

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THỐNG THÔNG TIN**



KHOÁ LUẬN TỐT NGHIỆP

**XÂY DỰNG ỨNG DỤNG QUẢNG CÁO VÀ
ĐÁNH GIÁ SẢN PHẨM TRÊN MẠNG XÃ HỘI
DỰA VÀO PHẢN HỒI CỦA NGƯỜI DÙNG**

Giảng viên hướng dẫn : **TS. NGUYỄN ANH TUẤN**

Sinh viên thực hiện : **NGUYỄN MINH CHIẾN**

TRẦN THANH TÙNG

Lớp : **HTTT02**

Khoá : **2007 – 2012**

TP. Hồ Chí Minh, tháng 2 năm 2012

MỞ ĐẦU

Khi tính cạnh tranh của các sản phẩm giữa các nhà kinh doanh diễn ra ngày càng khốc liệt thì cũng đồng nghĩa với các chiến dịch quảng cáo không ngừng ra đời. Tận dụng những lợi thế của Internet, quảng cáo trực tuyến xuất hiện ở hầu hết các trang web, người dùng có thể dễ dàng tiếp cận với sản phẩm, nhưng những thông tin về sản phẩm chưa đầy đủ và không đáng tin cậy.

Ứng dụng mà nhóm đã phát triển sẽ tạo nên một môi trường chứa đựng những thông tin sản phẩm đầy đủ và đáng tin cậy, hỗ trợ người dùng đưa ra các quyết định đúng đắn.

Với tác phong công nghiệp hiện đại, thời gian được xem là tiền bạc, thì ứng dụng sẽ hạn chế tối đa những thủ tục (đăng nhập, tạo hồ sơ chứa thông tin cá nhân...) cho người dùng có thể sử dụng hết tính năng của ứng dụng nhanh chóng thông qua việc lấy thông tin của người dùng sẵn có đâu đó trên Internet.

Đề tài được thực hiện trong khoảng thời gian tương đối ngắn, do đó nhóm thực hiện đề tài đã rất nỗ lực để đề tài được tốt nhất nhưng sẽ không tránh khỏi một số sai sót nào đó, nên rất mong quý thầy cô và các bạn thông cảm và đóng góp ý kiến để ứng dụng hoàn chỉnh hơn.

Chúng em xin chân thành cảm ơn!

Nhóm sinh viên thực hiện:

Nguyễn Minh Chiến

Trần Thanh Tùng

LỜI CẢM ƠN

Nhân cơ hội này, chúng em xin gửi lời cảm ơn chân thành đến khoa Hệ Thống Thông Tin và trường Đại Học Công Nghệ Thông Tin đã tạo điều kiện thuận lợi cho chúng em học tập và hoàn thành tốt đề tài của nhóm.

Chúng em xin chân thành cảm ơn TS. NGUYỄN ANH TUẤN. Dưới sự hướng dẫn của thầy, chúng em đã học hỏi được rất nhiều kiến thức mới và hay. Sự tận tình và nhiệt huyết của thầy là nguồn động lực dồi dào, thúc đẩy chúng em không ngừng nỗ lực để đề tài hoàn thành tốt nhất.

Chúng em xin chân thành cảm ơn các thầy cô trong khoa Hệ Thống Thông Tin, cũng như tất cả các thầy cô của trường Đại Học Công Nghệ Thông Tin đã tận tình chỉ dạy, giúp chúng em trang bị những kiến thức quý báu sẽ trở thành hành trang giúp chúng em tự tin bước vào xã hội sau khi rời ghế nhà trường.

Xin gửi lời cảm ơn đến tất cả bạn bè trong trường, mọi người luôn sẵn sàng giúp đỡ, động viên và ủng hộ nhóm trong quá trình học tập cũng như phát triển đề tài.

TPHCM, tháng 2 / 2012.

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

[illegible]

MỤC LỤC

Chương 1: GIỚI THIỆU	1
1.1. Giới thiệu đề tài:	1
1.1.1. Mục tiêu đề tài:.....	1
1.1.2. Động cơ thực hiện:	1
1.1.3. Phạm vi nghiên cứu:	2
1.1.4. Những đóng góp:	3
1.2. Cấu trúc bài báo cáo:	4
Chương 2: KIẾN THỨC NỀN TẢNG	5
2.1. Khảo sát:	5
2.2. Kiến thức nền:	6
2.2.1. Mạng xã hội:.....	6
2.2.2. Đánh giá và đề nghị (Evaluations & Recommendations):.....	8
2.2.3. Quảng cáo trực tuyến:	10
2.2.4. Ontology:	12
2.2.5. Ontology Web Language (OWL):	15
2.2.6. SPARQL:	17
2.2.7. Friend Of A Friend (FOAF):	20
2.2.8. Protégé API:	20
2.2.9. Facebook API:.....	20
2.3. Yêu cầu chức năng:	28
2.4. Yêu cầu phi chức năng:.....	29
2.4.1. Khả năng tái sử dụng và mở rộng:.....	29
2.4.2. Độ tin cậy:.....	29

2.4.3.	Tốc độ:.....	29
2.4.4.	Khả năng hỗ trợ người dùng:	29
2.5.	Kết chương:	29
Chương 3: XÂY DỰNG CƠ SỞ TRI THỨC		30
3.1.	Lược đồ use case:.....	30
3.1.1.	Danh sách các tác nhân:	30
3.1.2.	Danh sách các use case:	31
3.2.	Đặc tả các use case:	34
3.2.1.	Giới thiệu một quán cà phê:	34
3.2.2.	Cập nhật thông tin một quán cà phê:	35
3.2.3.	Chia sẻ hình ảnh về một quán cà phê hay một dịch vụ:.....	36
3.2.4.	Thêm một dịch vụ vào danh sách các dịch vụ do quán phục vụ: ...	37
3.2.5.	Chia sẻ một quán lên Facebook:	38
3.2.6.	Xác nhận là khách hàng của một quán cà phê:	39
3.2.7.	Xác nhận đang hay đã từng làm việc tại một quán cà phê:	39
3.2.8.	Xác nhận là chủ quán cà phê:.....	40
3.2.9.	Xác nhận quyền chỉnh sửa thông tin quán cà phê:	40
3.2.10.	Xem danh sách dịch vụ của một quán cà phê:	41
3.2.11.	Xem danh sách những khách hàng của một quán cà phê:	42
3.2.12.	Vẽ sơ đồ mối liên hệ giữa một người dùng và một quán cà phê: ...	43
3.2.13.	Xem chi tiết một quán cà phê:.....	44
3.2.14.	Xem danh sách các quán được chia sẻ theo thứ tự gần nhất:	45
3.2.15.	Xem mối liên hệ giữa người dùng và quán:	46
3.2.16.	Xác nhận dịch vụ của quán:.....	47
3.2.17.	Quản lý các dịch vụ của quán:	47

3.2.18.	Cho điểm một quán cà phê hay một dịch vụ:	48
3.2.19.	Bình luận:	48
3.2.20.	Thích một quán hay một loại cà phê:	49
3.2.21.	Xem danh sách các lời bình cho một quán cà phê:.....	50
3.2.22.	Xem điểm đánh giá một quán cà phê:	51
3.2.23.	Xem danh sách những khách hàng thích một quán cà phê:.....	51
3.2.24.	Hết thích một quán cà phê hay một dịch vụ:	52
3.2.25.	Tìm kiếm:.....	53
3.2.26.	Tìm kiếm người dùng:.....	53
3.2.27.	Tìm kiếm quán cà phê:	54
3.2.28.	Tìm kiếm dịch vụ:.....	54
3.2.29.	Gợi ý kết quả tìm kiếm:	55
3.2.30.	Đề nghị những quán nên đến:.....	55
3.2.31.	Đăng nhập:.....	56
3.2.32.	Đăng xuất:.....	57
3.2.33.	Cập nhật thông tin tài khoản:	57
3.2.34.	Xem danh sách các quán mà người dùng có quyền chỉnh sửa:	58
3.2.35.	Xem danh sách các quán mà người dùng thích:	59
3.2.36.	Xem danh sách các dịch vụ mà người dùng thích:	59
3.2.37.	Xem danh sách các quán mà người dùng đã đánh giá:	59
3.2.38.	Xem danh sách các hoạt động gần đây của người dùng:	60
3.2.39.	Xem mạng lưới mối quan hệ của người dùng:.....	60
3.2.40.	Xem danh sách các quán mà người dùng đã đến:.....	61
3.2.41.	Xem danh sách các quán mà người dùng có thể đã đến:	61
3.2.42.	Quản lý danh sách người dùng:.....	62

3.2.43.	Quản lý danh sách các quán cà phê:	62
3.2.44.	Quản lý danh sách các dịch vụ:	63
3.2.45.	Quản lý các thông số hệ thống:	63
3.3.	Ontology:	64
3.3.1.	Danh sách các khái niệm (concept):	64
3.3.2.	Sơ đồ:	66
3.3.3.	Danh sách các thuộc tính (slot):	67
3.3.4.	Các ràng buộc (facet):	69
3.4.	Kết chương:	70
Chương 4:	THIẾT KẾ CHƯƠNG TRÌNH	71
4.1.	Danh sách các use case sẽ hiện thực:	71
4.1.1.	Danh sách các use case sẽ hiện thực:	71
4.1.2.	Danh sách các đối tượng:	71
4.2.	Mô tả các tính năng:	74
4.2.1.	Giới thiệu một quán cà phê (Share shop):	74
4.2.2.	Chia sẻ hình ảnh về một quán cà phê hay dịch vụ (Share photo): ..	75
4.2.3.	Thêm một dịch vụ vào danh sách các dịch vụ do quán phục vụ (Share service):	77
4.2.4.	Xác nhận là khách hàng của một quán (Is customer):	78
4.2.5.	Xem danh sách những khách hàng của một quán cà phê (View shop's customers):	79
4.2.6.	Xem chi tiết một quán cà phê (View shop):	80
4.2.7.	Xem danh sách các quán cà phê được chia sẻ theo thứ tự gần nhất (List shops):	81
4.2.8.	Xác nhận dịch vụ của quán (Review shop's services):	82

4.2.9.	Xem mối liên hệ giữa người dùng và quán (User-shop relationship):	83
4.2.10.	Bình luận (Comment):.....	85
4.2.11.	Xem danh sách các lời bình (Get comments):	86
4.2.12.	Tìm kiếm quán cà phê (Search shops):	86
4.2.13.	Đề nghị những quán nên đến (Recommend shops):.....	87
4.2.14.	Đăng nhập (Log in):	91
4.2.15.	Xem danh sách các quán mà người dùng có quyền chỉnh sửa (Manage shops):.....	93
4.2.16.	Xem danh sách các hoạt động gần đây của người dùng (User's feeds):	93
4.2.17.	Xem mạng lưới mối quan hệ của người dùng (User-shop relationship):.....	94
4.3.	Hiện thực:.....	95
4.3.1.	Trang chủ:	95
4.3.2.	Trang chia sẻ một quán cà phê mới:.....	96
4.3.3.	Trang xem chi tiết một quán cà phê:	97
4.3.4.	Trang liệt kê danh sách các quán cà phê:.....	98
4.3.5.	Trang hiển thị kết quả tìm kiếm:.....	99
4.3.6.	Hồ sơ cá nhân của người dùng:.....	99
4.3.7.	Đề nghị quán nên đến:	101
4.3.8.	Các trang dành cho người dùng quản lý các quán cà phê:	102
4.3.9.	Trang quản lý người dùng dành cho thành viên quản trị:.....	103
4.3.10.	Trang quản lý quán dành cho thành viên quản trị:	103
4.3.11.	Trang quản lý dịch vụ dành cho thành viên quản trị:.....	103
4.3.12.	Các trang quản lý các thông số hệ thống:	104

4.4. Đánh giá:	106
Chương 5: KẾT LUẬN và HƯỚNG PHÁT TRIỂN	108
5.1. Tổng kết đề tài:.....	108
5.2. Kết luận:	109
5.3. Hướng phát triển:	109
DANH MỤC TÀI LIỆU THAM KHẢO	110
Tài liệu tiếng Anh.....	110
PHỤ LỤC	112
Bảng phân công công việc:	112
Nơi lưu trữ mã nguồn:	112
Làm việc với Graph API, PHP SDK:	112
Cài đặt và bắt đầu:	112
Xác thực và ủy quyền (<i>Authentication & Authorization</i>):.....	113
Kết hợp PHP SDK và Javascript SDK:	113
Một số đối tượng Facebook (Facebook Objects) quan trọng:	115
Một số câu hỏi và cách giải quyết bằng SPARQL:	128
1. Ai là bạn của A:	128
2. Bạn của bạn của A:	128
3. A đã từng đến thăm quán cà phê nào:.....	129
4. A đánh giá quán cà phê C bao nhiêu điểm:.....	129
5. Quán cà phê nào có phục vụ dịch vụ S:.....	129
6. Quán cà phê nào mà bạn của A đánh giá cao:	130
7. Quán cà phê nào mà bạn của A thích:	130
8. Quán cà phê nào mà bạn của A đã ghé thăm:	131
9. Quán cà phê nào mà bạn của A đã ghé thăm có phục vụ loại dịch vụ mà A và bạn của A cùng thích.....	131

10. Quán cà phê nào mà bạn của A đã ghé thăm có phục vụ loại dịch vụ mà A thích: 132
11. Quán cà phê nào mà bạn của A thích và có phục vụ loại dịch vụ A thích: 132
12. Quán cà phê nào mà bạn của A đánh giá cao và có phục vụ loại dịch vụ A thích: 133

DANH MỤC CÁC BẢNG

Bảng 1: các use case về quán và dịch vụ	31
Bảng 2: Các use case đánh giá và chia sẻ	32
Bảng 3: các use case tìm kiếm	32
Bảng 4: use case tư vấn.....	32
Bảng 5: các use case về hồ sơ người dùng	33
Bảng 6: các use case dành cho các thành viên quản trị	33
Bảng 7: Quyền sử dụng của từng use case với từng tác nhân.....	34
Bảng 8: mô tả use case Giới thiệu quán cà phê	35
Bảng 9: mô tả use case Cập nhật thông tin một quán cà phê	36
Bảng 10: mô tả use case Chia sẻ hình.....	37
Bảng 11: mô tả use case Chia sẻ dịch vụ.....	38
Bảng 12: mô tả use case Chia sẻ một quán lên Facebook.....	39
Bảng 13: mô tả use case Xác nhận khách hàng	39
Bảng 14: mô tả use case Xác nhận đang hay đã làm việc	40
Bảng 15: mô tả use case Xác nhận là chủ quán.....	40
Bảng 16: mô tả use case Xác nhận quyền chỉnh sửa.....	41
Bảng 17: mô tả use case Xem danh sách dịch vụ.....	41
Bảng 18: mô tả use case Xem danh sách khách hàng	42
Bảng 19: mô tả use case Vẽ sơ đồ mối liên hệ	43
Bảng 20: mô tả use case Xem chi tiết một quán.....	45
Bảng 21: mô tả use case Xem danh sách các quán	46

Bảng 22: mô tả use case Xem mối liên hệ.....	47
Bảng 23: mô tả use case Xác nhận dịch vụ	47
Bảng 24: mô tả use case Quản lý các dịch vụ.....	48
Bảng 25: mô tả use case Cho điểm quán	48
Bảng 26: mô tả use case Bình luận.....	49
Bảng 27: mô tả use case Thích	49
Bảng 28: mô tả use case Xem danh sách lời bình	50
Bảng 29: mô tả use case Xem điểm đánh giá	51
Bảng 30: mô tả use case Xem danh sách khách hàng thích quán.....	52
Bảng 31: mô tả use case Hết thích.....	52
Bảng 32: mô tả use case Tìm kiếm.....	53
Bảng 33: mô tả use case Tìm kiếm người dùng.	54
Bảng 34: tìm kiếm quán cà phê	54
Bảng 35: tìm kiếm dịch vụ	55
Bảng 36: gợi ý tìm kiếm	55
Bảng 37: đề nghị những quán nên đến.....	56
Bảng 38: đăng nhập.....	57
Bảng 39: đăng xuất.....	57
Bảng 40: cập nhật thông tin tài khoản	58
Bảng 41: Danh sách người dùng có quyền chỉnh sửa.....	58
Bảng 42: Danh sách các quán mà người dùng thích	59
Bảng 43: dịch vụ người dùng thích	59
Bảng 44: quán người dùng đã đánh giá	60
Bảng 45: hoạt động gần đây người dùng.....	60
Bảng 46: mạng lưới quan hệ người dùng	61
Bảng 47: quán người dùng đã đến.....	61
Bảng 48: quán có thể đã đến	62
Bảng 49: danh sách người dùng.....	62
Bảng 50: danh sách quán cà phê	63
Bảng 51: Danh sách dịch vụ.....	63
Bảng 52: quản lý thông số hệ thống.....	64

Bảng 53: danh sách các khai niệm	65
Bảng 54: danh sách các thuộc tính của person	68
Bảng 55: thuộc tính của coffeeshop	68
Bảng 56: thuộc tính của service	68
Bảng 57: thuộc tính của comment	69
Bảng 58: thuộc tính của rating	69
Bảng 59: thuộc tính của check-in	69
Bảng 60: use case đã thực hiện	71
Bảng 61: danh sách các đối tượng	73
Bảng 62: Bảng phân công công việc	112
Bảng 63: Các tham số Facebook API	124
Bảng 64: các tham số để đăng nhập Facebook	126

DANH MỤC CÁC HÌNH

Hình 1: nút Like	21
Hình 2: Recent Activity plugin	22
Hình 3: nút login	23
Hình 4: Các tác nhân	30
Hình 5: use case Giới thiệu quán cà phê	34
Hình 6: use case Cập nhật thông tin một quán cà phê	35
Hình 7: use case Chia sẻ hình ảnh	36
Hình 8: use case Chia sẻ dịch vụ	37
Hình 9: use case Chia sẻ một quán lên Facebook	38
Hình 10: use case Xác nhận quyền chỉnh sửa	40
Hình 11: use case Xem danh sách khách hàng	42
Hình 12: use case Vẽ sơ đồ mối liên hệ	43
Hình 13: use case Xem chi tiết một quán cà phê	44
Hình 14: use case Xem danh sách các quán	45
Hình 15: use case Xem mối liên hệ	46
Hình 16: use case Quản lý các dịch vụ	47
Hình 17: use case Thích	49
Hình 18: use case Xem danh sách lời bình	50
Hình 19: use case Xem điểm đánh giá	51
Hình 20: use case Tìm kiếm	53
Hình 21: đăng nhập	56
Hình 22: Danh sách người dùng có quyền chỉnh sửa	58
Hình 23: danh sách người dùng	62
Hình 24: cấu trúc ontology	66
Hình 25: giới thiệu quán cà phê	74
Hình 26: chia sẻ quán cà phê, dịch vụ	76
Hình 27: thêm dịch vụ vào quán	77
Hình 28: xác nhận là khách hàng	78
Hình 29: khách hàng của quán	79
Hình 30: xem chi tiết quán	80

Hình 31: quán được chia sẻ gần nhất	81
Hình 32: xác nhận dịch vụ của quán	82
Hình 33: mối quan hệ người dùng và quán	83
Hình 34: Bình luận	85
Hình 35: lời mời cho một quán cà phê	86
Hình 36: tìm kiếm quán cà phê	86
Hình 37: đề nghị quán nên đến	87
Hình 38: đăng nhập	91
Hình 39: kiểm tra đăng nhập	91
Hình 40: xác nhận đăng nhập	92
Hình 41: quán có quyền chỉnh sửa	93
Hình 42: hoạt động gần đây của người dùng	93
Hình 43: mạng lưới liên kết	94
Hình 44: Giao diện trang chủ	95
Hình 45: Giao diện trang chia sẻ quán	96
Hình 46: Một phần của trang liệt kê danh sách các quán	98
Hình 47: Hiển thị kết quả tìm kiếm với từ khóa "Br"	99
Hình 48: các hoạt động gần đây	99
Hình 49: các quán đã đến	99
Hình 50: các quán và dịch vụ người dùng thích	100
Hình 51: các quán đã đánh giá	100
Hình 52: "mạng lưới" của người dùng	100
Hình 53: thống kê quan hệ giữa người dùng và quán	100
Hình 54: những quán được đề nghị và lí do	101
Hình 55: các quán do người dùng chia sẻ	102
Hình 56: người dùng kiểm soát những người có quyền quản lý quán	102
Hình 57: Quản lý tất cả các thành viên	103
Hình 58: quản lý tất cả các quán	103
Hình 59: Liệt kê danh sách các dịch vụ	103
Hình 60: Giao diện cho phép thành viên quản trị thêm dịch vụ mới	104
Hình 61: các thông số để hệ thống hiển thị quán	104

Hình 62: thông số tùy biến trang chủ.....	105
Hình 63: tùy biến các hoạt động.....	105
Hình 64: điều chỉnh cách vẽ sơ đồ	106
Hình 65: điều chỉnh liên quan đến quán	106

Chương 1: GIỚI THIỆU

1.1. Giới thiệu đề tài:

1.1.1. Mục tiêu đề tài:

Tham gia vào mạng xã hội đã dần thành một phần của đời sống con người trong thời đại kỹ thuật số. Một trong những điểm quan trọng của mạng xã hội là sự tin cậy vào mối quan hệ, các liên kết giữa bạn bè, người thân, người quen biết. Trong khi đó, các quảng cáo trên các phương tiện thông tin đại chúng có phần thiên lệch và chưa được kiểm chứng.

Đề tài này nhằm tạo ra một ứng dụng chạy trên nền tảng Facebook, cho phép nhà cung cấp (quán café) tự quảng cáo cho mình. Phần đánh giá sẽ là do khách hàng (*facebook's users*, bạn bè) tự đánh giá và cho điểm (*vote*). Sự đánh giá trung thực giữa bạn bè của từng cá nhân sẽ góp phần giúp người dùng lựa chọn sản phẩm hợp lý cho mình.

Một ví dụ cụ thể đó là khi một người bạn của A muốn tìm một quán cafe với những tiêu chí đặt ra ban đầu như: ngon, đẹp và rẻ để họp mặt bạn bè. Tuy nhiên, A sẽ ưu tiên chọn ra các nhà hàng mà bạn bè mình đã từng ghé qua và có bình chọn. Căn cứ vào những bình chọn của bạn bè, A sẽ cảm thấy yên tâm hơn trong sự lựa chọn của mình vì nếu chỉ căn cứ vào sự tự quảng cáo của quán thì chưa đủ cơ sở để đánh giá.

1.1.2. Động cơ thực hiện:

Có 2 lí do chính để nhóm chọn thực hiện đề tài này:

- ✓ Sự thăng hoa của các mạng xã hội:
 - Trong vài năm gần đây, các mạng xã hội đang có những bước phát triển hết sức mạnh mẽ. Các mạng xã hội đang dần thay thế các diễn đàn, trang blog... để trở thành **trung tâm chia sẻ tin tức** của người dùng mạng [11] .
 - Tại Việt Nam, tính đến hết năm 2010, đã có 16 triệu người sử dụng các mạng xã hội (theo báo cáo VietnamNetCitizens Report 2011 của công ty Cimigo).

- Tận dụng số lượng người dùng đông đảo và có dấu hiệu không ngừng gia tăng trên mạng xã hội để phục vụ mục tiêu của đề tài.
- ✓ Lợi thế của mạng xã hội:
 - Nhóm cho rằng, một điểm khác biệt quan trọng giữa mạng xã hội và các diễn đàn hay trang blog, yếu tố quyết định giúp mạng xã hội thăng hoa, chính là **cách tiếp cận thông tin [12]**.
 - Để có được những thông tin mình cần, người dùng phải tự tìm đến các diễn đàn, tìm trong từng chuyên mục hay sử dụng các công cụ tìm kiếm trên diễn đàn.
 - Mạng xã hội cung cấp một cách thức khác hoàn toàn. Đó là, **thông tin tự tìm đến người dùng**.
 - Một người dùng sử dụng mạng xã hội sẽ chia sẻ những mối quan hệ của anh ta (bạn bè, người thân, đồng nghiệp...), công việc, những sở thích, các hoạt động thường nhật... Đây là những cơ sở để các mạng xã hội “biết” những thông tin nào mà người dùng sẽ quan tâm. Và thông tin đó sẽ được gửi đến người dùng bằng rất nhiều hình thức, qua thư, blog cá nhân, website cá nhân và ngay tại trang chủ của các mạng xã hội [13].

Với những đặc trưng nổi bật của mạng xã hội, nhóm phát triển sẽ tạo ra một ứng dụng nhằm khai thác mối liên kết giữa bạn bè để giúp khách hàng lựa chọn được những sản phẩm ưng ý cho mình được một cách dễ dàng.

1.1.3. Phạm vi nghiên cứu:

1.1.3.1. Phạm vi công nghệ:

- ✓ Nghiên cứu Friend of a Friend ontology (FOAF) trong việc mô tả các mối quan hệ trên mạng xã hội.
- ✓ Xây dựng ontology cho hệ thống quảng cáo và đánh giá sản phẩm trên mạng xã hội dựa trên phản hồi của người dùng.
- ✓ Sử dụng ngôn ngữ truy vấn SPARQL trong việc phát hiện tri thức.

- ✓ Nghiên cứu sử dụng API của một mạng xã hội hiện có (Facebook) nhằm xây dựng một ứng dụng minh họa cho thuật toán.
- ✓ Ngôn ngữ: OWL, PHP, Java.
- ✓ Hệ quản trị cơ sở tri thức: Protégé.

1.1.3.2. Phạm vi địa lý:

Các quán cà phê thuộc địa bàn TPHCM, mà tập trung ở các quận 1, 3, 4, 7, 10, Phú Nhuận, Bình Thạnh.

1.1.3.3. Đối tượng nghiên cứu:

- Xây dựng ontology bằng công cụ Protégé 3.4.7.
- Nghiên cứu cách khai thác tri thức thông qua các mối liên kết tìm ẩn bằng ngôn ngữ truy vấn tri thức Sparql.
- Nghiên cứu các API từ Facebook cung cấp để đơn giản hóa quá trình đăng ký tài khoản và đăng nhập. Ngoài ra còn nghiên cứu cách lấy các mối liên kết giữa các người dùng trong Facebook.
- Nghiên cứu các *Web Service* cho việc trao đổi dữ liệu qua lại giữa ngôn ngữ lập trình PHP và Java.

1.1.4. Những đóng góp:

- Ứng dụng của nhóm là một minh họa rõ nét cho phương pháp khám phá tri thức từ mạng xã hội trên nền tảng ontology.
- Ứng dụng cho thấy sự tiện dụng của việc sử dụng Facebook như một hệ thống đăng nhập tự động.
- Sản phẩm của nhóm, giúp người dùng có được một môi trường tiếp cận với các quán cà phê trong phạm vi nhất định với đầy những thông tin hữu ích và đáng tin cậy. Từ đó hỗ trợ tích cực trong việc lựa chọn địa điểm lý tưởng để tận hưởng hương vị tuyệt vời của thức uống mình ưa thích.

1.2. Cấu trúc bài báo cáo:

Trong chương 1, như bạn đọc đã theo dõi, đề cập đến các thông tin tổng quát về đề tài, bao gồm: tên đề tài, giảng viên hướng dẫn và nhóm thực hiện đề tài. Chương 1 cũng khái quát về mục tiêu của đề tài, động lực giúp nhóm thực hiện chọn đề tài cũng như những thành quả mà nhóm sẽ đóng góp khi đề tài hoàn thành.

Chương 2 cung cấp kiến thức về những khái niệm và phương pháp chính làm nền tảng cho đề tài. Chương 2 cũng giúp bạn đọc nắm được các công trình nghiên cứu, đã và đang được thực hiện, có liên quan đến đề tài. Một phần rất quan trọng trong chương này là nhóm sẽ xác định các yêu cầu chức năng và phi chức năng cho sản phẩm minh họa đề tài.

Chương 3 phân tích các chức năng cụ thể cần hiện thực để đạt được các yêu cầu nêu ra trong chương 2. Chương 3 sẽ mô tả toàn bộ quá trình xây dựng nên một hệ cơ sở tri thức nhằm đáp ứng được những yêu cầu của chương trình.

Chương 4 tiếp sau đó sẽ đề cập chi tiết đến phương pháp thực hiện các chức năng trong chương 3. Trong chương này, bạn đọc cũng sẽ được thấy các kết quả chính của chương trình.

Chương 5 là kết luận của nhóm về kết quả thực hiện đề tài, ưu khuyết điểm của đề tài cũng như hướng phát triển trong tương lai. Phần Phụ lục đề cập đến một số thông tin thêm về đề tài. Phần Tài liệu tham khảo là danh sách các tài liệu mà nhóm đã tham khảo trong quá trình thực hiện đề tài.

Lưu ý, để theo dõi đề tài được dễ dàng hơn, hãy sử dụng chức năng Navigation của phần mềm MSWord.

Sau đây, mời bạn đọc bước vào chương 2: Kiến thức nền tảng.

Chương 2: KIẾN THỨC NỀN TẢNG

2.1. Khảo sát:

Hiện tại có nhiều website chuyên về việc quảng cáo, giới thiệu các quán cà phê như website thienduongcafe.com, vnnavi.com, vv...

Tiến hành khảo sát website vnnavi.com, nhóm nhận thấy những ưu điểm của nó như:

- Mỗi quán cà phê, đều được giới thiệu đầy đủ những thông tin cơ bản: tên quán, địa chỉ, giờ tiếp khách, giá trung bình, ... và một lời giới thiệu hấp dẫn về các dịch vụ hay ưu đãi của quán. Ngoài ra, còn có phần trình bày danh mục các dịch vụ, thức ăn, thức uống của quán có kèm theo đơn giá.
- Người dùng, khách hàng có thể đưa ra những nhận xét, đánh giá về quán. Và những nhận xét, đánh giá này, những người khác đều có thể đọc được

So với vnnavi.com, thì website thienduongcafe.com cũng gồm các chức năng kể trên, ngoài ra còn có những điểm nổi bật như:

- Mỗi quán cà phê còn kèm theo một album các ảnh giới thiệu về quán và tất cả người dùng đều có thể chia sẻ thêm nhiều hình ảnh khác.
- Người dùng có thể đăng ký tài khoản và đăng nhập vào website để sử dụng được nhiều tính năng hơn.

Nhìn chung các website về quảng cáo cà phê đều có được sự nổi bật về cách trình bày và bố trí quảng cáo với đầy những hình ảnh quán sinh động. Thông tin và nội dung của các quán khá đầy đủ, cho người dùng có cái nhìn tổng quát về quán. Tuy nhiên những thông tin được đưa ra thông thường là từ những người có mối liên hệ mật thiết với quán đó và vì thế dễ dẫn đến thiếu đi tính khách quan.

Ngoài ra, một số website cho phép người dùng đánh giá về một quán cà phê, và xây dựng thêm tính năng tư vấn, hỗ trợ khách hàng trong việc xác định quán cà phê “lạ” nào tốt nhất để đến. Trong các website như vậy, thì foodio54.com là một ví dụ điển hình nhất.

Khi xem chi tiết về quán cà phê Plaza tại <http://foodio54.com/restaurant/Omaha-NE/43474/Plaza-Coffee-Shop>, của website foodio54.com. Ta có thể thấy, nó hỗ trợ người dùng đưa ra đánh giá (*rating*) trên sản phẩm quán Plaza, xem đánh giá của những người dùng khác, và xem biểu đồ quá trình người dùng khuyến nghị sử dụng quán đó. Những đánh giá này sẽ được dùng trong việc xử lý để đưa ra kết quả khuyến nghị cho các khách hàng khác. Tuy nhiên có một khuyết điểm tồn tại, là người dùng có thể đưa ra những đánh giá không chính xác như đánh giá tốt cho quán cà phê của mình làm chủ và đánh giá thấp cho các quán đối thủ, do đó các kết quả khuyến nghị có thể bị sai lệch.

2.2. Kiến thức nền:

2.2.1. Mạng xã hội:

2.2.1.1. Mạng xã hội là gì?

Mạng xã hội được tạo ra để tự thân nó lan rộng trong cộng đồng thông qua các tương tác của các thành viên trong chính cộng đồng đó. Mọi thành viên trong mạng xã hội cùng kết nối và mỗi người là một mắt xích để tạo nên một mạng lưới rộng lớn truyền tải thông tin trong đó. Mạng xã hội như một dịch vụ web nền tảng cho phép mỗi thành viên trong mạng xã hội có thể [7] : (1) xây dựng cho riêng mình một hồ sơ cá nhân cộng đồng (*public*) hoặc bán cộng đồng (*semi-public*) trong một hệ thống giới hạn, (2) liệt kê danh sách các người dùng mà ta chưa kết nối và ta có thể kết nối đến bất kỳ ai trong danh sách này, (3) cho phép người dùng hiện tại xem, duyệt qua danh sách các kết nối được tạo ra bởi những người dùng trong hệ thống.

Về cơ bản, mạng xã hội giống như một trang web mở với nhiều ứng dụng khác nhau. Mạng xã hội khác với trang web thông thường ở cách truyền tải thông tin và tích hợp ứng dụng. Trang web thông thường cũng giống như truyền hình, cung cấp càng nhiều thông tin, thông tin càng hấp dẫn càng tốt còn mạng xã hội tạo ra các ứng dụng mở, các công cụ tương tác để mọi người tự tương tác và tạo ra dòng tin rồi cùng lan truyền dòng tin đó.

Điều làm cho mạng xã hội trở nên độc đáo là người dùng không thể gặp được “người lạ”, “người lạ” ở đây là những người dùng trong hệ thống, nhưng chưa kết nối (kết bạn) với ta. Tuy nhiên, mạng xã hội cho phép ta tìm kiếm và kết bạn với bất kỳ ai trong hệ thống. Nhưng điều tuyệt vời hơn của mạng xã hội, là khám phá ra những mối quan hệ tiềm ẩn giữa những người dùng với nhau từ đó tạo ra cho người dùng cảm giác thích thú và không ngừng khám phá.

2.2.1.2. Cấu trúc của mạng xã hội:

Mạng xã hội dựa trên một cấu trúc vững chắc cho phép người dùng vừa diễn đạt thông tin từ cá nhân họ, lại vừa gặp được những người dùng khác có cùng sở thích [8] . Cấu trúc này gồm có: hồ sơ cá nhân (*profiles*), bạn bè (*friends*), *blogs*, *widgets* và những điều thú vị độc đáo khác tùy thuộc vào từng website mạng xã hội cụ thể.

- *Profile*: chứa thông tin cá nhân của người dùng, nơi mà người dùng chia sẻ thông tin về họ cho cả thế giới biết. Các thông tin thông thường gồm có: tên, năm sinh, đến từ đâu, sở thích gì...
- *Friends*: là những người dùng mà ta xem là đáng tin cậy, họ được phép viết lên tường (*wall*) của ta, bình luận lên hồ sơ cá nhân hoặc gửi các tin nhắn riêng tư với ta.
- *Groups*: hay nhóm, thường được mạng xã hội sử dụng để người dùng tìm kiếm những người có cùng sở thích với mình hoặc tham gia vào những cuộc thảo luận về một đề tài nào đó.
- *Discussion*: mục đích chính của nhóm là để tạo ra sự tương tác qua lại giữa các người dùng trong những chủ đề thảo luận. hầu hết các mạng xã hội đều hỗ trợ bảng thảo luận, nơi cho phép các thành viên trong nhóm có thể chia sẻ hình ảnh, video, âm nhạc, và các thông tin khác.
- *Blogs*: tính năng khác của mạng xã hội là cho phép người dùng viết các bài blog. Blog thông qua mạng xã hội trở nên hoàn hảo trong việc thông báo tình hình của bạn trong mọi lúc.

