

## LỜI CẢM ƠN

Trước tiên, em xin bày tỏ lòng biết ơn chân thành và sâu sắc nhất tới thạc sĩ Đỗ Thị Liên, người đã trực tiếp hướng dẫn, giúp đỡ em tận tình trong suốt quá trình thực hiện đồ án.

Em xin được gửi lời cảm ơn sâu sắc tới những giảng viên của Học viện Công nghệ Bưu chính viễn thông nói chung và khoa Công nghệ Đa phương tiện nói riêng, đã truyền đạt những kiến thức quý báu cho em trong suốt những năm học vừa qua.

Tài liệu này là kết quả thu được trong thời gian thực hiện khóa luận. Kính mong các thầy cô đóng góp ý kiến để em có thể hoàn thiện hơn.

Em xin chân thành cảm ơn.

Sinh viên

Nguyễn Tiến Đức

## MỤC LỤC

LỜI CẢM ƠN .....	1
MỤC LỤC.....	2
DANH MỤC HÌNH ẢNH .....	5
DANH MỤC BẢNG BIỂU .....	7
MỞ ĐẦU.....	8
1.        Đặt vấn đề.....	8
2.        Mục tiêu của đồ án .....	8
3.        Bố cục của đồ án .....	9
CHƯƠNG 1: TỔNG QUAN VỀ HỆ TƯ VÂN ĐA PHƯƠNG TIỆN TÍCH HỢP YẾU TỐ NGỮ CẢNH.....	10
1.1.        Tổng quan về hệ tư vấn truyền thống.....	10
1.1.1.        Hệ tư vấn dựa trên nội dung .....	11
1.1.2.        Hệ tư vấn dựa trên lọc cộng tác.....	12
1.1.2.1.        Phương pháp dựa trên bộ nhớ .....	14
1.1.2.2.        Phương pháp dựa trên mô hình .....	16
1.1.2.3.        Đánh giá chung về phương pháp lọc cộng tác .....	17
1.1.3.        Hệ tư vấn lai .....	18
1.1.4.        Những phương pháp đánh giá độ chính xác của hệ tư vấn .....	18
1.1.4.1.        Precision .....	19
1.1.4.2.        MAP (Mean Average Precision) .....	19
1.1.4.3.        Recall.....	20
1.2.        Tổng quan về hệ tư vấn đa phương tiện kết hợp ngữ cảnh .....	20
1.2.1.        Phương pháp mô hình hóa ngữ cảnh độc lập .....	23
1.2.2.        Phương pháp mô hình hóa ngữ cảnh phụ thuộc .....	24
1.2.2.1.        Độ tương quan ngữ cảnh độc lập .....	25
1.2.2.2.        Độ tương quan ngữ cảnh Latent.....	26
1.3.        Kết luận chương .....	27
CHƯƠNG 2: NGHIÊN CỨU PHƯƠNG PHÁP MÔ HÌNH HÓA NGỮ CẢNH CSLIM TRONG XÂY DỰNG HỆ TƯ VÂN ĐA PHƯƠNG TIỆN DỰA TRÊN NGỮ CẢNH.....	29

2.1.	Phương pháp tuyến tính thưa SLIM.....	29
2.2.	Mô hình hóa ngữ cảnh dựa trên độ tương quan đa chiều.....	32
2.3.	Thuật toán tích hợp ngữ cảnh đa chiều CSLIM_MCS.....	34
2.3.1.	Cập nhật tham số trọng số sản phẩm W .....	35
2.3.2.	Cập nhật tham số ngữ cảnh .....	36
2.3.3.	Thuật toán huấn luyện và dự đoán của CSLIM_MCS .....	38
2.4.	Kiểm nghiệm thuật toán .....	41
2.4.1.	Bộ dữ liệu kiểm nghiệm .....	41
2.4.2.	Phương pháp đánh giá .....	42
2.4.3.	Kết quả kiểm nghiệm .....	42
2.5.	Kết luận chương .....	45
<b>CHƯƠNG 3: PHÁT TRIỂN HỆ THỐNG TƯ VẤN ĐỊA ĐIỂM DU LỊCH TẠI THÀNH PHỐ HỒ CHÍ MINH DỰA TRÊN NGỮ CẢNH.....</b>		46
3.1.	Yêu cầu hoạt động của hệ thống .....	46
3.2.	Mô hình tổng quát hệ thống .....	46
3.3.	Công nghệ áp dụng.....	47
3.3.1.	Hệ điều hành Android .....	48
3.3.2.	Dịch vụ web REST.....	49
3.3.2.1.	Dịch vụ web là gì ? .....	49
3.3.2.2.	REST .....	49
3.3.3.	Thư viện CARSKit.....	51
3.4.	Phân tích thiết kế hệ thống tư vấn địa điểm du lịch tại thành phố Hồ Chí Minh .....	52
3.4.1.	Phân tích yêu cầu.....	52
3.4.1.1.	Phía client .....	52
3.4.1.2.	Phía server .....	53
3.4.2.	Mô hình phát triển phần mềm .....	53
3.4.3.	Hệ thống giao diện ứng dụng gợi ý địa điểm du lịch .....	54
3.4.4.	Phân tích thiết kế hệ thống cho ứng dụng gợi ý địa điểm du lịch tại thành phố Hồ Chí Minh .....	56
3.4.4.1.	Phía client .....	56
3.4.4.2.	Phía server .....	68
3.4.4.3.	Biểu đồ triển khai hệ thống .....	78
3.4.4.4.	Biểu đồ cơ sở dữ liệu quan hệ .....	79

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	81
1.         Kết luận .....	81
2.         Phương hướng phát triển.....	81
TÀI LIỆU THAM KHẢO.....	82
PHỤ LỤC: CÀI ĐẶT VÀ TRIỂN KHAI .....	84
1. Môi trường cài đặt.....	84
2. Triển khai ứng dụng .....	84

