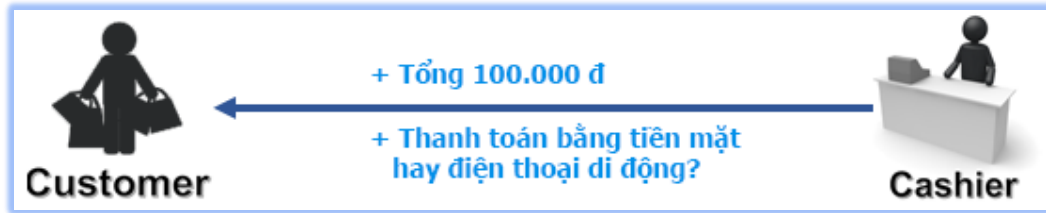
4.3.1. Mô hình thanh toán 1

- Bước 1: Khách hàng đi tới quầy thanh toán để tính tiền 1 cái áo sơ mi. Nhân viên thanh toán cầm cái áo



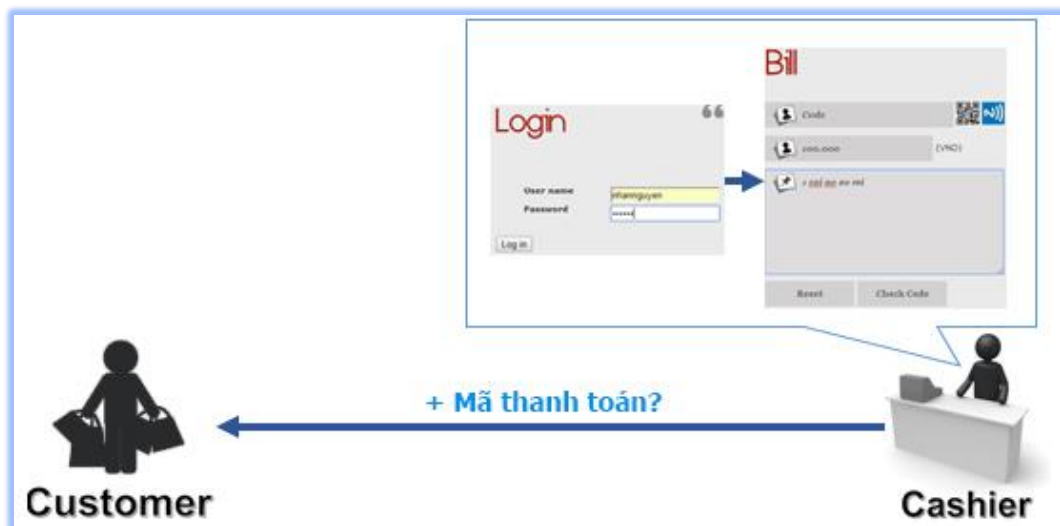
Hình 4-25: Hình mô tả bước khách hàng đem áo sơ mi tới quầy tính tiền

- Bước 2:
 - Nhân viên thanh toán xem giá thành của cái áo và nói với khách hàng là 100.000 đ
 - Nhân viên thanh toán hỏi khách hàng muốn trả bằng tiền mặt hay thanh toán qua điện thoại di động
 - Khách hàng trả lời là thanh toán qua điện thoại di động



Hình 4-26: Hình mô tả bước nhân viên thanh toán hỏi khách hàng loại hình thanh toán

- Bước 3:
 - Nhân viên thanh toán mở trình duyệt vào web của viPay và đăng nhập thành công
 - Tại màn hình chính, nhập vào Amount là “100.000” đ và Detail of Bill là “1 cái áo sơ mi” rồi nhấn vào nút NFC hoặc QRCode khi đó sẽ hiện lên chương trình WinForm Shop
 - Nhân viên thanh toán nói với khách hàng là đọc mã thanh toán hoặc đưa mã QRCode hoặc trạm vào thiết bị NFC



Hình 4-27: Hình mô tả mô hình thanh toán 1, bước nhân viên thanh toán hỏi khách hàng mã thanh toán

- Bước 4:
 - Khách hàng mở app → Nhập mã pin → Nhấn vào nút “Lấy mã thanh toán” → Màn hình điện thoại sẽ hiện lên mã thanh toán và hình mã QRCode
 - Khách hàng đọc mã thanh toán cho nhân viên thanh toán hoặc đưa mã QRCode vào Webcam hoặc chạm vào thiết bị NFC



Hình 4-28: Hình mô tả mô hình thanh toán 1, khách hàng lấy mã thanh toán và đọc cho nhân viên thanh toán

- Bước 5:
 - Nhân viên thanh toán nhập mã code rồi nhấn nút “Check Code” hoặc chương trình WinForm Shop sẽ tự động đọc mã QRCode hoặc thông qua NFC và ghi vào trong trình duyệt web, tự động nhấn nút enter để thực hiện thanh toán



Hình 4-29: Hình mô tả mô hình thanh toán 1, nhân viên thanh toán nhập mã thanh toán và gửi lên Server

- Bước 6:
 - Hóa đơn hiện lên trên màn hình điện thoại khách hàng → Khách hàng xác nhận hóa đơn bằng cách thanh toán bằng thẻ ngân hàng hoặc ví điện tử (nhấn vào nút tương ứng)



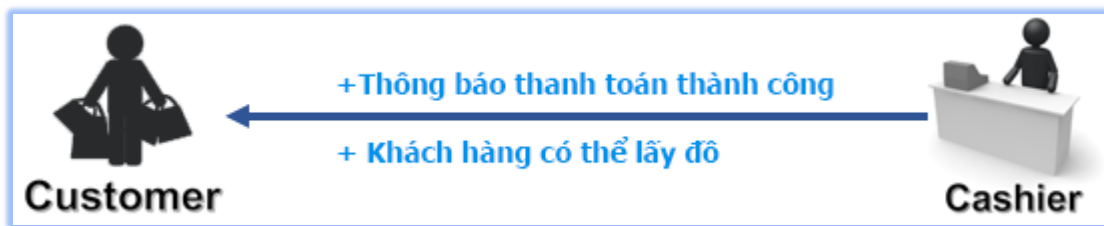
Hình 4-30: Hình mô tả mô hình thanh toán 1, khách hàng xác nhận hóa đơn thanh toán

- Bước 7:
 - Trên màn hình web của nhân viên thanh toán hiện thông báo thanh toán thành công
 - Điện thoại khách hàng cũng nhận được thông báo thành công, ngoài ra còn nhận được thông báo bằng tin nhắn SMS



Hình 4-31: Hình mô tả bước Server thông báo thanh toán thành công về nhân viên thanh toán và khách hàng

- Bước 8:
 - Nhân viên thanh toán thông báo với khách hàng là đã thanh toán thành công và tiến hành in hóa đơn giấy đưa cho khách hàng
 - Khách hàng nhận hóa đơn giấy và rời khỏi quầy thanh toán