- *Widgets*: *widget* làm cho cá tính của bạn trở nên nổi bật hơn thông qua việc trang trí hồ sơ cá nhân của bạn với các *widget* mà mạng xã hội hỗ trợ.

2.2.2. Đánh giá và đề nghị (Evaluations & Recommendations):

Hiện nay Collaborative Filtering (CF) được biết đến như là một trong số những phương pháp thành công nhất trong việc xây dựng một hệ thống tư vấn thông qua việc sử dụng dữ liệu của những người dùng này để tư vấn, dự đoán cho những người dùng khác [17] .

Tuy nhiên, CF cũng đang đối mặt với một số hạn chế và thách thức:

- Thiếu thông tin dữ liệu (*Data sparsity*)
 - Vấn đề thiếu thông tin dữ liệu có thể được gặp trong nhiều trường hợp, nhất là khi hệ thống có thêm người dùng mới hoặc sản phẩm mới. Khi đó hệ thống không có đầy đủ thông tin về người dùng và khó tìm ra được nhóm người dùng tương tự. Những sản phẩm mới thì không được tư vấn cho người dùng cho đến khi chúng được người dùng đánh giá.
 - Trong một số trường hợp, người dùng không mua hoặc mua sản phẩm rất ít và cũng ít khi đánh giá sản phẩm. Do đó thông tin về đánh giá sản phẩm và lịch sử mua hàng của người dùng ít và dẫn đến những tư vấn không chính xác của hệ thống.
 - Hệ thống dùng một con số phần trăm được gọi là ***coverage*** để xác định khi nào thì hệ thống có thể và không thể đưa ra tư vấn cho người dùng. Do đó với những người dùng đưa ra đánh giá trên số lượng sản phẩm quá ít so với tổng số sản phẩm có trong hệ thống thì hệ thống sẽ không thể đưa ra tư vấn cho họ.
- Khả năng mở rộng (*Scalability*): Khi số lượng người dùng và sản phẩm tăng vọt nhanh chóng, các thuật toán cơ bản của Collaborative filtering sẽ có độ phức tạp rất cao. Ngoài ra hệ thống tư vấn đòi hỏi

phải có sự phản hồi về cho người dùng ngay lập tức và đưa ra những tư vấn không phụ thuộc vào đánh giá hay lịch sử mua hàng của họ.

- Tính đồng nghĩa (*Synonym*)
 - Trong một số trường hợp khi mà các sản phẩm tương tự nhau, nhưng rất khác nhau về tên, mô tả hoặc một số thông tin khác. Khi đó các hệ thống tư vấn sẽ không có khả năng để phát hiện ra sự tương đồng giữa hai sản phẩm và từ đó đưa ra những tư vấn về hai sản phẩm rất khác nhau.
 - Minh họa cho tính đồng nghĩa, giả sử ta có 2 sản phẩm là “children movie” và “Children film” có nội dung rất giống nhau, nhưng hệ thống tư vấn sẽ không phát hiện ra được sự tương đồng từ 2 sản phẩm này.
- Quan điểm người dùng không nhất quán.
 - Một số người dùng không có sự nhất quán quan điểm với bất kỳ nhóm người nào. Do đó hệ thống tư vấn không thể sử dụng thông tin hiện có của các nhóm người dùng trong hệ thống để tư vấn cho họ.
 - Có một số người dùng không có bất cứ một quan điểm chung nào với bất cứ ai, điều này có nghĩa hệ thống tư vấn hoàn toàn không đưa ra được tư vấn cho họ. Trên thực tế thì với những người như vậy, các chuyên gia con người cũng không thể đưa ra bất kỳ tư vấn nào. Do đó, đây là một thất bại của hệ thống tư vấn nhưng là trường hợp thất bại được chấp nhận.
- Ý kiến đánh giá không khách quan: Đây là thách thức rất dễ gặp trong thương mại điện tử. Nhiều người cố tình đưa ra những đánh giá tốt cho sản phẩm của mình và đồng thời đánh giá rất thấp sản phẩm của đối thủ. Đây hoàn toàn là những đánh giá không khách quan. Điều này ảnh hưởng rất lớn đến sự chính xác của hệ thống tư vấn
- Những thách thức khác
 - Nhiều người dùng không muốn thói quen và quan điểm của họ bị đem ra phơi bày cho nhiều người biết.

- Trong một số trường hợp, người dùng đưa ra những đánh giá hoàn toàn không đúng với những gì họ nghĩ. Có rất nhiều nguyên nhân dẫn đến hành động đó, và nó đã ảnh hưởng đến sự chính xác của các tư vấn.

Sự phát triển mạnh mẽ của mạng xã hội với hệ thống cơ sở dữ liệu khổng lồ đã vô tình tạo ra một nhu cầu tất yếu về việc phải làm sao khai thác, tận dụng nguồn dữ liệu quý giá này.

Ontology đã mang đến sự chia sẻ cấu trúc dữ liệu giữa con người và các tác nhân máy tính. Nhờ vậy đã giải quyết vấn đề thiếu thông tin dữ liệu – một thách thức của hệ thống CF.

Hệ cơ sở tri thức được xây dựng trên ontology cho phép khai thác tri thức từ các mối quan hệ tìm ẩn và điều này rất có ý nghĩa trong việc đưa ra các kết quả khuyến nghị đến với người dùng.

Dựa vào mục tiêu đặt ra cho đề tài, nhận thấy hướng nghiên cứu và sử dụng ontology là phù hợp và đem lại hiệu quả tốt nhất.

2.2.3. Quảng cáo trực tuyến:

2.2.3.1. *Quảng cáo trực tuyến là gì?*

Cũng như các loại hình quảng cáo khác, quảng cáo trực tuyến trên mạng nhằm cung cấp thông tin, đẩy nhanh tiến độ giao dịch giữa người mua và người bán. Nhưng quảng cáo trực tuyến khác hẳn quảng cáo trên các phương tiện thông tin truyền thông đại chúng khác, nó giúp người tiêu dùng có thể tương tác với quảng cáo trực tuyến. Khách hàng có thể nhấn vào quảng cáo để lấy thông tin hoặc mua sản phẩm cùng mẫu mã trên logo quảng cáo đó, thậm chí họ còn có thể mua cả sản phẩm từ các quảng cáo online trên Website.

Quảng cáo trực tuyến đã tạo cơ hội cho các nhà quảng cáo nhắm chính xác vào khách hàng của mình, và giúp họ tiến hành quảng cáo theo đúng với sở thích và thị hiếu của người tiêu dùng. Các phương tiện thông tin đại chúng khác cũng có khả năng nhắm chọn, nhưng chỉ có mạng Internet mới có khả năng tuyệt vời như thế.

2.2.3.2. *Những ưu điểm của quảng cáo trực tuyến:*

- Khả năng nhắm chọn: Nhà quảng cáo trực tuyến trên mạng có rất nhiều khả năng nhắm chọn mới. Họ có thể nhắm vào các công ty, các quốc gia hay khu vực địa lý cũng như họ có thể sử dụng cơ sở dữ liệu để làm cơ sở cho tiếp thị trực tiếp. Họ cũng có thể dựa vào sở thích cá nhân và hành vi của người tiêu dùng để nhắm vào đối tượng thích hợp.
- Khả năng theo dõi: Các nhà tiếp thị online trên mạng có thể theo dõi hành vi của người sử dụng đối với nhãn hiệu của họ và tìm hiểu sở thích cũng như mối quan tâm của những khách hàng triển vọng. Ví dụ, một hãng sản xuất xe hơi có thể theo dõi hành vi của người sử dụng qua site của họ và xác định xem có nhiều người quan tâm đến quảng cáo của họ hay không?

Các nhà quảng cáo cũng có thể xác định được hiệu quả của một quảng cáo (thông qua số lần quảng cáo được nhấn, số người mua sản phẩm, và số lần tiến hành quảng cáo,...) nhưng điều này rất khó thực hiện đối với kiểu quảng cáo truyền thống như trên tivi, báo chí và bảng thông báo.

- Tính linh hoạt và khả năng phân phối: Một quảng cáo trực tuyến trên mạng được truyền tải 24/24 giờ một ngày, cả tuần, cả năm. Hơn nữa, chiến dịch quảng cáo có thể được bắt đầu cập nhật hoặc huỷ bỏ bất cứ lúc nào. Nhà quảng cáo có thể theo dõi tiến độ quảng cáo hàng ngày, xem xét hiệu quả quảng cáo ở tuần đầu tiên và có thể thay thế quảng cáo ở tuần thứ hai nếu cần thiết. Điều này khác hẳn kiểu quảng cáo trên báo chí, chỉ có thể thay đổi quảng cáo khi có đợt xuất bản mới, hay quảng cáo tivi với mức chi phí rất cao cho việc thay đổi quảng cáo thường xuyên.
- Tính tương tác: Mục tiêu của nhà quảng cáo online là gắn khách hàng triển vọng với nhãn hiệu hoặc sản phẩm của họ. Điều này có thể thực hiện hiệu quả trên mạng, vì khách hàng có thể tương tác với sản phẩm, kiểm tra sản phẩm và nếu thoả mãn thì có thể

mua. Ví dụ, một quảng cáo cho phần mềm máy tính có thể đưa khách hàng tới nơi trưng bày sản phẩm để lấy thông tin và kiểm tra trực tiếp. Nếu khách hàng thích phần mềm đó, họ có thể mua trực tiếp. Không có loại hình thông tin đại chúng nào lại có thể dẫn khách hàng từ lúc tìm hiểu thông tin đến khi mua sản phẩm mà không gặp trở ngại nào như mạng Internet.

2.2.3.3. Hạn chế của việc quảng cáo:

Thông tin về các sản phẩm được đưa lên từ nhà sản xuất nên thường được nói quá hơn so với thông tin thực sự.

Các bình luận, đánh giá được hỗ trợ trong nhiều mục quảng cáo trực tuyến, nhằm tạo tính tương tác cao giữa nhà sản xuất và người dùng. Tuy nhiên, vì một vài lý do nào đó, các bình luận và đánh giá đó bị làm cho sai lệch hoặc thiếu khách quan theo hướng có lợi cho nhà sản xuất và có tác dụng xấu đến quyết định sử dụng sản phẩm của người tiêu dùng.

2.2.4. Ontology:

2.2.4.1. Khái niệm:

Ontology mô tả các khái niệm cơ bản trong một ứng dụng và xác định mối liên hệ giữa các khái niệm này [3]. Trong bối cảnh của mạng ngữ nghĩa (*Semantic web*) hiện nay, thì ontology đóng một vai trò quan trọng trong việc giúp các xử lý tự động (hay còn được gọi là “*intelligent agents*”) trong việc truy vấn thông tin [18]. Một ontology được xây dựng từ các thành phần:

- Các lớp (*classes*) hoặc đôi khi còn được gọi là các khái niệm (*concepts*).
- Tính chất của mỗi khái niệm mô tả các tính năng và thuộc tính khác nhau của khái niệm (*slots*, đôi khi còn được gọi là *roles* hoặc *properties*).
- Giới hạn trên *slots* (*facets*, đôi khi còn được gọi là giới hạn trên *role*).

Trong phạm vi luận văn này, ontology được thiết kế thông qua công cụ là protégé phiên bản 3.4.7.

2.2.4.2. Tại sao tạo ra một ontology?

Một ontology cung cấp một vốn từ ngữ phổ biến cho các nhà nghiên cứu, và các nhà nghiên cứu này cần chia sẻ thông tin trong một ứng dụng [9]. Một số lý do để tạo ra một ontology là:

- Để chia sẻ những thông tin chung về cấu trúc của thông tin giữa con người và các tác nhân phần mềm.
- Để tái sử dụng các kiến thức miền.
- Để tạo ra các miền giả định rõ ràng.
- Để tách riêng kiến thức miền từ kiến thức vận hành.

Thông thường, một ontology được phát triển để dùng vào việc xác định tập các dữ liệu và cấu trúc dữ liệu cho các chương trình khác sử dụng.

2.2.4.3. Các bước đơn giản để xây dựng một ontology:

Việc phát triển một ontology có thể được thực hiện theo nhiều phương pháp khác nhau, và mỗi phương pháp có thể cho ra những ontology khác nhau nhưng đảm bảo đúng yêu cầu ứng dụng.

Phát triển một ontology là một quá trình lặp đi lặp lại. Đầu tiên là tạo ra một ontology căn bản, sau đó là các quá trình chỉnh sửa và tinh chỉnh lại và cuối cùng hoàn thành ontology bằng việc điền vào các chi tiết.

Trên thực tế, phát triển một ontology bao gồm các bước [9] :

- Xác định các lớp (*class*) trong ontology.
- Sắp xếp các lớp vào cây phả hệ gồm các lớp con (*subclass*) và lớp cha (*supclass*).
- Xác định các slot và mô tả giới hạn giá trị cho một số slot.
- Điền giá trị cho các slot của từng thực thể của lớp.

2.2.4.4. *Xác định một ontology được thiết kế đúng hay không đúng:*

Với mỗi lĩnh vực ứng dụng, chúng ta có thể tạo ra được nhiều ontology khác nhau. Và với bất kỳ một ontology cụ thể nào thì cũng chỉ có một cách để cấu trúc các khái niệm và các mối liên kết lại với nhau. Sau đây là một vài nguyên tắc đơn giản có thể giúp trong việc đưa ra các quyết định thiết kế trong nhiều trường hợp [9] :

- Với một ứng dụng, không phải chỉ có một cách để phát triển ontology, mà còn có nhiều cách khả thi khác để lựa chọn. Điều quan trọng để lựa chọn một phương pháp tốt nhất là dựa vào ứng dụng cụ thể nào sẽ khai thác ontology và khả năng mở rộng mà khi thiết kế chúng ta đã dự đoán trong đầu.
- Việc phát triển ontology cần một quá trình lặp đi lặp lại, không ngừng bổ sung và tinh chỉnh.
- Các khái niệm trong ontology nên gần gũi với các đối tượng (vật lý hoặc luận lý) và các mối liên kết thì trong phạm vi chúng ta quan tâm. Trong đó, hầu hết các đối tượng được biểu diễn là một danh từ và các mối liên kết là một động từ.

2.2.4.5. *Bắt đầu từ đâu để tạo một ontology?*

Đầu tiên chúng ta cần xác định, ứng dụng nào sẽ khai thác ontology mà chúng ta đang phát triển. Và ontology này sẽ phát triển mang nội dung chi tiết hay tổng quát [9] .

Sau đó, cần xác định rõ lựa chọn phát triển nào là khả thi nhất cho công việc đã dự kiến, lựa chọn nào có trực quan hơn mở rộng hơn, và duy trì lâu nhất.

Luôn nhớ rằng, một ontology là một mô hình mô phỏng lại một phạm vi trong thế giới thực, nên mỗi khái niệm trong ontology phải phản ánh đúng các đối tượng trong thế giới thực.

Sau khi đã tạo ra được phiên bản đầu tiên của ontology, chúng ta có thể đánh giá và gỡ lỗi bằng cách sử dụng nó trong các ứng dụng, hoặc sử

dụng phương pháp problem-solving, hoặc bàn bạc thảo luận với các chuyên gia.

2.2.5. **Ontology Web Language (OWL):**

Ontology Web Language (OWL) là ngôn ngữ đánh dấu được sử dụng để xuất bản và chia sẻ dữ liệu sử dụng các ontology trên Internet. OWL là một bộ từ vựng mở rộng của khung mô tả tài nguyên (RDF) và được kế thừa từ ngôn ngữ DAML+OIL Web ontology – một dự án được hỗ trợ bởi W3C.

OWL có thể khai báo các lớp, phân loại lớp theo dạng cây phả hệ (lớp con và lớp cha), khai báo các thuộc tính (*property*) và sắp xếp chúng dưới dạng cây phả hệ “*subproperty*”, và cung cấp miền (*domain*) và phạm vi (*range*) cho các thuộc tính đó. Các miền của thuộc tính OWL là các lớp của OWL và phạm vi của thuộc tính có thể là lớp OWL hoặc là một kiểu dữ liệu chuỗi hoặc số.

OWL có thể diễn tả đối tượng nào đó (có thể thể gọi là một “*individual*”) thuộc về lớp nào đó và giá trị thuộc tính nào thì thuộc về cá nhân (*individual*) nào đó.

RDF (*Resource Description Framework*) là cơ chế cho phép mô tả dữ liệu về dữ liệu (*metadata*). RDF coi các đối tượng trên Web (trang Web, đoạn văn, người, các đối tượng khác.v.v.) là các tài nguyên. Mỗi tài nguyên được mô tả bởi bộ ba đối tượng - thuộc tính – giá trị [1]. Ví dụ, mệnh đề “Phương là tác giả bài báo tại trang Web nào đó” sẽ được mô tả bởi bộ ba: [http:// www...](http://www...), tác giả, “Phương”. RDF Schema (RDFS) là một biến thể đơn giản sử dụng cơ chế RDF. RDFS cho phép mô tả các thuộc tính đặc thù cho ứng dụng, đồng thời định nghĩa lớp các đối tượng có cùng thuộc tính đó. Việc định nghĩa lớp đối tượng với thuộc tính và quan hệ rất cần thiết cho việc xây dựng ontology.

DAML + OIL. RDF và RDF Schema chỉ cho phép biểu diễn ngữ nghĩa ở mức độ đơn giản [19] [20]. Để biểu diễn ngữ nghĩa bao gồm nhiều đối tượng có quan hệ logic phức tạp với nhau cần các phương tiện biểu diễn mạnh hơn. DAML (*Darpa Agent Markup Language*) và OIL (*Ontology Interface Layer*) là các phương tiện như vậy. DAML+OIL là một mở rộng của RDFS. Trong DAML+OIL, ngữ nghĩa được mô tả thông qua logic mô tả (*descriptive logic*)

cho phép sử dụng logic bool khi mô tả quan hệ giữa các đối tượng và có nhiều kiểu quan hệ cơ sở hơn so với RDFS.

Các thông tin OWL có thể dễ dàng trao đổi giữa các kiểu hệ thống máy tính khác nhau, sử dụng các hệ điều hành và các ngôn ngữ ứng dụng khác nhau.

OWL gồm các phiên bản:

- OWL Lite: hỗ trợ cho những người dùng chủ yếu cần sự phân lớp theo thứ bậc và các ràng buộc đơn giản. Ví dụ: Trong khi nó hỗ trợ các ràng buộc về tập hợp, nó chỉ cho phép tập hợp giá trị của 0 hay 1. Điều này cho phép cung cấp các công cụ hỗ trợ OWL Lite dễ dàng hơn so với các bản khác.
- OWL DL (*OWL Description Logic*): hỗ trợ cho những người dùng cần sự diễn cảm tối đa trong khi cần duy trì tính tính toán toàn vẹn (tất cả các kết luận phải được đảm bảo để tính toán) và tính quyết định (tất cả các tính toán sẽ kết thúc trong khoảng thời gian hạn chế). OWL DL bao gồm tất cả các cấu trúc của ngôn ngữ OWL, nhưng chúng chỉ có thể được sử dụng với những hạn chế nào đó (Ví dụ: Trong khi một lớp có thể là một lớp con của rất nhiều lớp, một lớp không thể là một thể hiện của một lớp khác).
- OWL Full muốn đề cập tới những người dùng cần sự diễn cảm tối đa và sự tự do của RDF mà không cần đảm bảo sự tính toán của các biểu thức. Ví dụ, trong OWL Full, một lớp có thể được xem xét đồng thời như là một tập của các cá thể và như là một cá thể trong chính bản thân nó. OWL Full cho phép một ontology gia cố thêm ý nghĩa của các từ vựng được định nghĩa trước (RDF hoặc OWL).

Mối liên hệ giữa các ngôn ngữ con của OWL:

- Mọi ontology hợp lệ dựa trên OWL Lite đều là ontology hợp lệ trên OWL DL.
- Mọi ontology hợp lệ dựa trên OWL DL đều là ontology hợp lệ trên OWL Full.

- Mọi kết luận hợp lệ dựa trên OWL Lite đều là kết luận hợp lệ trên OWL DL.
- Mọi kết luận hợp lệ dựa trên OWL DL đều là kết luận hợp lệ trên OWL Full.

Và trong phạm vi luận văn của nhóm, ontology được thiết kế trên ngôn ngữ OWL phiên bản OWL DL.

2.2.6. SPARQL:

2.2.6.1. Định nghĩa:

SPARQL là một ngôn ngữ truy vấn dữ liệu RDF. **SPARQL** được tạo ra là một chuẩn để truy cập dữ liệu RDF được phát triển bởi nhóm *RDF Data Access Working Group* (DAWG) của World Wide Web Consortium, và được coi là một trong những công nghệ chủ chốt của *semantic web* [4] . SPARQL cho phép chúng ta:

- Lấy được giá trị từ dữ liệu có cấu trúc và cả dữ liệu bán cấu trúc.
- Khám phá dữ liệu bằng các mối liên kết chưa được biết đến.
- Biểu diễn sự liên kết qua lại phức tạp của các cơ sở dữ liệu riêng biệt bằng một truy vấn đơn giản.
- Chuyển đổi dữ liệu RDF từ một từ ngữ này sang từ ngữ khác.

Sơ lược về dữ liệu RDF:

- RDF là một mô hình dữ liệu đồ thị gồm có bộ ba *subject*, *predicate* và *object*.
- Tài nguyên có thể được đại diện bởi URIs, và có thể viết tắt bằng tên tiền tố.
- *Object* là các đối tượng có kiểu dữ liệu: *String*, *Integer*, *boolean*
...

Ontology được xây dựng trên OWL, OWL được phát triển từ XML và RDF, nên ta sẽ sử dụng Sparql để truy vấn, khai thác trên ontology.

2.2.6.2. *Cấu trúc của SPARQL:*

Cấu trúc một lệnh truy vấn SPARQL bao gồm (theo trình tự từ trên xuống) [4] :

- Định nghĩa tiền tố: được hiểu như việc tạo từ viết tắt cho URIs.
- Định nghĩa tập dữ liệu: xác định mô hình đồ thị nào của RDF sẽ được truy vấn.
- Xác định cụm kết quả: xác định các thông tin sẽ trả về từ lệnh truy vấn.
- Xác định mẫu truy vấn: xác định cái để truy vấn từ tập dữ liệu cơ bản.
- Bộ định nghĩa truy vấn: chứa thông tin cho biết việc sắp xếp (*order*), giới hạn (*limit*).. kết quả truy vấn.

```
# prefix declarations
PREFIX foo: <http://example.com/resources/>
...
# dataset definition
FROM ...
# result clause
SELECT ...
# query pattern
WHERE {
    ...
}
# query modifiers
ORDER BY ...
```

2.2.6.3. *Các kiểu truy vấn:*

SPARQL đặc tả 4 loại truy vấn khác nhau cho các mục đích khác nhau [10] .

- Truy vấn **SELECT**: Sử dụng để trích xuất các giá trị thô từ SPARQL endpoint, các kết quả được trả về trong một định dạng bảng. Ví dụ:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE {
    ?person foaf:name ?name .
}
```

- Truy vấn **CONSTRUCT**: Sử dụng để trích xuất thông tin từ SPARQL endpoint và chuyển kết quả thành dạng RDF hợp lệ. Ví dụ:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
CONSTRUCT { <http://example.org/person#Alice>
    vcard:FN ?name }
WHERE { ?x foaf:name ?name }
```

- Truy vấn **ASK**: Sử dụng để cung cấp các kết quả dạng True/False đơn giản cho các truy vấn trên SPARQL endpoint. Ví dụ:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
ASK { ?x foaf:name "Alice" }
```

- Truy vấn **DESCRIBE**: Sử dụng để trích xuất một đồ thị RDF từ SPARQL endpoint, các nội dung đó được đưa tới endpoint để quyết định dựa trên những thông tin có ích. Ví dụ:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
DESCRIBE ?x
WHERE { ?x foaf:mbox <mailto:alice@org> }
```

2.2.7. Friend Of A Friend (FOAF):

Là một bộ từ vựng chuẩn của RDF để mô tả một người và các mối liên kết của nó [4] .

FOAF được xác định bởi *namespace* 'http://xmlns.com/foaf/0.1/' [2] , do đó việc mở rộng và sửa đổi của FOAF phụ thuộc vào việc mở rộng và chỉnh sửa của *namespace* đó.

Một ví dụ về thông tin FOAF của người dùng có thể xem tại <http://dig.csail.mit.edu/2008/webdav/timbl/foaf.rdf>

2.2.8. Protégé API:

Protégé - OWL API là một thư viện Java mã nguồn mở hỗ trợ cho OWL và RDF [6] . API này gồm các lớp và các phương thức nhằm hỗ trợ:

- Đọc, chỉnh sửa, lưu các tập tin OWL.
- Truy vấn và thao tác các mô hình dữ liệu OWL.
- Biểu diễn các suy luận.

Hơn thế nữa, Protégé - OWL API được tối ưu hóa cho các giao diện đồ họa người dùng.

Trong phạm vi luận văn, Protégé - OWL API được sử dụng có phiên bản 3.4.7 vì có tính ổn định hơn so với các phiên bản khác.

2.2.9. Facebook API:

Hiện tại Facebook hỗ trợ rất nhiều các API cho người dùng trong nhiều lĩnh vực như website, ứng dụng trên Facebook, di động,... dưới nhiều hình thức như platform, hoặc SDK... Dưới đây sẽ trình bày về các Api quan trọng được sử dụng trong ứng dụng.

2.2.9.1. Facebook platform:

Facebook platform ra đời nhằm giúp xã hội hóa website của người dùng hơn [5] .

- *Social plugin*: các plugin này là những tính năng xã hội (social features) có thể nhúng vào website của người dùng bằng những

dòng HTML đơn giản. Những plugin này được lưu trữ trên Facebook, nên được cá nhân hóa cho từng người dùng đang đăng nhập, thậm chí đó là người dùng lần đầu tiên sử dụng website có tích hợp chúng.

Một số social plugin thông dụng:

- Nút “Like”: cho phép người dùng chia sẻ website của họ với bạn bè chỉ cần thông qua một cái nhấn chuột. Chúng ta có thể thêm nút “Like” vào website chỉ với một thẻ iframe như sau:

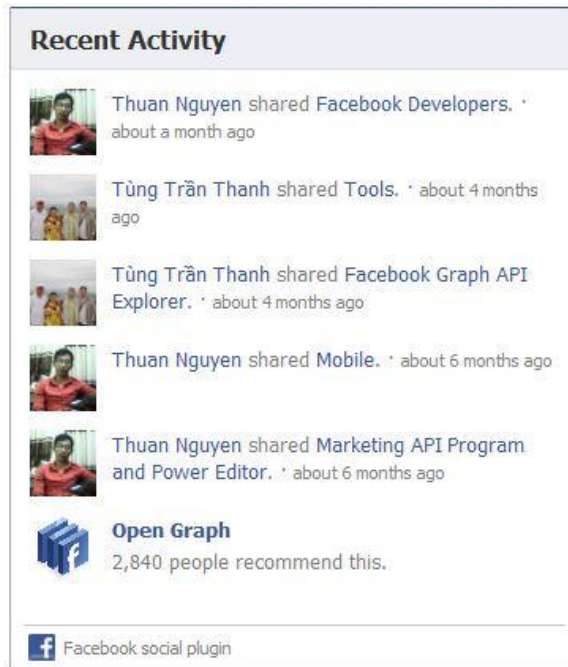
```
<iframe
src="http://www.facebook.com/plugins/like.php?
href=YOUR_URL"
scrolling="no" frameborder="0"
style="border:none;width:450px;
height:80px">
</iframe>
```

Sau khi thêm vào, ta có thể thấy nút “Like” hiện lên như hình bên dưới:



Hình 1: nút Like

- Activity feed plugin: dùng để hiển thị danh sách các sự kiện người dùng nhấn “Like” hoặc bình luận (comment) trên website của mình.



Hình 2: Recent Activity plugin

- **Xác thực (Authentication)** : Facebook giúp chúng ta đơn giản hóa việc đăng ký và đăng nhập bằng việc sử dụng Facebook như một hệ thống đăng nhập. Điều đó có nghĩa, Facebook sẽ chia sẻ tài khoản Facebook của người dùng với website của chúng ta, bất cứ khi nào người dùng đang đăng nhập vào Facebook thì khi đó họ tự động đăng nhập vào website của chúng ta. Để sử dụng chức năng này, chúng ta cần đăng ký website với Facebook để có được App Id (hoặc appid). Một ví dụ về cách sử dụng chức năng xác thực thông qua Facebook:

```
<html>
<head>
<title>My Facebook Login Page</title>
</head>
<body>
<div id="fb-root"></div>
<script>
    window.fbAsyncInit = function() {
        FB.init({
            appId      : 'YOUR_APP_ID',
            status      : true,
            cookie      : true,
```

```

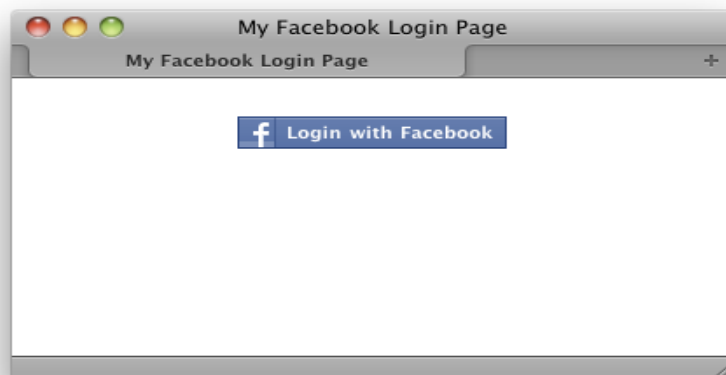
        xfbml      : true,
        oauth      : true,
    });
};
(function(d) {
    var js, id = 'facebook-jssdk'; if
(d.getElementById(id)) {return;}
    js = d.createElement('script'); js.id = id;
    js.async = true;
    js.src =
    '//connect.facebook.net/en_US/all.js';

    d.getElementsByTagName('head')[0].appendChild(js);
    }(document));
</script>
</body>
</html>

```

Sau khi tham chiếu javascript sdk vào trang ứng dụng của mình, bước kế tiếp đơn giản chỉ cần thêm nút ‘Login’ bằng thẻ div vào ứng dụng.

```
<div class="fb-login-button">Login with Facebook</div>
```



Hình 3: nút login

2.2.9.2. Graph Api:

Mỗi đối tượng trong soicial graph đều có một ID duy nhất. Chúng ta có thể truy cập thuộc tính của đối tượng bằng cách gửi một yêu cầu với cú pháp: <https://graph.facebook.com/ID> [5] .

Ví dụ, trang chính thức của Facebook platform có id là 9292868552, vì vậy bạn có thể lấy được nội dung của đối tượng <https://graph.facebook.com/19292868552>:

```
{
  "name": "Facebook Platform",
  "type": "page",
  "website": "http://developers.facebook.com",
  "username": "platform",
  "founded": "May 2007",
  "company_overview": "Facebook Platform enables anyone
to build...",
  "mission": "To make the web more open and social.",
  "products": "Facebook Application Programming
Interface (API)...",
  "likes": 449921,
  "id": 19292868552,
  "category": "Technology"
}
```

Tất cả đối tượng có thể được lấy thông tin theo cách tương tự như trên và kết quả trả về là một đối tượng JSON:

- Người dùng (User):

<https://graph.facebook.com/btaylor> (người dùng Bret Taylor).

- Trang (pages):

<https://graph.facebook.com/cocacola> (Coca-Cola page).

- Sự kiện (Events):

<https://graph.facebook.com/251906384206> (Facebook Developer Garage Austin).

- Nhóm (Groups):

<https://graph.facebook.com/195466193802264> (nhóm phát triển của Facebook).

- Ứng dụng (Applications):

<https://graph.facebook.com/2439131959> (the Graffiti app).

- Trạng thái tin nhắn:

<https://graph.facebook.com/367501354973> (một trạng thái tin nhắn từ Bret).

- Hình ảnh (Photos):

<https://graph.facebook.com/98423808305> (hình ảnh từ Coca-Cola page).

- Album hình ảnh (Photo albums):

<https://graph.facebook.com/99394368305> (các hình ảnh trên tường của Coca-Cola).

- Hình đại diện (Profile pictures):

<http://graph.facebook.com/btaylor/picture> (hình đại diện của người dùng Bret Taylor).

- Videos:

<https://graph.facebook.com/817129783203> (một video hướng dẫn về Graph API).

- Ghi chú (Notes):

<https://graph.facebook.com/122788341354> (ghi chú của Facebook thông báo về iPhone 3.0).

- Ghi dấu nơi đến (Checkins):

<https://graph.facebook.com/414866888308> (ghi dấu đang ở pizzeria).

Các đối tượng trong Facebook được kết nối với nhau thông qua các mối liên hệ, như Bret Taylor là một người hâm mộ(fan) của trang Coca Cola và Bret Taylor và Arjun Banker là bạn của nhau. Chúng ta có thể lấy

được thông tin về mối liên hệ giữa các đối tượng theo cấu trúc `https://graph.facebook.com/ID/CONNECTION_TYPE`. Các kết nối hỗ trợ giữa các đối tượng gồm có:

- Bạn của bạn (Friends):

`https://graph.facebook.com/me/friends?access_token=...`

- Tin tức hoạt động (News feed):

`https://graph.facebook.com/me/home?access_token=...`

- Hoạt động của đối tượng (Wall):

`https://graph.facebook.com/me/feed?access_token=...`

- Thích (Likes):

`https://graph.facebook.com/me/likes?access_token=...`

- Phim ảnh (Movies):

`https://graph.facebook.com/me/movies?access_token=...`

- Âm nhạc (Music):

`https://graph.facebook.com/me/music?access_token=...`

- Sách (Book):

`https://graph.facebook.com/me/books?access_token=...`

- Ghi chú (Notes):

`https://graph.facebook.com/me/notes?access_token=...`

- Sự cho phép (Permissions):

`https://graph.facebook.com/me/permissions?access_token=...`

- Đánh dấu ảnh (Photo Tags):

`https://graph.facebook.com/me/photos?access_token=...`

- Album hình ảnh (Photo Albums):

`https://graph.facebook.com/me/albums?access_token=...`

- Đánh dấu video (Video Tags):

`https://graph.facebook.com/me/videos?access_token=...`

- Video Uploads:

`https://graph.facebook.com/me/videos/uploaded?access_token=...`

- Sự kiện (Events):

`https://graph.facebook.com/me/events?access_token=...`

- Nhóm (Groups):

`https://graph.facebook.com/me/groups?access_token=...`

- Đánh dấu nơi đến (Checkins):

`https://graph.facebook.com/me/checkins?access_token=...`

Ủy quyền (Authorization):

Graph Api cho phép ta dễ dàng lấy được những thông tin công khai của một đối tượng như “first_name”, “last_name”, “username”,...