## DANH MỤC HÌNH ẢNH

Hình ảnh 1: Tổng quan ba loại phương pháp kết hợp ngữ cảnh .....	22
Hình ảnh 2: Phương pháp mô hình hóa ngữ cảnh độc lập .....	23
Hình ảnh 3: Ví dụ về phương pháp dự đoán đánh giá của SLIM .....	30
Hình ảnh 4: Không gian ngữ cảnh đa chiều .....	33
Hình ảnh 5: Phương pháp 3-fold cross validate .....	42
Hình ảnh 6: Biểu đồ so sánh chỉ số Prec@10 .....	44
Hình ảnh 7: Biểu đồ so sánh dựa trên chỉ số MAP@10 .....	44
Hình ảnh 8: Biểu đồ so sánh dựa trên chỉ số Recall@10 .....	45
Hình ảnh 9: Mô hình tổng quan hệ thống .....	47
Hình ảnh 10: Android Nougat .....	48
Hình ảnh 11: Mô hình của dịch vụ web .....	49
Hình ảnh 12: Dịch vụ web REST .....	50
Hình ảnh 13: Kiến trúc của CARSKit .....	51
Hình ảnh 14: Phát triển ứng dụng theo mô hình thác nước .....	54
Hình ảnh 15: Cây phân cấp giao diện ứng dụng .....	55
Hình ảnh 16: Biểu đồ use case phía client .....	58
Hình ảnh 17: Biểu đồ trạng thái của User .....	59
Hình ảnh 18: Biểu đồ tuần tự chức năng đăng nhập .....	60
Hình ảnh 19: Biểu đồ tuần tự chức năng đăng ký .....	61
Hình ảnh 20: Biểu đồ tuần tự chức năng đưa ra danh sách yêu thích .....	62
Hình ảnh 21: Biểu đồ tuần tự chức năng thêm vào danh sách yêu thích .....	62
Hình ảnh 22: Biểu đồ tuần tự chức năng xóa khỏi danh sách yêu thích .....	63
Hình ảnh 23: Biểu đồ tuần tự chức năng đưa ra top 10 gợi ý .....	63
Hình ảnh 24: Biểu đồ tuần tự chức năng gợi ý địa điểm liên quan .....	64
Hình ảnh 25: Biểu đồ tuần tự chức năng gợi ý dựa trên ngữ cảnh .....	64
Hình ảnh 26: Biểu đồ tuần tự chức năng tìm kiếm .....	65
Hình ảnh 27: Biểu đồ tuần tự chức năng lọc kết quả tìm kiếm .....	65
Hình ảnh 28: Biểu đồ tuần tự chức năng hiển thị thông tin chi tiết địa điểm .....	66
Hình ảnh 29: Biểu đồ tuần tự chức năng đánh giá địa điểm .....	66
Hình ảnh 30: Biểu đồ tuần tự chức năng liên hệ quản lý địa điểm .....	67
Hình ảnh 31: Biểu đồ lớp phía client .....	67
Hình ảnh 32: Biểu đồ use case phía server .....	69
Hình ảnh 33: Biểu đồ trạng thái của đối tượng Recommendation .....	70
Hình ảnh 34: Biểu đồ tuần tự chức năng đăng nhập .....	71
Hình ảnh 35: Biểu đồ tuần tự chức năng đăng ký .....	72
Hình ảnh 36: Biểu đồ tuần tự chức năng gợi ý top 10 .....	73
Hình ảnh 37: Biểu đồ tuần tự chức năng địa điểm liên quan .....	73
Hình ảnh 38: Biểu đồ tuần tự chức năng gợi ý dựa vào ngữ cảnh .....	74
Hình ảnh 39: Biểu đồ tuần tự chức năng đánh giá địa điểm du lịch .....	75

Hình ảnh 40: Biểu đồ tuần tự chức năng tìm kiếm .....	75
Hình ảnh 41: Biểu đồ tuần tự chức năng huấn luyện dữ liệu.....	76
Hình ảnh 42: Biểu đồ tuần tự chức năng đưa ra gợi ý và lưu vào cơ sở dữ liệu.....	77
Hình ảnh 43: Biểu đồ lớp phía server .....	78
Hình ảnh 44: Biểu đồ triển khai hệ thống .....	79
Hình ảnh 45: Biểu đồ cơ sở dữ liệu quan hệ .....	80
Hình ảnh 46: Giao diện đăng nhập.....	84
Hình ảnh 47: Giao diện đăng ký .....	85
Hình ảnh 48: Giao diện mà hình chính, tab top 10 .....	86
Hình ảnh 49: Giao diện màn hình chính tab gợi ý ngữ cảnh .....	87
Hình ảnh 50: Dialog chọn ngữ cảnh khi gợi ý .....	88
Hình ảnh 51: Tính năng tìm kiếm trên giao diện chính .....	89
Hình ảnh 52: Giao diện kết quả tìm kiếm .....	90
Hình ảnh 53: Giao diện cửa sổ điều hướng.....	91
Hình ảnh 54: Giao diện danh sách yêu thích .....	92
Hình ảnh 55: Giao diện chi tiết sản phẩm (1) .....	93
Hình ảnh 56: Giao diện chi tiết sản phẩm (2) .....	94
Hình ảnh 57: Giao diện đánh giá sản phẩm .....	95

## **DANH MỤC BẢNG BIỂU**

Bảng 1: Ví dụ về độ tương quan ngữ cảnh độc lập.....	26
Bảng 2: Ví dụ về độ tương quan Latent .....	27
Bảng 3: Thuật toán huấn luyện CSLIM_MCS.....	40
Bảng 4: Thuật toán dự đoán đánh giá CSLIM_MCS.....	41
Bảng 5: Thông tin về bộ dữ liệu InCarMusic .....	41
Bảng 6: Kết quả kiểm nghiệm và so sánh SLIM với CSLIM_MCS .....	43
Bảng 7: Bảng kết quả so sánh dựa trên chỉ số Prec@10.....	44
Bảng 8: Bảng so sánh dựa trên chỉ số MAP@10 .....	45
Bảng 9: Bảng so sánh dựa trên chỉ số Recall@10 .....	45
Bảng 10: Danh sách ca sử dụng phía client .....	57
Bảng 11: Danh sách các tác nhân phía server .....	68
Bảng 12: Danh sách các ca sử dụng phía server .....	69

## MỞ ĐẦU

### 1. Đặt vấn đề

Internet mang lại rất nhiều sự tiện ích khác nhau cho con người. Con người có thể học tập, tìm kiếm, tra cứu, giải trí, kết nối với nhau thông qua mạng Internet. Tuy nhiên đi kèm theo đó là sự bùng nổ về mặt số lượng thông tin. Điều này khiến cho người dùng khó có thể lựa chọn những thông tin hữu dụng trong cả một núi thông tin hỗn độn như vậy. Sẽ ra sao nếu một người dùng vào một trang web thương mại điện tử lớn như Amazon, Ebay lại không biết nên chọn mua một sản phẩm nào ? Một ví dụ đơn giản nhất là sáng ra bạn phải đi ra chợ chuẩn bị thực phẩm để ăn trong cả ngày nhưng không biết nên mua gì để cả nhà ngon miệng. Hay cả gia đình muốn đi du lịch nhưng không biết nên chọn địa điểm nào hợp lý nhất với điều kiện gia đình mình.

Xuất phát từ những nhu cầu giảm tải những thông tin thừa không cần thiết đó, những hệ thống tư vấn (Recommender System) đã ra đời nhằm đem lại những trải nghiệm tốt nhất cho người dùng khi sử dụng. Những hệ thống tư vấn truyền thống có khả năng ước lượng một cách tương đối mức độ ưa thích của một người dùng đối với một đối tượng cụ thể nào đó (như điện thoại, tin tức, bộ phim, ca khúc, địa điểm du lịch,...). Từ đó hệ thống có thể đưa ra những tư vấn, những gợi ý những đối tượng người dùng có thể muốn sử dụng. Tuy nhiên sở thích của người dùng có thể thay đổi tùy theo điều kiện hoàn cảnh hiện tại. Chẳng hạn có người trời mưa thì muốn đi tham quan bảo tàng, còn trời nắng lại muốn đi dã ngoại. Hay khi đi một mình thì muốn đi thăm các di tích, thăng cảnh, còn khi đi với bạn bè lại muốn tới những khu mua sắm, trung tâm thương mại, khu vui chơi. Vai trò của ngũ cảnh rất quan trọng trong lĩnh vực tư vấn sản phẩm cho người dùng. Vì vậy yếu tố ngũ cảnh đã được nghiên cứu một cách nghiêm túc trong lĩnh vực nghiên cứu về hệ tư vấn [1] [2].