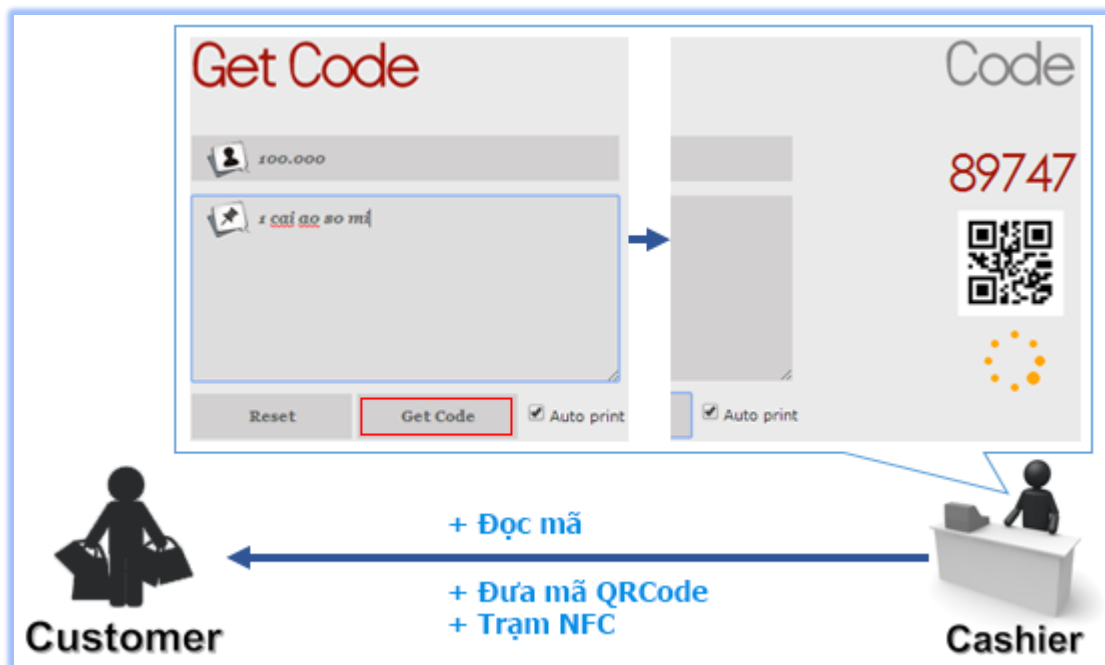


Hình 4-32: Hình mô tả bước nhân viên thanh toán thông báo cho khách hàng giao dịch đã hoàn tất

4.3.2. Mô hình thanh toán 2

- Bước 1 và bước 2 tương tự như Mô hình thanh toán 1

- Bước 3:
 - Nhân viên thanh toán mở trình duyệt vào web của viPay và đăng nhập thành công
 - Tại màn hình chính, nhập vào Amount là “100.000” đ và Detail of Bill là “1 cái ao so mi” rồi nhấn vào nút “Get code” khi đó mã thanh toán và QRCode sẽ hiện ra đồng thời chương trình WinForm Shop cũng được chạy
 - Nhân viên thanh toán đọc mã thanh toán cho khách hàng hoặc đưa mã QRCode cho khách hàng quét hoặc trạm vào thiết bị NFC



Hình 4-33: Hình mô tả mô hình thanh toán 2, nhân viên thanh toán đọc mã thanh toán cho khách hàng

- Bước 4:
 - Khách hàng mở app → Nhập mã pin → Tại màn hình chính nhập mã thanh toán và nhấn vào nút “Thanh toán” hoặc nhấn vào nút “Quét mã” để quét mã QRCode hoặc trạm điện thoại vào thiết bị NFC của nhân viên thanh toán

- Hóa đơn hiện lên trên màn hình điện thoại khách hàng → Khách hàng xác nhận hóa đơn bằng cách thanh toán bằng thẻ ngân hàng hoặc ví điện tử (nhấn vào nút tương ứng)

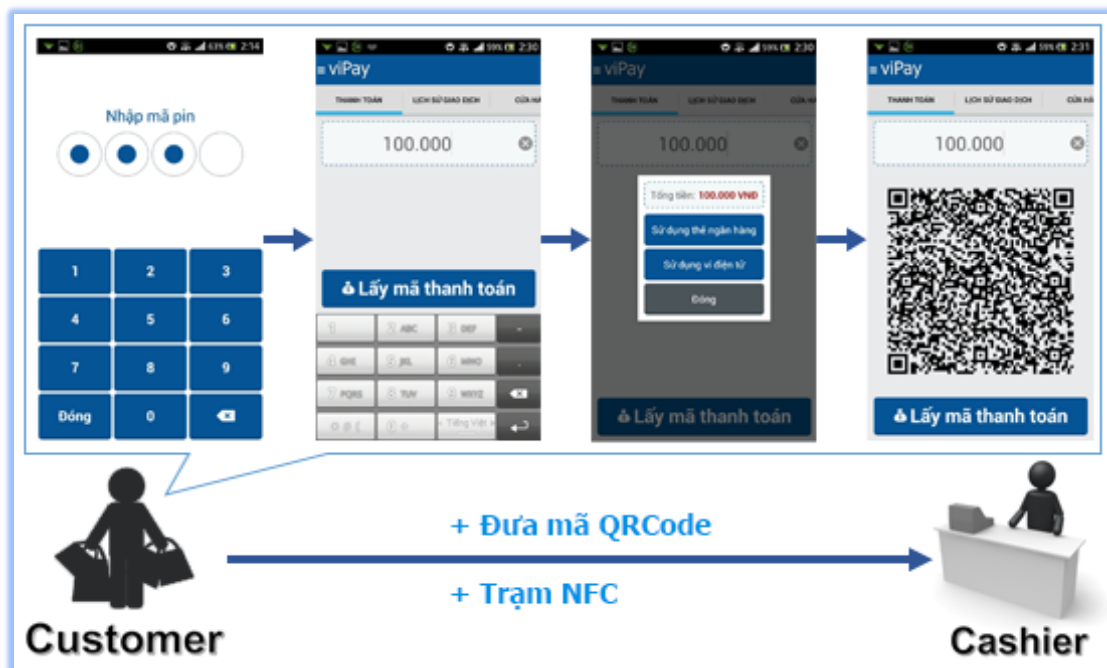


Hình 4-34: Hình mô tả mô hình thanh toán 2, khách hàng nhập mã thanh toán và xác nhận hóa đơn thanh toán

- Bước 5 và bước 6 tương tự bước 7 và bước 8 của Mô hình thanh toán 1

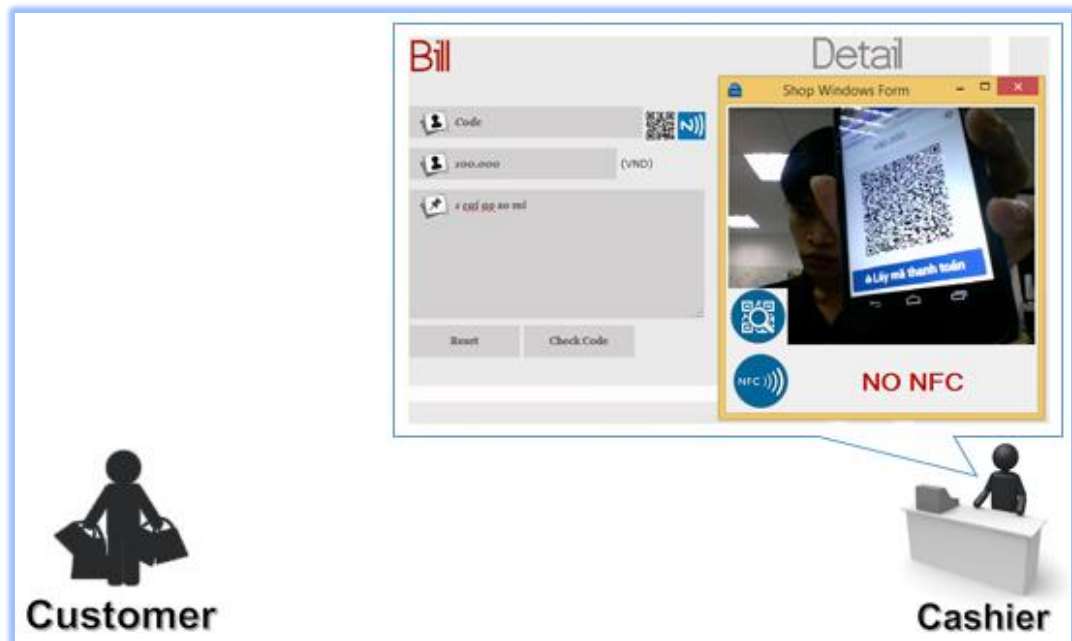
4.3.3. **Mô hình thanh toán 3**

- Bước 1, bước 2 và bước 3 tương tự như Mô hình thanh toán 1
- Bước 4:
 - Khách hàng mở app → Nhập mã pin → Nhập số tiền cần thanh toán → Nhấn vào nút “Lấy mã thanh toán” → Hóa đơn hiện lên trên màn hình điện thoại khách hàng → Khách hàng xác nhận hóa đơn bằng cách thanh toán bằng thẻ ngân hàng hoặc ví điện tử (nhấn vào nút tương ứng) → Màn hình điện thoại sẽ hiện lên mã thanh toán với định dạng là mã QRCode
 - Khách hàng đưa mã QRCode vào Webcam hoặc trạm vào thiết bị NFC của nhân viên thanh toán