Nhưng để lấy được nhiều thông tin khác thì cần phải có sự cho phép của đối tượng. Chúng ta cần “access token” từ người dùng Facebook. Sau khi có được “access token”, chúng ta có thể biểu diễn yêu cầu ủy quyền thay thế người dùng đó bằng cách thêm mã truy cập vào lệnh yêu cầu của chúng ta như cú pháp:

`https://graph.facebook.com/220439?access_token=...`

2.2.9.3. *PHP SDK:*

PHP SDK cung cấp cho lập trình viên rất nhiều những hàm hướng máy chủ (server-side) nhằm hỗ trợ cho việc gọi các Api của Facebook. PHP SDK bao gồm tất cả các tính năng của Graph Api, FQL (tìm hiểu thêm tại <https://developers.facebook.com/docs/reference/fql/>) và Deprecated REST API (tìm hiểu thêm tại <https://developers.facebook.com/docs/reference/rest/>). Chúng ta có thể tải gói thư viện PHP SDK tại <https://github.com/facebook/php-sdk>.

PHP SDK thường được sử dụng để thực hiện các hoạt động như một quản trị viên của ứng dụng, nhưng cũng có thể được sử dụng để thực hiện các hoạt động thay thế cho người sử dụng phiên hiện tại. Loại bỏ sự cần thiết của việc quản lý các mã (thẻ) truy cập, PHP SDK đơn giản hóa quá trình xác thực và ủy quyền cho người dùng trong các ứng dụng của bạn.

Sau khi chúng ta đăng ký ứng dụng với Facebook, Facebook sẽ cung cấp cho ứng dụng của chúng ta App Id và App secret. App Id và App secret là 2 tham số bắt buộc để có thể bắt đầu sử dụng SDK.

PHP SDK được sử dụng kết hợp cùng với JavaScript SDK để có được sự quản lý phiên liên mạch từ hướng máy chủ (server-side) đến hướng máy khách (client-side).

2.3. Yêu cầu chức năng:

Để thực hiện được các mục tiêu đã đề ra của đề tài, ứng dụng minh họa cần được hỗ trợ các tính năng chính sau:

- Cho phép người dùng giới thiệu các quán cà phê và các loại dịch vụ của quán, với quan tâm chính là các loại cà phê (*Coffee Shop & Coffee Management Functions*).
- Cho phép người dùng đánh giá và chia sẻ cảm nhận về các quán cà phê (*Comment and Rating Functions*).
- Chức năng tìm kiếm (*Searching Functions*).
- Giới thiệu các quán cà phê phù hợp với người dùng (*Coffee Shop Recommendation Functions*).
- Quản trị (*Administrative Functions*): chức năng này được chia thành ba chức năng nhỏ hơn là:
 - Quản lý thông tin người dùng (*Profile Management Functions*).
 - Quản trị hệ thống (*System Management Functions*).
- Quản lý tri thức (*Knowledge Management Functions*).

Các tính năng này sẽ được mô tả chi tiết trong chương 3 và chương 4.

2.4. Yêu cầu phi chức năng:

2.4.1. Khả năng tái sử dụng và mở rộng:

- Ứng dụng phải đảm bảo có thể được tái sử dụng cho các sản phẩm khác như quảng cáo thức ăn, quần áo, xe máy, hoặc cả địa điểm du lịch.
- Ứng dụng phải đảm bảo được xây dựng trên nền tảng tốt, việc mở rộng nội dung sẽ không ảnh hưởng đến các chức năng hiện có.

2.4.2. Độ tin cậy:

- Khi người dùng chấp nhận chia sẻ hồ sơ cá nhân của họ cho ứng dụng dựa trên chính sách của Facebook, ứng dụng đảm bảo các thông tin đó được dùng vào đúng mục đích của nội bộ chương trình chứ không chia sẻ thông tin này cho bất kỳ một bên thứ ba nào khác.
- Các thông tin về sản phẩm đến với người dùng được lập trình sao cho độ khách quan và đáng tin cậy nhất.

2.4.3. Tốc độ:

- Hạn chế lớn nhất của ontology là làm chậm tốc độ xử lý, để bù đắp lại điều này, nhóm lập trình phải đảm bảo các đoạn mã được viết với những thuật toán hay, đảm bảo về tốc độ xử lý.

2.4.4. Khả năng hỗ trợ người dùng:

- Để người dùng dễ dàng thao tác các chức năng của ứng dụng, ứng dụng phải có giao diện dễ hiểu, bố cục hợp lý.
- Xây dựng các tài liệu hướng dẫn.
- Có các mục hỏi và đáp. Và địa chỉ liên lạc khi có vấn đề thắc mắc.

2.5. Kết chương:

Qua chương 2 ta đã có được cái nhìn tổng quát về hiện trạng các ứng dụng, sản phẩm có điểm tương đồng với ứng dụng của nhóm. Ngoài ra cũng hiểu rõ các kiến thức nền tảng mà nhóm cần chuẩn bị cũng như những yêu cầu đã đặt ra cho đề tài. Trong chương kế tiếp (chương 3) sẽ trình bày chi tiết về nội dung xây dựng một hệ cơ sở tri thức.

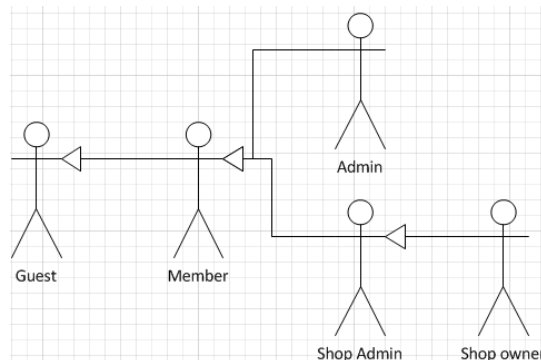
Chương 3: XÂY DỰNG CƠ SỞ TRI THỨC

3.1. Lược đồ use case:

3.1.1. Danh sách các tác nhân:

Các đối tượng giao tiếp với hệ thống gồm:

- Thành viên quản trị hệ thống (*Admin*): là những thành viên có toàn quyền sử dụng mọi chức năng thông qua giao diện của ứng dụng.
- Thành viên (*Member*): là những thành viên thông thường của hệ thống (đã đăng nhập vào hệ thống). Người dùng được sử dụng hầu hết các chức năng của hệ thống.
- Chủ quán (*Shop owner*): là các thành viên làm chủ ít nhất một quán cà phê. Những thành viên này sẽ được các thành viên quản trị xác minh trong thực tế. Chủ quán có quyền chỉnh sửa thông tin quán và danh sách các dịch vụ của quán. Chủ quán còn có quyền chỉ định những thành viên khác có quyền quản lý quán hay từ chối quyền này.
- Thành viên quản lý quán (*Shop admin*): là những thành viên được chỉ định để quản lý thông tin của một quán nào đó, những dịch vụ mà quán phục vụ. Khác với chủ quán, thành viên quản lý không có quyền chỉ định những thành viên khác để quản lý quán.
- Khách (*Guest*): là những người dùng chưa đăng nhập vào hệ thống. Các chức năng dành cho khách sẽ rất hạn chế, chủ yếu chỉ là xem các bài viết được chia sẻ cho cả cộng đồng (*public*).



Hình 4: Các tác nhân

Các chức năng cụ thể và quyền sử dụng của các đối tượng sẽ được nêu trong phần tiếp theo.

3.1.2. Danh sách các use case:

Các *use case* sẽ được trình bày theo từng nhóm các chức năng chính (xem phần [2.1](#)).

STT	Chức năng	Tên use case
1	Giới thiệu các quán cà phê và các loại cà phê (<i>Coffee Shop and Coffee type Management Functions</i>).	
1.1	Đăng thông tin giới thiệu về một quán cà phê.	<i>Share coffee shop</i>
1.2	Cập nhật thông tin của một quán cà phê.	<i>Update shop info</i>
1.3	Chia sẻ hình ảnh về một quán cà phê hay một dịch vụ.	<i>Share photo</i>
1.4	Thêm một dịch vụ vào danh sách các dịch vụ do quán phục vụ.	<i>Share service</i>
1.5	Chia sẻ một quán lên Facebook.	<i>Post shop to Wall</i>
1.6	Xác nhận là khách hàng của một quán cà phê.	<i>Is customer</i>
1.7	Xác nhận đang hay đã từng làm việc tại một quán cà phê.	<i>Is employee</i>
1.8	Xác nhận là chủ quán cà phê	<i>Is owner</i>
1.9	Xác nhận quyền chỉnh sửa thông tin quán cà phê	<i>Authorize admin permission</i>
1.10	Xem danh sách các dịch vụ của một quán cà phê.	<i>Viewshop's services</i>
1.11	Xem danh sách những khách hàng của một quán cà phê.	<i>View shop's customers</i>
1.12	Vẽ sơ đồ mối liên hệ giữa một người dùng và một quán cà phê.	<i>Build user-shop relationship graph</i>
1.13	Xem chi tiết một quán cà phê.	<i>View shop</i>
1.14	Xem danh sách các quán theo thứ tự được chia sẻ gần nhất.	<i>List shops</i>
1.15	Xem mối liên hệ giữa người dùng và quán	<i>User-shop relationship</i>
1.16	Xác nhận dịch vụ của quán	<i>Review shop's services</i>
1.17	Quản lý danh sách các dịch vụ của quán.	<i>Manage shop's services</i>

Bảng 1: các use case về quán và dịch vụ

STT	Chức năng	Tên use case
-----	-----------	--------------

2	Đánh giá và chia sẻ cảm nhận về các quán cà phê hay các loại cà phê (<i>Comment and Rating Functions</i>).	
2.1	Cho điểm một quán cà phê, một dịch vụ do quán cung cấp.	<i>Rate</i>
2.2	Bình luận.	<i>Comment</i>
2.3	Thích một quán hay một loại cà phê	<i>Like</i>
2.4	Xem danh sách các lời bình cho một quán cà phê.	<i>Get comments</i>
2.5	Xem điểm đánh giá một quán cà phê.	<i>Get ratings</i>
2.6	Xem danh sách những khách hàng thích một quán cà phê.	<i>Get likes</i>
2.7	Hết thích một quán hay một dịch vụ.	<i>Unlike</i>

Bảng 2: Các use case đánh giá và chia sẻ

STT	Chức năng	Tên use case
3	Tìm kiếm (<i>Searching Functions</i>).	
3.1	Tìm kiếm	<i>Search</i>
3.2	Tìm kiếm người dùng	<i>Search users</i>
3.3	Tìm kiếm quán cà phê	<i>Search shops</i>
3.4	Tìm kiếm dịch vụ	<i>Search services</i>
3.5	Gợi ý kết quả tìm kiếm	<i>Suggest</i>

Bảng 3: các use case tìm kiếm

STT	Chức năng	Tên use case
4	Giới thiệu các quán cà phê phù hợp với người dùng (<i>Coffee Shop Recommendation Functions</i>).	
4.1	Giới thiệu các quán nên đến.	<i>Recommend shops</i>

Bảng 4: use case tư vấn

STT	Chức năng	Tên use case
5	Quản lý thông tin người dùng (<i>Profile Management Functions</i>).	
5.1	Đăng nhập vào ứng dụng.	<i>Log in</i>
5.2	Đăng xuất.	<i>Log out</i>
5.3	Cập nhật thông tin tài khoản.	<i>Update profile info</i>
5.4	Xem danh sách các quán cà phê mà người dùng có quyền chỉnh sửa.	<i>Manage shops</i>
5.5	Xem danh sách các quán cà phê mà người dùng thích.	<i>Manage liked shops</i>
5.6	Xem danh sách các dịch vụ mà người dùng thích.	<i>Manage liked services</i>
5.7	Xem danh sách các quán mà người dùng đã đánh giá.	<i>See rated shops</i>
5.8	Xem danh sách các hoạt động gần đây của người dùng.	<i>User's feeds</i>
5.9	Xem mạng lưới mối quan hệ của một	<i>View network</i>

	người dùng.	
5.10	Xem các hoạt động gần đây của hệ thống.	<i>Activity feeds</i>
5.11	Xem danh sách các quán mà người dùng đã đến	<i>Visited shops</i>
5.12	Xem danh sách các quán mà người dùng có thể đã đến	<i>Possibly-visited shops</i>

Bảng 5: các use case về hồ sơ người dùng

STT	Chức năng	Tên use case
6	Quản trị hệ thống (<i>System Management Functions</i>).	
6.1	Quản lý danh sách người dùng.	<i>Users management</i>
6.2	Quản lý danh sách các quán cà phê.	<i>Shops management</i>
6.3	Quản lý danh sách các dịch vụ.	<i>Services management</i>
6.4	Quản lý các thông số hệ thống.	<i>Variables management</i>

Bảng 6: các use case dành cho các thành viên quản trị

Sau đây là quyền sử dụng của từng đối tượng với từng chức năng:

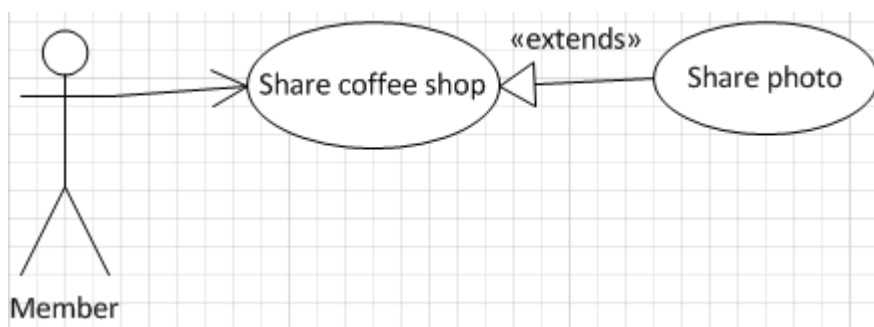
STT của use case	Quản trị viên	Thành viên	Chủ quán	Thành viên quản lý quán	Khách
1.1	x	x	x	x	
1.2	x		x	x	
1.3	x		x	x	
1.4	x	x	x	x	
1.5	x	x	x	x	
1.6	x	x	x	x	
1.7	x	x	x	x	
1.8	x		x		
1.9	x		x		
1.10	x	x	x	x	x
1.11	x	x	x		x
1.12	x	x	x	x	x
1.13	x	x	x	x	x
1.14	x	x	x	x	x
1.15	x	x	x	x	
1.16	x		x	x	
1.17	x		x	x	
2.1	x	x	x	x	
2.2	x	x	x	x	
2.3	x	x	x	x	
2.4	x	x	x	x	
2.5	x	x	x	x	x
2.6	x	x	x	x	x

2.7	x	x	x	x	
3.1	x	x	x	x	x
3.2	x	x	x	x	x
3.3	x	x	x	x	x
3.4	x	x	x	x	x
3.5	x	x	x	x	x
4.1	x	x	x	x	
5.1	x				x
5.2	x	x	x	x	
5.3	x	x	x	x	
5.4	x	x	x	x	
5.5	x	x	x	x	x
5.6	x	x	x	x	x
5.7	x	x	x	x	x
5.8	x	x	x	x	x
5.9	x	x	x	x	
5.10	x	x	x	x	x
5.11	x	x	x	x	
5.12	x	x	x	x	
6.1	x				
6.2	x				
6.3	x				
6.4	x				

Bảng 7: Quyền sử dụng của từng use case với từng tác nhân

3.2. Đặc tả các use case:

3.2.1. Giới thiệu một quán cà phê:



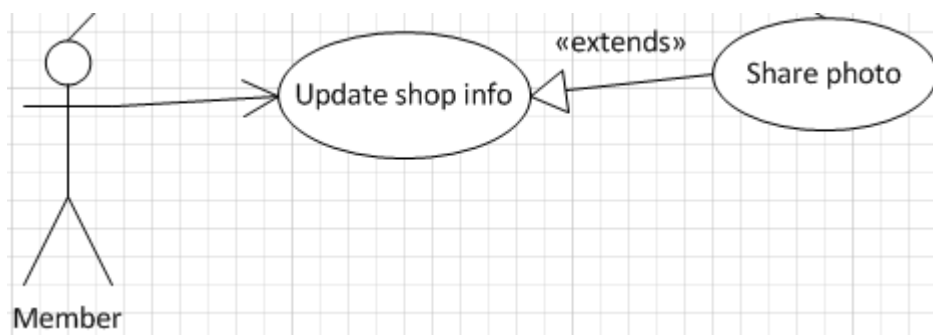
Hình 5: use case Giới thiệu quán cà phê

Use case	Share coffee shop
Mô tả	Use case này cho phép người dùng chia sẻ những quán cà phê mà họ đã từng đến.
Tác nhân	Thành viên

Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng điền các thông tin về quán. Các thông tin bắt buộc là tên quán, mô tả và địa chỉ của quán. - Người dùng bấm nút “Chia sẻ” để hoàn tất quá trình.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	<ul style="list-style-type: none"> - Nếu tên quán đã tồn tại, người dùng sẽ nhận được một thông báo. Người dùng phải chọn một tên quán khác để tiếp tục quá trình chia sẻ. - Nếu quá trình chia sẻ thành công, người dùng sẽ được đưa tới trang chia sẻ các dịch vụ của quán (use case 3.2.14). - Người dùng sẽ được trở thành một thành viên quản lý quán (<i>shop admin</i>). Cho tới khi quán có chủ (<i>shop owner</i>) thì người chủ này có quyền tiếp tục hay ngừng quyền quản lý quán của người dùng đã chia sẻ quán.
Điểm mở rộng	Người dùng có thể sử dụng tính năng chia sẻ hình ảnh (use case 3.2.1) để chia sẻ hình ảnh đại diện cho quán.

Bảng 8: mô tả use case Giới thiệu quán cà phê

3.2.2. Cập nhật thông tin một quán cà phê:



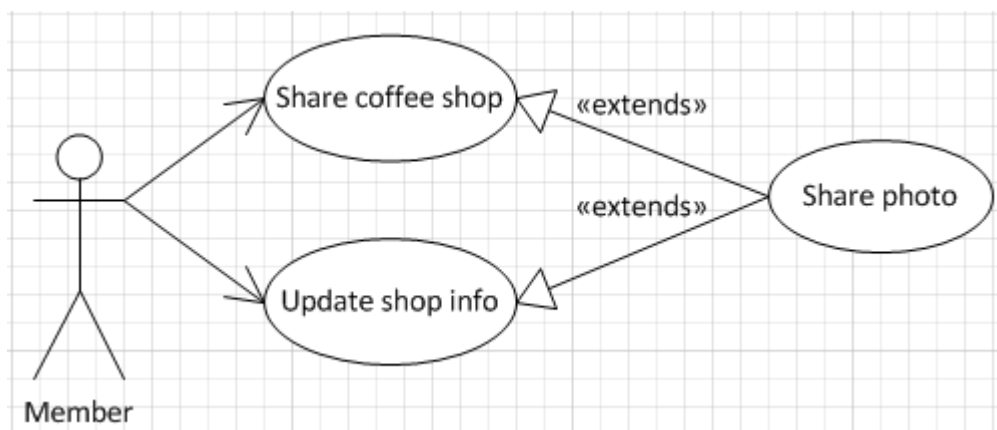
Hình 6: use case Cập nhật thông tin một quán cà phê

Use case	Update coffee shop
Mô tả	Use case này cho phép người dùng chia sẻ những quán cà phê mà họ đã từng đến.
Tác nhân	Quản trị viên (<i>admin</i>), chủ quán (<i>shop owner</i>), thành viên

	quản lý quán (<i>shop admin</i>).
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng chỉnh sửa các thông tin theo ý muốn. - Người dùng có thể sửa tên quán, tuy nhiên, tên mới phải chưa tồn tại trong CSDL. - Người dùng bấm “Lưu” để hoàn tất quá trình.
Các điều kiện đặc biệt	Chỉ có những thành viên quản trị và những thành viên được quyền mới có thể chỉnh sửa thông tin.
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng vào trang liệt kê những quán mà người dùng có quyền chỉnh sửa. - Người dùng vào trang xem chi tiết một quán để chỉnh sửa.
Hậu điều kiện	<ul style="list-style-type: none"> - Nếu quá trình chỉnh sửa có một vài lỗi, người dùng sẽ nhận được thông báo của hệ thống. Người dùng sửa các lỗi này để hoàn thành quá trình. - Nếu quá trình thành công, người dùng sẽ được đưa tới trang xem chi tiết quán.
Điểm mở rộng	Người dùng có thể dùng tính năng chia sẻ hình ảnh (use case 3.2.3) để thay đổi hình ảnh đại diện cho quán.

Bảng 9: mô tả use case Cập nhật thông tin một quán cà phê

3.2.3. Chia sẻ hình ảnh về một quán cà phê hay một dịch vụ:



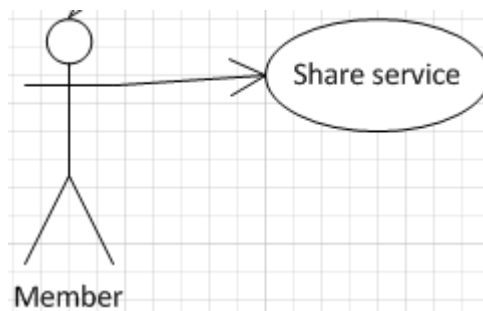
Hình 7: use case Chia sẻ hình ảnh

Use case	Share photo
Mô tả	Use case này cho phép người dùng chia sẻ hình đại diện cho các quán cà phê hay các dịch vụ.

Tác nhân	Thành viên
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng chọn hình muốn chia sẻ. - Tải hình ảnh lên <i>server</i> của hệ thống. - Từ <i>server</i>, tải hình lên tài khoản Facebook của người dùng. - Nếu quá trình tải hình lên Facebook thất bại, giữ hình ảnh trên <i>server</i>. Nếu thành công, xóa hình trên <i>server</i> của hệ thống.
Các điều kiện đặc biệt	<ul style="list-style-type: none"> - Nếu quán mới được chia sẻ, người dùng chia sẻ quán sẽ được dùng tính năng này. - Nếu tính năng này được dùng để cập nhật hình đại diện cho một quán đang có, chỉ có thành viên quản trị, thành viên quản lý quán và chủ quán mới được sử dụng. - Quyền được đăng bài lên tường của người dùng (<i>publish_stream</i>).
Tiền điều kiện	Người dùng đang trong quá trình chia sẻ một quán cà phê hay đang chỉnh sửa thông tin của quán.
Hậu điều kiện	Người dùng có thể tiếp tục hoàn thành quá trình chia sẻ một quán cà phê hay chỉnh sửa thông tin.
Điểm mở rộng	Không có.

Bảng 10: mô tả use case Chia sẻ hình

3.2.4. Thêm một dịch vụ vào danh sách các dịch vụ do quán phục vụ:



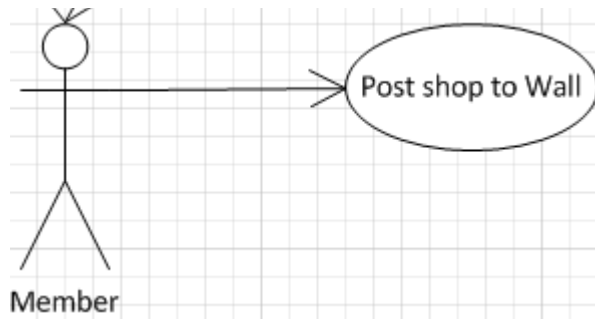
Hình 8: use case Chia sẻ dịch vụ

Use case	Share service
Mô tả	Use case này cho phép người dùng chia sẻ những dịch vụ mà một quán cà phê phục vụ.

Tác nhân	Thành viên
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng chọn những dịch vụ mà quán có. - Người dùng bấm “Lưu” để hoàn tất.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đang xem danh sách các quán cà phê; hoặc - Người dùng đang xem chi tiết một quán.
Hậu điều kiện	<ul style="list-style-type: none"> - Nếu là người dùng là quản trị viên hay thành viên quản lý quán hay chủ quán thì những dịch vụ chia sẻ sẽ vào danh sách các dịch vụ của quán ngay lập tức. - Nếu người dùng không thuộc trường hợp trên, các dịch vụ chia sẻ sẽ cần được một quản trị viên hay một thành viên quản lý quán hay chủ quán chấp nhận.
Điểm mở rộng	Không có.

Bảng 11: mô tả use case Chia sẻ dịch vụ

3.2.5. Chia sẻ một quán lên Facebook:



Hình 9: use case Chia sẻ một quán lên Facebook

Use case	Post shop to Wall
Mô tả	Use case này cho phép người dùng chia sẻ những quán cà phê yêu thích lên tường của họ.
Tác nhân	Thành viên
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng chọn quán cà phê muốn chia sẻ. - Người dùng ghi thông điệp. - Người dùng bấm “Chia sẻ lên tường” để hoàn tất. - Quán cà phê cùng thông điệp của người dùng sẽ xuất

	hiện trên tường của người dùng tại Facebook.
Các điều kiện đặc biệt	Quyền được đăng bài lên tường của người dùng (<i>publish_stream</i>).
Tiền điều kiện	Người dùng đang ở tại trang chủ.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 12: mô tả use case Chia sẻ một quán lên Facebook

3.2.6. Xác nhận là khách hàng của một quán cà phê:

Use case	Is customer
Mô tả	Use case này cho phép người dùng xác nhận họ là khách hàng của một quán cà phê.
Tác nhân	Thành viên
Dòng sự kiện	Người dùng chọn “ <i>You were here</i> ”.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Người dùng đang xem chi tiết một quán.
Hậu điều kiện	Người dùng sẽ được thêm vào danh sách khách hàng của quán.
Điểm mở rộng	Không có.

Bảng 13: mô tả use case Xác nhận khách hàng

3.2.7. Xác nhận đang hay đã từng làm việc tại một quán cà phê:

Use case	Is employee
Mô tả	Use case này cho phép người dùng xác nhận họ là nhân viên của quán hay đã từng là.
Tác nhân	Thành viên
Dòng sự kiện	Người dùng chọn “ <i>I am working at this shop</i> ” để xác định người dùng đang làm việc tại quán. Người dùng chọn “ <i>I worked at this shop</i> ” để xác định người dùng đã từng làm việc tại quán.
Các điều kiện đặc biệt	Không có.

Tiền điều kiện	Người dùng đang xem chi tiết một quán.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

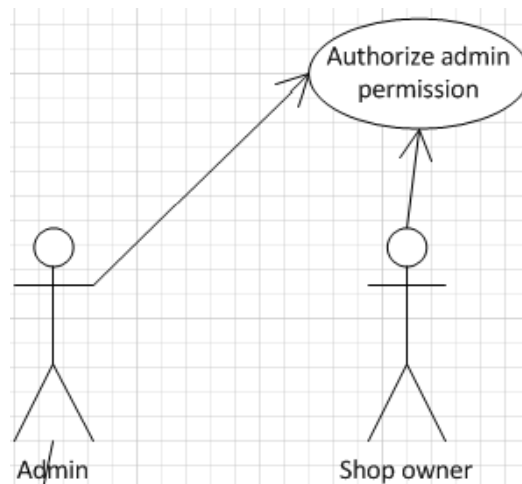
Bảng 14: mô tả use case Xác nhận đang hay đã làm việc

3.2.8. Xác nhận là chủ quán cà phê:

Use case	Is owner
Mô tả	Use case này cho phép người dùng xác nhận họ là chủ của một quán cà phê.
Tác nhân	Thành viên
Dòng sự kiện	Người dùng chọn “ <i>I am the shop owner</i> ”.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Người dùng đang xem chi tiết một quán.
Hậu điều kiện	Người dùng sẽ được thêm vào danh sách chờ để được xác nhận (use case 3.29).
Điểm mở rộng	Không có.

Bảng 15: mô tả use case Xác nhận là chủ quán

3.2.9. Xác nhận quyền chỉnh sửa thông tin quán cà phê:



Hình 10: use case Xác nhận quyền chỉnh sửa

Use case	Authorize admin permission
Mô tả	Use case này cho phép người dùng quản lý danh sách những thành viên được quyền chỉnh sửa thông tin các quán mà họ

	quản lý.
Tác nhân	Quản trị viên, chủ quán.
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng chọn “<i>Review Permissions</i>”. - Chọn một quyền bất kỳ để xác nhận (<i>approve</i>) hay từ chối (<i>reprove</i>) quyền đó.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Người dùng đang xem hồ sơ (<i>profile</i>) của mình.
Hậu điều kiện	<ul style="list-style-type: none"> - Người được xác nhận sẽ có quyền chỉnh sửa thông tin, thay đổi danh sách dịch vụ của quán đó. - Nếu người dùng được xác nhận là chủ quán, người đó còn có quyền xác định những người khác có quyền quản lý quán đó.
Điểm mở rộng	Không có.

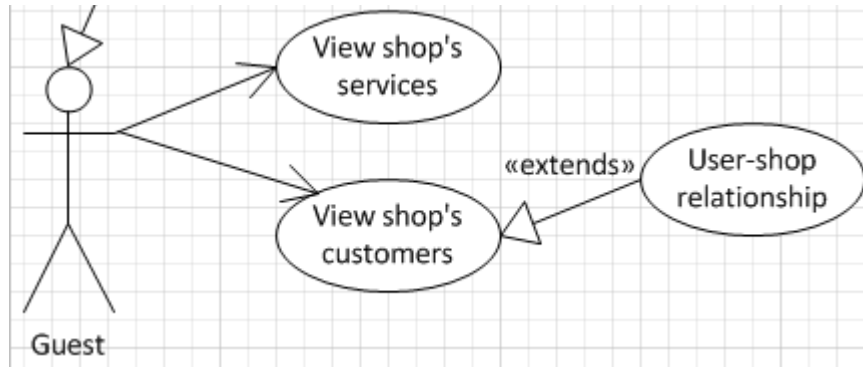
Bảng 16: mô tả use case Xác nhận quyền chỉnh sửa

3.2.10. Xem danh sách dịch vụ của một quán cà phê:

Use case	View shop's services
Mô tả	Use case này cho phép người dùng biết các dịch vụ của một quán.
Tác nhân	Khách
Dòng sự kiện	Người dùng xem danh sách các dịch vụ của quán dưới dạng <i>slideshow</i> .
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Người dùng đang xem chi tiết một quán.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 17: mô tả use case Xem danh sách dịch vụ

3.2.11. Xem danh sách những khách hàng của một quán cà phê:

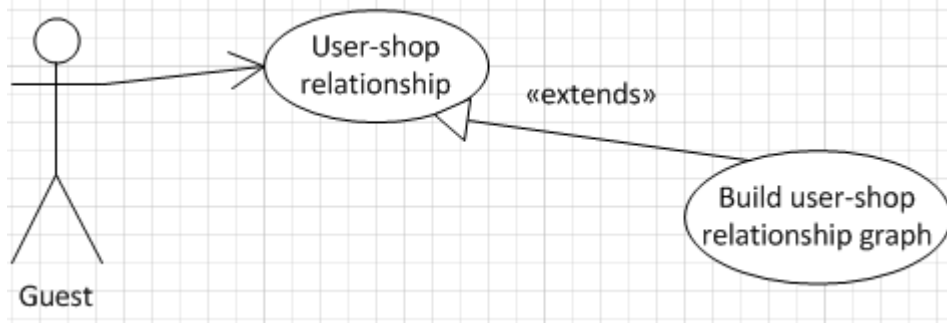


Hình 11: use case Xem danh sách khách hàng

Use case	View shop's customers
Mô tả	Use case này cho phép lấy danh sách những khách hàng đã đến quán.
Tác nhân	Khách
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng được xem trước một số khách hàng và tổng số khách hàng của quán mà ứng dụng ghi nhận được. Những khách hàng mà người dùng được xem trước là bản thân người dùng (nếu là khách hàng) cùng với những bạn bè của người dùng (nếu có). - Người dùng bấm “Xem thêm” để xem được nhiều thông tin hơn.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Người dùng đang xem chi tiết một quán.
Hậu điều kiện	Không có.
Điểm mở rộng	Nếu người dùng là một thành viên và người dùng bấm xem thêm, người dùng sẽ được đưa đến trang thống kê mối quan hệ giữa người dùng với quán (use case 3.2.15).

Bảng 18: mô tả use case Xem danh sách khách hàng

3.2.12. Vẽ sơ đồ mối liên hệ giữa một người dùng và một quán cà phê:

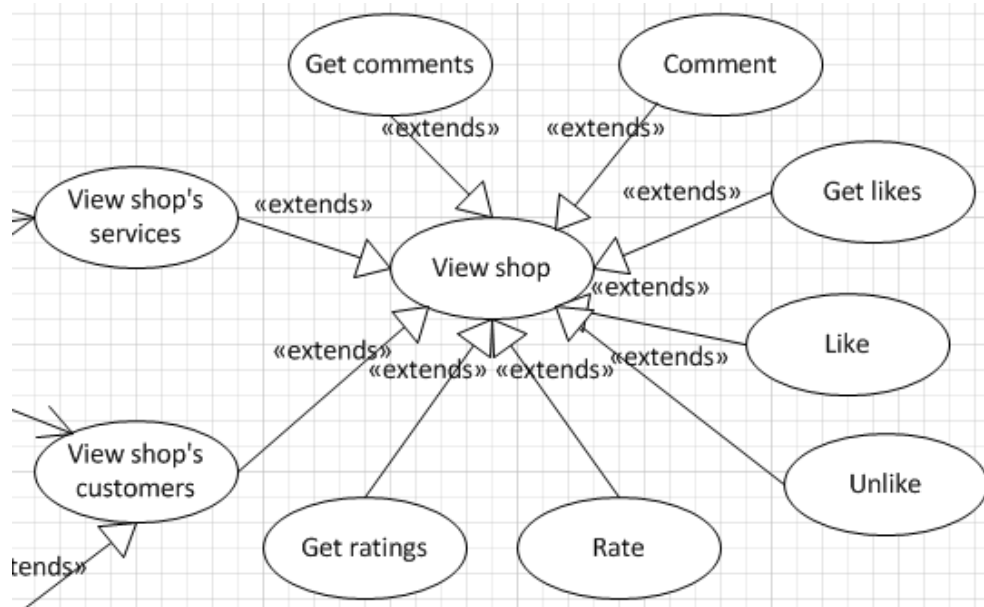


Hình 12: use case Vẽ sơ đồ mối liên hệ

Use case	Build user-shop relationship graph
Mô tả	Use case này cho phép người dùng xem mối liên hệ giữa một người dùng bất kỳ với một quán bằng một sơ đồ.
Tác nhân	Khách
Dòng sự kiện	<ul style="list-style-type: none"> - Hệ thống lựa chọn một số người bạn của người dùng để thể hiện trên sơ đồ. Những người bạn được lựa chọn là khách hàng của quán, đang làm việc tại quán, hay thích quán.. (cách thức chọn lựa sẽ được mô tả kĩ hơn trong chương 4).
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Người dùng đang xem trang mối liên hệ giữa người dùng và một quán.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 19: mô tả use case Vẽ sơ đồ mối liên hệ

3.2.13. Xem chi tiết một quán cà phê:



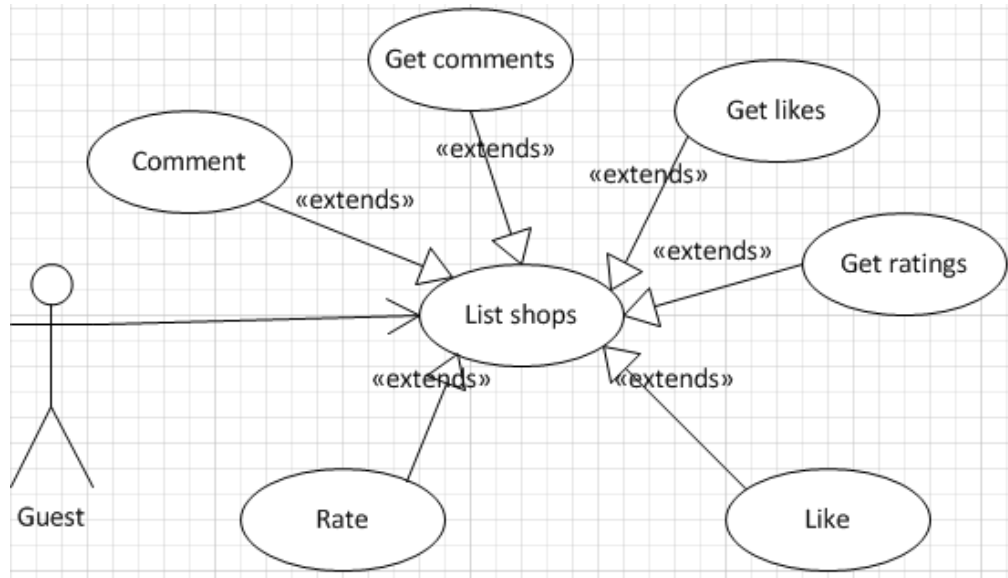
Hình 13: use case Xem chi tiết một quán cà phê

Use case	View shop
Mô tả	Use case này cho phép người dùng xem mối liên hệ giữa một người dùng bất kỳ với một quán bằng một sơ đồ.
Tác nhân	Khách
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng sẽ được xem những thông tin cơ bản về quán, hình ảnh đại diện. - Các use case mở rộng sẽ giúp người dùng biết danh sách các dịch vụ của quán, khách hàng của quán, các lời bình của người dùng với quán...
Các điều kiện đặc biệt	Nếu người dùng là khách, người dùng không được phép sử dụng các use case <i>Like</i> , <i>Unlike</i> , <i>Comment</i> và <i>Rate</i> .
Tiền điều kiện	Người dùng đang xem trang mối liên hệ giữa người dùng và một quán.
Hậu điều kiện	Không có.
Điểm mở rộng	<ul style="list-style-type: none"> - Các use case <i>Comment</i>, <i>Like</i>, <i>Unlike</i>, <i>Rate</i> cho phép người dùng đánh giá, thích hay cho điểm quán (khách không được sử dụng). - Các use case <i>Get comments</i>, <i>Get likes</i>, <i>Get ratings</i>, <i>View</i>

shop's customers, View shop's services sẽ cho phép người dùng xem thêm các thông tin khác về quán.