Đề tài: “*Nghiên cứu xây dựng hệ tư vấn đa phương tiện dựa trên ngũ cảnh*” được thực hiện với mục tiêu nghiên cứu tổng quan về hệ tư vấn đa phương tiện dựa trên ngũ cảnh và áp dụng xây dựng hệ tư vấn địa điểm du lịch tại thành phố Hồ Chí Minh dựa trên ngũ cảnh cho khách du lịch.

### 2. Mục tiêu của đồ án

Mục tiêu của đồ án này là:

- Tìm hiểu tổng quan về hệ tư vấn truyền thống và và hệ thống tư vấn thông tin đa phương tiện dựa trên ngũ cảnh.
- Nghiên cứu phương pháp mô hình hóa ngũ cảnh trong xây dựng hệ tư vấn đa phương tiện dựa trên ngũ cảnh.
- Áp dụng phương pháp mô hình hóa trong việc phát triển hệ thống tư vấn địa điểm du lịch tại thành phố Hồ Chí Minh dựa trên ngũ cảnh.

### 3. Bộ cục của đồ án

Quyển đồ án này bao gồm ba chương:

- **Chương 1: Tổng quan về hệ tư vấn đa phương tiện tích hợp yếu tố ngữ cảnh.** Nội dung chính của chương trình bày tập trung vào:
  - Những nghiên cứu cơ bản về các phương pháp lọc thông tin trong xây dựng hệ tư vấn truyền thống nói chung: lọc cộng tác, lọc nội dung, lọc kết hợp. Phân tích những ưu nhược điểm của mỗi phương pháp.
  - Nghiên cứu cơ bản về yếu tố ngữ cảnh và các phương pháp tích hợp ngữ cảnh vào bài toán tư vấn thông tin đa phương tiện.
- **Chương 2: Nghiên cứu phương pháp mô hình hóa ngữ cảnh CSLIM trong xây dựng hệ tư vấn đa phương tiện dựa trên ngữ cảnh.** Chương này sẽ trình bày những phần sau:
  - Tìm hiểu về một thuật toán tư vấn thông tin truyền thống sử dụng phương pháp mô hình hóa. Cụ thể là thuật toán tuyến tính thưa (Sparse Linear Method – SLIM). Thuật toán này sẽ được sử dụng để nghiên cứu phương pháp kết hợp với kỹ thuật tích hợp ngữ cảnh.
  - Tìm hiểu về kỹ thuật mô hình hóa ngữ cảnh dựa trên độ tương quan ngữ cảnh đa chiều và thuật toán kết hợp kỹ thuật mô hình hóa ngữ cảnh với thuật toán SLIM, thuật toán CSLIM\_MCS. Đây cũng chính là thuật toán sẽ sử dụng cho hệ thống tư vấn địa điểm du lịch tại thành phố Hồ Chí Minh.
  - Trình bày kết quả kiểm nghiệm thuật toán CSLIM\_MCS.
- **Chương 3: Phát triển hệ thống tư vấn địa điểm du lịch tại thành phố Hồ Chí Minh dựa trên ngữ cảnh.** Đây là chương cuối của quyển đồ án. Nội dung chương này tập trung vào hệ thống tư vấn địa điểm du lịch tại thành phố Hồ Chí Minh có tích hợp ngữ cảnh. Chương này trình bày thiết kế và xây dựng hệ tư vấn địa điểm du lịch tại thành phố Hồ Chí Minh dựa trên ngữ cảnh sử dụng phương pháp mô hình hóa được nghiên cứu trong chương 2.

# CHƯƠNG 1: TỔNG QUAN VỀ HỆ TƯ VẤN ĐA PHƯƠNG TIỆN TÍCH HỢP YẾU TỐ NGỮ CẢNH

Chương này sẽ trình bày tổng quan về hệ tư vấn truyền thống và những phương pháp tư vấn truyền thống, bao gồm hệ tư vấn dựa trên nội dung (Content-based Recommender System), hệ tư vấn dựa trên lọc cộng tác (Collaborative Filtering Recommender System) và hệ tư vấn lai (Hybrid Recommender System), đồng thời cũng phân tích ưu nhược điểm của những phương pháp này.

Chương này còn trình bày tổng quan về những kỹ thuật tích hợp yếu tố ngữ cảnh vào hệ tư vấn truyền thống nhằm tăng cường hiệu quả tư vấn.

## 1.1. Tổng quan về hệ tư vấn truyền thống

Hệ tư vấn là những hệ thống và công nghệ thông minh, có khả năng đưa ra những tư vấn về một sản phẩm nào đó (hàng hóa, phim, sách, bài hát, ...) cho một người dùng [1]. Quá trình đưa ra những tư vấn có thể coi là một quá trình đưa ra quyết định, như là gợi ý những bộ phim nào cho người dùng, gợi ý những bài hát nào, ... Những thuật toán của hệ tư vấn được sử dụng để đưa ra những gợi ý, những tư vấn hữu dụng cho người dùng một cách hiệu quả.

Hệ tư vấn là một mảng nghiên cứu lớn, đã được nghiên cứu trong thời gian dài. Những nhà nghiên cứu tập trung vào những phương pháp nhằm đưa ra những gợi ý chính xác cho người dùng. Những thuật toán của hệ tư vấn truyền thống được tạo ra nhằm khai thác mối quan hệ giữa những người dùng, những sản phẩm nhằm đưa ra những tư vấn hiệu quả nhất.

Bài toán hệ tư vấn sẽ khởi đầu là những đánh giá của người dùng với những sản phẩm. Những đánh giá này có thể là tường minh hoặc không tường minh. Ví dụ ta có tập dữ liệu đầu vào là danh sách những đánh giá tường minh của người dùng như sau:

- Người dùng A đánh giá phim “Doctor Stranger” 2/5 điểm.
- Người dùng A đánh giá phim “Suicide Squad” 4/5 điểm.
- Người dùng A đánh giá phim “Civil War” 4/5 điểm.
- Người dùng B đánh giá phim “Doctor Stranger” 3/5 điểm.
- Người dùng C đánh giá “Suicide Squad” 2/5 điểm.
- Người dùng C đánh giá “Civil War” 3/5 điểm.

Những đánh giá của người dùng thực tế thường rất thưa (còn rất nhiều phim mà người dùng chưa từng đánh giá). Từ những thông tin đầu vào như vậy, người ta thiết kế nên những thuật toán để dự đoán đánh giá của một người dùng với những sản phẩm mà họ chưa đánh giá. Như trong ví dụ trên thì hệ tư vấn sẽ dự đoán đánh giá của người dùng B với hai bộ phim chưa được đánh giá là “Suicide Squad” và “Civil War”. Những hệ tư vấn trước đây chỉ tập trung khai thác thông tin của hai đối tượng

là người dùng và sản phẩm, được gọi là những hệ tư vấn truyền thống. Công thức dưới mô tả bài toán của hệ tư vấn truyền thống:

$$R: Users \times Items \rightarrow Ratings$$

Trong đó *Users* là những người dùng, *Items* là những sản phẩm và *Ratings* là những đánh giá của những người dùng. Hệ tư vấn sẽ ước lượng một hàm *R* để tìm ra những giá trị đánh giá còn thiếu. Sau khi các giá trị đánh giá đã đầy đủ, hệ tư vấn sẽ đưa ra những gợi ý cho người dùng là những sản phẩm mà người đó chưa đánh giá và được dự đoán đánh giá tốt nhất:

$$\forall u \in Users, rec(u) = argmax(R(u, i))$$

Những hệ tư vấn truyền thống có thể chia thành ba nhóm chính: hệ tư vấn dựa trên nội dung, hệ tư vấn dựa trên lọc cộng tác và hệ tư vấn lai. Tác giả sẽ lần lượt trình bày về những hệ thống này.