Hình 4-35: Hình mô tả mô hình thanh toán 3, khách hàng nhập số tiền, xác nhận hóa đơn thanh toán và lấy mã thanh toán

- Bước 5:
 - Chương trình WinForm Shop sẽ tự động đọc mã QRCode hoặc thông qua NFC và ghi vào trong trình duyệt web, tự động nhấn nút enter để thực hiện thanh toán



Hình 4-36: Hình mô tả mô hình thanh toán 3, chương trình WinForm Shop tự động đọc mã QRCode hoặc thông qua NFC và ghi vào trong trình duyệt web

- Bước 6 và bước 7 tương tự bước 7 và bước 8 của Mô hình thanh toán 1

4.4. Khảo sát khách hàng sử dụng phần mềm

- Địa điểm: Công ty Cổ Phần Dịch Vụ Trực Tuyến Cộng Đồng Việt – VIETUNION.
- Thành phần tham gia: nhân viên phòng Dự Án, nhân viên phòng Kinh Doanh, nhân viên phòng Kỹ Thuật, nhân viên phòng Marketing.
- Số lượng: 31 người.
- Lý do:
 - Khảo sát nhu cầu của người dùng về một mô hình thanh toán mới.
 - Khảo sát tính khả dụng của từng mô hình thanh toán (3 mô hình).
 - Thu thập ý kiến người dùng, từ đó cải thiện các mô hình.
 - Đề xuất mô hình thanh toán khả thi nhất.
- Nội dung:
 - Khảo sát tính hoàn thiện của mô hình.
 - Khảo sát tính tiện dụng quy trình thanh toán.
 - Khảo sát tính bảo mật.
 - Khảo sát tính tiện dụng của giao diện màn hình.
 - Khảo sát độ thân thiện của giao diện.

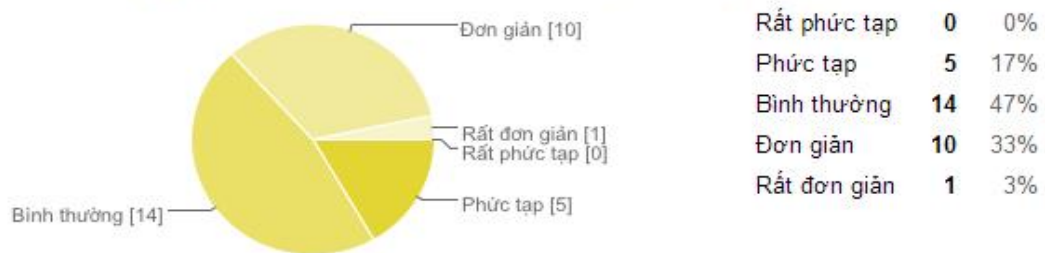
(Xem thêm ở phần phụ lục A)

4.4.1. Khảo sát Mô hình thanh toán 1.

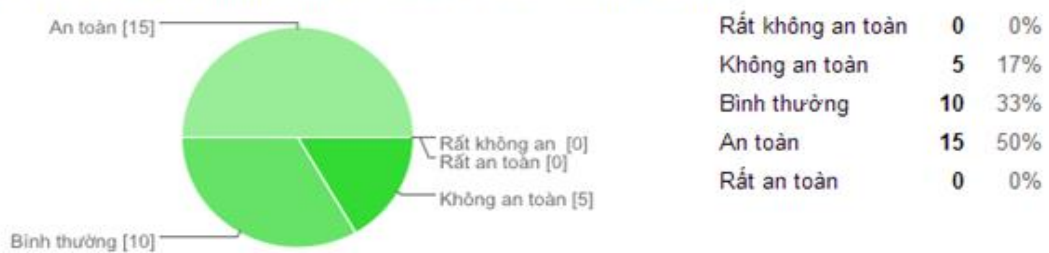
Ứng dụng đã cung cấp đủ chức năng cho quá trình giao dịch hay chưa?



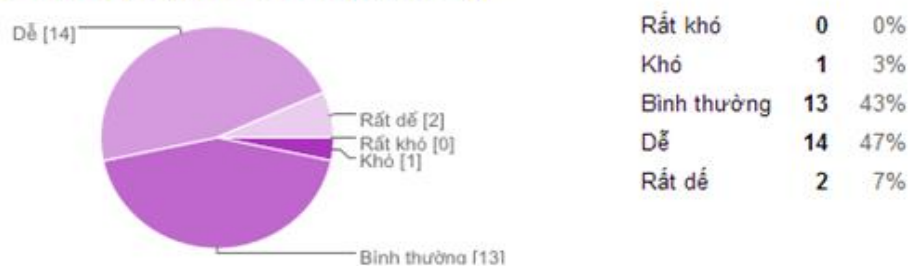
Bạn cảm thấy quy trình thanh toán này có đơn giản và tiện dụng hay không?



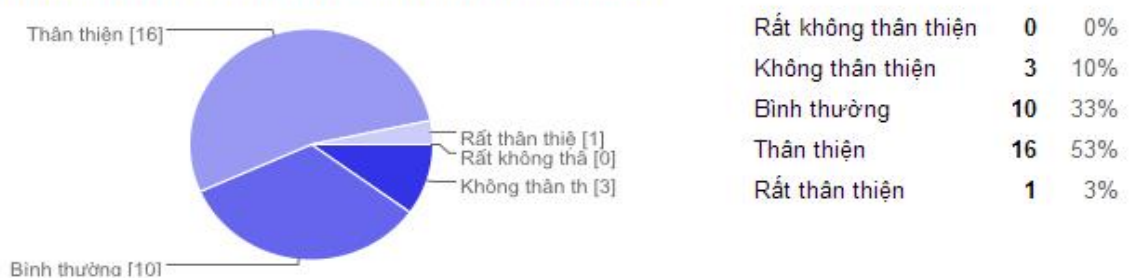
Bạn cảm thấy mô hình thanh toán này có an toàn hay không?



Bạn cảm thấy ứng dụng có dễ sử dụng không?



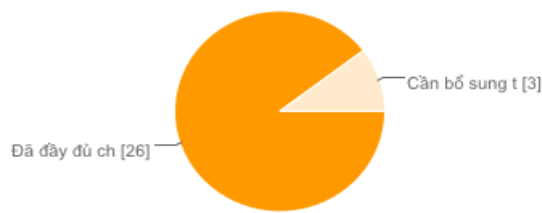
Đánh giá tính thân thiện của giao diện người dùng?



Hình 4-37: Kết quả khảo sát ý kiến người dùng mô hình thanh toán 1.

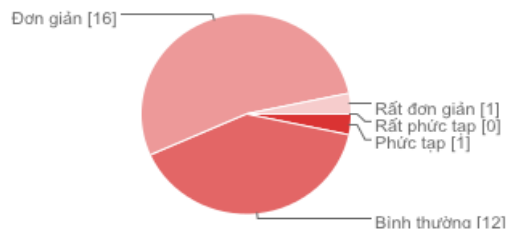
4.4.2. Khảo sát Mô hình thanh toán 2

Ứng dụng đã cung cấp đủ chức năng cho quá trình giao dịch hay chưa?



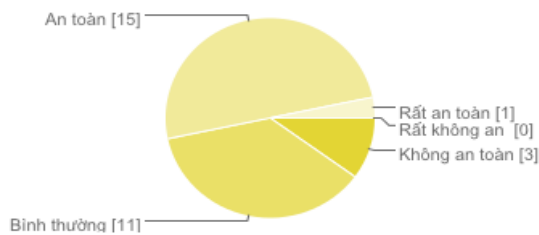
Đã đầy đủ chức năng	26	90%
Cần bổ sung thêm chức năng	3	10%

Bạn cảm thấy quy trình thanh toán này có đơn giản và tiện dụng hay không?