Bảng 20: mô tả use case Xem chi tiết một quán

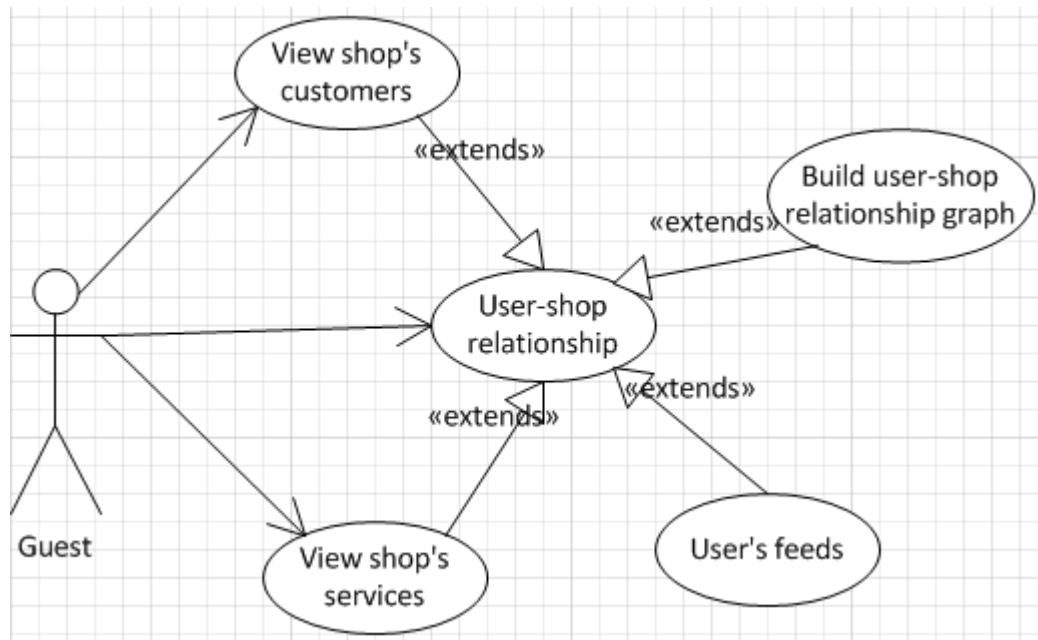
3.2.14. Xem danh sách các quán được chia sẻ theo thứ tự gần nhất:



Hình 14: use case Xem danh sách các quán

Use case	List shops
Mô tả	Use case này cho phép người dùng xem danh sách tất cả các quán theo thứ tự được chia sẻ gần nhất.
Tác nhân	Khách
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng sẽ được xem danh sách những quán có trên hệ thống được sắp xếp theo thứ tự chia sẻ gần nhất. - Các use case mở rộng sẽ giúp người dùng biết danh sách các lời bình của người dùng với quán, những người thích quán, điểm đánh giá của người dùng...
Các điều kiện đặc biệt	Nếu người dùng là khách, người dùng không được phép sử dụng các use case <i>Like</i> , <i>Unlike</i> , <i>Comment</i> và <i>Rate</i> .
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	<ul style="list-style-type: none"> - Các use case <i>Comment</i>, <i>Like</i>, <i>Unlike</i>, <i>Rate</i> cho phép người dùng đánh giá, thích hay cho điểm quán (khách không được sử dụng).

Bảng 21: mô tả use case Xem danh sách các quán

3.2.15. Xem mối liên hệ giữa người dùng và quán:

Hình 15: use case Xem mối liên hệ

Use case	User-shop relationship
Mô tả	Use case này cho phép người dùng xem danh sách tất cả các quán theo thứ tự được chia sẻ gần nhất.
Tác nhân	Khách
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng sẽ được xem thông tin tóm gọn về quán cũng như những thống kê về sự liên hệ giữa người dùng với quán.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	<ul style="list-style-type: none"> - Các use case mở rộng như <i>View shop's customers</i>, <i>View shop's services</i> giúp người dùng biết được danh sách khách hàng cũng như các dịch vụ của quán. - Use case <i>User's feed</i> giúp người dùng biết được những hoạt động của người dùng có liên quan đến quán. - Use case <i>Build user-shop relationship graph</i> cung cấp

cho người dùng một hình vẽ về mối liên hệ giữa người dùng và quán.

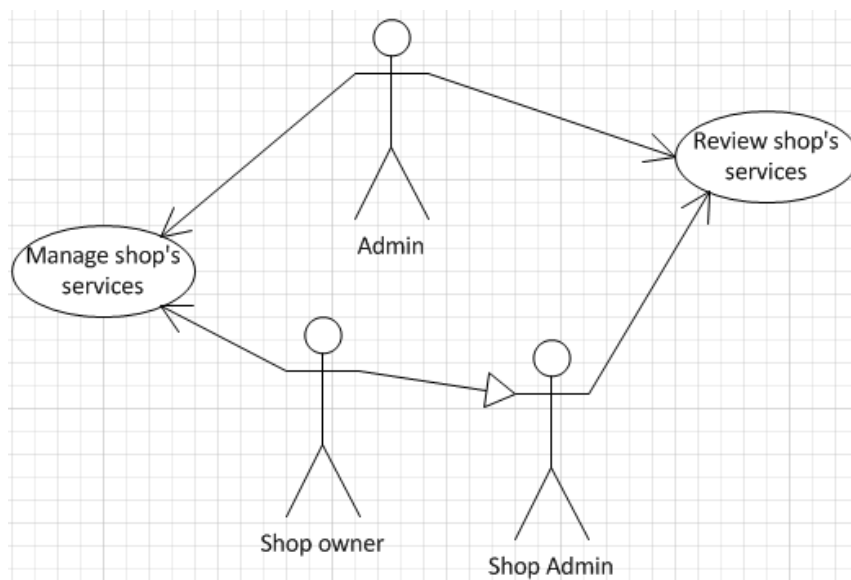
Bảng 22: mô tả use case Xem mối liên hệ

3.2.16. Xác nhận dịch vụ của quán:

Use case	Review shop's services
Mô tả	Use case này giúp người dùng xác nhận hay từ chối các dịch vụ của quán do những thành viên khác chia sẻ.
Tác nhân	Thành viên quản trị, chủ quán, thành viên quản lý quán.
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng xem danh sách các dịch vụ của quán do những người dùng khác chia sẻ. - Người dùng chọn “Chấp nhận” để xác nhận dịch vụ đó quán có cung cấp. Ngược lại thì bấm “Từ chối”.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 23: mô tả use case Xác nhận dịch vụ

3.2.17. Quản lý các dịch vụ của quán:



Hình 16: use case Quản lý các dịch vụ

Use case	Manage shop's services
----------	------------------------

Mô tả	Use case này giúp người dùng xác nhận hay từ chối các dịch vụ của quán do những thành viên khác chia sẻ.
Tác nhân	Thành viên quản trị, chủ quán, thành viên quản lý quán.
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng xem danh sách những dịch vụ hiện tại của quán. - Người dùng có thể xóa một số dịch vụ hay thêm vào một số dịch vụ khác.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 24: mô tả use case Quản lý các dịch vụ

3.2.18. Cho điểm một quán cà phê hay một dịch vụ:

Use case	Rate
Mô tả	Use case này cho phép người dùng đánh giá một quán cà phê bằng điểm số từ 1 tới 5.
Tác nhân	Thành viên
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng đánh giá quán từ 1 sao tới 5 sao.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Người dùng đang xem chi tiết quán hay danh sách các quán.
Hậu điều kiện	Điểm trung bình của quán sẽ được cập nhật.
Điểm mở rộng	Không có.

Bảng 25: mô tả use case Cho điểm quán

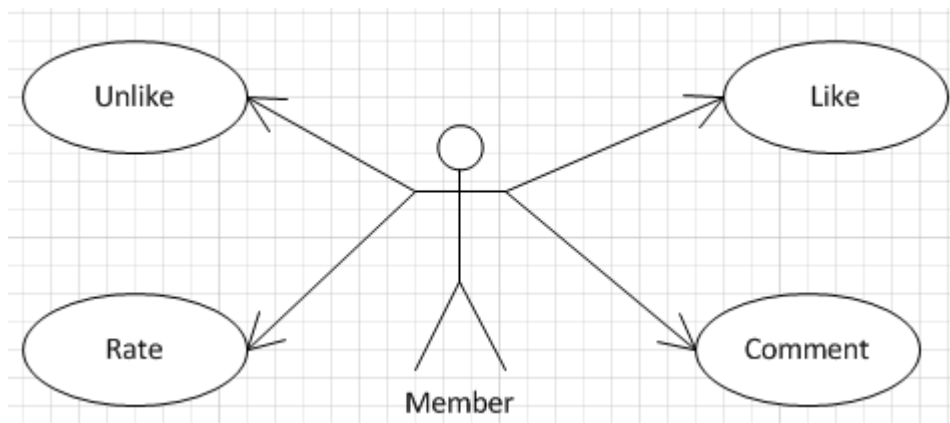
3.2.19. Bình luận:

Use case	Comment
Mô tả	Use case này giúp người dùng đưa ra những lời bình cho một quán cà phê.
Tác nhân	Thành viên
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng đưa ra lời bình luận của mình rồi bấm <i>Enter</i>.

	- Người dùng có thể đưa ra lời bình trên một quán hay trả lời một lời bình khác.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Người dùng đang xem chi tiết quán hay danh sách các quán.
Hậu điều kiện	Lời bình sẽ được xuất hiện ngay tại danh sách lời bình.
Điểm mở rộng	Không có.

Bảng 26: mô tả use case Bình luận

3.2.20. Thích một quán hay một loại cà phê:

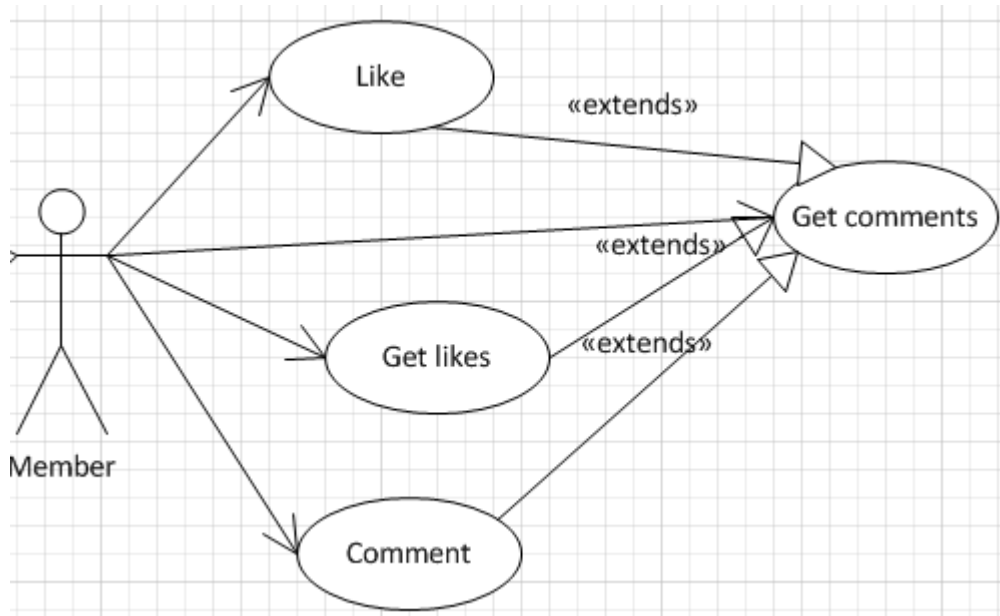


Hình 17: use case Thích

Use case	Like
Mô tả	Use case này giúp người dùng đưa ra những lời bình cho một quán cà phê.
Tác nhân	Thành viên
Dòng sự kiện	- Người dùng bấm “Thích”.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Người dùng đang xem chi tiết quán hay danh sách các quán.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 27: mô tả use case Thích

3.2.21. Xem danh sách các lời bình cho một quán cà phê:

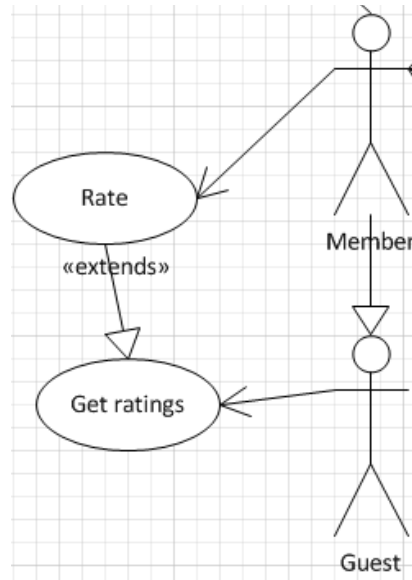


Hình 18: use case Xem danh sách lời bình

Use case	Get comments
Mô tả	Use case này giúp người dùng xem danh sách những lời bình của một quán.
Tác nhân	Thành viên
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng xem danh sách các lời bình của quán. - Nếu danh sách quá dài, hệ thống sẽ tự động hiển thị từng phần một của danh sách. Phần được hiển thị sẽ là những lời bình mới nhất. Người dùng chọn “Xem những lời bình trước” để xem những lời bình trước đó. - Người dùng có thể xem những lời bình trả lời cho những lời bình.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Người dùng đang xem chi tiết quán hay danh sách các quán.
Hậu điều kiện	Không có.
Điểm mở rộng	Các use case <i>Comment</i> , <i>Like</i> , <i>Get likes</i> giúp cho người dùng có thể trả lời cho một lời bình, thích lời bình đó hay biết được những người thích lời bình đó.

Bảng 28: mô tả use case Xem danh sách lời bình

3.2.22. Xem điểm đánh giá một quán cà phê:



Hình 19: use case Xem điểm đánh giá

Use case	Get ratings
Mô tả	Use case này giúp người dùng xem thống kê về điểm trung bình của một quán.
Tác nhân	Khách
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng xem điểm trung bình của quán. - Người dùng xem có bao nhiêu người đánh giá 5 sao, 4 sao...
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Người dùng đang xem chi tiết quán hay danh sách các quán.
Hậu điều kiện	Không có.
Điểm mở rộng	Use case <i>Rate</i> cho phép người dùng cho điểm đánh giá của riêng mình.

Bảng 29: mô tả use case Xem điểm đánh giá

3.2.23. Xem danh sách những khách hàng thích một quán cà phê:

Use case	Get likes
Mô tả	Use case này giúp người dùng xem danh sách những người thích quán.
Tác nhân	Thành viên

Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng được xem danh sách tóm gọn những người dùng thích quán. - Người dùng có thể bấm vào “Xem thêm” hay số lượng người thích còn lại để xem toàn bộ danh sách.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Người dùng đang xem chi tiết quán hay danh sách các quán.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

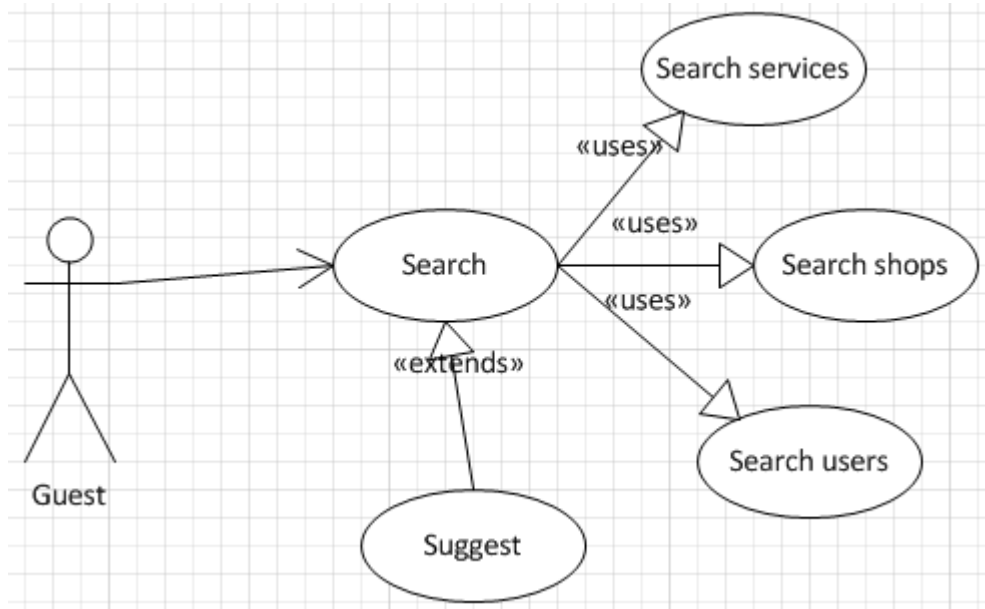
Bảng 30: mô tả use case Xem danh sách khách hàng thích quán

3.2.24. Hết thích một quán cà phê hay một dịch vụ:

Use case	Unlike
Mô tả	Use case này giúp người dùng xác định “hết thích” một quán.
Tác nhân	Thành viên
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng bấm “Hết thích”.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Người dùng đang xem chi tiết quán hay danh sách các quán và người dùng đã thích quán đó.
Hậu điều kiện	Người dùng bị loại ra khỏi danh sách những người thích quán.
Điểm mở rộng	Không có.

Bảng 31: mô tả use case Hết thích

3.2.25. Tìm kiếm:



Hình 20: use case Tìm kiếm

Use case	Search
Mô tả	Use case này giúp người dùng tìm kiếm được những người dùng khác hay quán, dịch vụ.
Tác nhân	Khách
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng gõ từ khóa cần tìm kiếm. - Người dùng bấm “Tìm kiếm” để xem kết quả.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	<ul style="list-style-type: none"> - Use case <i>Suggest</i> giúp hiển thị những gợi ý về kết quả cho người dùng.

Bảng 32: mô tả use case Tìm kiếm

3.2.26. Tìm kiếm người dùng:

Use case	Search users
Mô tả	Use case này giúp người dùng tìm kiếm được những người dùng khác.
Tác nhân	Khách

Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng gõ từ khóa cần tìm kiếm. - Người dùng bấm “Tìm kiếm” để xem kết quả. - Từ khóa tìm kiếm có thể nằm trong họ, tên, địa chỉ <i>email</i>...
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	<ul style="list-style-type: none"> - Use case <i>Suggest</i> giúp hiển thị những gợi ý về kết quả cho người dùng.

Bảng 33: mô tả use case Tìm kiếm người dùng.

3.2.27. Tìm kiếm quán cà phê:

Use case	Search shops
Mô tả	Use case này giúp người dùng tìm kiếm được những quán cà phê.
Tác nhân	Khách
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng gõ từ khóa cần tìm kiếm. - Người dùng bấm “Tìm kiếm” để xem kết quả. - Từ khóa tìm kiếm có thể nằm trong tên quán, mô tả quán, loại quán...
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	<ul style="list-style-type: none"> - Use case <i>Suggest</i> giúp hiển thị những gợi ý về kết quả cho người dùng.

Bảng 34: tìm kiếm quán cà phê

3.2.28. Tìm kiếm dịch vụ:

Use case	Search services
Mô tả	Use case này giúp người dùng tìm kiếm được những dịch vụ.

Tác nhân	Khách
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng gõ từ khóa cần tìm kiếm. - Người dùng bấm “Tìm kiếm” để xem kết quả. - Từ khóa tìm kiếm có thể nằm trong tên dịch vụ, mô tả...
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	<ul style="list-style-type: none"> - Use case <i>Suggest</i> giúp hiển thị những gợi ý về kết quả cho người dùng.

Bảng 35: tìm kiếm dịch vụ

3.2.29. Gợi ý kết quả tìm kiếm:

Use case	Suggest
Mô tả	Use case này đưa ra các kết quả gợi ý cho người dùng trong lúc người dùng gõ từ khóa.
Tác nhân	Khách
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng gõ từ khóa cần tìm kiếm. - Hệ thống hiển thị ra một số kết quả gợi ý cho người dùng để giúp việc tìm kiếm nhanh hơn.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Người dùng đang gõ từ khóa vào ô tìm kiếm.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 36: gợi ý tìm kiếm

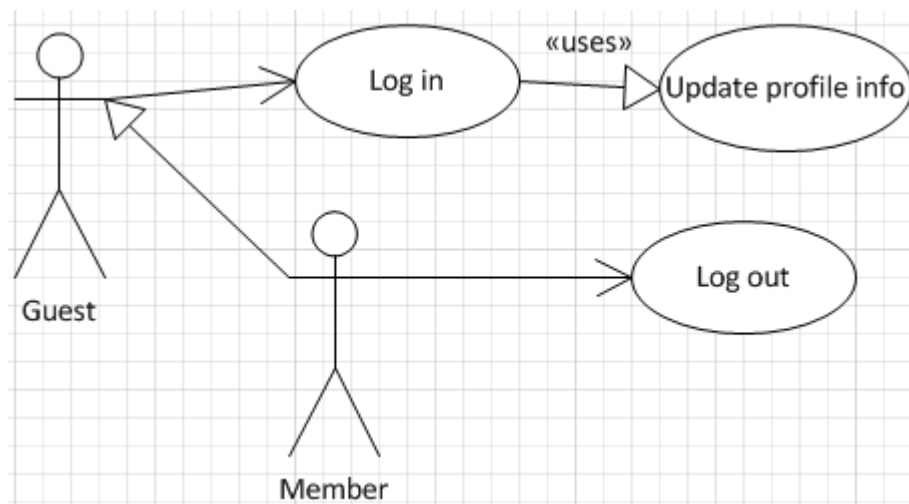
3.2.30. Đề nghị những quán nên đến:

Use case	Recommend shops
Mô tả	Use case này đưa ra danh sách các quán người dùng nên thử đến.
Tác nhân	Thành viên
Dòng sự kiện	<ul style="list-style-type: none"> - Hệ thống hiển thị danh sách các quán mà người dùng

	nên đến. Kết quả đưa ra dựa trên dữ liệu ghi nhận được về người dùng cũng như bạn của người dùng. Chi tiết cách chọn kết quả sẽ được mô tả trong chương 4.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 37: đề nghị những quán nên đến

3.2.31. Đăng nhập:



Hình 21: đăng nhập

Use case	Log in
Mô tả	Use case này cho phép người dùng đăng nhập vào hệ thống.
Tác nhân	Khách
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng bấm vào “Join us with Facebook” để bắt đầu đăng nhập. - Một yêu cầu sẽ được gửi đến Facebook. Facebook sẽ yêu cầu người dùng đăng nhập vào tài khoản Facebook của người dùng. - Nếu người dùng đã đăng nhập hay tài khoản đăng nhập thành công, Facebook kiểm tra các quyền (<i>permission</i>)

	<p>mà hệ thống đòi hỏi so với những quyền mà người dùng đã cho phép hệ thống thực thi. Nếu có quyền nào người dùng cho phép, Facebook sẽ hỏi và chờ người dùng chấp nhận.</p> <ul style="list-style-type: none"> - Nếu quá trình trên thành công, Facebook sẽ gửi thông tin đăng nhập thành công (cùng một số thông tin khác) cho hệ thống để xác nhận quá trình đăng nhập thành công. - Hệ thống sử dụng use case “<i>Update profile info</i>” để cập nhật thông tin cơ bản của người dùng vào hệ thống.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	<ul style="list-style-type: none"> - Người dùng trở thành một thành viên của hệ thống. - Người dùng được đưa trở về trang trước đó mà người dùng đang xem.
Điểm mở rộng	Không có.

Bảng 38: đăng nhập

3.2.32. Đăng xuất:

Use case	Log in
Mô tả	Use case này cho phép người dùng đăng xuất khỏi hệ thống.
Tác nhân	Thành viên
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng bấm vào nút “Đăng xuất”.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	<ul style="list-style-type: none"> - Người dùng được đưa lại trang chủ.
Điểm mở rộng	Không có.

Bảng 39: đăng xuất

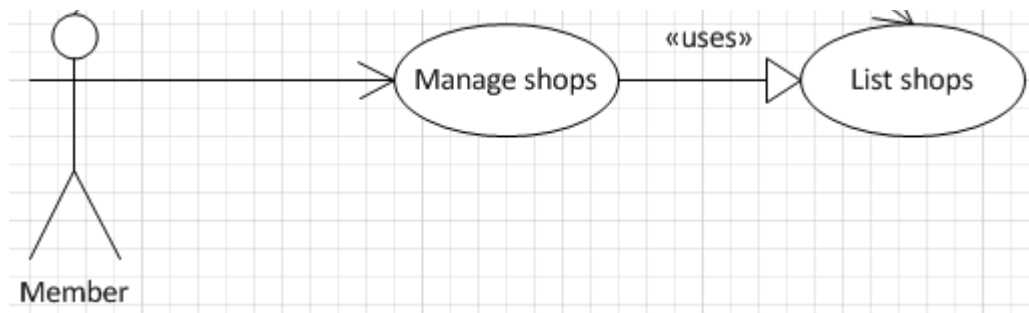
3.2.33. Cập nhật thông tin tài khoản:

Use case	Log in
Mô tả	Use case này giúp hệ thống luôn có được thông tin mới nhất

	về người dùng.
Tác nhân	Khách
Dòng sự kiện	Sau khi Facebook xác nhận quá trình đăng nhập thành công, hệ thống sẽ gửi một yêu cầu lấy thông tin cơ bản của người dùng và cập nhật vào hệ thống.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Người dùng đăng nhập thành công.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 40: cập nhật thông tin tài khoản

3.2.34. Xem danh sách các quán mà người dùng có quyền chỉnh sửa:



Hình 22: Danh sách người dùng có quyền chỉnh sửa

Use case	Manage shops
Mô tả	Use case này giúp người dùng quản lý danh sách những quán mà người dùng có quyền.
Tác nhân	Thành viên
Dòng sự kiện	Người dùng được xem danh sách những quán mà người dùng là chủ quán hay thành viên quản lý.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 41: Danh sách người dùng có quyền chỉnh sửa

3.2.35. Xem danh sách các quán mà người dùng thích:

Use case	Manage liked shops
Mô tả	Use case này giúp người dùng xem danh sách các quán mà người dùng thích.
Tác nhân	Thành viên
Dòng sự kiện	Người dùng được xem danh sách những quán cà phê mà người dùng thích.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 42: Danh sách các quán mà người dùng thích

3.2.36. Xem danh sách các dịch vụ mà người dùng thích:

Use case	Manage liked services
Mô tả	Use case này giúp người dùng xem và thay đổi danh sách các dịch vụ mà người dùng thích.
Tác nhân	Thành viên
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng được xem danh sách những dịch vụ mà người dùng thích. - Người dùng có thể thay đổi danh sách bằng cách thêm hay bớt các dịch vụ.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 43: dịch vụ người dùng thích

3.2.37. Xem danh sách các quán mà người dùng đã đánh giá:

Use case	Seerated shops
Mô tả	Use case này giúp người dùng xem danh sách các dịch vụ

	mà người dùng thích.
Tác nhân	Thành viên
Dòng sự kiện	Người dùng được xem danh sách những dịch vụ mà người dùng đã đánh giá.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 44: quán người dùng đã đánh giá

3.2.38. Xem danh sách các hoạt động gần đây của người dùng:

Use case	User's feeds
Mô tả	Use case này giúp liệt kê tất cả các hoạt động gần đây của người dùng.
Tác nhân	Khách
Dòng sự kiện	Người dùng được xem tất cả các hoạt động gần đây của người dùng, ví dụ như mỗi lần <i>check-in</i> , đưa ra lời bình, thích quán hay chia sẻ quán mới.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 45: hoạt động gần đây người dùng

3.2.39. Xem mạng lưới mối quan hệ của người dùng:

Use case	View network
Mô tả	Use case này tạo ra một hình vẽ thể hiện các mối quan hệ của người dùng.
Tác nhân	Thành viên
Dòng sự kiện	<ul style="list-style-type: none"> - Người dùng được xem tất cả các bạn bè của mình. - Người dùng có thể chọn để xem tất cả những quán mà

	người dùng và bạn bè đã từng đến và những dịch vụ của quán đó.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 46: mạng lưới quan hệ người dùng

3.2.40. Xem danh sách các quán mà người dùng đã đến:

Use case	Visited shops
Mô tả	Use case này liệt kê danh sách những quán mà người dùng đã đến.
Tác nhân	Thành viên
Dòng sự kiện	Người dùng được xem danh sách những quán mà người dùng đã đến.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 47: quán người dùng đã đến

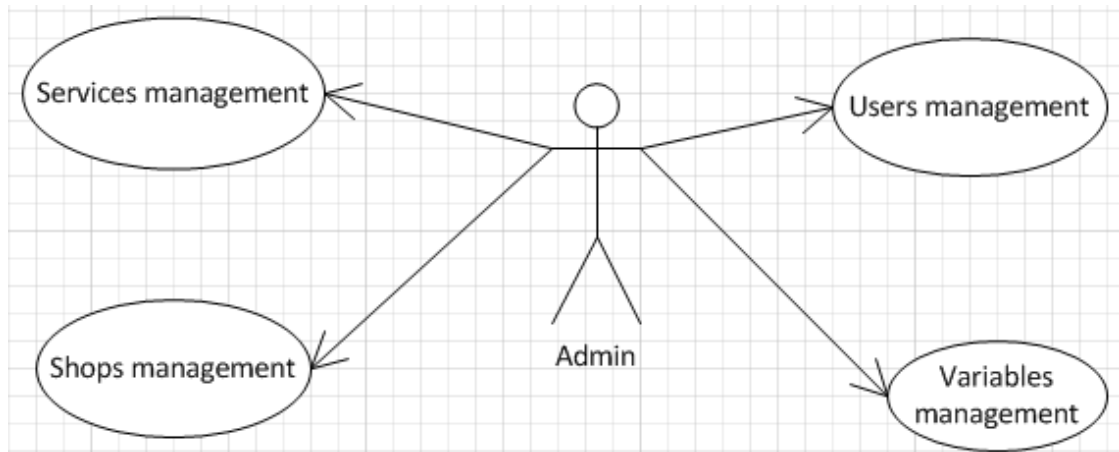
3.2.41. Xem danh sách các quán mà người dùng có thể đã đến:

Use case	Visited shops
Mô tả	Use case này liệt kê danh sách những quán mà người dùng có thể đã đến.
Tác nhân	Thành viên
Dòng sự kiện	Người dùng được xem danh sách những quán mà người dùng có thể đã đến. Người dùng chọn một quán trong danh sách để xác định đã đến quán đó.
Các điều kiện đặc biệt	Không có.

Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 48: quán có thể đã đến

3.2.42. Quản lý danh sách người dùng:



Hình 23: danh sách người dùng

Use case	Users management
Mô tả	Use case này cho phép các thành viên quản trị quản lý danh sách các thành viên trong hệ thống.
Tác nhân	Thành viên quản trị
Dòng sự kiện	Thành viên quản trị được xem danh sách tất cả các thành viên. Thành viên quản trị có quyền xóa một thành viên ra khỏi hệ thống.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 49: danh sách người dùng

3.2.43. Quản lý danh sách các quán cà phê:

Use case	Shops management
Mô tả	Use case này cho phép các thành viên quản trị quản lý danh

	sách các quán cà phê trong hệ thống.
Tác nhân	Thành viên quản trị
Dòng sự kiện	Thành viên quản trị được xem danh sách tất cả các quán cà phê. Thành viên quản trị có quyền xóa một quán ra khỏi hệ thống, chỉnh sửa thông tin quán hay chọn quán đó làm tiêu điểm (<i>favorite shop</i>).
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 50: danh sách quán cà phê

3.2.44. Quản lý danh sách các dịch vụ:

Use case	Services management
Mô tả	Use case này cho phép các thành viên quản trị quản lý danh sách các thành viên trong hệ thống.
Tác nhân	Thành viên quản trị
Dòng sự kiện	Thành viên quản trị được xem danh sách tất cả các dịch vụ. Thành viên quản trị có quyền xóa một dịch vụ ra khỏi hệ thống, chỉnh sửa một dịch vụ hay thêm một dịch vụ mới.
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 51: Danh sách dịch vụ

3.2.45. Quản lý các thông số hệ thống:

Use case	Services management
Mô tả	Use case này cho phép các thành viên quản trị quản lý các thông số hoạt động của hệ thống.
Tác nhân	Thành viên quản trị

Dòng sự kiện	Thành viên quản trị được xem và chỉnh sửa danh sách tất cả các thông số của hệ thống, ví dụ như số lượng quán liệt kê trong mỗi trang, số lượng quán đề nghị cho người dùng...
Các điều kiện đặc biệt	Không có.
Tiền điều kiện	Không có.
Hậu điều kiện	Không có.
Điểm mở rộng	Không có.