### 1.1.1. Hệ tư vấn dựa trên nội dung

Những hệ thống tư vấn có khả năng đưa ra những gợi ý cho người dùng những sản phẩm mà hệ thống dựa đoán người dùng sẽ thích nó dựa trên những tương tác của người dùng trong quá khứ. Hệ tư vấn dựa trên nội dung là những hệ tư vấn có khả năng tính toán độ tương tự giữa những sản phẩm người dùng đã tương tác trước đó với những sản phẩm chưa được tương tác, từ đó đưa ra những gợi ý về những sản phẩm phù hợp nhất [1]. Trong phần này, tác giả sẽ trình bày tổng quan về hệ tư vấn dựa trên nội dung.

Để minh họa về hệ tư vấn dựa trên nội dung, tác giả sẽ lấy ví dụ một website bán sách. Để gợi ý cho người dùng *u*, hệ thống sẽ tìm ra những điểm tương đồng giữa những cuốn sách mà người dùng *u* đã thích, coi nó như sở thích cá nhân của người dùng *u*. Quá trình này chính là quá trình trích chọn đặc trưng (Feature Extraction). Từ đó sẽ chọn ra những cuốn sách mà người dùng *u* chưa có tương tác mà phù hợp nhất với sở thích của người dùng *u* đã tính toán được ở trên rồi đưa ra gợi ý.

Việc trích chọn đặc trưng được miêu tả bằng một hàm *Content(i)*, trả về một tập các thuộc tính đặc trưng của sản phẩm *i*. Đa phần những hệ tư vấn dựa trên nội dung được thiết kế dựa trên văn bản (text-based). Những hệ tư vấn này sẽ trích chọn đặc trưng dựa vào những văn bản mô tả sản phẩm. Những văn bản mô tả này sẽ được trích chọn ra những từ khóa bằng cách sử dụng những phương pháp đánh giá từ khóa như TF-IDF (Term Frequency – Inverse Document Frequency) [3]. Hàm đánh giá *R(u,i)* của hệ tư vấn tích hợp ngữ cảnh sẽ được viết lại như sau:

$$R(u, i) = sim(ContentBasedProfile(u), Content(i))$$

Trong đó *ContentBasedProfile(u)*, *Content(i)* là những vector đặc trưng mô tả đặc tính của người dùng *u* và sản phẩm *i*. Hàm *sim* là hàm đánh giá độ tương tự giữa hai vector đặc trưng, sau đó sẽ chọn ra danh sách những sản phẩm có độ tương đồng cao nhất để gợi ý cho người dùng *u*. Để tính độ tương đồng, ta có thể sử dụng một số công thức heuristic (đánh giá ước lượng) nhằm ước lượng độ tương đồng giữa hai vector như cosine coefficient hay pearson coefficient [4].

Ngoài những phương pháp heuristic, hệ tư vấn dựa trên nội dung còn có thể sử dụng những kỹ thuật học máy như phân lớp Bayes (Bayes Classifier [5]), những thuật toán gom cụm (Cluster[6]), mạng nơ ron nhân tạo (Artificial Neural Network [7]), ... Điểm khác biệt ở đây là những thuật toán học máy này sẽ tìm ra bộ tham số phù hợp dựa trên tập dữ liệu để dự đoán những trường hợp chưa biết chứ không chỉ sử dụng những công thức ước lượng.

Bằng cách sử dụng độ tương đồng giữa những sản phẩm dựa trên mô tả của chúng, những hệ tư vấn dựa trên nội dung có điểm mạnh sau:

- Người dùng độc lập với nhau nên việc tư vấn không quá đang ngại khi đối mặt với vấn đề dữ liệu thưa (dữ liệu ít đánh giá).
- Vấn đề sản phẩm mới dù có ít được đánh giá nhưng vẫn có thể được đem ra gợi ý do độ tương đồng với những sản phẩm trong quá khứ của người dùng.

Bên cạnh những điểm mạnh như vậy, hệ tư vấn dựa trên nội dung cũng có những mặt hạn chế:

- Sự đa dạng của những tư vấn là không nhiều. Ví dụ người dùng u đã thích sản phẩm  $i_1$ , hệ thống gợi ý sản phẩm  $i_2$  thì  $i_2$  chắc chắn sẽ giống  $i_1$  về thể loại, chủ đề. Điều này dẫn đến sự thiếu đa dạng trong tư vấn khi mà con người thích nhiều chủ đề chứ không chỉ tập trung ở một chủ đề.
- Những hệ tư vấn dựa trên nội dung thường là dựa vào văn bản mô tả, các thông tin dưới dạng văn bản. Trong khi những sản phẩm dạng hình ảnh, video thì văn bản mô tả chưa chắc đã mô tả chính xác nội dung đa phương tiện. Vì vậy hệ tư vấn sản phẩm đa phương tiện dựa trên nội dung sẽ không hoạt động hiệu quả.
- Vấn đề người dùng mới có ít đánh giá thì đặc trưng được trích chọn từ sở thích của người dùng u có thể không chính xác, không thể hiện được sở thích của người dùng u.

Như vậy hệ tư vấn dựa trên nội dung sẽ đạt hiệu quả cao đối với những sản phẩm được biểu diễn dưới dạng đặc trưng nội dung như tin tức, sách, báo, ..., nhưng lại gặp vấn đề với các dạng thông tin đa phương tiện – dạng thông tin không được biểu diễn dưới dạng đặc trưng nội dung. Vì vậy đối với những sản phẩm đa phương tiện, người ta thường sử dụng phương pháp lọc cộng tác cho hệ tư vấn nhằm tìm ra sở thích của cộng đồng ẩn để tư vấn. Phương pháp hệ tư vấn dựa trên lọc cộng tác sẽ được trình bày ở phần tiếp theo.

### **1.1.2. Hệ tư vấn dựa trên lọc cộng tác**

Hệ tư vấn dựa trên nội dung tập trung khai thác những khía cạnh liên quan đến đặc trưng nội dung mô tả sản phẩm đã được người dùng đánh giá trong quá khứ để gợi ý những sản phẩm tương tự. Còn hệ tư vấn dựa trên lọc cộng tác (Collaborative Filtering) thì không như vậy, hệ tư vấn dựa trên lọc cộng tác sẽ tập trung khai thác những thông tin tiềm ẩn trong tập dữ liệu đánh giá giữa người dùng và sản phẩm. Cụ thể là kỹ thuật lọc cộng tác sẽ giúp hệ tư vấn tìm ra những khuôn mẫu (pattern) về

hành vi tương tác của người dùng trong quá khứ [1]. Từ đó dự đoán hành vi của người dùng với những sản phẩm và đưa ra gợi ý những sản phẩm được dự đoán tốt.

So với hệ tư vấn dựa trên nội dung, hệ tư vấn dựa trên lọc cộng tác đơn giản hơn do chỉ phụ thuộc vào tập dữ liệu là đánh giá của người dùng mà không cần đến những thông tin bổ sung như thông tin về sản phẩm. Trong đồ án này tác giả tập trung nghiên cứu về hệ tư vấn dựa trên lọc cộng tác.