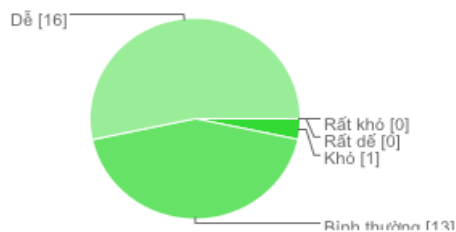
Rất phức tạp	0	0%
Phức tạp	1	3%
Bình thường	12	40%
Đơn giản	16	53%
Rất đơn giản	1	3%

Bạn cảm thấy mô hình thanh toán này có an toàn hay không?



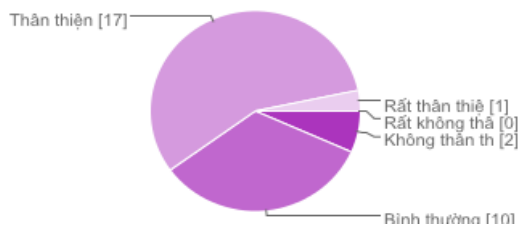
Rất không an toàn	0	0%
Không an toàn	3	10%
Bình thường	11	37%
An toàn	15	50%
Rất an toàn	1	3%

Bạn cảm thấy ứng dụng có dễ sử dụng không?



Rất khó	0	0%
Khó	1	3%
Bình thường	13	43%
Dễ	16	53%
Rất dễ	0	0%

Đánh giá tính thân thiện của giao diện người dùng?

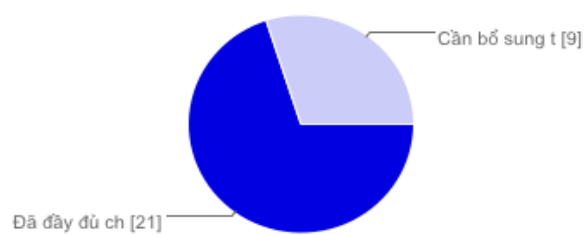


Rất không thân thiện	0	0%
Không thân thiện	2	7%
Bình thường	10	33%
Thân thiện	17	57%
Rất thân thiện	1	3%

Hình 4-38: Kết quả khảo sát ý kiến người dùng mô hình thanh toán 2.

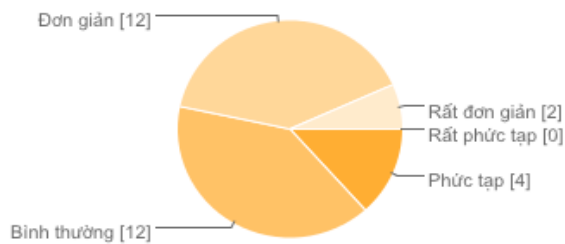
4.4.3. Khảo sát Mô hình thanh toán 3

Ứng dụng đã cung cấp đủ chức năng cho quá trình giao dịch hay chưa?



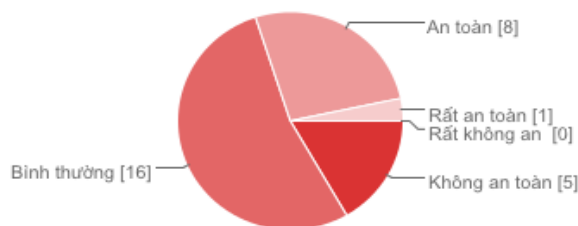
Đã đầy đủ chức năng	21	70%
Cần bổ sung thêm chức năng	9	30%

Bạn cảm thấy quy trình thanh toán này có đơn giản và tiện dụng hay không?



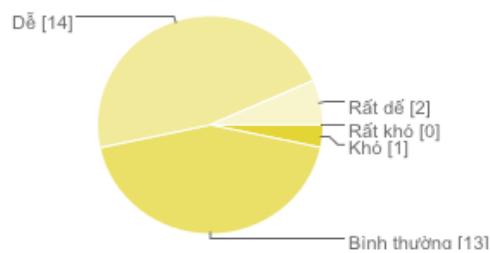
Rất phức tạp	0	0%
Phức tạp	4	13%
Bình thường	12	40%
Đơn giản	12	40%
Rất đơn giản	2	7%

Bạn cảm thấy mô hình thanh toán này có an toàn hay không?



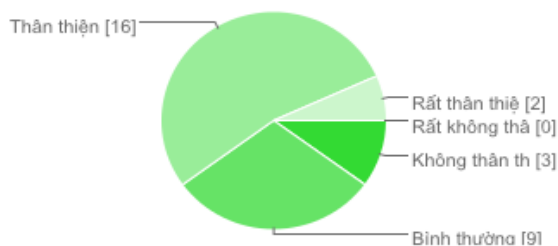
Rất không an toàn	0	0%
Không an toàn	5	17%
Bình thường	16	53%
An toàn	8	27%
Rất an toàn	1	3%

Bạn cảm thấy ứng dụng có dễ sử dụng không?



Rất khó	0	0%
Khó	1	3%
Bình thường	13	43%
Dễ	14	47%
Rất dễ	2	7%

Đánh giá tính thân thiện của giao diện người dùng?



Rất không thân thiện	0	0%
Không thân thiện	3	10%
Bình thường	9	30%
Thân thiện	16	53%
Rất thân thiện	2	7%

Hình 4-39: Kết quả khảo sát ý kiến người dùng mô hình thanh toán 3.

4.4.4. So sánh giữa 3 mô hình thanh toán.

Thông tin dữ liệu từ Hình 4-37, Hình 4-38 và Hình 4-39 được tổng quát ở bảng sau:

(Thang đo mức độ hoàn thiện: 0%-100%)

	Mô hình thanh toán 1	Mô hình thanh toán 2	Mô hình thanh toán 3
Mức độ hoàn thiện của ứng dụng	77%	90%	70%
Mức độ tiện dụng của qui trình thanh toán	83%	97%	87%
Mức độ bảo mật của mô hình	83%	90%	83%
Mức độ tiện dụng của ứng dụng thanh toán	97%	97%	97%
Mức độ thân thiện giao diện ứng dụng	90%	93%	90%

Bảng 4-1: Tổng kết số liệu khảo sát 3 mô hình thanh toán.

Với số liệu khảo sát riêng lẻ từng mô hình như bảng 4-1, có thể nhận thấy mô hình thanh toán 2 vượt trội hơn mô hình thanh toán 1 và mô hình thanh toán 3.

Kết quả khảo sát tính khả thi của các mô hình được thể hiện ở hình sau:



Hình 4-40: Kết quả khảo sát mức độ khả thi giữa các mô hình.

Như vậy, theo nhận định từ số liệu ở bảng thì mô hình thanh toán 2 mang các đặc tính tích cực trội hơn 2 mô hình thanh toán còn lại. Bên cạnh đó, kết quả khảo sát người dùng từ hình 4-40, một lần nữa khẳng định độ khả thi của mô hình thanh toán 2 ở mức cao.

Chương 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Từ những thông tin đã phân tích cùng với kết quả thu được sau khi đưa từng mô hình thanh toán vào thực nghiệm và tiến hành khảo sát (mục 4.4), một vài nhận định về lý do dẫn đến mức độ khả thi khác nhau của các mô hình thanh toán được đúc kết như sau:

Mô hình 1, khách hàng sau khi cùng cấp mã thanh toán cho nhân viên thanh toán phải thụ động chờ thông tin hóa đơn cần thanh toán, việc tạo ra tâm lý phải chờ đợi cho khách hàng làm ảnh hưởng không nhỏ đến tính khả thi của mô hình, vì tâm lý người dùng là một trong những nhân tố quyết định kết quả thành công hay thất bại của mô hình khi mô hình được triển khai.