Bảng 52: quản lý thông số hệ thống

3.3. Ontology:

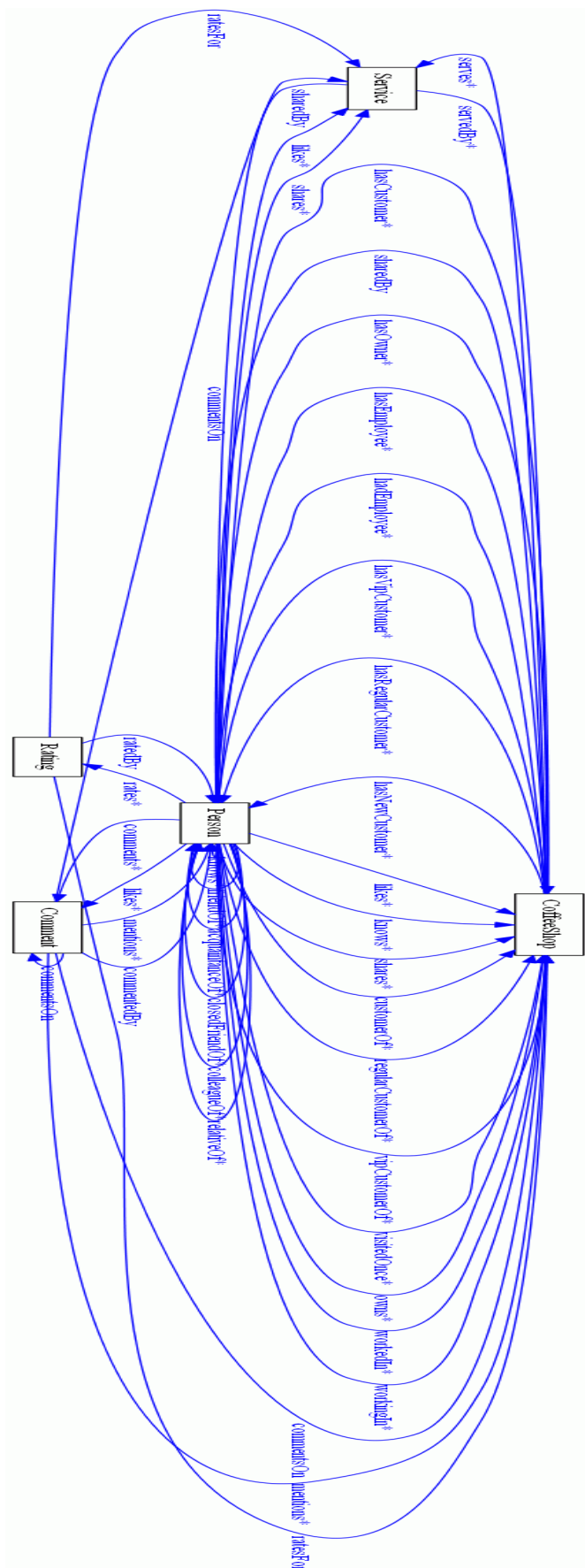
3.3.1. Danh sách các khái niệm (concept):

STT	Khái niệm (tiếng Việt)	Khái niệm (tiếng Anh)	Giải thích	Thừa kế
1	Thành viên	<i>Person</i>	Thành viên xác định cụ thể những người đã hay đang sử dụng ứng dụng.	<i>owl:Thing</i>
2	Quán cà phê	<i>CoffeeShop</i>	Quán cà phê thể hiện một địa điểm kinh doanh các loại thức uống được chế biến từ cà phê. Đây là điều kiện bắt buộc để địa điểm kinh doanh được gọi là quán cà phê. Ngoài ra, địa điểm này còn có thể kinh doanh các dịch vụ khác như phục vụ các loại thức ăn, nước uống khác, biểu diễn ca nhạc, đặt tiệc... Hay địa điểm đó có thể là một quầy hay cửa hàng trong một khách sạn, nhà hàng, siêu thị...	<i>owl:Thing</i>
3	Dịch vụ	<i>Service</i>	Đây là các dịch vụ mà các quán cà phê có thể cung cấp.	<i>owl:Thing</i>
4	Loại cà phê	<i>Coffee</i>	Các loại thức uống được pha chế từ cà phê.	<i>Drink</i>
5	Lời nhận xét	<i>Comment</i>	Lời nhận xét về một đối tượng nào đó trong ứng dụng, như về một người dùng, một quán cà phê, một loại cà	<i>owl:Thing</i>

			phê, một lời mời... hay là một lời nhận xét cho một lời nhận xét khác.	
6	Điểm đánh giá	<i>Rating</i>	Là một số nguyên có giá trị trong đoạn từ 1 đến 5. Điểm đánh giá thể hiện mức độ hài lòng (bằng số) về một quán cà phê hay một dịch vụ nào đó ở một quán. -	<i>owl:Thing</i>
7	Thức ăn	<i>Food</i>	Các loại thức ăn.	<i>Service</i>
8	Nước uống	<i>Drink</i>	Các loại nước uống.	<i>Service</i>
9	Lần đi	<i>Check-in</i>	Một lần đi đến một quán cà phê nào đó, tương ứng với mỗi lần người dùng <i>check-in</i> trên Facebook.	<i>owl:Thing</i>

Bảng 53: danh sách các khai niệm

3.3.2. Sơ đồ:



Hình 24: cấu trúc ontology

3.3.3. Danh sách các thuộc tính (slot):

Các thuộc tính được liệt kê theo từng khái niệm mà nó thuộc về. Các khái niệm “con” thừa hưởng các thuộc tính từ khái niệm “cha” nên các thuộc tính đã liệt kê trong khái niệm cha sẽ không được liệt kê lại trong các khái niệm con.

Khái niệm		Person	Thừa kế	owl:Thing
STT	Thuộc tính	Kiểu giá trị	Loại	Thừa kế
	personFullname	Kiểu chuỗi	Datatype	
	Mô tả: tên đầy đủ của người dùng.			
	joinDate	Kiểu ngày giờ	Datatype	
	Mô tả: ngày người dùng bắt đầu sử dụng ứng dụng.			
	likes	CoffeeShop, Comment, Service, Check-in.	Object	
	Mô tả: người dùng thích một nội dung nào đó, như một quán, dịch vụ, lời bình...			
	comments	Comment.	Object	
	Mô tả: người dùng đưa ra một lời bình luận, đánh giá.			
	rates	CoffeeShop, Service.	Object	
	Mô tả: người dùng đánh giá một quán hay một dịch vụ bằng một mức điểm cụ thể.			
	knows	CoffeeShop, Person	Object	
	Mô tả: biết đến một ai đó hay một quán.			
	customerOf	CoffeeShop	Object	knows
	Mô tả: là khách hàng nói chung, không phân biệt rõ là khách mới, khách quen...			
	visitedOnce	CoffeeShop	Object	customerOf
	Mô tả: khách hàng lần đầu đến quán.			
	regularCustomerOf	CoffeeShop	Object	customerOf
	Mô tả: khách hàng thường xuyên.			
	vipCustomerOf	CoffeeShop	Object	customerOf
	Mô tả: khách VIP.			
	friendOf	Person	Object	knows
	Mô tả: quan hệ bạn bè (nói chung).			
	acquaintanceOf	Person	Object	friendOf
	Mô tả: có gặp nhau một vài lần.			
	closedFriendOf	Person	Object	friendOf
	Mô tả: quan hệ bạn thân.			
	colleagueOf	Person	Object	friendOf
	Mô tả: quan hệ đồng nghiệp.			
	relativeOf	Person	Object	
	Mô tả: gia đình, họ hàng.			
	workedIn	CoffeeShop	Object	knows

	Mô tả: đã từng làm ở quán.	Đồng nghĩa:	
	workingIn	CoffeeShop	Object
	Mô tả: đang làm ở quán.	Đồng nghĩa:	
	checkedIn	Check-in	Object
	Mô tả: người dùng đến một quán nào đó.		
	shares	CoffeeShop, Service	Object
	Mô tả: chia sẻ một quán cà phê hay một dịch vụ.		
	owns	CoffeeShop	Object
	Mô tả: xác định người dùng là chủ một quán.		

Bảng 54: danh sách các thuộc tính của person

Khái niệm	CoffeeShop	Thừa kế	owl:Thing
STT	Thuộc tính	Kiểu giá trị	Loại
	shopTitle	Kiểu chuỗi	Datatype
	Mô tả: tên của quán.		
	serves	Service	Object
	Mô tả: xác định quán có cung cấp những dịch vụ gì.		
	hasCustomer	Person	Object
	Mô tả: xác định quán có những khách hàng là ai.		
	hasNewCustomer	Person	Object
	Mô tả: xác định quán có những khách hàng mới nào.		
	hasRegularCustomer	Person	Object
	Mô tả: xác định quán có những khách hàng thường xuyên nào.		
	hasVipCustomer	Person	Object
	Mô tả: xác định quán có những khách hàng VIP nào.		
	isSharedBy	Person	Object
	Mô tả: xác định quán được chia sẻ bởi người dùng nào.		
	hasEmployee	Person	Object
	Mô tả: xác định những người dùng là nhân viên của quán.		
	hadEmployee	Person	Object
	Mô tả: xác định những người dùng đã từng làm việc tại quán.		
	hasOwner	Person	Object
	Mô tả: xác định chủ của quán.		

Bảng 55: thuộc tính của coffeeshop

Khái niệm	Service	Thừa kế	owl:Thing
STT	Thuộc tính	Kiểu giá trị	Loại
	serviceTitle	Kiểu chuỗi	Datatype
	Mô tả: tên của dịch vụ.		
	servedBy	CoffeeShop	Object
	Mô tả: xác định những quán nào cung cấp dịch vụ này.		
	sharedBy	Person	Object
	Mô tả: người dùng đã chia sẻ dịch vụ.		

Bảng 56: thuộc tính của service

Khái niệm	Comment		Thừa kế	owl:Thing
STT	Thuộc tính	Kiểu giá trị	Loại	Thừa kế
	commentBrief	Kiểu chuỗi	Datatype	
	Mô tả: đoạn trích nội dung lời bình.			
	commentedBy	Person	Object	
	Mô tả: xác định người viết lời bình.			
	mentions	Person, CoffeeShop, Service.	Object	
	Mô tả: những đối tượng được lời bình nhắc tới.			
	commentsOn	CoffeeShop, Comment, Service.	Object	
	Mô tả: lời bình được thực hiện trên đối tượng nào.			

Bảng 57: thuộc tính của comment

Khái niệm	Rating		Thừa kế	owl:Thing
STT	Thuộc tính	Kiểu giá trị	Loại	Thừa kế
	ratePoint	Kiểu số thực	Datatype	
	Mô tả: điểm số đánh giá.			
	ratedBy	Person	Object	
	Mô tả: xác định người dùng cho điểm.			
	ratesFor	CoffeeShop, Service.	Object	
	Mô tả: xác định điểm số được đánh giá trên đối tượng nào.			

Bảng 58: thuộc tính của rating

Khái niệm	Check-in		Thừa kế	owl:Thing
STT	Thuộc tính	Kiểu giá trị	Loại	Thừa kế
	checkinDate	Kiểu ngày giờ	Datatype	
	Mô tả: xác định ngày đến quán			
	checkedInto	CoffeeShop	Object	
	Mô tả: xác định quán mà người dùng đã đến			
	checkedInToTry	Service	Object	
	Mô tả: xác định các dịch vụ mà người dùng đã sử dụng trong lần đến này.			
	checkedInWith	Person	Object	
	Mô tả: xác định những người đi cùng.			

Bảng 59: thuộc tính của check-in

3.3.4. Các ràng buộc (facet):

- Person *disjoint* {CoffeeShop, Service, Comment, Rating, Check-in}.
- CoffeeShop *disjoint* {Person, Service, Comment, Rating, Check-in}.
- Service *disjoint* {Person, CoffeeShop, Comment, Rating, Check-in}.
- Comment *disjoint* {Person, Service, CoffeeShop, Rating, Check-in}.
- Rating *disjoint* {Person, Service, Comment, CoffeeShop, Check-in}.
- Check-in *disjoint* {Person, Service, Comment, Rating, CoffeeShop}.
- Miền giá trị của *slot knows* là CoffeeShop, Person.

3.4. Kết chương:

Trong chương 3, chúng ta đã liệt kê ra đầy đủ các tính năng cần có của ứng dụng, cách từng tính năng hoạt động dựa trên góc nhìn của người dùng cuối. Chúng ta cũng đã xem cấu trúc của ontology được dùng để phục vụ cho các tính năng.

Trong chương 4, chúng ta sẽ đi sâu vào quá trình xử lý các tính năng, cách sử dụng ontology trong việc xử lý. Chúng ta cũng sẽ được xem các kết quả của các tính năng và những đánh giá về các kết quả này trong chương 4.

Chương 4: THIẾT KẾ CHƯƠNG TRÌNH

4.1. Danh sách các use case sẽ hiện thực:

Trong phần này, nhóm thực hiện sẽ chọn một số use case chính hay có một số xử lý đặc biệt để mô tả.

4.1.1. Danh sách các use case sẽ hiện thực:

Use case được hiện thực	Phần mô tả
Share shop	4.2.1
Share photo	4.2.2
Share service	4.2.3
Is customer	4.2.4
View shop's customers	4.2.5
User-shop relationship	4.2.9
List shops	4.2.7
Review shop's services	4.2.8
Comment	4.2.10
Get comments	4.2.11
Search shops	4.2.12
Recommend shops	4.2.13
Log in	4.2.14
Manage shops	4.2.15
User's feed	4.2.16
View network	4.2.17

Bảng 60: use case đã thực hiện

4.1.2. Danh sách các đối tượng:

STT	Tên lớp	Loại	Diễn giải
	Guest	Actor	Khách, đại diện cho tác nhân Khách.
	Member	Actor	Thành viên.
	Shop admin	Actor	Thành viên quản lý quán.
	Shop owner	Actor	Chủ quán.

	Admin	Actor	Thành viên quản trị.
	Facebook	Actor	Facebook, đối tượng này xuất hiện khi hệ thống cần trao đổi dữ liệu với Facebook.
	AddNewShopPage	Boundary	Trang chia sẻ quán mới.
	EditShopPage	Boundary	Trang chỉnh sửa thông tin quán.
	PhotoForm	Boundary	<i>Form</i> cho phép chia sẻ hình.
	AddServiceToShopPage	Boundary	Trang chia sẻ dịch vụ.
	ViewShopPage	Boundary	Trang xem quán.
	UserShopPage	Boundary	Trang mối liên hệ giữa người dùng và quán.
	ListShopsPage	Boundary	Trang liệt kê danh sách quán.
	NetworkPage	Boundary	Trang “mạng lưới”.
	FeedPage	Boundary	Trang liệt kê danh sách hoạt động.
	ReviewServicesPage	Boundary	Trang xác nhận/từ chối dịch vụ.
	ShopShouldVisitPage	Boundary	Trang tư vấn quán.
	CommentBox	Boundary	Khung bình luận.
	CommentHolder	Boundary	Nơi chứa các lời bình.
	ShopController	Control	Điều khiển các thao tác về quán.
	OntoController	Control	Điều khiển các thao tác về cơ sở tri thức.
	PhotoController	Control	Điều khiển các thao tác về chia sẻ hình, lấy đường dẫn hình...
	ServiceController	Control	Điều khiển các thao tác về dịch vụ.
	ProfileController	Control	Điều khiển các thao tác về hồ sơ cá nhân (<i>profile</i>).
	ActivityController	Control	Điều khiển các thao tác về hoạt động (<i>activity</i>).
	GraphController	Control	Điều khiển việc vẽ sơ đồ.
	CommentController	Control	Điều khiển các thao tác về lời

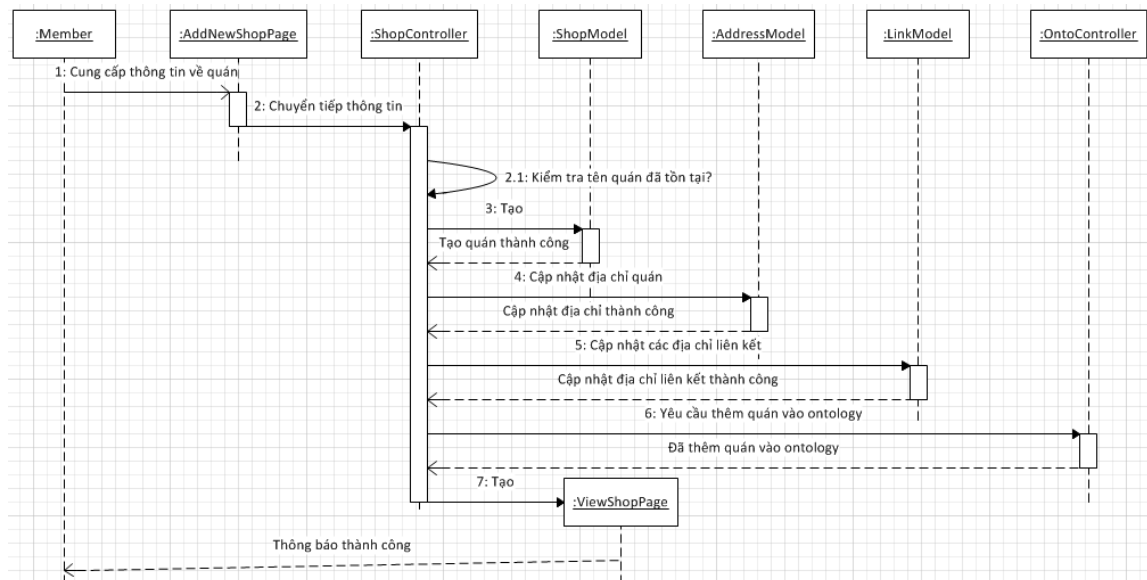
			bình.
	IndexController	Control	Điều khiển các thao tác về đăng nhập, đăng xuất...
	ShopPermissionController	Control	Điều khiển các thao tác về quyền của người dùng với quán.
	ShopModel	Entity	Đảm trách các thao tác với dữ liệu về các quán.
	AddressModel	Entity	Đảm trách các thao tác với dữ liệu về các địa chỉ.
	LinkModel	Entity	Đảm trách các thao tác với dữ liệu về các đường dẫn (<i>link</i>).
	PhotoModel	Entity	Đảm trách các thao tác với dữ liệu về hình ảnh.
	ServiceModel	Entity	Đảm trách các thao tác với dữ liệu về các dịch vụ.
	LikeModel	Entity	Đảm trách các thao tác với dữ liệu về sở thích.
	RatingModel	Entity	Đảm trách các thao tác với dữ liệu về các điểm đánh giá.
	CommentModel	Entity	Đảm trách các thao tác với dữ liệu về các lời bình.
	ActivityModel	Entity	Đảm trách các thao tác với dữ liệu về các hoạt động.
	CheckinModel	Entity	Đảm trách các thao tác với dữ liệu về các lần đến thăm quán.
	UserModel	Entity	Đảm trách các thao tác với dữ liệu về người dùng.
	ShopPermissionModel	Entity	Đảm trách các thao tác với dữ liệu về các quyền hạn với quán.

Bảng 61: danh sách các đối tượng

4.2. Mô tả các tính năng:

4.2.1. Giới thiệu một quán cà phê (Share shop):

Quá trình từ lúc người dùng cung cấp (*submit*) các thông tin về một quán cà phê vào hệ thống được mô tả bằng sơ đồ trình tự (*sequence diagram*) như sau:



Hình 25: giới thiệu quán cà phê

Lưu ý, quá trình trên giả định là tên quán chưa tồn tại trong CSDL và mọi xử lý đều thành công.

Mô tả:

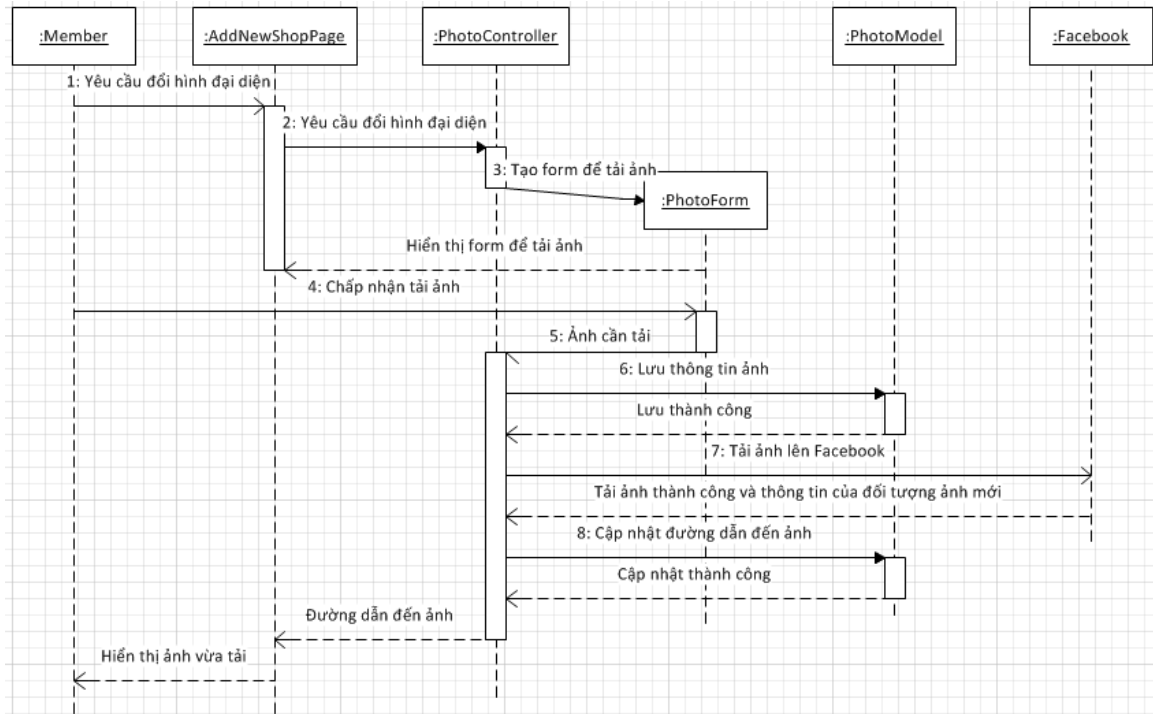
- Người dùng cung cấp (*submit*) thông tin về quán mới cho hệ thống.
- Hệ thống sẽ kiểm tra xem tên quán mà người dùng chọn đã dùng cho một quán khác trong hệ thống chưa. Nếu có rồi, thông báo lỗi cho người dùng biết. Kết thúc quá trình tạo quán và chờ lệnh tạo tiếp theo.
- Nếu chưa, một quán mới sẽ được tạo ra với những thông tin cơ bản như: tên, mô tả ngắn gọn, mô tả đầy đủ về quán và hình đại diện (nếu có) của quán.
- Để phục vụ cho nhu cầu mở rộng sau này, nhóm phát triển đã quyết định lưu các thông tin về địa chỉ của quán vào một bảng riêng. Do

đó, tiếp theo, hệ thống sẽ lấy các thông tin về địa chỉ để lưu vào CSDL.

- Nếu việc lưu thất bại, một thông báo lỗi sẽ được chuẩn bị hiển thị cho người dùng. Người dùng có thể cập nhật thông tin địa chỉ của quán sau.
- Tiếp theo, hệ thống sẽ lưu các địa chỉ liên kết quán. Các địa chỉ liên kết này bao gồm: một địa chỉ trang web chính thức, *fan page* trên Facebook, tài khoản Twitter và Google+. Mục tiêu của việc lưu những địa chỉ này là cho nhu cầu mở rộng sau này, nhóm phát triển sẽ cho phép người dùng đồng bộ thông tin về quán của họ trên tất cả các kênh thông tin mà họ đăng ký với hệ thống của nhóm.
- Nếu việc lưu các địa chỉ liên kết thất bại, một thông báo lỗi sẽ được chuẩn bị hiển thị cho người dùng. Người dùng có thể cập nhật thông tin này sau.
- Hệ thống sẽ chuyển người dùng tới trang xem chi tiết quán mà họ vừa tạo. Hệ thống hiển thị thông báo tạo quán thành công cùng các thông báo lỗi (nếu có, do quá trình lưu địa chỉ quán hay các đường liên kết thất bại).

4.2.2. Chia sẻ hình ảnh về một quán cà phê hay dịch vụ (Share photo):

Quá trình từ lúc người dùng tải (*upload*) ảnh lên hệ thống được mô tả bằng sơ đồ trình tự (*sequence diagram*) như sau:



Hình 26: chia sẻ quán cà phê, dịch vụ

Lưu ý, sơ đồ trên mô tả quá trình chọn một hình đại diện cho quán khi đang chia sẻ quán mới. Đối với quá trình cập nhật thông tin quán, quy trình cập nhật hình đại diện cũng gần tương tự như trên. Chỉ khác tác nhân không phải là thành viên thường mà là thành viên quản lý quán, chủ quán hay thành viên quản trị.

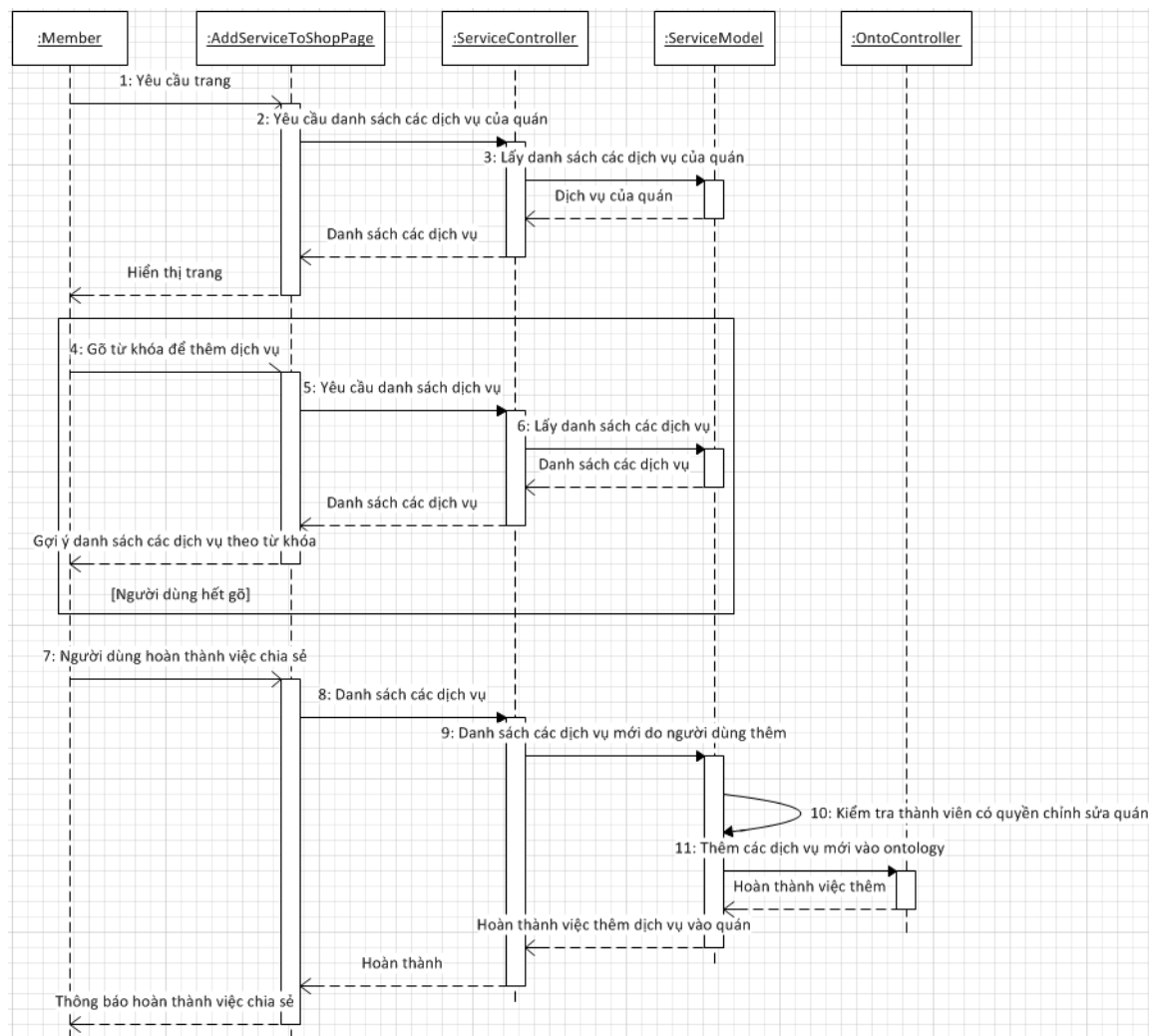
Mô tả:

- Người dùng có nhu cầu chọn hình ảnh đại diện cho quán. Hệ thống sẽ tạo động một *form* để người dùng chọn hình ảnh.
- Người dùng chấp nhận (*submit*) tải ảnh, hệ thống sẽ gửi một yêu cầu động (dùng kĩ thuật *AJAX*) để tải ảnh lên hệ thống.
- Nếu quá trình tải ảnh lên máy chủ của hệ thống thành công, hệ thống sẽ gọi một API để tải ảnh lên tài khoản Facebook cá nhân của người dùng (Xem phần Phụ lục để biết cách thực hiện API này).
- Nếu quá trình tải ảnh lên Facebook thành công, hệ thống sẽ nhận được thông tin về đối tượng ảnh mới trên Facebook. Hệ thống cập nhật những thông tin này vào CSDL và xóa phiên bản của ảnh này trên máy chủ hệ thống đi.

- Nếu quá trình tải lên Facebook thất bại, phiên bản tại máy chủ sẽ được giữ lại.
- Hệ thống tự động hiển thị hình ảnh vừa tải ngay tại trang chia sẻ quán.

4.2.3. Thêm một dịch vụ vào danh sách các dịch vụ do quán phục vụ (Share service):

Quá trình được mô tả bằng sơ đồ trình tự (*sequence diagram*) như sau:



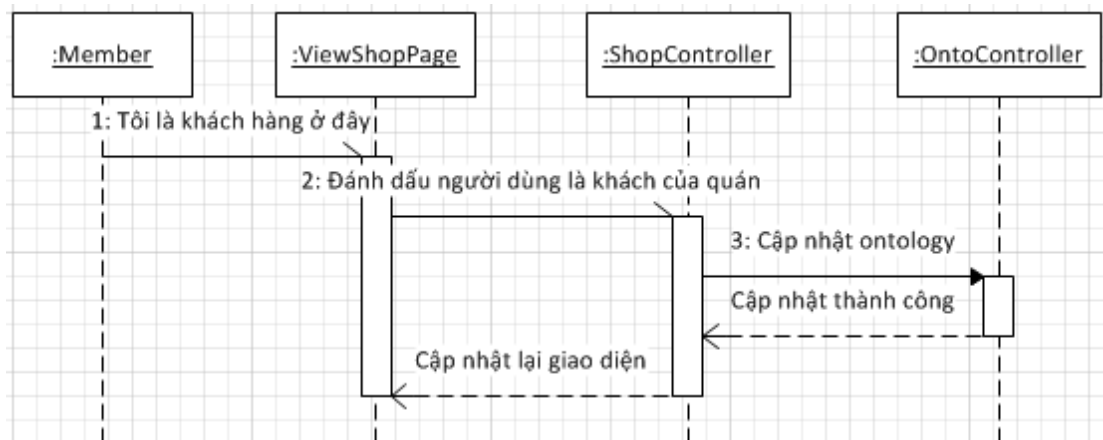
Hình 27: thêm dịch vụ vào quán

Mô tả:

- Khi người dùng muốn chia sẻ những dịch vụ mà một quán có phục vụ. Hệ thống sẽ lấy danh sách những dịch vụ hiện tại của quán để hiển thị lên để giúp người dùng biết cần thêm những dịch vụ gì.

- Người dùng gõ tên dịch vụ muốn chia sẻ vào. Trong quá trình gõ, hệ thống sẽ đưa ra danh sách những dịch vụ trùng với từ khóa của người dùng.
- Người dùng bấm vào dịch vụ cần chia sẻ để thêm vào danh sách các dịch vụ đang có của quán.
- Người dùng bấm “Lưu” để hoàn tất.
- Hệ thống lọc ra những dịch vụ mới để thêm vào danh sách dịch vụ của quán.
- Nếu người dùng là thành viên của quán, chủ quán hay thành viên quản trị, các dịch vụ sẽ được thêm vào danh sách ngay. Nếu người dùng không thuộc các đối tượng trên, các dịch vụ do người dùng thêm vào sẽ cần được một thành viên thuộc một trong những đối tượng trên chấp nhận để vào danh sách.
- Thông báo việc hoàn thành chia sẻ các dịch vụ. Nếu các dịch vụ chưa được thêm ngay vào danh sách dịch vụ, thông báo cho người dùng biết.

4.2.4. Xác nhận là khách hàng của một quán (Is customer):

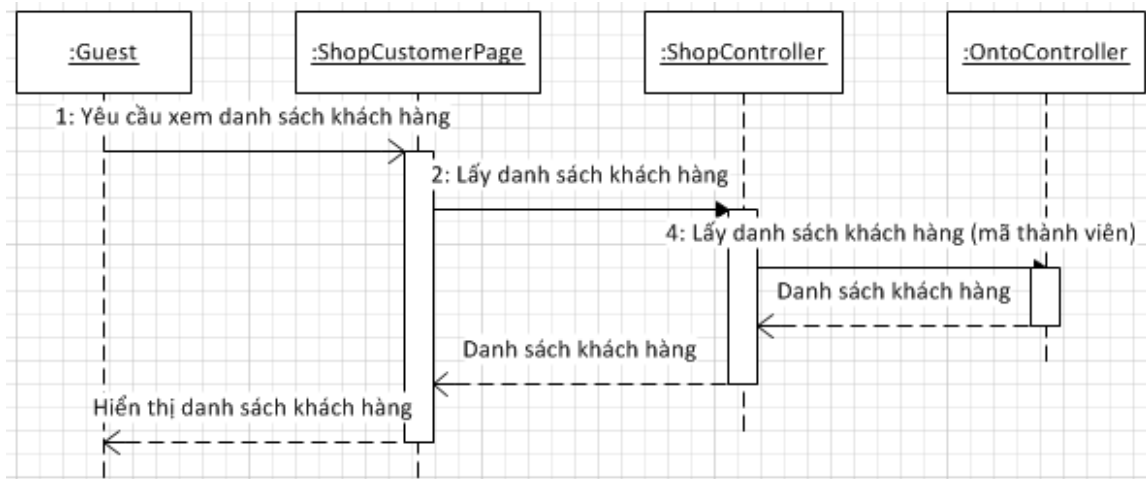


Hình 28: xác nhận là khách hàng

Mô tả:

- Để xác nhận mình là khách hàng tại một quán, người dùng vào trang xem chi tiết quán đó và bấm “*You were here*”. Ngoài ra, người dùng còn có thể xác nhận cụ thể hơn rằng là một khách hàng mới hay khách quen, khách VIP.