Bên cạnh những lợi thế của hệ tư vấn dựa trên lọc cộng tác so với dựa trên nội dung, những kỹ thuật lọc cộng tác cũng có những hạn chế:

- Bộ dữ liệu đầu vào thưa thớt sẽ ảnh hưởng rất nhiều tới hiệu quả của hệ tư vấn do quá trình tìm ra xu hướng của người dùng trong tập dữ liệu bị thiếu nhiều thông tin.
- Sự đa dạng trong hệ tư vấn lọc cộng tác cũng là một hạn chế. Mặc dù hệ tư vấn dựa trên lọc cộng tác có thể gợi ý sản phẩm đa dạng về thể loại, nhưng do việc tư vấn cho người dùng dựa trên sở thích của một tập những người dùng nên những sản phẩm được tư vấn sẽ chỉ tập trung ở một số chủ đề nhất định.

Hệ tư vấn dựa trên lọc cộng tác được tiếp cận theo hai hướng chính: phương pháp dựa trên bộ nhớ (Memory based) và phương pháp dựa vào mô hình thống kê (Model based).

Phương pháp dựa trên bộ nhớ tập trung xây dựng mối quan hệ giữa sản phẩm – sản phẩm (Item based) hoặc người dùng – người dùng (User based) bằng những công thức heuristic. Sau đó việc dự đoán đánh giá của người dùng sẽ dựa vào đánh giá của tất cả những người dùng hoặc những sản phẩm (sử dụng toàn bộ ma trận đánh giá để dự đoán). Độ tương đồng giữa những người dùng hay sản phẩm lúc này đóng vai trò như một trọng số. Những công thức heuristic được sử dụng là những công thức tính độ liên quan (correlation) như Pearson correlation (PC) [4], cosine correlation, euclidean distance, mean square deviation.

Khác với phương pháp dựa trên bộ nhớ, phương pháp dựa trên mô hình không chỉ sử dụng những công thức heuristic để xây dựng mối quan hệ giữa các đối tượng mà còn sử dụng những mô hình học máy nhằm tìm ra mối quan hệ ẩn trong tập dữ liệu thông qua những bước huấn luyện. Từ đó tìm ra bộ tham số thích hợp để có thể đưa ra những dự đoán chính xác. Quá trình huấn luyện chính là quá trình tìm ra bộ tham số để tối ưu độ chính xác của kết quả dự đoán.

Ưu điểm của phương pháp dựa trên mô hình là mô hình huấn luyện có kích thước nhỏ hơn rất nhiều so với toàn bộ ma trận đánh giá nên quá trình dự đoán sẽ nhanh hơn rất nhiều so với phương pháp dựa trên bộ nhớ. Ưu điểm thứ hai là phương pháp dựa trên mô hình tách biệt hoàn toàn hai quá trình huấn luyện và dự đoán. Vậy nên khi mô hình cần cập nhật lại thì chỉ cần cập nhật là quá trình huấn luyện.

### 1.1.2.1. Phương pháp dựa trên bộ nhớ

Cho ma trận R có n hàng và m cột tương ứng với n người dùng và m sản phẩm. Mỗi giá trị  $r_{ij}$  là đánh giá của người dùng i với sản phẩm j. Vấn đề là ma trận R còn thiếu rất nhiều đánh giá do một người dùng trên thực tế chỉ đánh giá một số lượng rất nhỏ sản phẩm so với tổng số sản phẩm. Vì vậy người ta đã xây dựng những thuật toán để dự đoán những đánh giá còn thiếu. Phương pháp lọc cộng tác dựa trên bộ nhớ sẽ dựa vào ma trận đánh giá R này để ước lượng độ tương đồng giữa những người dùng hay sản phẩm bằng những công thức heuristic, từ đó dự đoán những đánh giá còn thiếu dựa vào toàn bộ những đánh giá đã có trong ma trận R. Công thức đánh giá độ tương quan thường sử dụng là công thức Pearson.

#### 1.1.2.1.1. Độ tương quan Pearson

Pearson là một công thức heuristic, ước lượng độ tương đồng giữa hai vector. Nếu coi những sản phẩm người dùng  $u_1, u_2$  đã đánh giá là hai vector thì ta hoàn toàn có thể tính độ tương quan giữa hai người dùng  $u_1$  và  $u_2$  theo công thức Pearson:

$$p_{12} = \cos((\vec{u}_1 - \vec{b}_1), (\vec{u}_2 - \vec{b}_2)) = \frac{(\vec{u}_1 - \vec{b}_1) \cdot (\vec{u}_2 - \vec{b}_2)}{\|(\vec{u}_1 - \vec{b}_1)\| \cdot \|(\vec{u}_2 - \vec{b}_2)\|}$$

Trong đó:

$\vec{u}_1, \vec{u}_2$  là vector đánh giá của người dùng với tất cả sản phẩm (bằng 0 nếu chưa đánh giá)

$\vec{b}_1, \vec{b}_2$  là vector đánh giá trung bình của người dùng, tức là  
 $b_1 = \frac{\sum_i r_{1i}}{\|I\|}$  ( $\forall i \in I | r_{1i} > 0$ )

Công thức trên cũng dễ để nhận ra rằng bản chất đo độ tương quan Pearson là đo góc lệch giữa những vector người dùng. Tuy nhiên mỗi người dùng lại có dải đánh giá khác nhau. Chẳng hạn có người dễ tính đánh giá một sản phẩm tốt 5/5 điểm, sản phẩm kém 3 điểm, nhưng người khó tính chỉ đánh giá sản phẩm tốt 4/5 còn sản phẩm kém 1/5 điểm. Vậy nên vector người dùng sẽ được biểu diễn bằng hiệu số của đánh giá và trung bình đánh giá của người dùng đó như công thức trên. Để rõ ràng hơn, công thức Pearson còn được triển khai ra như sau:

$$p_{12} = \frac{\sum_{i \in I} (r_{1i} - b_1)(r_{2i} - b_2)}{\sqrt{\sum_{i \in I} (r_{1i} - b_1)^2} \cdot \sqrt{\sum_{i \in I} (r_{2i} - b_2)^2}}$$

Trong đó:

I là tập tất cả sản phẩm.

$r_{1i}, r_{2i}$  là đánh giá của người dùng  $u_1$  và  $u_2$  với sản phẩm i.

$b_1, b_2$  là đánh giá trung bình của người dùng  $u_1$  và  $u_2$ .

Như vậy với công thức Pearson, ta có thể đánh giá độ tương đồng giữa những người dùng (hoặc sản phẩm). Giá trị Pearson được sử dụng như một trọng số phục vụ cho quá trình dự đoán những đánh giá còn thiếu bằng thuật toán User-based hoặc Item-based.

#### **1.1.2.1.2. Thuật toán User-based và Item-based**

User-based và Item-based là hai thuật toán dự đoán dựa trên bộ nhớ. Ý tưởng của thuật toán này là tìm ra dựa vào độ tương đồng giữa những người dùng (User-based) hoặc những sản phẩm (Item-based) để ước lượng đánh giá của một người dùng với một sản phẩm. Sau đó sẽ đưa ra gợi ý cho người dùng những sản phẩm mà họ chưa đánh giá mà có dự đoán đánh giá cao nhất.

Người ta thường sử dụng công thức Pearson để tính toán độ tương quan giữa từng cặp đôi tượng (người dùng hoặc sản phẩm), sau đó với từng đối tượng chọn ra K đối tượng khác có độ tương quan cao nhất để dự đoán những đánh giá còn thiếu.