Mô hình 2, bên cạnh người dùng có thể hoàn thành giao dịch mà không phải thao tác quá nhiều, người dùng còn là người chủ động yêu cầu thông tin hóa đơn và tiến hành thanh toán mà không cần phải cung cấp thêm thông tin. Tạo tâm lý thoải mái và yên tâm cho người dùng.

Mô hình 3, tuy khách hàng có thể thực hiện giao dịch một cách dễ dàng và nhanh chóng, nhưng cách thức thanh toán này còn khá mới mẻ với phần đông người tiêu dùng ở nước ta, mang lại tâm lý lo ngại, e dè cho khách hàng.

Nhìn chung cả 3 mô hình đề xuất, mỗi mô hình đều có ưu và nhược điểm riêng, nhưng vấn đề đặt ra là phải chú trọng vào tâm lý của người dùng, phải lựa chọn mô hình có khả năng đáp ứng nhu cầu người dùng cao nhất, mô hình hạn chế được sự e dè của người dùng đến mức thấp nhất.

Mặc dù có cố gắng khắc phục các khuyết điểm trong việc thanh toán trên thiết bị di động thì các mô hình cũng không thể tránh khỏi các khuyết điểm đặc trưng của hình thức này là việc tâm lý dè dặt của người dùng. Nguyên nhân là do:

- Trình độ bảo mật ở nước ta còn hạn chế.
- Việc tìm các đối tác cung cấp dịch vụ ngân hàng cũng như Ví điện tử là rất khó khăn nếu không có tổ chức đứng ra bảo đảm, cùng với thói

quen dùng tiền mặt của người dân Việt Nam ta từ xưa đến nay khó có thể thay đổi được.

Tuy nhiên, bằng cách mang công nghệ hiện đại tiếp cận với người dùng với các thao tác đơn giản, giúp người dùng thực hiện thanh toán dễ dàng và thuận tiện hơn, tác giả hy vọng rằng mô hình thanh toán đề xuất có thể thay đổi cách nhìn của người dùng về hình thức thanh toán này và đưa mô hình áp dụng vào thực tế.

5.2. Hướng phát triển

Ngoài việc đáp ứng nhu cầu thanh toán ở thị trường Việt Nam, hệ thống thanh toán “viPay” sẽ mở rộng liên kết với một số nhà cung cấp dịch vụ trên phạm vi toàn cầu như: Visa, MasterCard, PayPal...

Để mở rộng số lượng người dùng, ứng dụng trên điện thoại di động của mô hình cần sử dụng được trên nhiều thiết bị khác nhau, nhiều nền tảng khác nhau. Do đó ứng dụng “viPay” sẽ được thử nghiệm trên nhiều thiết bị và triển khai trên nền tảng iOS và Windows Phone.

Với tìm năng phát triển của máy POS, mô hình thanh toán “viPay” sẽ có thể phát triển mạnh mẽ hơn nếu có thể tích hợp máy POS vào hệ thống, cụ thể máy POS sẽ thay thế vai trò của Web-Client. Khi máy POS thay thế Web-Client, tính bảo mật và tiện dụng của hệ thống sẽ được cải thiện đáng kể.

Xây dựng framework hỗ trợ phát triển ứng dụng di động có nhu cầu sử dụng dịch vụ thanh toán, bán hàng. Trên nhiều nền tảng như: Android, iOS, Windows Phone.

Xây dựng framework hỗ trợ xây dựng hệ thống website cho nhu cầu thực hiện giao dịch trực tuyến như các website chợ điện tử hay website bán hàng trực tuyến.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Azuma. A survey of augmented reality. ACM SIGGRAPH, 1-38, 1997
- [2] Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, pp. 360-365, 1997.
- [3] Google Cloud Messaging for Android, Available:
<http://developer.android.com/google/gcm/index.html>
- [4] Google Wallet. Available:
<https://www.google.com/wallet/>
- [5] Gson, Available:
<https://code.google.com/p/google-gson/>
- [6] Kumar, D., Mobile payment architectures for India,
- [7] MasterCard PayPass. Available:
<http://www.mastercard.com/us/paypass/phonetrial/whatispaypass.html>
- [8] Mopay. Available:
<http://www.mopay.com>
- [9] NFC-Windows 8, Available:
<https://software.intel.com/en-us/articles/using-nfc-from-windows-8-applications>
- [10] Nguyễn Văn Lê Tây, Nguyễn Xuân Thành, “Xây dựng hệ thống thực tại ảo tăng cường trong tìm kiếm dịch vụ trên điện thoại Android”, Khoa Công Nghệ Thông Tin, Khoa học Tự nhiên thành phố Hồ Chí Minh, 2013.
- [11] Pay by phone, Available:
<https://paybyphone.com/>
- [12] R. Rivest, A. Shamir, L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, Vol. 21 (2), pp. 120–126. 1978. Previously released as an MIT "Technical Memo" in April 1977. Initial publication of the RSA scheme.
- [13] Signalr. Available:
<http://www.signalr.net>

- [14] Tình hình kinh tế - xã hội. Available:
<http://gso.gov.vn/default.aspx?tabid=621&ItemID=13907>
- [15] Website Ngân Hàng Nhà Nước Việt Name, Available:
<http://www.sbv.gov.vn/>
- [16] Zxing, Available:
<https://github.com/zxing/zxing>

PHỤ LỤC A

Ý KIẾN ĐÁNH GIÁ CÁC MÔ HÌNH THANH TOÁN

Với mong muốn tìm hiểu nhu cầu người dùng về một mô hình thanh toán mới và thu thập ý kiến người dùng về tính hoàn thiện, khả thi của từng mô hình thanh toán. Để có thể tiếp thu ý kiến người dùng một cách khách quan nhất, bài khảo sát được tiến hành với 30 người, với nhiều độ tuổi khác nhau, nhiều chuyên môn khác nhau.

A.1 Nội dung bảng khảo sát

Bảng khảo sát ý kiến người dùng

Mô hình "Khách hàng chủ động lấy mã thanh toán"

1. Ứng dụng đã cung cấp đủ chức năng cho quá trình giao dịch hay chưa?
☐ Đã đầy đủ chức năng
☐ Cần bổ sung thêm chức năng
2. Bạn cảm thấy quy trình thanh toán này có đơn giản và tiện dụng hay không?
☐ Rất phức tạp
☐ Phức tạp
☐ Bình thường
☐ Đơn giản
☐ Rất đơn giản
3. Bạn cảm thấy mô hình thanh toán này có an toàn hay không?
☐ Rất không an toàn
☐ Không an toàn
☐ Bình thường
☐ An toàn
☐ Rất an toàn
4. Bạn cảm thấy ứng dụng có dễ sử dụng không?
☐ Rất khó
☐ Khó
☐ Bình thường
☐ Dễ
☐ Rất dễ
5. Đánh giá tính thân thiện của giao diện người dùng?
☐ Rất không thân thiện
☐ Không thân thiện
☐ Bình thường
☐ Thân thiện
☐ Rất thân thiện

Mô hình "Nhân viên thanh toán chủ động lấy mã thanh toán"

6. Ứng dụng đã cung cấp đủ chức năng cho quá trình giao dịch hay chưa?
- ☐ Đã đầy đủ chức năng
 - ☐ Cần bổ sung thêm chức năng
7. Bạn cảm thấy quy trình thanh toán này có đơn giản và tiện dụng hay không?
- ☐ Rất phức tạp
 - ☐ Phức tạp
 - ☐ Bình thường
 - ☐ Đơn giản
 - ☐ Rất đơn giản
8. Bạn cảm thấy mô hình thanh toán này có an toàn hay không?
- ☐ Rất không an toàn
 - ☐ Không an toàn
 - ☐ Bình thường
 - ☐ An toàn
 - ☐ Rất an toàn
9. Bạn cảm thấy ứng dụng có dễ sử dụng không?
- ☐ Rất khó
 - ☐ Khó
 - ☐ Bình thường
 - ☐ Dễ
 - ☐ Rất dễ
10. Đánh giá tính thân thiện của giao diện người dùng?
- ☐ Rất không thân thiện
 - ☐ Không thân thiện
 - ☐ Bình thường
 - ☐ Thân thiện
 - ☐ Rất thân thiện