4.2.5. Xem danh sách những khách hàng của một quán cà phê (View shop's customers):

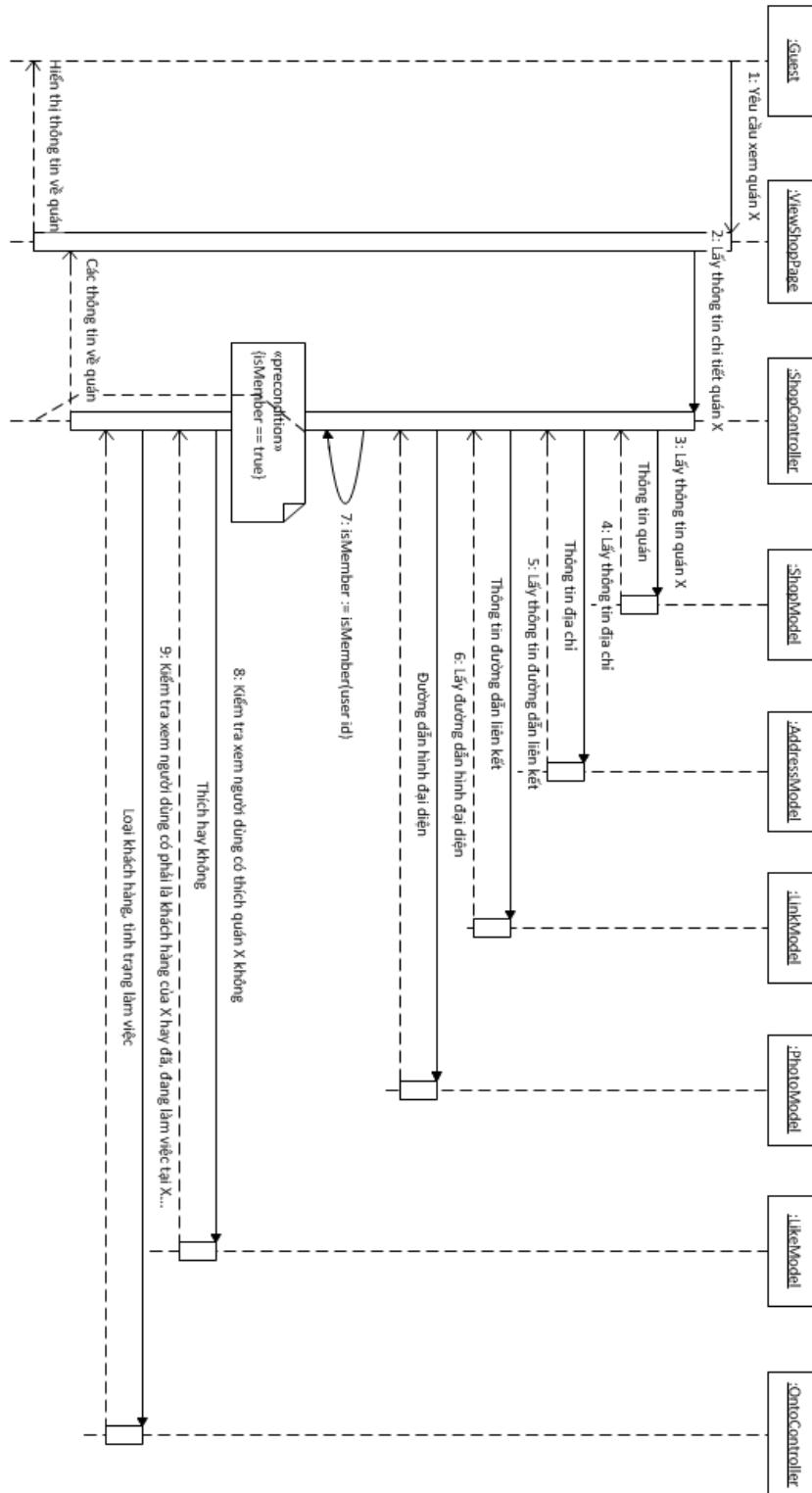


Hình 29: khách hàng của quán

Mô tả:

- Người dùng yêu cầu xem danh sách khách hàng của một quán.
- Nếu người dùng đang xem danh sách này trong trang xem chi tiết quán hay ngay trong hồ sơ cá nhân (*profile*) của mình (người dùng là thành viên) thì mã thành viên sẽ là mã của người dùng.
- Nếu người dùng đang xem mối liên hệ giữa một người dùng khác với quán, mã thành viên sẽ là mã của người dùng đó.
- Nếu người dùng là khách, mã thành viên sẽ được gán bằng rỗng (*null*).
- Nếu mã thành viên khác rỗng, hệ thống sẽ tiến hành chọn ra những khách hàng của quán mà là bạn của thành viên đó trước. Sau đó, hệ thống mới lấy toàn bộ danh sách khách hàng. Kết quả trả về là danh sách được trộn giữa hai danh sách này, với hình ảnh của thành viên đó ở đầu danh sách (nếu thành viên đó là khách hàng).
- Nếu mã thành viên bằng rỗng, hệ thống sẽ trả về kết quả là danh sách tất cả các khách hàng của quán.
- Tùy vào ngữ cảnh mà người dùng sẽ được thấy một phần của danh sách hay toàn bộ danh sách.

4.2.6. Xem chi tiết một quán cà phê (View shop):



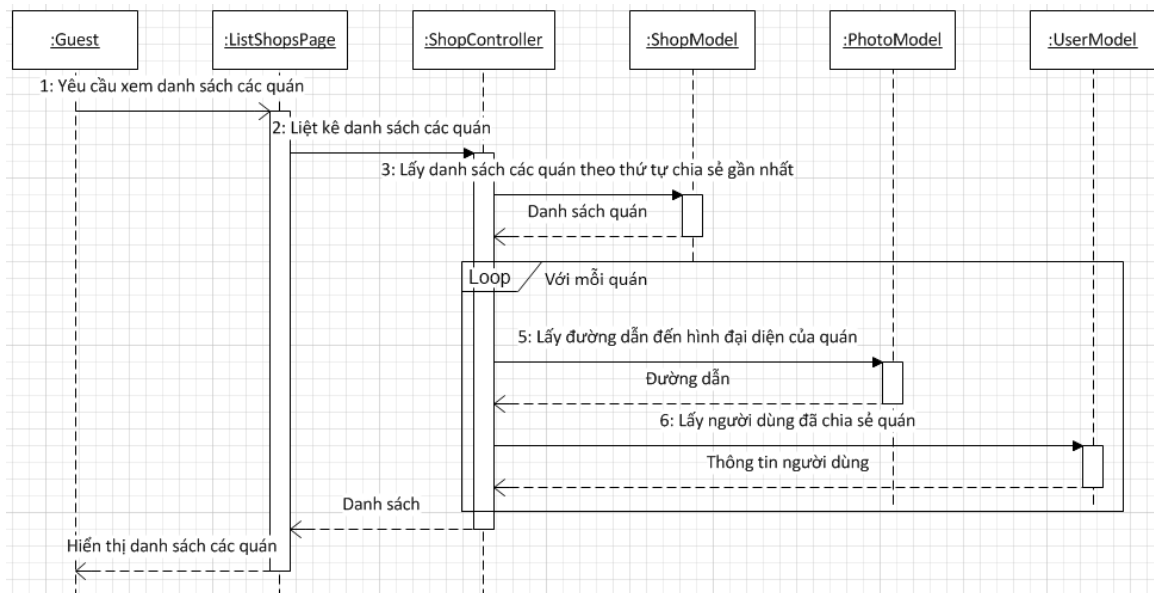
Hình 30: xem chi tiết quán

Mô tả:

- Người dùng yêu cầu xem thông tin quán cà phê X.
- Hệ thống lấy các thông tin của quán X bao gồm:
 - o Các thông tin cơ bản: tên quán, mô tả...

- Các thông tin thống kê: số lượng người dùng thích, số lượt xem...
- Các thông tin về địa chỉ.
- Các thông tin về đường dẫn liên kết.
- Nếu người dùng là thành viên, hệ thống sẽ kiểm tra xem người dùng đã thích quán hay chưa, đã là khách hàng của quán hay chưa... Người dùng có thể thay đổi các thông tin này trên trang này.
- Hệ thống dùng các use case *Get comments*, *Get shop's services*, *Get likes*, *Get shop's customers* để hiển thị thêm thông tin về quán.
- Hệ thống dùng use case *Recommend shops* để đưa ra gợi ý về những quán cà phê mà người dùng nên đến thử.

4.2.7. Xem danh sách các quán cà phê được chia sẻ theo thứ tự gần nhất (List shops):



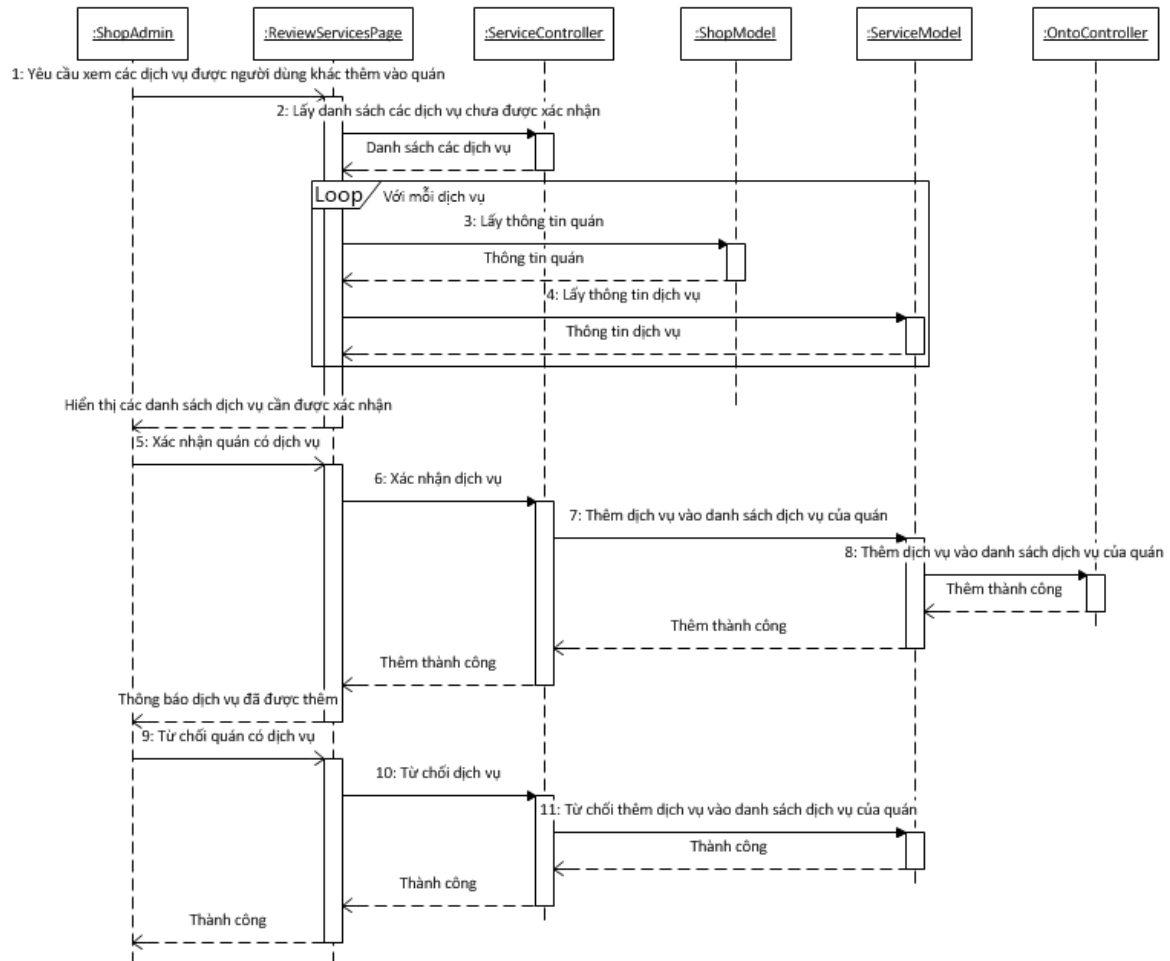
Hình 31: quán được chia sẻ gần nhất

Mô tả:

- Người dùng yêu cầu xem danh sách các quán.
- Hệ thống lấy ra danh sách tất cả các quán, sắp xếp theo thứ tự chia sẻ gần nhất. Mỗi lần chỉ lấy một số lượng nhất định (do thành viên quản trị tùy chỉnh).
- Với mỗi quán, lấy đường dẫn (*link*) đến hình đại diện của quán. Lấy thông tin về người dùng chia sẻ quán này.

- Hiển thị danh sách trên ra. Hệ thống hiển thị thêm một nút “Xem thêm” để người dùng xem được thêm nhiều quán nữa.

4.2.8. Xác nhận dịch vụ của quán (Review shop's services):

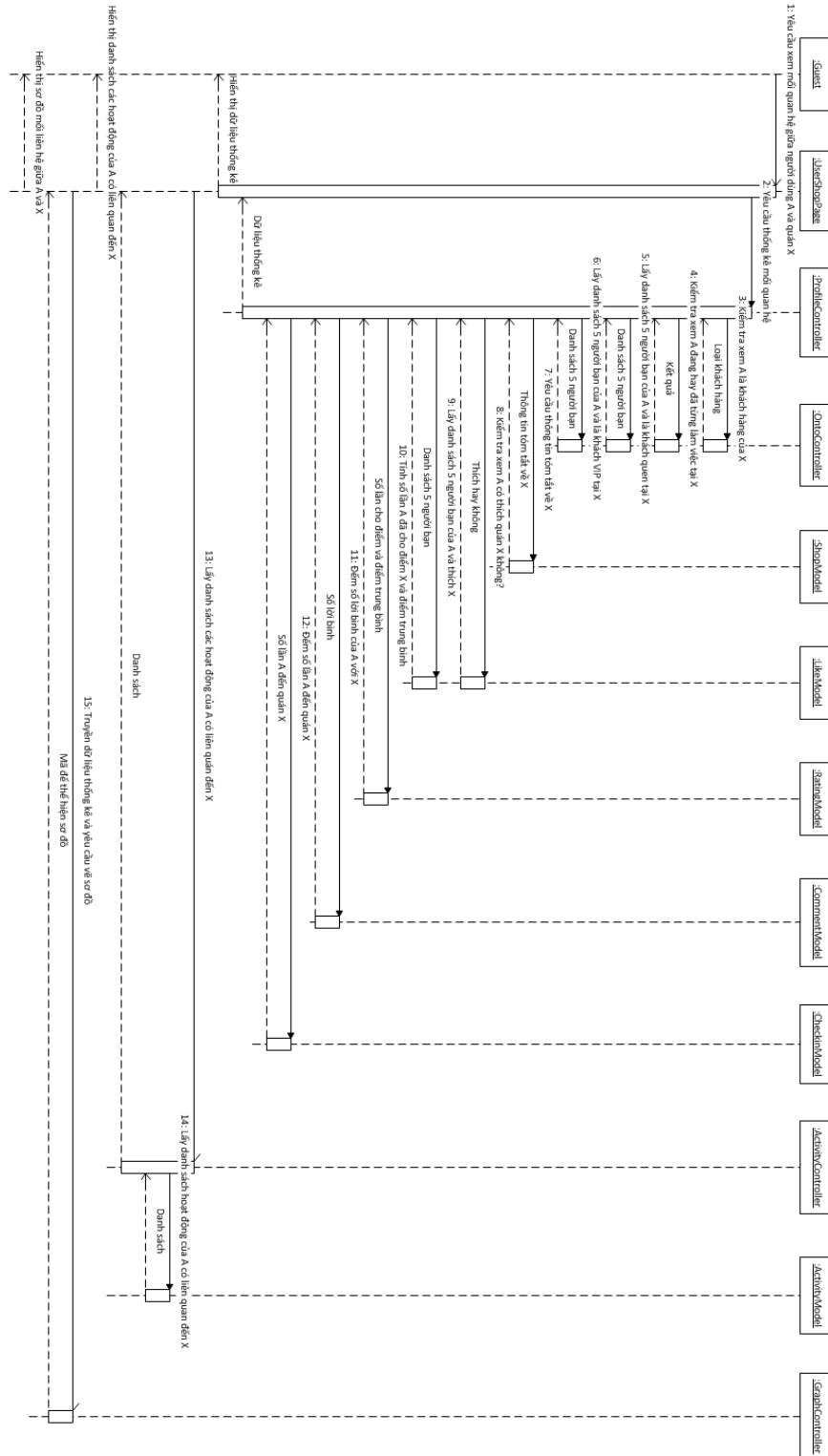


Hình 32: xác nhận dịch vụ của quán

Mô tả:

- Người dùng yêu cầu kiểm tra danh sách các dịch vụ do những người dùng khác thêm vào các quán mà họ có quyền quản lý (là thành viên quản lý quán hay chủ quán hay là thành viên quản trị).
- Hệ thống lọc ra danh sách các quán mà họ có quyền quản lý. Lấy ra danh sách các dịch vụ chưa được xác nhận của các quán này.
- Nếu người dùng chấp nhận thêm một dịch vụ vào một quán. Cập nhật trạng thái của dịch vụ này vào CSDL và ontology.
- Nếu người dùng từ chối thêm một dịch vụ vào quán. Cập nhật trạng thái của dịch vụ này vào CSDL.

4.2.9. Xem mối liên hệ giữa người dùng và quán (User-shop relationship):



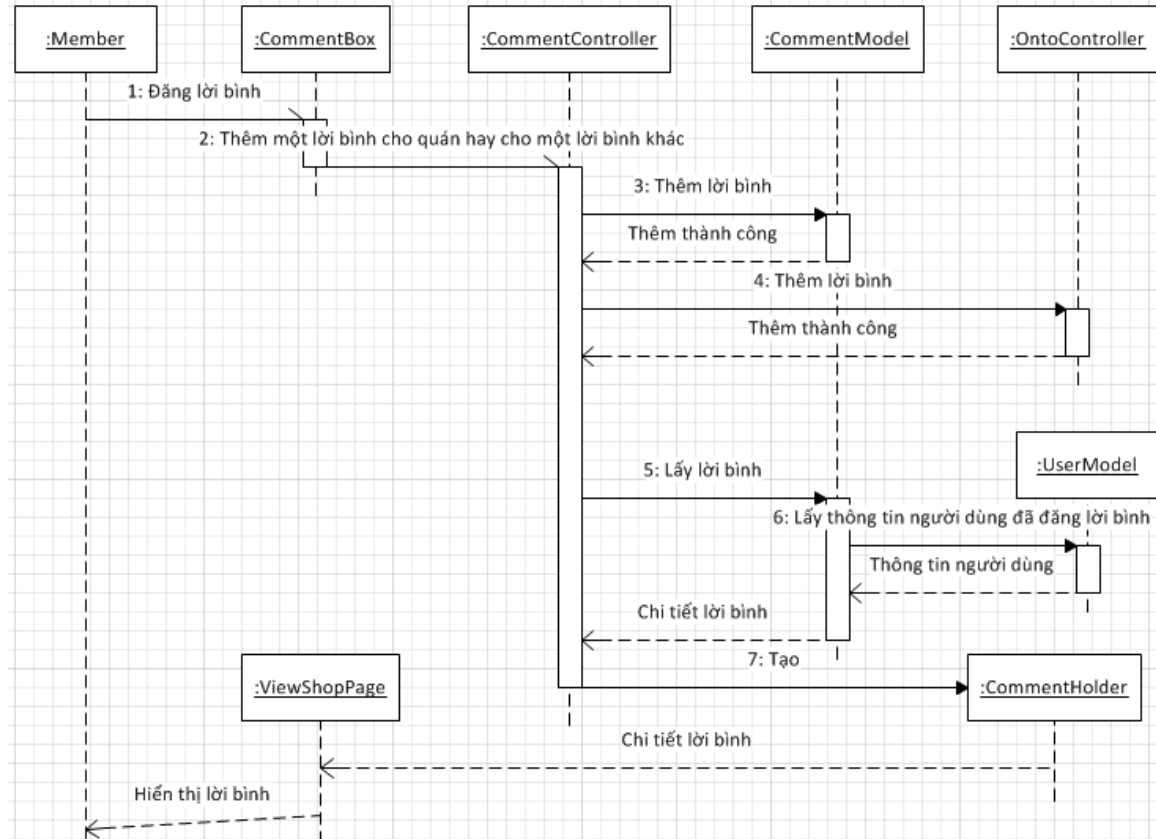
Hình 33: mối quan hệ người dùng và quán

Mô tả:

- Người dùng yêu cầu được xem mối quan hệ giữa thành viên A và quán cà phê X.

- Hệ thống sẽ thống kê mối quan hệ này bằng cách trích ra những thông tin sau:
 - Thống kê bản thân A và X:
 - A có là khách hàng của X hay không? Khách hàng loại gì (khách mới, khách quen, khách VIP)?
 - A có đang làm việc hay đã làm việc tại X? Hay A là chủ quán X?
 - A có thích quán X không?
 - A đã từng cho điểm (*rate*) quán X mấy lần? Và điểm trung bình của những lần đó là bao nhiêu?
 - A đã từng đến (*check-in*) quán X bao nhiêu lần?
 - A đã từng đưa ra bao nhiêu lời bình (*comment*) cho quán X?
 - Những người bạn của A và quán X:
 - Lấy danh sách 5 người bạn của A là khách quen tại quán X.
 - Lấy danh sách 5 người bạn của A là khách VIP tại quán X.
 - Đếm số lượng những người bạn của A thích X.
- Hệ thống lấy những thông tin tóm tắt về quán X như tên quán, hình đại diện, mô tả ngắn gọn... để kết hợp cùng với các thông tin trên để hiển thị cho người dùng.
- Hệ thống lấy động danh sách các hoạt động (*activity*) của người dùng A có liên quan đến quán X để hiển thị.
- Hệ thống dùng dữ liệu thống kê như trên để tạo động một sơ đồ mô tả mối quan hệ A và X.

4.2.10. Bình luận (Comment):

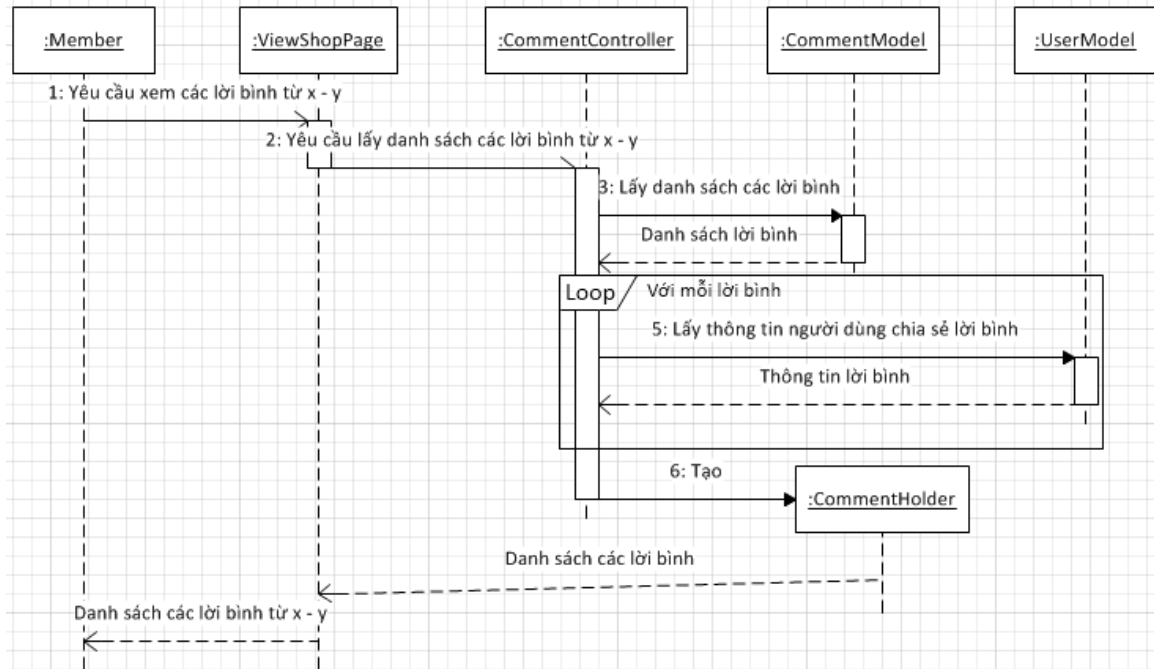


Hình 34: Bình luận

Mô tả:

- Người dùng đánh nội dung lời bình (*comment*) vào ô lời bình (*comment box*). Người dùng bấm “Enter” để đăng lời bình. Người dùng có thể bình luận cho một quán hay cho một lời bình khác.
- Hệ thống thêm lời bình vào CSDL và ontology. Hệ thống thêm động lời bình đó vào trang mà người dùng vừa nhập lời bình. Trong hình lấy ví dụ là trang xem chi tiết quán.

4.2.11. Xem danh sách các lời bình (Get comments):

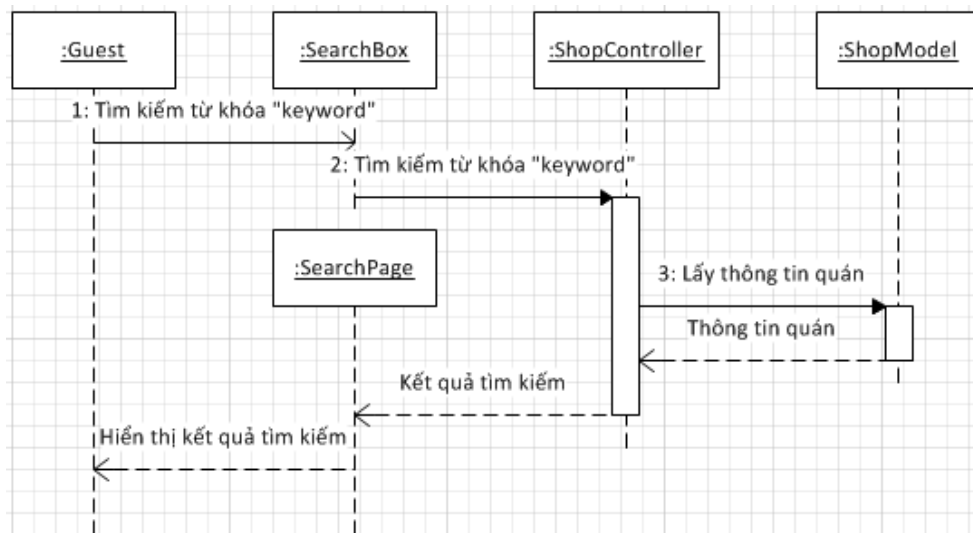


Hình 35: lời mời cho một quán cà phê

Mô tả:

- Người dùng yêu cầu xem danh sách những lời bình từ vị trí thứ x đến y. Mặc định, hệ thống sẽ cho người dùng xem $(y - x + 1)$ lời bình mới nhất. Người dùng có thể bấm xem thêm để xem tiếp những lời bình trước đó.

4.2.12. Tìm kiếm quán cà phê (Search shops):

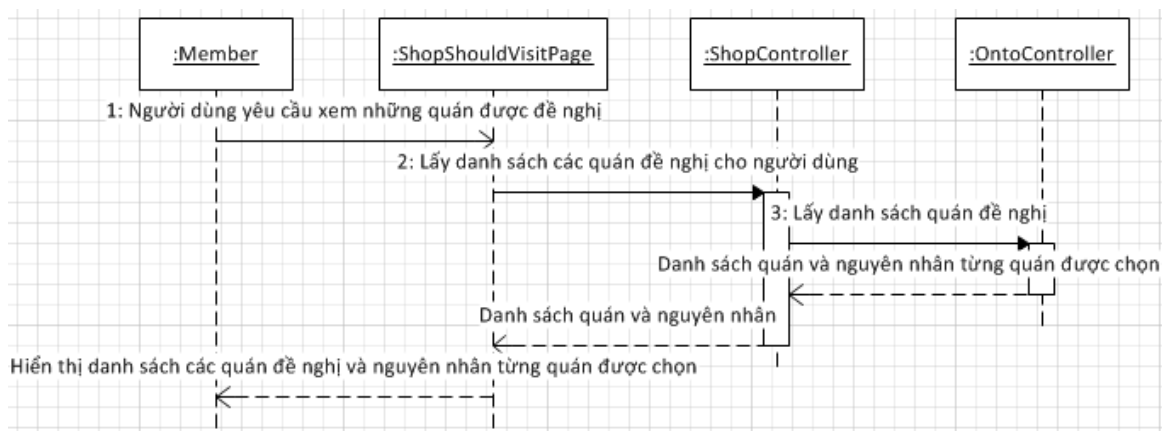


Hình 36: tìm kiếm quán cà phê

Mô tả:

- Người dùng tìm kiếm với từ khóa “*keyword*”. Hệ thống sẽ đưa ra các kết quả tìm kiếm theo các ưu tiên sau:
 - Quán có một trong những thuộc tính đúng bằng *keyword*.
 - Quán có một trong những thuộc tính có dạng “*keyword %*”.
 - Quán có một trong những thuộc tính có dạng “*% keyword %*”.
 - Quán có một trong những thuộc tính có dạng “*keyword%*”.
 - Quán có một trong những thuộc tính có dạng “*%keyword%*”.

4.2.13. Đề nghị những quán nên đến (Recommend shops):



Hình 37: đề nghị quán nên đến

Mô tả:

- Khi người dùng yêu cầu được xem những quán mà hệ thống đề nghị họ đến. Hệ thống sẽ truy vấn từ cơ sở tri thức và trả về kết quả cùng nguyên nhân cho người dùng.
- Các quán cả phê được chọn lần lượt theo bảy tiêu chí. Các quán thỏa các tiêu chí đầu sẽ được chọn trước.
- Loại bỏ những quán mà người dùng đã đi để ra được danh sách những quán thỏa một tiêu chí.
- Nếu số lượng quán thỏa đã đủ số quán cần (do thành viên quản trị tùy chỉnh) thì ngưng truy vấn. Nếu chưa thì tiếp tục truy vấn theo tiêu chí tiếp theo. Các tiêu chí được sắp theo độ ưu tiên giảm dần lần lượt là:
 - Giả sử A là người dùng đang được tư vấn. B là bạn của A. Các quán C được chọn theo tiêu chí này phải thỏa:

- C phục vụ dịch vụ S.
- A và B cùng thích dịch vụ S.
- B là khách hàng của C.

Câu truy vấn dùng để tìm kiếm những tiêu chí này như sau:

```
SELECT distinct ?shop ?serves ?friends
where
{
    ?person rdf:type:Person;
    :likes ?likes;
    :friendOf ?friends.
    filter(?person=:A)
    ?shop rdf:type:CoffeeShop.
    ?shop :serves ?serves.
    Filter(?likes = ?serves).
    ?friends :likes ?fLikes;
    :customerOf ?fShops.
    filter(?serves=?fLikes).
    filter(?shop = ?fShops)
}
```

○ Các quán thỏa tiêu chí 2 thì cần đáp ứng:

- Quán C phục vụ dịch vụ S.
- A thích S.
- B thích C.

Truy vấn:

```
SELECT distinct ?shop ?serves ?friends
where
{
    ?person rdf:type:Person;
    :likes ?likes;
    :friendOf ?friends.
    filter(?person=:A)
    ?shop rdf:type:CoffeeShop.
    ?shop :serves ?serves.
    Filter(?likes = ?serves).
    ?friends :likes ?fLikes.
    filter(?shop = ?fLikes)
}
```

- Quán C thỏa tiêu chí 3 sẽ cần đáp ứng:
 - C phục vụ dịch vụ S.
 - A thích dịch vụ S.
 - B đánh giá cao (từ 4 sao trở lên) quán C.

Ta truy vấn như sau để tìm được quán C:

```
SELECT distinct ?shop ?serves ?friends ?point
where
{
    ?person rdf:type:Person;
    :likes ?likes;
    :friendOf ?friends.
    filter(?person=:A)
    ?shop rdf:type:CoffeeShop.
    ?shop :serves ?serves.
    Filter(?likes = ?serves).
    ?rating rdf:type:Rating;
    :ratePoint ?point;
    :ratedBy ?ratedBy;
    :ratesFor ?ratesFor.
    FILTER(?point >= 4 || ?point >= '4')
    FILTER(?ratedBy = ?friends)
    FILTER(?shop = ?ratesFor)
}
```

- Các quán thỏa tiêu chí 4 chỉ cần thỏa điều kiện dễ hơn:
 - Quán C phục vụ dịch vụ S.
 - A thích dịch vụ S.
 - B là khách hàng của C.