Đối với User-based, sau khi tính được độ tương quan giữa các cặp người dùng, việc dự đoán sẽ được tính toán theo công thức sau:

$$\bar{R}_{ui} = b_u + \frac{\sum_{v \in U} p_{uv} \cdot (r_{vi} - b_v)}{\sum_{v \in U} |p_{uv}|}$$

Trong đó:

$\bar{R}_{ui}$  là dự đoán đánh giá của người dùng u với sản phẩm i.

$b_u, b_v$  là đánh giá trung bình của người dùng u và người dùng v.

$p_{uv}$  là độ tương quan pearson giữa người dùng u và người dùng v.

$r_{vi}$  là đánh giá của người dùng v với sản phẩm i.

U là tập tất cả những người dùng có trong tập dữ liệu.

Công thức trên có thể hiểu là dựa vào đánh giá của những người dùng khác, tính ra độ ưa thích của những người dùng này với sản phẩm bằng cách lấy hiệu giá trị đánh giá thật và đánh giá trung bình. Giá trị này có thể coi là độ lệch so với đánh giá trung bình, và sau đó đánh giá dự đoán sẽ được tính bằng tổng độ lệch và đánh giá trung bình.

Tương tự đối với Item-based:

$$\bar{R}_{ui} = b_i + \frac{\sum_{v \in I} p_{iv} \cdot (r_{uv} - b_v)}{\sum_{v \in I} |p_{iv}|}$$

Trong đó:

$\bar{R}_{ui}$  là dự đoán đánh giá của người dùng u với sản phẩm i.

$b_i, b_v$  là trung bình đánh giá của sản phẩm i và sản phẩm v.

$p_{iv}$  là độ tương quan pearson giữa sản phẩm i và sản phẩm v.

$r_{uv}$  là đánh giá của người dùng  $u$  với sản phẩm  $v$ .

$I$  là tập tất cả sản phẩm.

### 1.1.2.2. Phương pháp dựa trên mô hình

Khác với phương pháp dựa trên bộ nhớ, thay vì sử dụng các công thức heuristic tính toán độ tương đồng giữa các đối tượng để đưa ra dự đoán, kỹ thuật lọc cộng tác dựa trên mô hình sẽ sử dụng những kỹ thuật học máy và các thuật toán khai phá dữ liệu để nhận ra các mối quan hệ phức tạp thông qua những bước huấn luyện dữ liệu, sau đó mới đưa ra những gợi ý thông minh.

Cũng tương tự những phương pháp dựa trên bộ nhớ, phương pháp dựa trên mô hình sử dụng ma trận đánh giá  $R$  giữa những người dùng và những sản phẩm. Những thuật toán dự đoán sử dụng mô hình sẽ sử dụng công thức ước lượng đánh giá của người dùng khác nhau với những tham số (mô hình). Quá trình huấn luyện thực chất là tìm ra bộ tham số (mô hình) để công thức ước lượng đánh giá đạt giá trị chính xác nhất. Độ chính xác trong quá trình huấn luyện sử dụng những hàm lỗi (loss function) để đánh giá độ lỗi rồi cập nhật các tham số cho phù hợp. Tóm những thuật toán dự đoán dựa trên mô hình sẽ chia rõ ràng thành hai quá trình riêng biệt:

- *Quá trình dự đoán:* Dựa vào bộ tham số tìm được và công thức ước lượng để tính toán đánh giá dự đoán của người dùng với sản phẩm.
- *Quá trình huấn luyện:*
  - Khởi tạo ngẫu nhiên các tham số (mô hình).
  - Lặp các bước sau:
    - Thực hiện quá trình dự đoán cho tất cả người dùng và sản phẩm.
    - Đánh giá độ lỗi của bộ tham số hiện tại dựa vào những dự đoán trên.
    - Nếu độ lỗi đã quá nhỏ (so với một ngưỡng nào đó) thì dừng lại và bộ tham số sử dụng đã là bộ tham số tối ưu. Nếu độ lỗi vẫn còn lớn thì sử dụng những thuật toán tối ưu tham số để lần lặp tiếp theo đạt độ lỗi ít hơn.

Có thể thấy rõ ràng việc huấn luyện mô hình thực chất là tìm ra tham số để đạt kết quả dự đoán chính xác nhất. Một thuật toán tối ưu tham số thường được sử dụng trong lĩnh vực hệ tư vấn nói riêng và học máy nói chung là thuật toán Stochastic Gradient Descent (SGD).

SGD (Stochastic Gradient Descent [8]) là kỹ thuật tối ưu một hàm mục tiêu bằng cách giảm độ dốc của phương trình tiếp tuyến của hàm mục tiêu. Ý tưởng của kỹ thuật này là tìm ra tham số tối ưu cục bộ để có kết quả tối ưu toàn cục. Quá trình tìm tham số tối ưu là những vòng lặp và tại mỗi bước lặp, ta sẽ cập nhật ước lượng các tham số để tối ưu hàm mục tiêu cục bộ theo công thức:

$$w = w - \mu \cdot \frac{\partial L}{\partial w}$$

Trong đó:

$w$  là tham số cần tìm để tối ưu  $L$ .

$L$  là hàm mục tiêu cần tối ưu.

$\mu$  là tham số tỉ lệ học, quyết định tốc độ học.

$\frac{\partial L}{\partial w}$  là giá trị độ dốc (gradient) của phương trình tiếp tuyến.

Ví dụ ta có hàm dự đoán  $\tilde{y}_i = w_1 + w_2 \cdot x_i$ , trong đó  $w_1$  và  $w_2$  là hai tham số cần ước lượng. Và ta cũng có hàm đánh giá lỗi của dự đoán là hàm lỗi toàn phương trung bình (Mean Square Error [9]):  $L = \frac{1}{N} \sum_{i=1}^N (y_i - \tilde{y}_i)^2$ , trong đó  $\tilde{y}_i$  là giá trị dự đoán còn  $y_i$  là giá trị thực tế theo tập dữ liệu.

Như vậy, cách tính độ dốc sẽ là:

$$\frac{\partial L_i}{\partial w_1} = 2(y_i - w_1 + w_2 \cdot x_i)$$

$$\frac{\partial L_i}{\partial w_2} = 2x_i(y_i - w_1 + w_2 \cdot x_i)$$

Và qua mỗi bước lặp, ta sẽ cập nhật các tham số như sau:

$$w_1 = w_1 - 2\mu(y_i - w_1 + w_2 \cdot x_i)$$

$$w_2 = w_2 - 2\mu x_i(y_i - w_1 + w_2 \cdot x_i)$$

Quá trình lặp sẽ kết thúc sau một số bước lặp do ta định trước hoặc độ thay đổi của hàm mục tiêu giữa hai lần lặp là rất bé.

Kỹ thuật SGD rất thường được sử dụng trong lĩnh vực máy học như trong những mô hình hồi quy [10], mô hình mạng nơ ron [7], gom cụm [6], ... Ưu điểm của phương pháp dựa trên mô hình là tách biệt hai quá trình huấn luyện và dự đoán nên nếu có sự cập nhật mô hình thì chỉ cần cập nhật ở bước huấn luyện. Ưu điểm thứ hai là quá trình dự đoán không dựa trên toàn bộ dữ liệu như phương pháp dựa trên bộ nhớ nên tốc độ sẽ nhanh hơn.