Mô hình Thanh Toán Khách Hàng Không Kết Nối Internet

11. Ứng dụng đã cung cấp đủ chức năng cho quá trình giao dịch hay chưa?
- ☐ Đã đầy đủ chức năng
 - ☐ Cần bổ sung thêm chức năng
12. Bạn cảm thấy quy trình thanh toán này có đơn giản và tiện dụng hay không?
- ☐ Rất phức tạp
 - ☐ Phức tạp
 - ☐ Bình thường
 - ☐ Đơn giản
 - ☐ Rất đơn giản
13. Bạn cảm thấy mô hình thanh toán này có an toàn hay không?
- ☐ Rất không an toàn
 - ☐ Không an toàn
 - ☐ Bình thường
 - ☐ An toàn
 - ☐ Rất an toàn

23. Bạn cảm thấy ứng dụng có dễ sử dụng không?

- ☐ Rất khó
- ☐ Khó
- ☐ Bình thường
- ☐ Dễ
- ☐ Rất dễ

24. Đánh giá tính thân thiện của giao diện người dùng?

- ☐ Rất không thân thiện
- ☐ Không thân thiện
- ☐ Bình thường
- ☐ Thân thiện
- ☐ Rất thân thiện

So Sánh Giữa Các Mô Hình Thanh Toán

25. Mô hình thanh toán đơn giản, tiện dụng nhất?

- ☐ Mô hình "Khách hàng chủ động lấy mã thanh toán"
- ☐ Mô hình "Nhân viên thanh toán chủ động lấy mã thanh toán"
- ☐ Mô hình Thanh Toán Khách Hàng Không Kết Nối Internet

26. Mô hình thanh toán có độ bảo mật tốt nhất?

- ☐ Mô hình "Khách hàng chủ động lấy mã thanh toán"
- ☐ Mô hình "Nhân viên thanh toán chủ động lấy mã thanh toán"
- ☐ Mô hình Thanh Toán Khách Hàng Không Kết Nối Internet

27. Mô hình thanh toán dễ sử dụng nhất?

- ☐ Mô hình "Khách hàng chủ động lấy mã thanh toán"
- ☐ Mô hình "Nhân viên thanh toán chủ động lấy mã thanh toán"
- ☐ Mô hình Thanh Toán Khách Hàng Không Kết Nối Internet

28. Mô hình thanh toán có giao diện thân thiện nhất?

- ☐ Mô hình "Khách hàng chủ động lấy mã thanh toán"
- ☐ Mô hình "Nhân viên thanh toán chủ động lấy mã thanh toán"
- ☐ Mô hình Thanh Toán Khách Hàng Không Kết Nối Internet

Kết Luận

29. Nếu mô hình được triển khai vào thực tế bạn có muốn sử dụng không?

- ☐ Có
- ☐ Không

30. Mô hình thanh toán bạn muốn sử dụng để thực hiện giao dịch?

- ☐ Mô hình "Khách hàng chủ động lấy mã thanh toán"
- ☐ Mô hình "Nhân viên thanh toán chủ động lấy mã thanh toán"
- ☐ Mô hình Thanh Toán Khách Hàng Không Kết Nối Internet

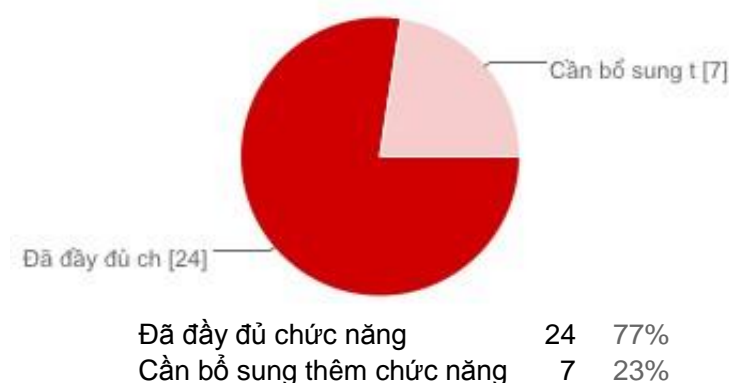
A.2 Kết quả khảo sát.

31 câu trả lời

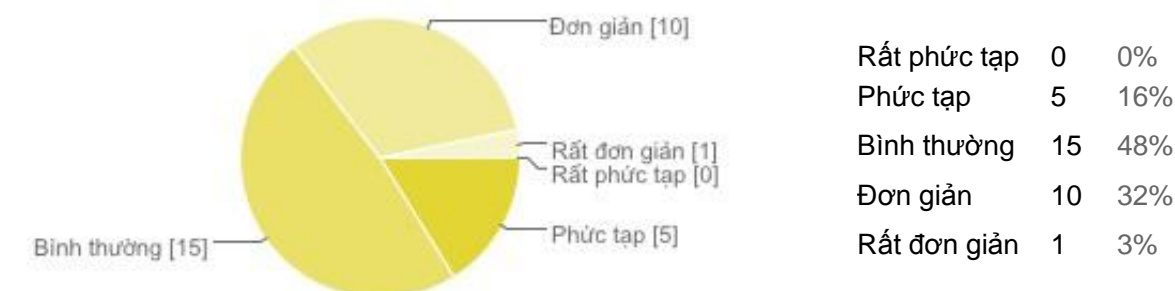
Phản Kiểm Tra Trải Nghiệm Người Dùng Với 3 Mô hình thanh toán

Mô hình "Khách hàng chủ động lấy mã thanh toán"

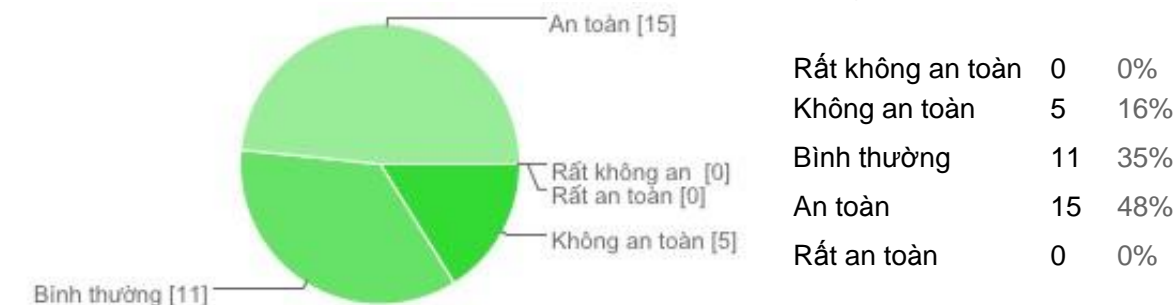
Ứng dụng đã cung cấp đủ chức năng cho quá trình giao dịch hay chưa?



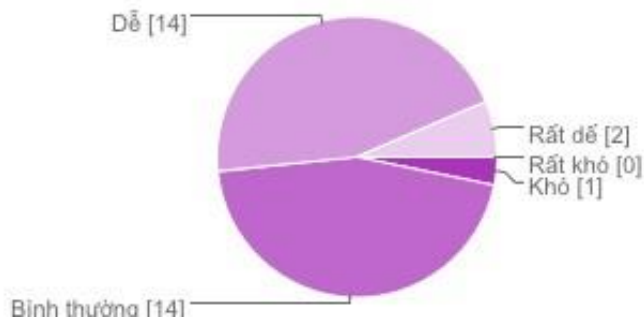
Bạn cảm thấy quy trình thanh toán này có đơn giản và tiện dụng hay không?



Bạn cảm thấy mô hình thanh toán này có an toàn hay không?