Để lọc ra những quán thỏa điều kiện này, ta truy vấn như sau:

```
SELECT distinct ?shop ?serves ?friends
Where
{
    ?person rdf:type:Person;
    :likes ?likes;
    :friendOf ?friends.
    filter(?person=:A)
    ?shop rdf:type:CoffeeShop.
    ?shop :serves ?serves.
```

```

Filter(?likes = ?serves).
?friends :customerOf ?fShops.
filter(?shop = ?fShops)
}

```

- Nếu B thích quán C thì quán C thỏa tiêu chí 5. Câu truy vấn sau sẽ tìm ra được những quán thỏa tiêu chí này:

```

SELECT distinct ?shop ?friends
where
{
    ?person rdf:type:Person;
    :friendOf ?friends.
    filter(?person=:A)
    ?friends :likes ?fLikes.
    ?shop rdf:type:CoffeeShop.
    FILTER (?shop = ?fLikes)
}

orderby ASC(?shop )

```

- Nếu B đánh giá cao quán C (từ 4 sao trở lên) thì quán C thỏa tiêu chí 6. Ta truy vấn như sau:

```

SELECT distinct ?shop ?friends ?point
where
{
    ?person rdf:type:Person;
    :friendOf ?friends.
    filter(?person=:A)
    ?rating rdf:type:Rating;
    :ratePoint ?point;
    :ratedBy ?ratedBy;
    :ratesFor ?ratesFor.
    FILTER(?ratedBy = ?friends)
    ?shop rdf:type:CoffeeShop.
    FILTER(?shop = ?ratesFor)
    FILTER (?point >= '4' || ?point >= 4 )
}

```

- Những quán mà B đã ghé thăm sẽ thỏa tiêu chí 7:

```

SELECT distinct ?shop ?friends
where
{

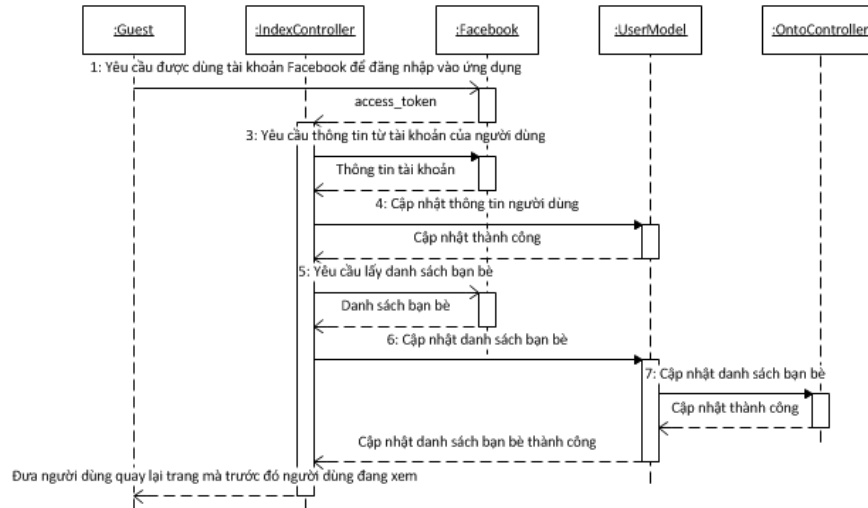
```

```

?person rdf:type:Person;
:friendOf ?friends.
filter(?person=:A)
?friends :customerOf ?fcustomerOf.
?shop rdf:type:CoffeeShop.
FILTER (?shop = ?fcustomerOf)
}

```

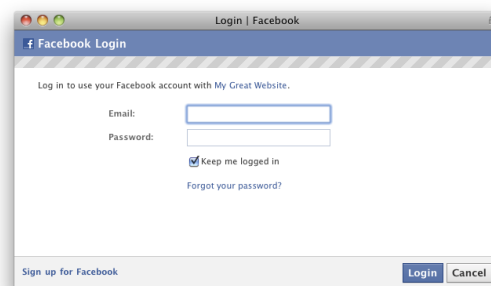
4.2.14. Đăng nhập (Log in):



Hình 38: đăng nhập

Mô tả:

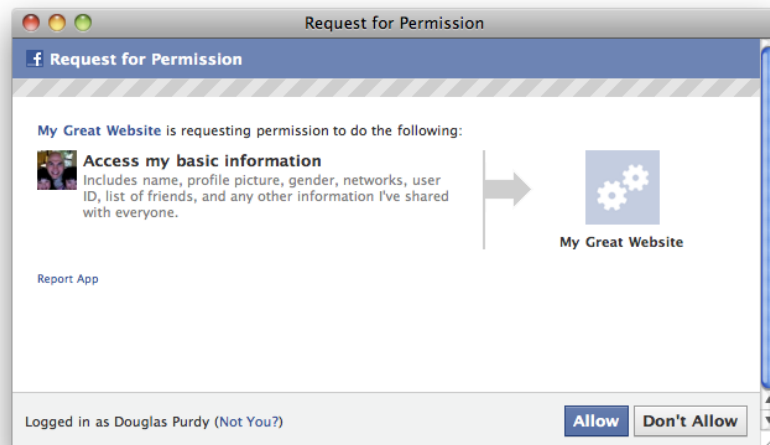
- Người dùng bấm vào “Join us with Facebook” để bắt đầu việc yêu cầu Facebook cho phép người dùng dùng tài khoản cá nhân của mình để đăng nhập vào ứng dụng.
- Những bước tiếp theo do Facebook xử lý. Tóm tắt các bước do Facebook xử lý như sau:
 - o Kiểm tra (*authenticate*) tài khoản người dùng. Nếu người dùng đã đăng nhập, tới bước tiếp theo. Nếu chưa, hiện ra hộp thoại:



Hình 39: kiểm tra đăng nhập

Trong đó, *My Great Website* sẽ được thay thế bằng tên ứng dụng mà hệ thống đăng ký với Facebook, *Coffee Go*.

- Tiếp theo, Facebook sẽ hiển thị một hộp thoại để yêu cầu người dùng chấp nhận những quyền mà ứng dụng yêu cầu. Nếu người dùng, trước đó, đã chấp nhận tất cả các quyền, hộp thoại này sẽ được bỏ qua.

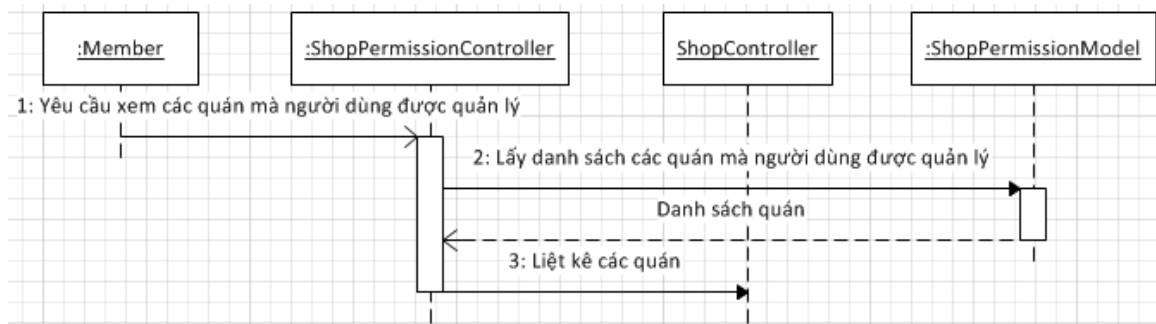


Hình 40: xác nhận đăng nhập

- Nếu người dùng chọn “*Don't Allow*”, Facebook sẽ không cho phép ứng dụng dùng tài khoản của người dùng để đăng nhập hệ thống. Nếu người dùng chọn “*Allow*”, Facebook sẽ trả về cho hệ thống một chuỗi gọi là *access_token* và thời gian chuỗi này hết hiệu lực.
- Chuỗi *access_token* sẽ là chuỗi mà hệ thống sẽ phải dùng phần lớn trong các hàm gọi Facebook để lấy dữ liệu từ tài khoản người dùng. Lưu ý, những dữ liệu được phép lấy chỉ nằm trong khuôn khổ những quyền mà người dùng đã cấp cho ứng dụng.
- Nếu nhận được một chuỗi *access_token* hợp lệ, hệ thống sẽ gửi một yêu cầu cho Facebook để lấy danh sách bạn bè của người dùng. Khi nhận được danh sách này, hệ thống sẽ cập nhật vào ontology.

Để có thêm thông tin về quá trình này, các bạn có thể ghé thăm trang [Facebook Authentication](#). Phần Phụ lục của báo cáo cũng hướng dẫn các bạn các thao tác căn bản để giao tiếp với Facebook.

4.2.15. Xem danh sách các quán mà người dùng có quyền chỉnh sửa (Manage shops):

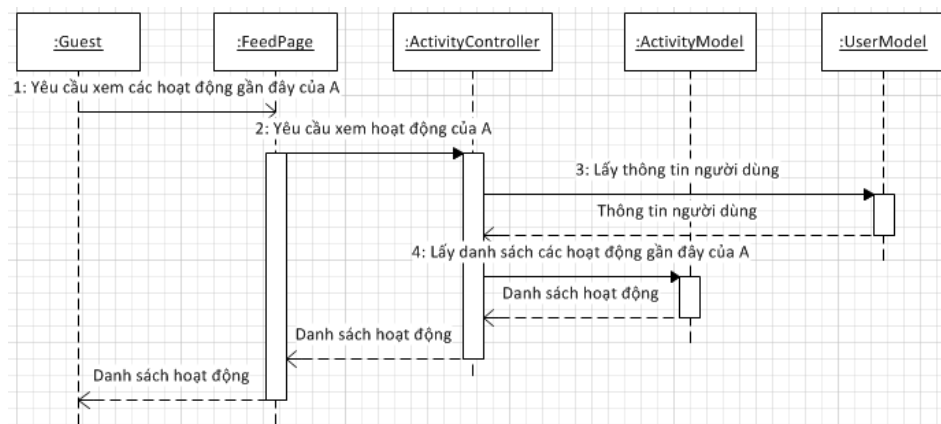


Hình 41: quán có quyền chỉnh sửa

Mô tả:

- Người dùng yêu cầu xem các quán mà người dùng có quyền quản lý (người dùng là thành viên quản lý của quán, chủ quán hay là thành viên quản trị).
- Hệ thống lấy ra danh sách các quán thỏa điều kiện rồi dùng use case *List shops* để thể hiện ra danh sách cho người dùng xem.

4.2.16. Xem danh sách các hoạt động gần đây của người dùng (User's feeds):

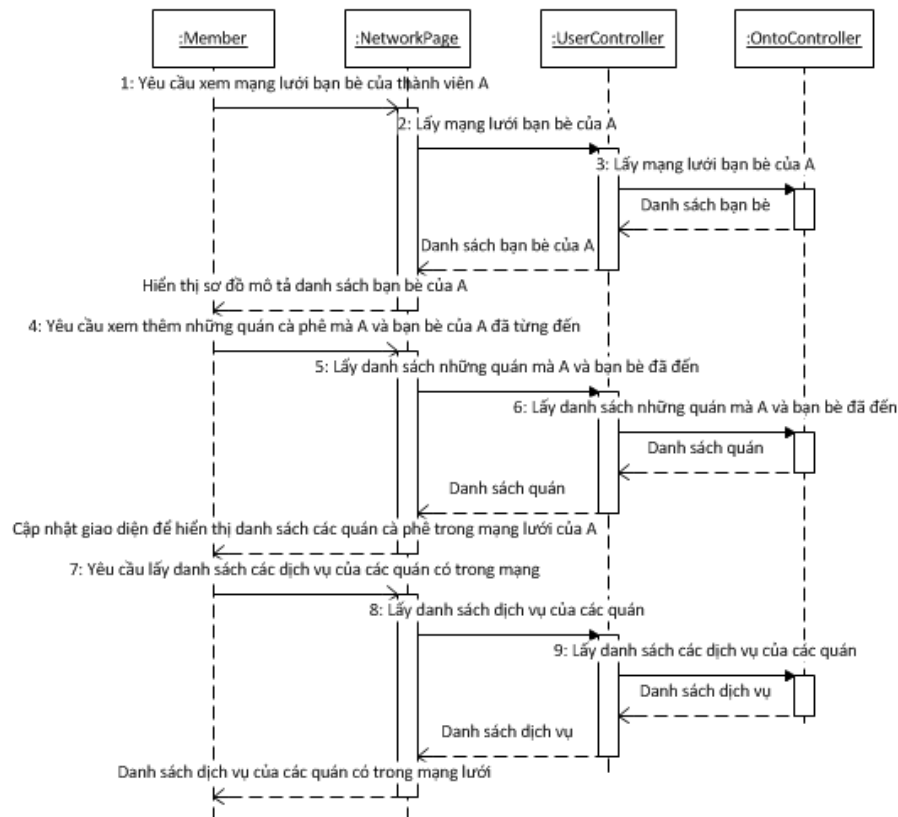


Hình 42: hoạt động gần đây của người dùng

Mô tả:

- Người dùng yêu cầu xem danh sách các hoạt động (*activity*) gần đây của thành viên A. Hệ thống sẽ lấy ra danh sách các hoạt động của A và xếp theo thứ tự mới nhất.

4.2.17. Xem mạng lưới mối quan hệ của người dùng (User-shop relationship):



Hình 43: mạng lưới liên kết

Mô tả:

- Người dùng yêu cầu được xem mạng lưới (*network*) của thành viên A. Hệ thống sẽ hiển thị mạng lưới A và những người bạn của A.

```
SELECT distinct ?friends ?FOF
```

```
where
```

```
{
```

```
    ?person rdf:type:Person;
```

```
    :friendOf ?friends.
```

```
    FILTER(?person=:person)
```

```
    ?friends :friendOf ?FOF.
```

```
}
```

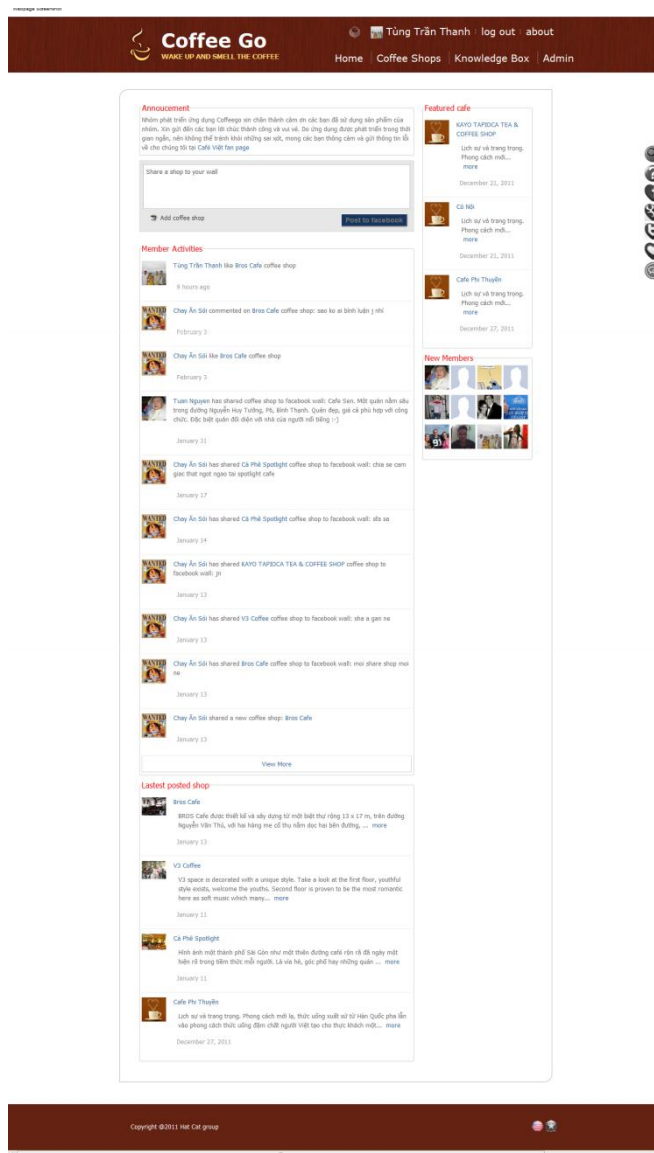
```
order by desc(?friends)
```

- Người dùng có thể yêu cầu xem thêm danh sách các quán cà phê mà A và những người bạn đã đến.
- Người dùng cũng có thể yêu cầu xem thêm danh sách các dịch vụ của các quán cà phê trên.

4.3. Hiện thực:

Phần này sẽ liệt kê một số giao diện chính và nổi bật của chương trình.

4.3.1. Trang chủ:



Hình 44: Giao diện trang chủ

Trang chủ cho phép người dùng:

- Chia sẻ một quán yêu thích của họ lên tường (wall) Facebook.
- Xem danh sách những hoạt động mới nhất của cộng đồng.
- Xem danh sách những quán được chia sẻ gần nhất.
- Xem danh sách những thành viên mới nhất.
- Phần tiêu đề cho phép người dùng xem hồ sơ cá nhân của mình, đăng xuất (hay đăng nhập)...

- Phần nút bên cạnh cho phép người dùng tìm kiếm, chia sẻ trang lên Facebook, Google+, Twitter, đồng thời cung cấp sự hướng dẫn cho người dùng. Trang cho phép người dùng gửi phản hồi.
- Phần cuối chứa dấu hiệu bản quyền của nhóm thực hiện. Người dùng cũng có thể thay đổi ngôn ngữ hiển thị tại đây.

4.3.2. Trang chia sẻ một quán cà phê mới:

Café bình minh

? likes - Average rating: 4.3 - ? people were here - ? views

[Edit summary](#)



Café bình minh chỉ mở cửa từ 3h30 sáng cho đến 8h sáng. Café nằm tại trên sân thượng (tầng 68) tòa nhà Bitexco đảm bảo mang lại cho bạn một cảm giác không thể quên được khi ngắm bình minh. Còn gì tuyệt hơn khi được nhâm nhi tách cà phê thơm ngon bên bạn bè và ngắm cảnh cả thành phố vươn mình trong ánh bình minh?

Hãy đến với Café bình minh để cảm giác trải nghiệm tuyệt vời này.

Choose an image

Không...hơn

Pick an avatar for this shop.

Title

Give this photo a title

Description

Say something about this

Format

Insert

B

I

S

x₂

AaBbCc

Paragraph

AaBb

Heading 1

AaBb

Heading 2

AaBb

Heading 3

Address info	
Number	45
Street	Ngô Đức Kế
Ward	Bến Nghé
District	1
City	Ho Chi Minh
Country	Viet Nam
Full address	45 <u>Ngô Đức Kế street</u> , Ben Nghe ward, <u>district 1</u> , Ho Chi Minh City, Vietnam
Style	Terrace coffee shop
Hours	3h30 AM - 8h00 AM
Off days	
Delivery	<input checked="" type="checkbox"/>
Reservation	<input checked="" type="checkbox"/>
Children	
Phone	
Fax	

<http://www.facebook.com/Bitexco>

Hình 45: Giao diện trang chia sẻ quán.

Trang chia sẻ quán sử dụng *Aloha Editor* cho phép người dùng có thể điền thông tin trong một bố cục sẽ xuất hiện đúng như khi xem quán. Điều này giúp người dùng khỏi phải xem trước như khi dùng các công cụ khác.

Aloha editor còn có một đặc điểm nổi bật: dù trang có nhiều nơi cần nhập dữ liệu (*input*) thì cũng chỉ có một *instance* của Aloha editor được tạo ra. Điểm này khác biệt so với những công cụ khác như tinyMCE, ở mỗi ô nhập dữ


liệu phải cần một *instance* của công cụ đó. Do đó, *Aloha editor* giúp giảm thời gian tải trang.

Trang cho phép người dùng chọn hình ảnh đại diện cho quán. Hình này sau khi được tải lên (*upload*) xong thì sẽ hiện ra ngay tại trang. Điều này giúp cho người dùng dễ dàng chọn hình sao cho kích thước phù hợp với trang.

4.3.3. Trang xem chi tiết một quán cà phê:

Bros Cafe

2 likes - Average rating: 4.3 - 0 people were here - 43 views






BROS Cafe được thiết kế và xây dựng từ một biệt thự rộng 13 x 17 m, trên đường Nguyễn Văn Thủ, với hai hàng me cổ thụ nằm dọc hai bên đường, đoạn giữa Hai Bà Trưng và Phùng Khắc Khoan. Quán được thiết kế theo phong cách sang trọng và tươi mát. Không gian của BROS Cafe được chia làm 03 phần :

1. Không gian ngoài trời, có tầm nhìn ra hai hàng me cổ thụ bên đường, ngắm nhìn sinh hoạt của khu phố.
2. Không gian cây xanh và hồ nước với ghế mây, có máy điều hòa.
3. Không gian để làm việc văn phòng, chuyện trò thân mật với bạn bè, thưởng thức âm nhạc ... với ghế đệm rộng, thoải mái, có máy điều hòa.

BROS Cafe mở cửa từ 7.00 am tới 11.00 pm, phục vụ các loại thức uống đa dạng, điểm tâm sáng và các món ăn Âu Á với giá cả hợp lý.
BROS Cafe có 140 chỗ ngồi, là nơi cho mọi người hàn huyên, trao đổi tâm tình, làm việc ... Ngoài ra quán còn tổ chức phục vụ sinh nhật, tiệc gia đình, tiệc hội họp bạn bè, hội thảo ...

BROS Cafe có wifi miễn phí, sử dụng cáp quang. Với kinh nghiệm thực tế sử dụng wifi ở quán này, tốc độ truy cập cao gấp 4-5 lần so với đường truyền ADSL thông thường, tốc độ download có thể lên tới 1.2 M/sec.
Giữ xe miễn phí.

 Unlike
  You were here
  Recommend

Address info

Full address

Style **Office coffee shop**

Hours

Off days





Delivery ☒

Reservation ☒



Children

Phone


Fax


2 Likes

1 Customer



Services



White Chocolate Latte
Nắp dẫn thiết: chiều chiều dao chơi cùng bạn gái, rồi kết thúc buổi tối bên góc quán thưởng thức White Chocolate Latte thì rất tuyệt

Comment - Edit - Delete

☆☆☆☆☆

View 1 comment

Hình 46: Trang xem chi tiết một quán

Trang cho người dùng biết những thông tin về quán, những dịch vụ của quán, các lời bình của người dùng với quán...

4.3.4. Trang liệt kê danh sách các quán cà phê:

Coffee shops

Chay Ăn Sói shared a shop :



Bros Cafe

BROS Cafe được thiết kế và xây dựng từ một biệt thự rộng 13 x 17 m, trên đường Nguyễn Văn Thủ, với hai hàng me cổ thụ nằm dọc hai bên đường, đoạn giữa Hai Bà Trưng và Phùng Khắc Khoan. ... [more](#)

January 13

2 users like

[View 1 comment](#)

Chay Ăn Sói shared a shop :



V3 Coffee

V3 space is decorated with a unique style. Take a look at the first floor, youthful style exists, welcome the youths. Second floor is proven to be the most romantic here as soft music which many famous French, British love

songs belong to is perform

January 11

2 users like

[View 3 comments](#)

Hình 46: Một phần của trang liệt kê danh sách các quán

4.3.5. Trang hiển thị kết quả tìm kiếm:

Services
This feature will be coming soon!

Shops

Chay Ăn Sói shared a shop :

Bros Cafe
BROS Cafe được thiết kế và xây dựng từ một biệt thự rộng 13 x 17 m, trên đường Nguyễn Văn Thủ, với hai hàng me cổ thụ nằm dọc hai bên đường, đoạn giữa Hai Bà Trưng và Phùng Khắc Khoan. ...
[more](#)

January 13 - [Unlike](#) - [Comment](#) - [Customers](#) - [Services](#)
☆☆☆☆☆

You and 1 user likes
View 1 comment

Chay Ăn Sói shared a shop :

Cà Phê Spotlight
Hình ảnh một thành phố Sài Gòn như một thiên đường café rộng rãi đã ngày một hiện rõ trong tiềm thức mỗi người. Là vỉa hè, góc phố hay những quán café sang trọng, thú café như trở thành một...
[more](#)

January 11 - [Like](#) - [Comment](#) - [Customers](#) - [Services](#)
☆☆☆☆☆

Chay Ăn Sói shared a shop :

V3 Coffee
V3 space is decorated with a unique style. Take a look at the first floor, youthful style exists, welcome the youths. Second floor is proven to be the most romantic here as soft music which many famous French, British love songs belong to is perform

January 11 - [Like](#) - [Comment](#) - [Customers](#) - [Services](#)
☆☆☆☆☆

2 users like
View 3 comments

Hình 47: Hiển thị kết quả tìm kiếm với từ khóa "Br"

4.3.6. Hồ sơ cá nhân của người dùng:

Feeds
Shops
Visited shops
Liked shops
Rated shops
Network

Recent activities

Chay Ăn Sói Chay Ăn Sói commented on Bros Cafe coffee shop: sao ko ai bình luận j nhỉ
February 3

Chay Ăn Sói Chay Ăn Sói like Bros Cafe coffee shop
February 3

Chay Ăn Sói Chay Ăn Sói has shared Cà Phê Spotlight coffee shop to facebook wall: chia sẻ cảm giác thật ngọt ngào tại spotlight cafe
January 17

Chay Ăn Sói Chay Ăn Sói has shared Cà Phê Spotlight coffee shop to facebook wall: sfa
January 14

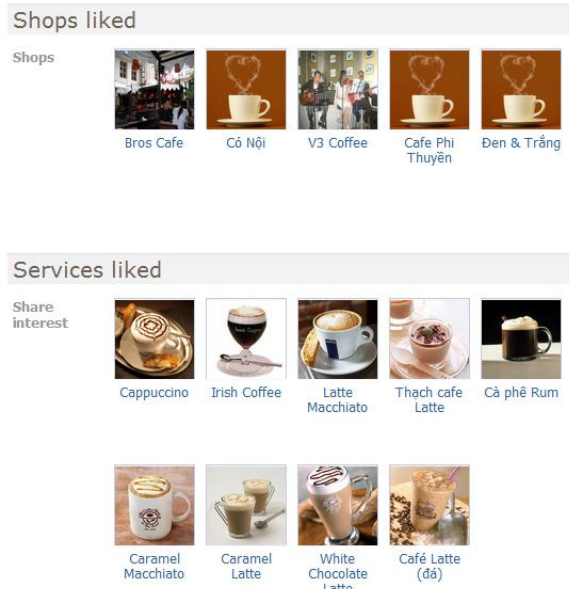
Chay Ăn Sói Chay Ăn Sói has shared KAYO TAPIOCA TEA & COFFEE SHOP coffee shop to facebook wall: jn
January 13

Hình 48: các hoạt động gần đây

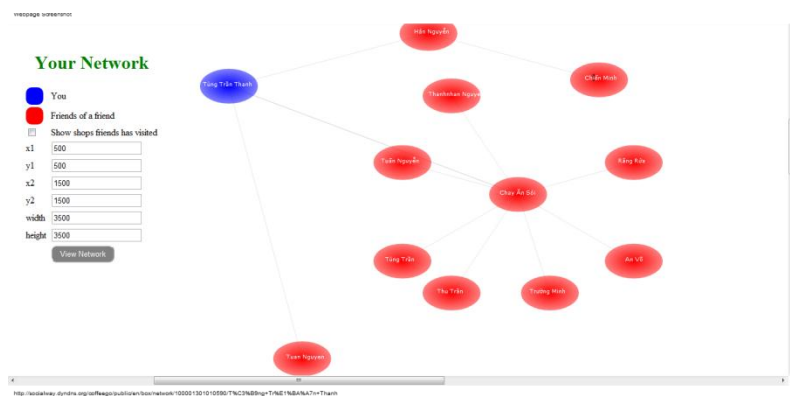
Shops have been visited

	Shop name	Shared by	
1	Bros Cafe	Chay Ăn Sói	
2	Cà Phê Spotlight	Chay Ăn Sói	
3	KAYO TAPIOCA TEA & COFFEE SHOP	Chay Ăn Sói	
4	Có Nội	Chay Ăn Sói	
5	V3 Coffee	Chay Ăn Sói	
6	Cafe Phi Thuyền	Chay Ăn Sói	
7	Penguin ice cream	Thanhnhân Nguyễn	

Hình 49: các quán đã đến



Hình 50: các quán và dịch vụ người dùng thích



Hình 52: "mạng lưới" của người dùng

Feeds

Shops

Visited shops

Liked shops

Rated shops

Chay An Soi and Bros Cafe

Network

Bros Cafe

January 13 - Shared by Chay An Soi

BROS Cafe được thiết kế và xây dựng từ một biệt thự rộng 13 x 17 m, trên đường Nguyễn Văn Thủ, với hai hàng me cổ thụ nằm dọc hai bên đường, đoạn giữa Hai Bà Trưng và Phùng Khắc Khoan.

Quán được thiết kế theo phong cách

Likes: 2	Customer: 1
Chay An Soi is a new customer.	1 friends like this shop.
Commented 1 time(s).	Rated 4 (1 time(s)).

02-03-2012

Chay An Soi commented on Bros Cafe coffee shop: sao ko ai bình luận j nhi - February 3

Chay An Soi like Bros Cafe coffee shop - February 3

01-13-2012

Chay An Soi has shared Bros Cafe coffee shop to facebook wall: moi share shop moi ne - January 13

Chay An Soi shared a new coffee shop: Bros Cafe - January 13

01-12-2012

Chay An Soi shared a new coffee shop: Gugi cafe - January 12

Hình 53: thống kê quan hệ giữa người dùng và quán

Rated shops

	Shop name	Point	
1	Bros Cafe	★★★★☆	
2	Cafe Phi Thuyền	★★★★★	
3	Khuc Giao Mua cafe	★★★★☆	
4	Penguin ice cream	★★★★☆	

Hình 51: các quán đã đánh giá

4.3.7. Đề nghị quán nên đến:

Shops that you might be interested in



Bros Cafe

Because :
your friend(Chay Ăn Sói) likes Bros Cafe



V3 Coffee

Because :
your friends(Chay Ăn Sói, Hân Nguyễn) like V3 Coffee



Cafe Phi Thuyền

Because :
your friend(Chay Ăn Sói) likes Cafe Phi Thuyền



Cỏ Nội

Because :
your friend(Chay Ăn Sói) likes Cỏ Nội



Đen & Trắng

Because :
your friend(Chay Ăn Sói) likes Đen & Trắng



KAYO TAPIOCA TEA & COFFEE SHOP


Because :
your friend appreciates KAYO TAPIOCA TEA & COFFEE SHOP (Chay Ăn Sói appreciated this shop was perfect)


Hình 54: những quán được đề nghị và lí do

4.3.8. Các trang dành cho người dùng quản lý các quán cà phê:

Coffee shops

[Browse](#)
[My Shops](#)
[Add New Shop](#)



Tùng Trần Thanh shared a shop :



 Café thời gian
Café thời gian

December 31, 2011 - [Unlike](#) - [Comment](#) - [Edit](#) - [Delete](#) - [Customers](#) - [Services](#)
 ★★★★★☆ - [View rating result](#)

You and 5 users like

[View 27 comments](#)


Tùng Trần Thanh shared a shop :


 Café Bạch Đằng
Café Bạch Đằng






December 31, 2011 - [Unlike](#) - [Comment](#) - [Edit](#) - [Delete](#) - [Customers](#) - [Services](#)
 ★★★★★☆

You like

[View 19 comments](#)

Hình 55: các quán do người dùng chia sẻ

Choose user to admin shop's information

 Trương Minh can edit Café công nghệ, January 30 Tùng Trần Thanh revoked this permission, January 30.	<input checked="" type="checkbox"/> <input type="button" value="Approve"/> <input type="button" value="Reprove"/>
Răng Rứa can edit Café công nghệ, January 29 Tùng Trần Thanh revoked this permission, January 30.	<input checked="" type="checkbox"/> <input type="button" value="Approve"/> <input type="button" value="Reprove"/>
 Thanhnhân Nguyễn can edit Era coffee, January 29 Tùng Trần Thanh revoked this permission, January 30.	<input checked="" type="checkbox"/> <input type="button" value="Approve"/> <input type="button" value="Reprove"/>
 Đại Dương test can edit Era coffee, January 29 Tùng Trần Thanh revoked this permission, January 28.	<input checked="" type="checkbox"/> <input type="button" value="Approve"/> <input type="button" value="Reprove"/>
 Nguyễn Thanh Danh test is an owner of Café thời gian, January 25 Tùng Trần Thanh revoked this permission, January 30.	<input checked="" type="checkbox"/> <input type="button" value="Approve"/> <input type="button" value="Reprove"/>
 Đại Dương test is an owner of Café thời gian, January 25 Trần Thanh Tung test granted this permission, February 1.	<input checked="" type="checkbox"/> <input type="button" value="Approve"/> <input type="button" value="Reprove"/>
Tran Thanh Tung test can edit Café công nghệ, January 23 Tùng Trần Thanh granted this permission, January 28.	<input checked="" type="checkbox"/> <input type="button" value="Approve"/> <input type="button" value="Reprove"/>
Tran Thanh Tung test can edit Café thời gian, January 22 Tùng Trần Thanh granted this permission, January 27.	<input checked="" type="checkbox"/> <input type="button" value="Approve"/> <input type="button" value="Reprove"/>

Hình 56: người dùng kiểm soát những người có quyền quản lý quán

4.3.9. Trang quản lý người dùng dành cho thành viên quản trị:

<input type="checkbox"/>	ID	Name	User Name	Gender	Join Date	Link	Deleted	
<input type="checkbox"/>	100000271151735	Chay Ấn Sỏi	nguyenbanggia	male	November 28, 2011	http://www.facebook.com/nguyenbanggia	active	Login As
<input type="checkbox"/>	100000947778325	Thanhnhân Nguyễn	nhaniuchien	female	November 28, 2011	http://www.facebook.com/nhaniuchien	active	Login As
<input type="checkbox"/>	100001301010590	Tùng Trần Thanh	tt.tungtt	male	December 5, 2011	http://www.facebook.com/tt.tungtt	active	Login As
<input type="checkbox"/>	100002396126840	Răng Rửa		male	December 13, 2011	http://www.facebook.com/profile.php?id=100002396126840	active	Login As

Hình 57: Quản lý tất cả các thành viên

4.3.10. Trang quản lý quán dành cho thành viên quản trị:

Coffee Shops/View all shops

<input type="checkbox"/>	Shop ID	Title	Shop style	Working hours	Reservation	Deleted	Modified date	Shared By	
<input type="checkbox"/>	54	Bros Cafe	office	not update	not update	active	4 minutes ago	100000271151735	<input type="checkbox"/> <input type="star"/> <input type="edit"/>
<input type="checkbox"/>	53	V3 Coffee	office	not update	not update	active	January 13	100000271151735	<input type="checkbox"/> <input type="star"/> <input type="edit"/>
<input type="checkbox"/>	52	Cà Phê Spotlight	other	not update	not update	active	February 4	100000271151735	<input type="checkbox"/> <input type="star"/> <input type="edit"/>
<input type="checkbox"/>	48	Cafe Phi Thuyền	4teen	not update	not update	active	January 4	100000271151735	<input type="checkbox"/> <input checked="" type="star"/> <input type="edit"/>
<input type="checkbox"/>	47	Pha lê đá	4teen	not update	not update	deleted	January 4	100000271151735	<input checked="" type="checkbox"/>
<input type="checkbox"/>	46	Cổ Nội	4teen	not update	not update	active	January 13	100000271151735	<input type="checkbox"/> <input checked="" type="star"/> <input type="edit"/>
<input type="checkbox"/>	45	KAYO TAPIOCA TEA & COFFEE SHOP	4teen	not update	not update	active	February 4	100000271151735	<input type="checkbox"/> <input checked="" type="star"/> <input type="edit"/>
<input type="checkbox"/>	44	Đen & Trắng	4teen	not update	not update	active	January 4	100000271151735	<input type="checkbox"/> <input type="star"/> <input type="edit"/>
<input type="checkbox"/>	43	Cafe Thủy Thiên Đường	4teen	not update	not update	active	January 4	100000271151735	<input type="checkbox"/> <input type="star"/> <input type="edit"/>
<input type="checkbox"/>	42	Mojo coffee	4teen	not update	not update	active	January 4	100000271151735	<input type="checkbox"/> <input type="star"/> <input type="edit"/>
<input type="checkbox"/>	41	Cat Dang	4teen	not update	not update	active	January 4	100000271151735	<input type="checkbox"/> <input type="star"/> <input type="edit"/>
<input type="checkbox"/>	40	Hideaway	4teen	not update	not update	active	January 4	100000271151735	<input type="checkbox"/> <input type="star"/> <input type="edit"/>
<input type="checkbox"/>	39	Hi-end coffee	4teen	not update	not update	active	January 4	100000271151735	<input type="checkbox"/> <input type="star"/> <input type="edit"/>
<input type="checkbox"/>	38	Tam Giao coffee	4teen	not update	not update	active	January 4	100000947778325	<input type="checkbox"/> <input type="star"/> <input type="edit"/>
<input type="checkbox"/>	37	Zenta Coffee & Hot pot	4teen	not update	not update	active	January 4	100000947778325	<input type="checkbox"/> <input type="star"/> <input type="edit"/>

Hình 58: quản lý tất cả các quán

4.3.11. Trang quản lý dịch vụ dành cho thành viên quản trị:

Service/View All

<input type="checkbox"/>	Service ID	Title	Type	Summary	Created Date	Modified date	Deleted	
<input type="checkbox"/>	1	Cappuccino	Coffee	Equal parts espresso, steamed milk, and frothed milk, often with cinnamon or flaked chocolate sprinkled on top. Sometimes coffee shops will add more milk, so that the customer will get a bigger drink	December 15, 2011	December 25, 2011	active	<input type="checkbox"/> <input type="edit"/>
<input type="checkbox"/>	2	Irish Coffee	Coffee	Irish coffee (theo tiếng Anh nghĩa là "cà phê theo kiểu của Ireland") là loại đồ uống nóng có pha rượu whisky đặc trưng của Ireland. Irish coffee (tiếng Ai-len: Caife Gaelach) là một loại đồ uống pha (cocktail) có chứa cồn.	December 15, 2011	December 25, 2011	active	<input type="checkbox"/> <input type="edit"/>
<input type="checkbox"/>	3	Latte Macchiato	Coffee	no summary	December 15, 2011	December 15, 2011	active	<input type="checkbox"/> <input type="edit"/>
<input type="checkbox"/>	14	Espresso	Coffee	Cái tên Espresso bắt nguồn từ nước Ý. Trong tiếng Ý, Espresso được gọi là "express", nghĩa là cà phê có thể được phục vụ cho khách hàng ngay lập tức. Cà	December 23, 2011	December 24, 2011	active	<input type="checkbox"/> <input type="edit"/>

Hình 59: Liệt kê danh sách các dịch vụ

Service name This field is requiredThis field is required

Service type:

Description: This field is requiredThis field is required

Summary:

Image Upload JPEG, GIF 2MB max per image

Hình 60: Giao diện cho phép thành viên quản trị thêm dịch vụ mới

4.3.12. Các trang quản lý các thông số hệ thống:

Coffee Shops/Setting

Shop count per page

How many shops will be shown per page? (Enter a number between 1 and 999)

Shop count per page

How many shops will be shown per page in admin view? (Enter a number between 1 and 999)

Visited shop count per page

How many shops will be shown for user checking visiting? (Enter a number between 1 and 999)

Shops Per Page in knowledge box

How many shops will be shown in knowledge box? (Enter a number between 1 and 999)

Shops number will be display for recommending to user

How many shops will be shown? (Enter a number between 1 and 999)

Hình 61: các thông số để hệ thống hiển thị quán

Settings/General

Favourite shops count

 How many featured shops will be displayed? (Enter a number between 1 and 999)

Top comment this week

 How many shops are the top comment of this week?