Trong phần này, tác giả đã trình bày tổng quan về phương pháp tư vấn dựa trên mô hình. Chi tiết thuật toán sử dụng phương pháp dựa trên mô hình sẽ được trình bày ở chương tiếp theo.

### 1.1.2.3. Đánh giá chung về phương pháp lọc cộng tác

So với hệ tư vấn dựa trên nội dung, hệ tư vấn dựa trên lọc cộng tác có những ưu điểm vượt trội hơn trong bài toán tư vấn sản phẩm đa phương tiện do những sản phẩm này không được biểu diễn rõ dưới dạng đặc trưng nội dung. Tuy nhiên phương pháp lọc cộng tác cũng có những mặt hạn chế:

- *Sự thưa thớt về dữ liệu:* Khi tập dữ liệu có quá ít đánh giá hoặc cần tư vấn cho người dùng trong khi có rất ít thông tin về người đó thì việc khai thác để tìm ra sở thích của họ là rất khó khăn, độ sai lệch sẽ là rất cao. Chẳng

hạn như đối với trường hợp “người dùng mới” và “sản phẩm mới”. Hệ thống không có thông tin về những đối tượng mới thì làm sao có thể tìm ra mối tương quan với những đối tượng khác. Để hạn chế trường hợp này, người ta đã nghiên cứu ra những kỹ thuật thu giảm số chiều như Singular Value Decomposition (SVD) [14], Latent Semantic Indexing (LSI) [15].

- *Vấn đề những người dùng cá biệt:* Có những người dùng có những sở thích không giống bất cứ một nhóm người dùng nào. Điều này sẽ gây ảnh hưởng tới chất lượng của hệ thống khi tư vấn cho những người dùng này. Để giải quyết vấn đề này, người ta sử dụng hệ tư vấn lai.
- *Vấn đề sở thích người dùng thay đổi theo ngữ cảnh:* Đánh giá của người dùng phụ thuộc rất nhiều vào tâm trạng của họ khi đánh giá cũng như những yếu tố ngoại cảnh. Chẳng hạn trời mưa thì họ thích nghe nhạc buồn, trời nắng nhẹ lại thích nghe nhạc mạnh mẽ, vui tươi. Để có thể đưa ra gợi ý tốt nhất, người ta nghiên cứu các kỹ thuật tư vấn kết hợp yếu tố ngữ cảnh [2].

### 1.1.3. Hệ tư vấn lai

Như đã đề cập ở trên, hệ tư vấn dựa trên nội dung có nhược điểm là việc tính toán đưa ra gợi ý dựa trên văn bản mô tả sản phẩm. Nếu văn bản mô tả không chính xác thì sẽ ảnh hưởng rất lớn tới chất lượng gợi ý. Đặc biệt là đối với loại dữ liệu như hình ảnh, âm thanh, loại dữ liệu khó có thể có văn bản mô tả chính xác. Còn chất lượng hệ tư vấn lọc cộng tác sẽ bị ảnh hưởng rất lớn nếu bộ dữ liệu thưa (có ít đánh giá). Bộ dữ liệu thưa khiến những mô hình không bắt được đúng sở thích của từng nhóm người dùng.

Để giải quyết vấn đề trên, người ta kết hợp ưu điểm của hai phương pháp lọc cộng tác và lọc nội dung để tạo nên hệ tư vấn lai. Hệ tư vấn lai chính là hệ tư vấn kết hợp phương pháp lọc nội dung và phương pháp lọc cộng tác nhằm khai thác những điểm mạnh của hai phương pháp này. Ví dụ hệ thống gợi ý phim rút trích đặc trưng từ thông tin cá nhân người dùng, sau đó tìm ra nhóm người dùng có sở thích giống nhau rồi gợi ý bằng phương pháp lọc cộng tác. Hoặc cũng có thể thực hiện riêng rẽ hai phương pháp rồi tổng hợp kết quả lại để gợi ý tới người dùng.

Người ta cũng đã nghiên cứu và nhận xét rằng hệ tư vấn lai sẽ cho ra kết quả gợi ý tốt hơn so với riêng rẽ từng hệ tư vấn dựa trên nội dung hay hệ tư vấn dựa trên lọc cộng tác thuần túy [1].

### 1.1.4. Những phương pháp đánh giá độ chính xác của hệ tư vấn

Trong lĩnh vực truy vấn thông tin (information retrieval), người ta đặt ra các chỉ số để đánh giá tính đúng đắn của kết quả truy vấn như precision@k, recall@k, AP@k, MAP@k, F-measure, ... Đối với hệ thống gợi ý, mục tiêu là đưa ra những gợi ý chính xác tới người dùng, vậy nên cũng có thể sử dụng những chỉ số trên để đánh giá kết quả gợi ý. Tác giả sẽ lần lượt trình bày ba chỉ số được sử dụng phổ biến nhất trong bài toán của hệ thống gợi ý là precision@k, recall@k và MAP@k.

#### 1.1.4.1. Precision

Precision là một chỉ số đánh giá kết quả truy vấn có chính xác hay không dựa vào tỉ lệ giữa số lượng kết quả chính xác và tổng số kết quả. Nói một cách đơn giản, chỉ số Precision sẽ trả lời câu hỏi Ví dụ hệ thống đưa ra 100 kết quả thì có 2 kết quả người dùng lựa chọn vậy precision sẽ bằng 0,02.

$$Prec = \frac{|P_{true}|}{|P|}$$

Trong đó:

Prec : Chỉ số Precision.

$|P_{true}|$  : Tổng số lượng những kết quả chính xác.

$|P|$  : Tổng số lượng kết quả.

Tuy nhiên trên thực tế thì số lượng kết quả trả về sẽ rất lớn mà người dùng không cần thiết lựa chọn hết nên chỉ số precision không còn mấy ý nghĩa. Vì vậy người ta thường sử dụng chỉ số Precision@k để có thể đánh giá kết quả trả về chuẩn xác hơn. Precision@k khác với Precision ở chỗ chỉ lựa chọn top k kết quả chính xác nhất sử dụng để đánh giá.

$$Prec@k = \frac{|P_{true}|}{\min(k, |P|)}$$

Trong đó:

Prec : Chỉ số Precision.

$|P_{true}|$  : Tổng số lượng những kết quả chính xác.

$|P|$  : Tổng số lượng kết quả.

k : Tham số được lựa chọn để đánh giá kết quả.

Đồ án này sẽ lựa chọn chỉ số Precision@10 để đánh giá. Điều này có nghĩa là Precision@10 sẽ thể hiện rằng cứ 10 gợi ý tới người dùng thì sẽ có bao nhiêu gợi ý được người dùng lựa chọn.

#### 1.1.4.2. MAP (Mean Average Precision)

MAP là một chỉ số đánh giá phổ biến khác của hệ tư vấn. Ý tưởng của MAP là ngoài vấn đề thể hiện sự hiệu quả của những gợi ý như Precision, MAP còn thể hiện tính đúng đắn về thứ hạng của những gợi ý. Chẳng hạn như đối với Precision, người dùng chỉ lựa chọn một trong 10 gợi ý thì dù đó là gợi ý thứ 1 hay thứ 10 thì giá trị Precision vẫn là 0,1. Nhưng đối với MAP thì khác. Nếu người dùng lựa chọn gợi ý thứ 1 thì giá trị MAP sẽ lớn hơn nếu người dùng lựa chọn gợi ý thứ 10.