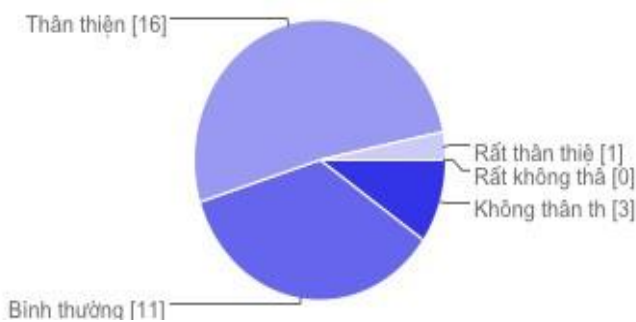


Bạn cảm thấy ứng dụng có dễ sử dụng không?



Rất khó	0	0%
Khó	1	3%
Bình thường	14	45%
Dễ	14	45%
Rất dễ	2	6%

Đánh giá tính thân thiện của giao diện người dùng?



Rất không thân thiện	0	0%
Không thân thiện	3	10%
Bình thường	11	35%
Thân thiện	16	52%
Rất thân thiện	1	3%

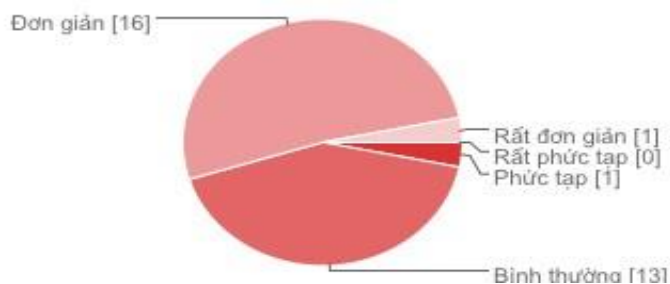
Mô hình "Nhân viên thanh toán chủ động lấy mã thanh toán"

Ứng dụng đã cung cấp đủ chức năng cho quá trình giao dịch hay chưa?



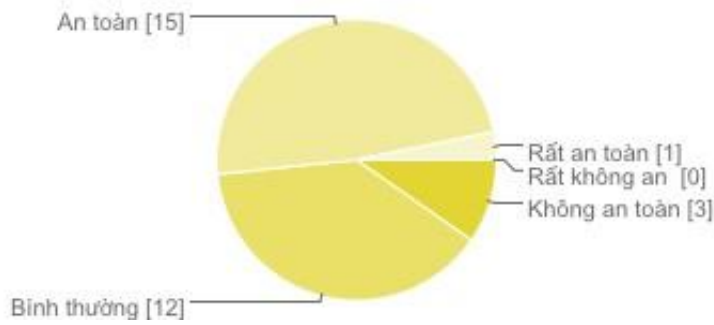
Đã đầy đủ chức năng	27	90%
Cần bổ sung thêm chức năng	3	10%

Bạn cảm thấy quy trình thanh toán này có đơn giản và tiện dụng hay không?



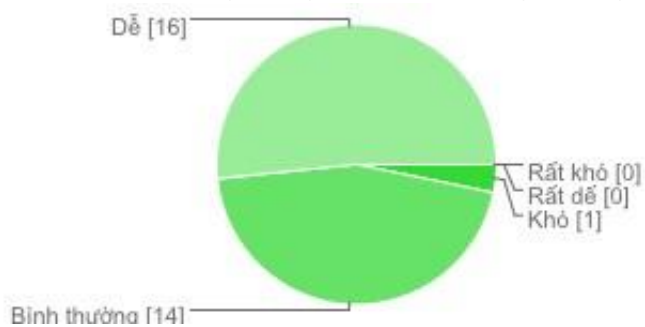
Rất phức tạp	0	0%
Phức tạp	1	3%
Bình thường	13	42%
Đơn giản	16	52%
Rất đơn giản	1	3%

Bạn cảm thấy mô hình thanh toán này có an toàn hay không?



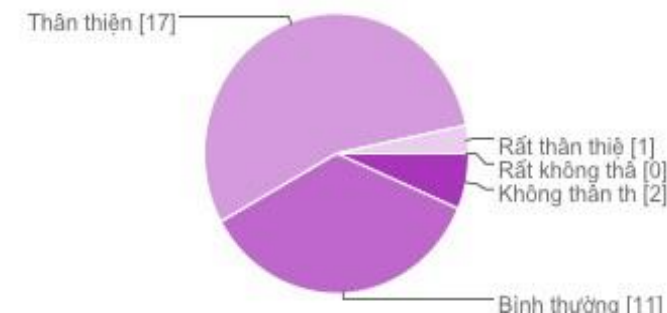
Rất không an toàn	0	0%
Không an toàn	3	10%
Bình thường	12	39%
An toàn	15	48%
Rất an toàn	1	3%

Bạn cảm thấy ứng dụng có dễ sử dụng không?



Rất khó	0	0%
Khó	1	3%
Bình thường	14	45%
Dễ	16	52%
Rất dễ	0	0%

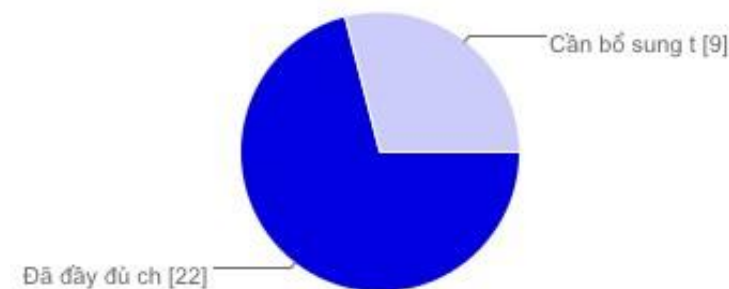
Đánh giá tính thân thiện của giao diện người dùng?



Rất không thân thiện	0	0%
Không thân thiện	2	6%
Bình thường	11	35%
Thân thiện	17	55%
Rất thân thiện	1	3%

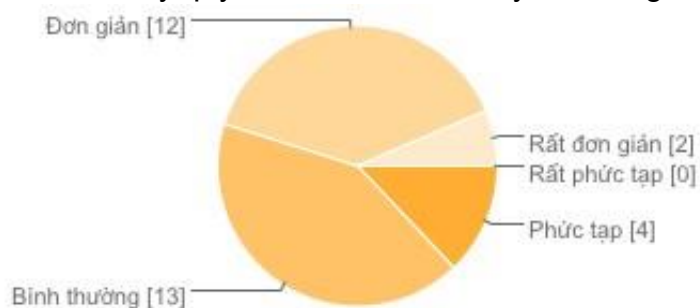
Mô hình Thanh Toán Khách Hàng Không Kết Nối Internet

Ứng dụng đã cung cấp đủ chức năng cho quá trình giao dịch hay chưa?



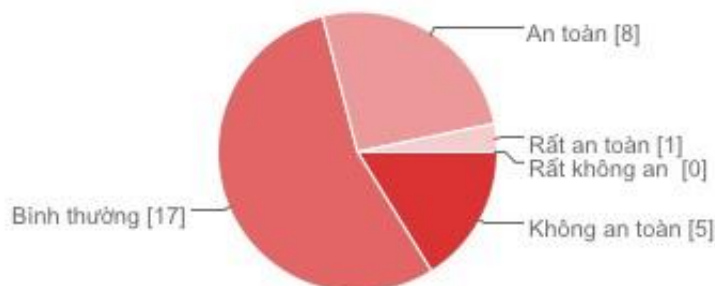
Đã đầy đủ chức năng	22	71%
Cần bổ sung thêm chức năng	9	29%

Bạn cảm thấy quy trình thanh toán này có đơn giản và tiện dụng hay không?



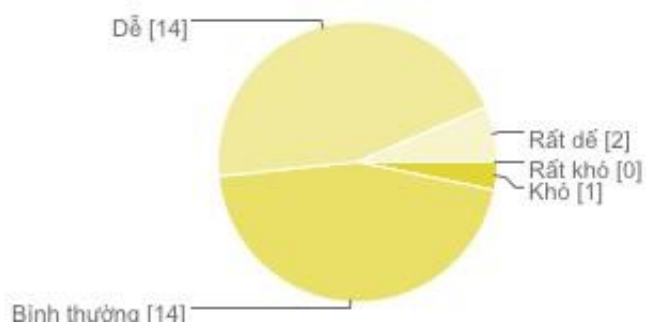
Rất phức tạp	0	0%
Phức tạp	4	13%
Bình thường	13	42%
Đơn giản	12	39%
Rất đơn giản	2	6%

Bạn cảm thấy mô hình thanh toán này có an toàn hay không?