Top like this week

 How many shops are the top like of this week?

Top rating this week

 How many shops are the top rating of this week?

Lastest posted shops

 How many shops are the lastest posted shops?

Member actived per page

 How many items will be displayed?

Newest member count

 How many newest members will be displayed?

Save Changes

Hình 62: thông số tùy biến trang chủ

Settings/activity

Display new shop on activity

 Fill 1 for enable and 0 for disable

Display new comment on activity

 Fill 1 for enable and 0 for disable

Display like event on activity

 Fill 1 for enable and 0 for disable

Display when user shared about a shop on activity

 Fill 1 for enable and 0 for disable

Save Changes

Hình 63: tùy biến các hoạt động

Settings/User network

Radius x of ellipse

 Integer value for size of radius x

Radius y of ellipse

 Integer value for size of radius y

Text color

 What color is the ellipse text?

Line color

 What color is the line connect 2 ellipse?

Color of root ellipse

 What color is the root ellipse(root ellipse is the logged user)

Color of friends ellipse

 What color is friends ellipse?

Color of shops

 What color is the shops ellipse

Color of services

 What color is the service ellipse?

Hình 64: điều chỉnh cách vẽ sơ đồ

Service/Settings

Service count per page

 How many services will be shown per page? (Enter a number between 1 and 999)

User share service mode

 Are you allow user to share service that a particular shop serves? (Enter a number 1 or 0)

Album Name

 What is the album that img will be uploaded to?

Facebook uid

 Whose will images be uploaded to album of?

Hình 65: điều chỉnh liên quan đến quán

4.4. Đánh giá:

- Đưa vào sử dụng từ 6/2/2012, đến 15/2/2012 ứng dụng đã có được 1604 hồ sơ người dùng.

⇒ Để có 1 lượng lớn người dùng trong 1 thời gian ngắn, thì mạng xã hội là lựa chọn số 1.

- Từ 6/2/2012 đến 11/2/2012 đã có hơn 2000 lượt truy cập.
- Điểm nổi bật:
 - Kết quả tư vấn thú vị và đáng tin cậy.
 - Biểu diễn mối liên kết của người dùng được đánh giá cao.

Chương 5: KẾT LUẬN và HƯỚNG PHÁT TRIỂN

5.1. Tổng kết đề tài:

Đề tài đã hoàn thành với những ưu điểm có được:

- Đơn giản hóa quá trình đăng nhập và đăng ký cho người dùng.
- Giao diện đẹp, các chức năng được bố trí hợp lý, tiện dụng.
- Thông tin của quán cà phê được biểu diễn đầy đủ và tin cậy.
- Khả năng khai thác các mối quan hệ tiềm ẩn của người dùng và đưa ra những kết quả khuyến nghị thú vị và đáng tin cậy.
- Mọi người dùng có thể xem lại các mối liên kết của riêng mình với bạn bè, quán cà phê dưới dạng đồ thị mạng lưới.

Đề tài còn một số điểm hạn chế:

- Tốc độ hoạt động của hệ thống và quá trình khai thác tri thức còn chậm.
- Có hai nguyên nhân chính dẫn đến hạn chế này:
 - o Quá trình khai thác tri thức còn chậm công với việc phải truyền nhận dữ liệu giữa hai ứng dụng khiến thời gian đáp ứng của hệ thống kém.
 - o Do tại thời điểm thực hiện đề tài này, Facebook tại Việt Nam bị các nhà mạng chặn truy cập hay hạn chế tốc độ, nên các tác vụ cần đến Facebook thường tốn thời gian nhiều không cần thiết.

Hiện nay nhóm đã cơ bản hoàn thành được những mục tiêu đã đề ra. Ứng dụng của nhóm đã tạo ra, còn chứa đựng ý nghĩa và đóng góp:

- Hiểu và ứng dụng được hệ cơ sở tri thức biểu diễn bằng OWL.
- Hiểu được tầm quan trọng và thế mạnh của mạng xã hội, từ đó có những đối sách thích hợp cho việc quảng bá thương hiệu nói riêng và quản lý xã hội nói chung.
- Đề tài thực hiện hệ thống giúp người dùng không mất nhiều thời gian mà vẫn có thể tiếp cận được những sản phẩm mình có thể thích.
- Tận dụng được tri thức của cộng đồng trong việc đánh giá và phân loại sản phẩm để từ đó góp phần làm lành mạnh hoá môi trường kinh doanh

quảng cáo: giảm bớt những công ty có quảng cáo quá sai với sự thật và bảo vệ người tiêu dùng [14] [15] .

- Đóng góp vào sự phát triển của mạng xã hội, hệ thống thông minh, hỗ trợ người dùng dựa trên sự đóng góp của cộng đồng, cụ thể là bạn bè trên mạng xã hội.

5.2. Kết luận:

Với những mục tiêu đã đặt ra ban đầu của nhóm như phải nắm bắt được khái niệm và ứng dụng của ontology, nắm được khả năng tận dụng nguồn dữ liệu khổng lồ của Facebook để xây dựng một ứng dụng hỗ trợ người dùng trong việc tìm kiếm sản phẩm mà họ có thể thích.

Trong mạng xã hội, thông tin quý giá nhất là mối quan hệ bạn bè và tự tin tương lẫn nhau giữa họ. Một thông tin quảng cáo cho dù có hay đến mấy cũng chưa đủ độ tin cậy do người quảng cáo bao giờ cũng có xu hướng nói tốt cho họ. Với sự phản hồi của bạn bè, thông qua sự giới thiệu của bạn bè cũng như sự trải nghiệm dịch vụ của bạn bè, các thông tin quảng cáo sẽ được kiểm chứng. Điều này mang lại lợi ích cho cả 2 phía: nhà cung cấp dịch vụ và người dùng. Giúp nhà cung cấp dịch vụ nâng cao chất lượng dịch vụ và chăm sóc khách hàng tốt hơn

5.3. Hướng phát triển:

- Tiếp tục hoàn thiện hơn nữa hệ thống và giao diện để khuyến khích nhiều người sử dụng.
- Lợi thế của việc sử dụng Ontology là cung cấp khả năng suy luận tự động. Trong tương lai, chúng tôi sẽ khai thác tốt hơn cơ sở tri thức đã thu thập được để đưa ra những gợi ý chính xác nhất cho người dùng dựa theo các liên kết bạn bè.
- Đề tài có thể phát triển đa dạng các lĩnh vực bình chọn: nhà hàng, khách sạn, khu mua sắm, hay một sản phẩm cụ thể.
- Hơn nữa, với các *semantic links* giữa bạn bè, ta có thể truy vấn ra những câu phức tạp như “bạn của bạn của bạn” để tìm các thông tin hữu ích [16] .

DANH MỤC TÀI LIỆU THAM KHẢO

Tài liệu tiếng Anh

- [1] Jos de Bruijn, Using Ontologies - Enabling Knowledge Sharing and Reuse on the Semantic Web, DERI Technical Report DERI-2003-10-29, **2003**, 21-31.
- [2] Brickley, D. & Miller, L., FOAF Specification, **2007**, available at <http://xmlns.com/foaf/spec/>.
- [3] Natalya F. Noy and Deborah L. McGuinness, Ontology Development 101: A Guide to Creating Your First Ontology, **2001**.
- [4] Lee Feigenbaum & Eric Prud'hommeaux, SPARQL By Example A Tutorial, **2011**, available at <http://www.cambridgesemantics.com/2008/09/sparql-by-example/>.
- [5] Wayne Graham, Facebook API Developers Guide, **2008**, available at <https://developers.facebook.com/docs/>.
- [6] Holger Knublauch, Protege-OWL API Programmer's Guide, **2010**, available at http://protegewiki.stanford.edu/wiki/ProtegeOWL_API_Programmers_Guide.
- [7] Danah M. Boyd & Nicole B. Ellison, Social Network Sites: Definition, History, and Scholarship, Article first, **2007**, available at <http://onlinelibrary.wiley.com/doi/10.1111/j.1083-6101.2007.00393.x/full>
- [8] Daniel Nations, Social Networking Explained, available at <http://webtrends.about.com/od/socialnetworking/a/social-network.htm>
- [9] Eliza Sachs, Getting Started with Protege-Frames, **2006**, available at http://protege.stanford.edu/doc/tutorial/get_started/table_of_content.html.
- [10] W3C author group, SPARQL Query Language for RDF, **2008**, available at <http://www.w3.org/TR/rdf-sparql-query/#construct>.
- [11] Douglass, F., It's All About the (Social) Network, *Internet Computing, IEEE*, **2010**, 14, 4 -6
- [12] Amichai-Hamburger, Y. & Vinitzky, G., Social network use and personality, *Computers in Human Behavior*, **2010**, 26, 1289 – 1295.
- [13] Zaphiris, P. & Ang, C. S. (Eds.), Social Computing and Virtual Communities, *Chapman & Hall/CRC*, **2009**.
- [14] Berjani, B. & Strufe, T., A recommendation system for spots in location-based online social networks, *Proceedings of the 4th Workshop on Social Network Systems, ACM*, **2011**, 4:1-4:6
- [15] Nguyen, T.; Loke, S. W.; Torabi, T. & Lu, H., PlaceAware: A Tool for Enhancing Social Interactions in Urban Places, *Proceeding of the 10th International Symposium on Pervasive Systems, Algorithms, and Networks (ISPAN)*, **2009**, 143 -147.

- [16] Wang, F.-Y.; Carley, K. M.; Zeng, D. & Mao, W., Social Computing: From Social Informatics to Social Intelligence, Intelligent Systems, IEEE, **2007**, 22, 79 -83.
- [17] Taghi M. Khoshgoftaar&Xiaoyuan Su, A Survey of Collaborative Filtering Techniques, Artificial Intelligence, **2009**, available at <http://www.hindawi.com/journals/aai/2009/421425/>
- [18] Horrocks, I.; Patel-Schneider, P. F. & van Harmelen, F. From SHIQ and RDF to OWL: The Making of a Web Ontology Language *Journal of Web Semantics, Elsevier*, **2003**, 1, 7-26
- [19] Decker, S.; van Harmelen, F.; Broekstra, J.; Erdmann, M.; Fensel, D.; Horrocks, I.; Klein, M. & Melnik, S. The Semantic Web: The Roles of XML and RDF *IEEE Internet Computing*, **2000**, 4, 63-74
- [20] Ian Horrocks, DAML+OIL: A Description Logic for the Semantic Web *IEEE Data Engineering Bulletin*, **2002**, 25, 4-9

PHỤ LỤC

Bảng phân công công việc:

	Nguyễn Minh Chiến	Trần Thanh Tùng
Khảo sát	X	
Xác định yêu cầu chức năng và yêu cầu phi chức năng		X
Xây dựng giao diện	X	
Tìm hiểu về Facebook API		X
Tìm hiểu về Ontology		X
Tìm hiểu về SPARQL	X	
Tìm hiểu về Protégé API	X	
Xây dựng cơ sở tri thức (ontology)		X
Tìm hiểu về webservice (java web service)		X
Tìm hiểu về SVG (Scalable Vector Graphics) và xây dựng chức năng biểu diễn mối liên kết bằng đồ họa	X	
Phát triển ứng dụng	X	X

Bảng 62: Bảng phân công công việc

Nơi lưu trữ mã nguồn:

- Google code: <http://code.google.com/p/khoa-luan-tot-nghiep-nhom-hat-cat/>
- Trang quản lý dự án: <https://sites.google.com/site/igoforcoffee/>
- Trang web: <http://socialway.dyndns.org/coffeeego/public/>
- Địa chỉ ứng dụng trên Facebook: <http://apps.facebook.com/coffeeego>

Làm việc với Graph API, PHP SDK:

Cài đặt và bắt đầu:

Sau khi tải gói thư viện PHP SDK, ta trích xuất (*extract*) các tập tin (*files*) và sao chép các tập tin trong thư mục `src/` vào thư mục trên máy chủ mà chúng ta đặt ứng dụng của chúng ta, ví dụ như vào thư mục `php-sdk`. Sau đó, bất cứ khi nào muốn sử dụng SDK, chỉ cần thêm vào `php-`

sdk/facebook.php. Để sử dụng SDK, việc đầu tiên cần làm là tạo ra một đối tượng Facebook, một ví dụ đơn giản về việc tạo ra đối tượng Facebook với App Id và App secret:

```
require_once("php-sdk/facebook.php");
$config = array();
$config['appId'] = 'YOUR_APP_ID';
$config['secret'] = 'YOUR_APP_SECRET';
$config['fileUpload'] = false; // optional
$facebook = new Facebook($config);
```

Xác thực và ủy quyền (*Authentication & Authorization*):

PHP SDK được sử dụng để hỗ trợ quá trình đăng ký và đăng nhập cho người dùng vào website của bạn thông qua tài khoản Facebook của người dùng. Về hướng máy chủ, PHP SDK cung cấp các hàm giúp đỡ (*helper methods*) để tối giản việc đăng nhập, gửi yêu cầu cho phép, và đăng xuất. Các tính năng đó thì được cung cấp thông qua các phương thức

`Facebook::getUser()`,

`Facebook::getLoginUrl()` và `Facebook::getLogoutUrl()`.

Kết hợp PHP SDK và Javascript SDK:

Sự kết hợp giữa PHP SDK và Javascript SDK tạo nên sự mạch lạc giữa phía máy chủ và máy khách. Một khi người dùng đang đăng nhập vào Facebook, Javascript SDK sẽ tự động lấy thông tin phiên đăng nhập đó và gửi nó đến máy chủ thông qua cookie.

Một ví dụ về sự kết hợp giữa PHP SDK và Javascript SDK từ tác giả Jerry Cain – một thành viên thuộc nhóm phát triển của Facebook:

```
<?php
require 'php-sdk/src/facebook.php';
$facebook = new Facebook(array(
    'appId' => 'YOUR_APP_ID',
    'secret' => 'YOUR_APP_SECRET',
));
// See if there is a user from a cookie
```

```

$user = $facebook->getUser();
if ($user) {
    try {
        // Proceed knowing you have a logged in
        user who's authenticated.
        $user_profile = $facebook->api('/me');
    } catch (FacebookApiException $e) {
        echo '<pre>'.htmlspecialchars(print_r($e,
true)).'</pre>';
        $user = null;
    }
}
?>
<!DOCTYPE html>
<html
xmlns:fb="http://www.facebook.com/2008/fbml">
<body>
<?php if ($user_profile) { ?>
    Your user profile is
    <pre>
    <?php print
    htmlspecialchars(print_r($user_profile, true))
    ?>
    </pre>
    <?php } else { ?>
    <fb:login-button></fb:login-button>
    <?php } ?>
    <div id="fb-root"></div>
    <script>
        window.fbAsyncInit = function() {
            FB.init({
                appId: '<?php echo $facebook-
>getAppID() ?>',

```

```

        cookie: true,
        xfbml: true,
        oauth: true
    });

    FB.Event.subscribe('auth.login',
function(response) {
    window.location.reload();
});

    FB.Event.subscribe('auth.logout',
function(response) {
    window.location.reload();
});
};

(function() {
    var e =
document.createElement('script'); e.async =
true;

    e.src = document.location.protocol +

'//connect.facebook.net/en_US/all.js';

    document.getElementById('fb-
root').appendChild(e);

    }());
</script>
</body>
</html>

```

Một số đối tượng Facebook (Faceook Objects) quan trọng:

Api: gọi các phương thức từ Graph Api, gọi các truy vấn FQL, và gọi các phương thức (**DEPRECATED**) REST API tùy thuộc vào tham số truyền vào.

Để gọi các phương thức Graph Api, ta truyền tham số đầu tiên như một đường dẫn, để chạy các truy vấn FQL hoặc REST API ta truyền mảng các tham số cần thiết cho lời gọi như một số ví dụ mô tả bên dưới:

- Lấy thông tin người dùng thông qua Graph Api và in tên người dùng ra màn hình:

```
<?
    // Remember to copy files from the SDK's
src/ directory to a
    // directory in your application on the
server, such as php-sdk/
    require_once('php-sdk/facebook.php');
    $config = array(
        'appId' => 'YOUR_APP_ID',
        'secret' => 'YOUR_APP_SECRET',
    );
    $facebook = new Facebook($config);
    $user_id = $facebook->getUser();
?>
<html>
<head></head>
<body>
<?
    if($user_id) {
        // We have a user ID, so probably a
logged in user.
        // If not, we'll get an exception,
which we handle below.
        try {
            $user_profile = $facebook-
>api('/me','GET');
            echo "Name: ".
$user_profile['name'];
```

```

    } catch(FacebookApiException $e) {
        // If the user is logged out, you
can have a
        // user ID even though the access
token is invalid.
        // In this case, we'll get an
exception, so we'll
        // just ask the user to login again
here.

        $login_url      =      $facebook-
>getLoginUrl();
        echo 'Please <a href="' . $login_url
. '">login.</a>';
        error_log($e->getType());
        error_log($e->getMessage());
    }
} else {
    // No user, print a link for the user
to login
    $login_url = $facebook->getLoginUrl();
    echo 'Please <a href="' . $login_url .
'">login.</a>';
}
?>
</body>
</html>

```

- Lấy tên người dùng sử dụng FQL và in ra màn hình.

```

<?
    // Remember to copy files from the SDK's
src/ directory to a
    // directory in your application on the
server, such as php-sdk/

```

```

require_once('php-sdk/facebook.php');
$config = array(
    'appId' => 'YOUR_APP_ID',
    'secret' => 'YOUR_APP_SECRET',
);
$facebook = new Facebook($config);
$user_id = $facebook->getUser();
?>
<html>
<head></head>
<body>
<?
    if($user_id) {
        // We have a user ID, so probably a
logged in user.
        // If not, we'll get an exception,
which we handle below.
        try {
            $fql = 'SELECT name from user where
uid = ' . $user_id;
            $ret_obj = $facebook->api(array(
                'method'
=> 'fql.query',
                'query'
=> $fql,
            ));
            // FQL queries return the results in
an array, so we have
            // to get the user's name from the
first element in the array.
            echo '<pre>Name: ' .
$ret_obj[0]['name'] . '</pre>';
        } catch(FacebookApiException $e) {

```

```

        // If the user is logged out, you
        can have a
        // user ID even though the access
        token is invalid.
        // In this case, we'll get an
        exception, so we'll
        // just ask the user to login again
        here.

        $login_url = $facebook-
>getLoginUrl();
        echo 'Please <a href="' . $login_url
        . '>login.</a>';
        error_log($e->getType());
        error_log($e->getMessage());
    }
} else {
    // No user, so print a link for the
    user to login
    $login_url = $facebook->getLoginUrl();
    echo 'Please <a href="' . $login_url .
    '>login.</a>';
}
?>
</body>
</html>

```

- Viết một đường dẫn lên tường người dùng sử dụng Graph Api.

```

<?
    // Remember to copy files from the SDK's
    src/ directory to a
    // directory in your application on the
    server, such as php-sdk/
    require_once('php-sdk/facebook.php');

```



```

$config = array(
    'appId' => 'YOUR_APP_ID',
    'secret' => 'YOUR_APP_SECRET',
);
$facebook = new Facebook($config);
$user_id = $facebook->getUser();
?>
<html>
<head></head>
<body>
<?
    if($user_id) {
        // We have a user ID, so probably a
logged in user.
        // If not, we'll get an exception,
which we handle below.
        try {
            $ret_obj = $facebook-
>api('/me/feed', 'POST',
                                array(
                                    'link'
=> 'www.example.com',

'message' => 'Posting with the PHP SDK!'
                                ));
            echo '<pre>Post ID: ' .
$ret_obj['id'] . '</pre>';
        } catch(FacebookApiException $e) {
            // If the user is logged out, you
can have a
            // user ID even though the access
token is invalid.

```

```

        // In this case, we'll get an
exception, so we'll
        // just ask the user to login again
here.

        $login_url = $facebook->getLoginUrl(
array(

                                'scope' =>
'publish_stream'

                                ));
        echo 'Please <a href="' . $login_url
. '">login.</a>';
        error_log($e->getType());
        error_log($e->getMessage());
    }
    // Give the user a logout link
    echo '<br /><a href="' . $facebook-
>getLogoutUrl() . '">logout</a>';
} else {
    // No user, so print a link for the
user to login
    // To post to a user's wall, we need
publish_stream permission
    // We'll use the current URL as the
redirect_uri, so we don't
    // need to specify it here.
    $login_url = $facebook->getLoginUrl(
array( 'scope' => 'publish_stream' ) );
    echo 'Please <a href="' . $login_url .
'">login.</a>';
}
?>
</body>
</html>

```

- Tải hình ảnh lên hồ sơ cá nhân người dùng:

```
<?
    // Remember to copy files from the SDK's
src/ directory to a
    // directory in your application on the
server, such as php-sdk/
    require_once('php-sdk/facebook.php');
    $config = array(
        'appId' => 'YOUR_APP_ID',
        'secret' => 'YOUR_APP_SECRET',
        'fileUpload' => true,
    );
    $facebook = new Facebook($config);
    $user_id = $facebook->getUser();
    $photo = './mypic.png'; // Path to the
photo on the local filesystem
    $message = 'Photo upload via the PHP
SDK!';
?>
<html>
<head></head>
<body>
<?
    if($user_id) {
        // We have a user ID, so probably a
logged in user.
        // If not, we'll get an exception,
which we handle below.
        try {
            // Upload to a user's profile. The
photo will be in the
```

```

        // first album in the profile. You
        can also upload to
        // a specific album by using
        /ALBUM_ID as the path
        $ret_obj = $facebook->
>api('/me/photos', 'POST', array(

'source' => '@' . $photo,

'message' => $message,

                                )
                                );

        echo '<pre>Photo ID: ' .
$ret_obj['id'] . '</pre>';
    } catch (FacebookApiException $e) {
        // If the user is logged out, you
        can have a
        // user ID even though the access
        token is invalid.
        // In this case, we'll get an
        exception, so we'll
        // just ask the user to login again
        here.

        $login_url = $facebook->getLoginUrl(
array(

                                'scope' =>

'sphoto_upload'

                                ));

        echo 'Please <a href="' . $login_url
. '">login.</a>';
        error_log($e->getType());
        error_log($e->getMessage());
    }

```

```

        echo '<br /><a href="' . $facebook-
>getLogoutUrl() . '">logout</a>';
    } else {
        // No user, print a link for the user
to login
        // To upload a photo to a user's wall,
we need photo_upload permission
        // We'll use the current URL as the
redirect_uri, so we don't
        // need to specify it here.
        $login_url = $facebook->getLoginUrl(
array( 'scope' => 'photo_upload' ) );
        echo 'Please <a href="' . $login_url .
'">login.</a>';
    }
?>
</body>
</html>

```

Để gọi phương thức của Graph Api, truyền đường dẫn cuối mà ta muốn nhận kết quả từ nó như tham số đầu tiên, các tham số lựa chọn khác như mô tả:

```
$ret = $facebook->api($path, $method, $params);
```

Tên	Mô tả
path	Đường dẫn truyền vào khi gọi hàm, ví dụ: “/me” khi lấy thông tin hồ sơ cá nhân
method	Phương thức lựa chọn cho phương thức HTML của yêu cầu: GET, POST, hoặc DELETE
params	Tham số lựa chọn, tùy thuộc vào phương thức mà chúng ta gọi mà truyền vào mảng với cặp thẻ phù hợp “name”=> “value”

Bảng 63: Các tham số Facebook API

Để chạy một truy vấn FQL, truyền tham số vào trong mảng với thuộc tính “method” là ‘fql.query’:

```
$ret = $facebook->api( array(
    'method' => 'fql.query',
    'query' => 'SELECT . . . ',
));
```

REST APIs được gọi giống như cách chạy một truy vấn FQL, nhưng khác ở chỗ, giá trị của thuộc tính ‘method’ được thay thế tương ứng với tên của REST API, và các tham số khác được truyền phù hợp với lời gọi API, ví dụ như:

```
$ret = $facebook->api( array(
    'method' => 'link.getStats',
    'urls' =>
    facebook.com,developers.facebook.com',
));
```

- `getAccessToken`: `Facebook::getAccessToken()` , được dùng để lấy mã truy cập thông báo của một người dùng hay một ứng dụng, ví dụ như:

```
// Get the current access token
$access_token = $facebook->getAccessToken();
```

- `getApiSecret`: `Facebook::getApiSecret()` , lấy App secret mà SDK đang sử dụng, ví dụ như:

```
// Get current App secret
$secret = Facebook->getApiSecret();
```

- `getAppId`: `Facebook::getAppId()` , lấy App Id mà SDK đang sử dụng, ví dụ như:

```
// Get the current app id
$appId = $facebook->getAppId();
```

- `getLoginUrl`: `Facebook::getLoginUrl($params = array())` , trả về đường dẫn url mà khi người dùng phía máy khách nhấn vào sẽ tự động chuyển người dùng đăng nhập vào Facebook và Facebook ủy quyền lại về ứng dụng của chúng ta nếu cần, sau đó, chuyển hướng trở lại website của ta. Nếu người dùng đăng nhập vào Facebook thất bại hoặc Facebook không nhận thấy khả năng nhận ủy quyền từ ứng dụng của chúng ta, người dùng sẽ được chuyển hướng thông qua mã HTML 302 đến đường dẫn mà ta truyền vào cho giá trị `'request_uri'` với nội dung `'error'`, `'error_reason'`, và `'error_description'` như những tham số trong đường dẫn. Ví dụ:

```
$params = array(
    'scope' => 'read_stream, friends_likes',
    'redirect_uri' =>
    'https://www.myapp.com/post_login_page'
);

$loginUrl = $facebook->getLoginUrl($params);
```

Trong đó:

Name	Description
scope	Các điều khoản yêu cầu từ người dùng. Nếu giá trị này không được truyền thì các quyền cơ bản sẽ được yêu cầu từ người dùng.
redirect_uri	Đường dẫn sẽ được chuyển hướng tới khi quá trình đăng nhập/ủy quyền hoàn tất.
display	Chế độ hiển thị, giá trị mặc định là <code>'page'</code> , nhưng có thể thay đổi sang giá trị khác như <code>'popup'</code> .

Bảng 64: các tham số để đăng nhập Facebook

- `getLogoutUrl`: `Facebook::getLogoutUrl($params = array())` , trả về đường dẫn mà khi người dùng nhấn vào sẽ kết thúc phiên đăng nhập của người dùng Facebook.

```
$params = array('next' =>
    'https://www.myapp.com/after_logout' );
```

```
$facebook->getLogoutUrl($params); // $params
is optional.
```

Trong đó, ‘next’ là đường dẫn mà Facebook sẽ chuyển hướng tới sau khi người dùng đăng xuất hoàn tất.

- `getUser: Facebook::getUser()` , trả về Facebook Id của người dùng đang đăng nhập, nếu chưa đăng nhập sẽ trả về 0.

```
$uid = $facebook->getUser();
```

- `setAccessToken: Facebook::setAccessToken($access_token)`, gán giá trị mã truy cập thông báo đang được sử dụng bởi SDK.

```
// Set a new access token, by first getting it
via means other than the SDK
```

```
$facebook->setAccessToken($new_access_token);
```

Trong đó ‘access_token’ là giá trị mã truy cập thông báo mới.

- `setApiSecret: Facebook::setApiSecret($app_secret)` , gán giá trị app secret mà SDK đang sử dụng

```
// Set a new app secret
```

```
$facebook->setApiSecret("APP_SECRET");
```

```
// If you do above, also set the app id
```

```
$facebook->setAppId("APP_ID");
```

Trong đó ‘app_secret’ là giá trị app secret mới.

- `setAppId: Facebook::setAppId($app_id)`, gán giá trị app id mà SDK đang sử dụng.

```
// Set a new app id to use
```

```
$facebook->setAppId("APP_ID");
```

```
// If you do the above, also set a new app
secret
```

```
$facebook->setApiSecret("APP_SECRET");
```

Trong đó, ‘app_id’ là giá trị app id mới.

- `setFileUploadSupport:Facebook::setFileUploadSupport($fileUploadSupport)`, gán tập tin hỗ trợ tải lên trong SDK. Máy chủ của chúng ta phải cấu hình hỗ trợ tải lên để có thể sử dụng tính năng này.

```
// Upload a photo to a user's profile

// Your app needs photo_upload permission for
this to work

$facebook->setFileUploadSupport(true);

$img = '/tmp/mypic.png';

$photo = $facebook->api('/me/photos', 'POST',
    array( 'source' => '@' . $img,
          'message' => 'Photo uploaded via the PHP
          SDK!'
        ));
```

Trong đó, ‘fileUploadSupport’ gán bằng ‘true’ nếu máy chủ hỗ trợ tải lên và bằng ‘false’ nếu ngược lại.

Một số câu hỏi và cách giải quyết bằng SPARQL:

1. Ai là bạn của A:

```
SELECT ?person
where
{
    ?person rdf:type:Person;
    :friendOf ?friend.
    filter(?friend =:A)
}
```

2. Bạn của bạn của A:

```
SELECT distinct ?person
where
```

```
{
    ?person rdf:type:Person;
    :friendOf ?friend.
    ?friend :friendOf ?A.
    filter(?A =:A)
}
```

3. A đã từng đến thăm quán cà phê nào:

```
SELECT distinct ?shop
where
{
    ?shop rdf:type:CoffeeShop;
    :hasCustomer ?customer.
    filter(?customer =:A)
}
```

4. A đánh giá quán cà phê C bao nhiêu điểm:

```
SELECT ?Point
where
{
    ?x rdf:type:Rating;
    :ratedBy ?rb;
    :ratesFor ?rf;
    :ratePoint ?Point.
    Filter(?rb =:A).
    Filter(?rf =:C).
}
```

5. Quán cà phê nào có phục vụ dịch vụ S:

```
SELECT ?Shop
where
{
```

```

    ?Shop rdf:type:CoffeeShop;
    :serves ?s;
    filter(?s =:S)
}

```

6. Quán cà phê nào mà bạn của A đánh giá cao:

```
SELECT distinct ?shop ?friends ?point
```

```
where
```

```

{
    ?person rdf:type:Person;
    :friendOf ?friends.
    filter(?person=:A)
    ?rating rdf:type:Rating;
    :ratePoint ?point;
    :ratedBy ?ratedBy;
    :ratesFor ?ratesFor.
    FILTER(?ratedBy = ?friends)
    ?shop rdf:type:CoffeeShop.
    FILTER(?shop = ?ratesFor)
    FILTER (?point >= '4' || ?point >= 4 )
}

```

7. Quán cà phê nào mà bạn của A thích:

```
SELECT distinct ?shop ?friends
```

```
where
```

```

{
    ?person rdf:type:Person;
    :friendOf ?friends.
    filter(?person=:A)
    ?friends :likes ?fLikes.
    ?shop rdf:type:CoffeeShop.
}

```

```

    FILTER (?shop = ?fLikes)
}
orderby ASC(?shop )

```

8. Quán cà phê nào mà bạn của A đã ghé thăm:

```

SELECT distinct ?shop ?friends
where
{
    ?person rdf:type:Person;
    :friendOf ?friends.
    filter(?person=:A)
    ?friends :customerOf ?fcustomerOf.
    ?shop rdf:type:CoffeeShop.
    FILTER (?shop = ?fcustomerOf)
}

```

9. Quán cà phê nào mà bạn của A đã ghé thăm có phục vụ loại dịch vụ mà A và bạn của A cùng thích.

```

SELECT distinct ?shop ?serves ?friends
where
{
    ?person rdf:type:Person;
    :likes ?likes;
    :friendOf ?friends.
    filter(?person=:A)
    ?shop rdf:type:CoffeeShop.
    ?shop :serves ?serves.
    Filter(?likes = ?serves).
    ?friends :likes ?fLikes;
    :customerOf ?fShops.
    filter(?serves=?fLikes).
}

```

```

    filter(?shop = ?fShops)
}

```

10. Quán cà phê nào mà bạn của A đã ghé thăm có phục vụ loại dịch vụ mà

A thích:

```

SELECT distinct ?shop ?serves ?friends

```

where

```

{
    ?person rdf:type:Person;

    :likes ?likes;

    :friendOf ?friends.

    filter(?person=:A)

    ?shop rdf:type:CoffeeShop.

    ?shop :serves ?serves.

    Filter(?likes = ?serves).

    ?friends :customerOf ?fShops.

    filter(?shop = ?fShops)
}

```

11. Quán cà phê nào mà bạn của A thích và có phục vụ loại dịch vụ A

thích:

```

SELECT distinct ?shop ?serves ?friends

```

where

```

{
    ?person rdf:type:Person;

    :likes ?likes;

    :friendOf ?friends.

    filter(?person=:A)
}

```

```

?shop rdf:type:CoffeeShop.
?shop :serves ?serves.
Filter(?likes = ?serves).
?friends :likes ?fLikes.
filter(?shop = ?fLikes)
}

```

12. Quán cà phê nào mà bạn của A đánh giá cao và có phục vụ loại dịch vụ A thích:

```

SELECT distinct ?shop ?serves ?friends ?point
where
{
    ?person rdf:type:Person;
    :likes ?likes;
    :friendOf ?friends.
    filter(?person=:A)
    ?shop rdf:type:CoffeeShop.
    ?shop :serves ?serves.
    Filter(?likes = ?serves).
    ?rating rdf:type:Rating;
    :ratePoint ?point;
    :ratedBy ?ratedBy;
    :ratesFor ?ratesFor.
    FILTER(?point >= 4 || ?point >= '4')
    FILTER(?ratedBy = ?friends)
    FILTER(?shop = ?ratesFor)
}

```