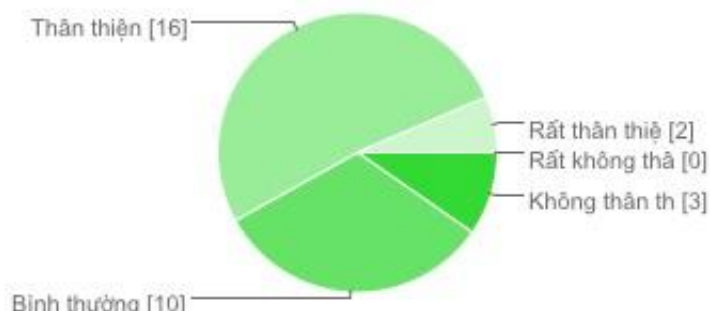
Rất không an toàn	0	0%
Không an toàn	5	16%
Bình thường	17	55%
An toàn	8	26%
Rất an toàn	1	3%

Bạn cảm thấy ứng dụng có dễ sử dụng không?



Rất khó	0	0%
Khó	1	3%
Bình thường	14	45%
Dễ	14	45%
Rất dễ	2	6%

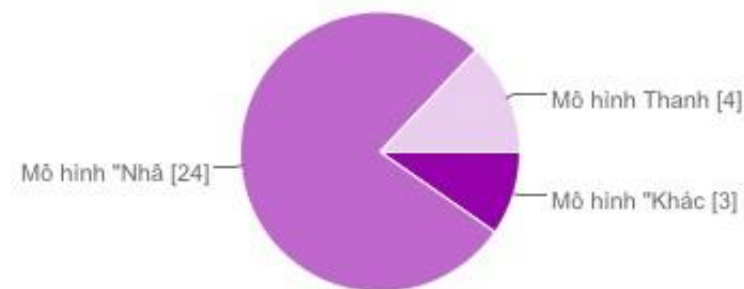
Đánh giá tính thân thiện của giao diện người dùng?



Rất không thân thiện	0	0%
Không thân thiện	3	10%
Bình thường	10	32%
Thân thiện	16	52%
Rất thân thiện	2	6%

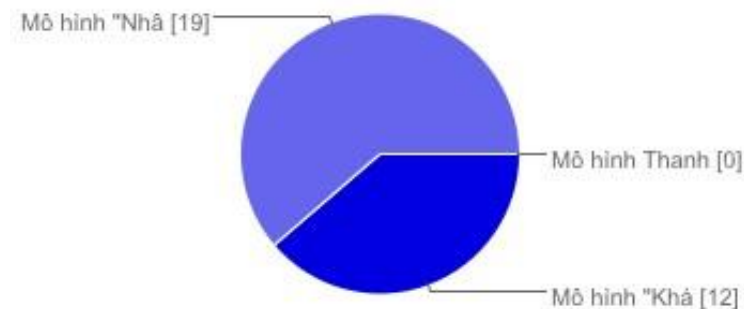
So Sánh Giữa Các Mô Hình Thanh Toán

Mô hình thanh toán đơn giản, tiện dụng nhất?



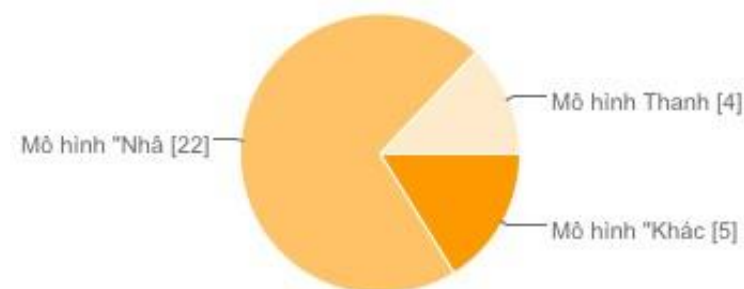
Mô hình "Khách hàng chủ động lấy mã thanh toán"	3	10%
Mô hình "Nhân viên thanh toán chủ động lấy mã thanh toán"	24	77%
Mô hình Thanh Toán Khách Hàng Không Kết Nối Internet	4	13%

Mô hình thanh toán có độ bảo mật tốt nhất?



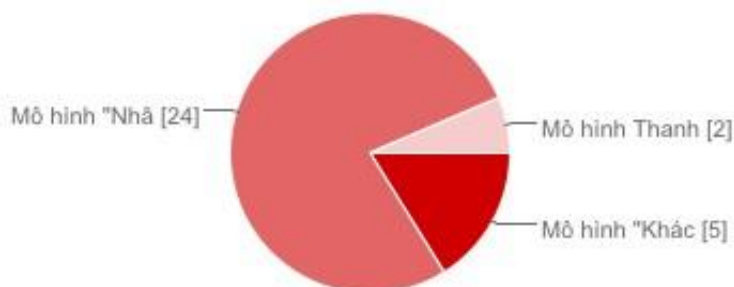
Mô hình "Khách hàng chủ động lấy mã thanh toán"	12	39%
Mô hình "Nhân viên thanh toán chủ động lấy mã thanh toán"	19	61%
Mô hình Thanh Toán Khách Hàng Không Kết Nối Internet	0	0%

Mô hình thanh toán dễ sử dụng nhất?



Mô hình "Khách hàng chủ động lấy mã thanh toán"	5	16%
Mô hình "Nhân viên thanh toán chủ động lấy mã thanh toán"	22	71%
Mô hình Thanh Toán Khách Hàng Không Kết Nối Internet	4	13%

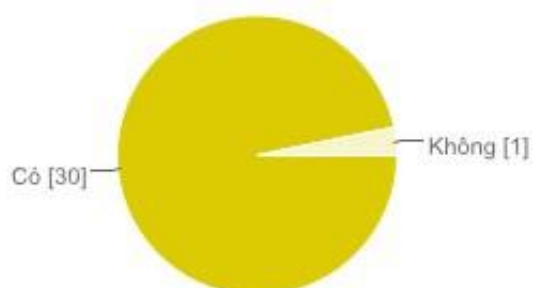
Mô hình thanh toán có giao diện thân thiện nhất?



Mô hình "Khách hàng chủ động lấy mã thanh toán"	5	16%
Mô hình "Nhân viên thanh toán chủ động lấy mã thanh toán"	24	77%
Mô hình Thanh Toán Khách Hàng Không Kết Nối Internet	2	6%

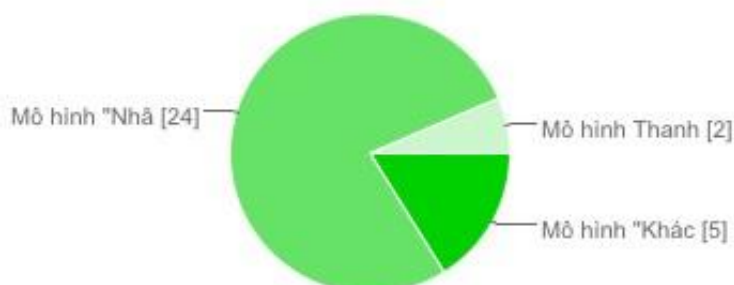
Kết Luận

Nếu mô hình được triển khai vào thực tế bạn có muốn sử dụng không?



Có	30	97%
Không	1	3%

Mô hình thanh toán bạn muốn sử dụng để thực hiện giao dịch?



Mô hình "Khách hàng chủ động lấy mã thanh toán"	5	16%
Mô hình "Nhân viên thanh toán chủ động lấy mã thanh toán"	24	77%
Mô hình Thanh Toán Khách Hàng Không Kết Nối Internet	2	6%