Cũng như Precision, người ta thường sử dụng MAP@k với k là tham số lựa chọn để đánh giá những gợi ý. Công thức tính MAP@k như sau:

$$MAP@k = \sum_1^n \frac{AP@k}{n}$$

$$AP@k = \sum_{i=1}^k \frac{Prec@i}{\min(k, M)}$$

Trong đó:

$AP@k$  : Chỉ số đánh giá hiệu quả của gợi ý đối với từng người dùng.

$Prec@i$  : Chỉ số precision nếu chỉ sử dụng top  $i$  gợi ý.

$M$  : Tổng số lựa chọn của người dùng đang xét.

$k$  : Tham số được lựa chọn để đánh giá kết quả.

$n$  : Tổng số lượng người dùng.

Trong công thức trên, bằng cách tính  $AP@k$ , chỉ số precision sẽ được lấy trung bình tại từng vị trí từ 1 tới  $k$ . Điều này nhằm chắc chắn rằng người dùng lựa chọn gợi ý có thứ hạng càng tốt thì  $AP@k$  càng cao. Sau đó MAP sẽ được tính bằng trung bình của  $AP@k$  đối với từng người dùng như công thức.

#### 1.1.4.3. Recall

Recall cũng là một chỉ số đánh giá phổ biến. Khác với precision, recall thể hiện xác suất gợi ý thành công của hệ thống, được tính bằng công thức sau:

$$Recall@k = \frac{|P_{true}|}{\min(k, M)}$$

Trong đó:

$|P_{true}|$  : Tổng số lượng những kết quả chính xác.

$M$  : Tổng số lượng lựa chọn của người dùng.

$k$  : Tham số được lựa chọn để đánh giá kết quả.

## 1.2. Tổng quan về hệ tư vấn đa phương tiện kết hợp ngữ cảnh

Trong phần trước, tác giả đã trình bày tổng quan về hệ tư vấn truyền thống với ba phương pháp là lọc nội dung, lọc cộng tác và hệ tư vấn lai. Nhưng dù sử dụng phương pháp nào thì hệ tư vấn truyền thống vẫn chỉ quan tâm tới những đối tượng của hệ tư vấn là người dùng (user), sản phẩm (item) và những bình luận của người dùng với sản phẩm. Tuy nhiên trên thực tế, sở thích của người dùng lại không cố định. Ví dụ một người trời nóng thì thích ăn kem, uống sinh tố, nhưng khi trời lạnh lại thích ăn phở, uống cà phê nóng. Hoặc cùng một bộ phim nhưng trời mưa thì thích xem còn trời khô ráo thì có khi lại không thích. Có thể nói sở thích của người dùng bị tác động nhiều bởi những yếu tố ngữ cảnh bên ngoài. Trong chương này tác giả sẽ

trình bày về yếu tố ngữ cảnh (context) trong hệ tư vấn và giới thiệu khái quát những phương pháp kết hợp yếu tố ngữ cảnh vào một hệ tư vấn truyền thống.

Đầu tiên ta cần hiểu khái niệm ngữ cảnh là gì. Theo như [1]: “Thông tin ngữ cảnh là những thông tin có thể mô tả được hoàn cảnh của một thực thể. Thực thể ở đây có thể là người, là vật hoặc là đối tượng có liên quan tới sự tương tác giữa người dùng và ứng dụng, bao gồm cả bản thân người dùng và ứng dụng đó.” Để rõ ràng hơn, ta có thể lấy ví dụ những yếu tố như thời tiết, địa điểm, tâm trạng, ... đều là những thông tin ngữ cảnh có thể ảnh hưởng tới đánh giá của người dùng tới một bộ phim. Chính vì vậy nên yếu tố ngữ cảnh được nghiên cứu rộng rãi trong lĩnh vực hệ tư vấn [2].

Đối với hệ tư vấn truyền thống, ta chỉ quan tâm tới hai đối tượng là người dùng là sản phẩm để đưa ra dự đoán. Có thể mô phỏng bài toán của hệ tư vấn truyền thống như sau:

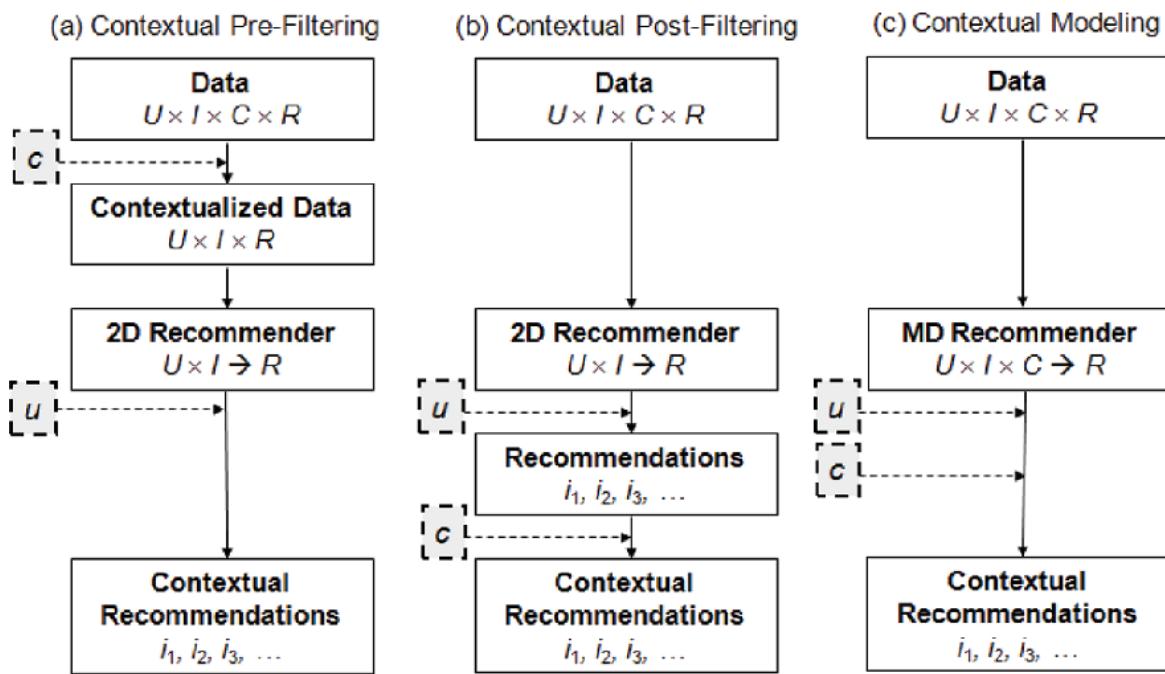
$$R: Users \times Items \rightarrow Ratings$$

Còn đối với hệ tư vấn kết hợp ngữ cảnh, ngoài thông tin về hai đối tượng người dùng và sản phẩm, hệ thống còn quan tâm tới những yếu tố ngữ cảnh khi người dùng đánh giá một sản phẩm. Bài toán của hệ tư vấn lúc này sẽ như sau:

$$R: Users \times Items \times Contexts \rightarrow Ratings$$

Chẳng hạn đối với hệ tư vấn du lịch, yếu tố ngữ cảnh có thể là thời gian (buổi trong ngày, thời gian trong tuần, mùa), bạn đồng hành (một mình, gia đình, bạn bè). Những yếu tố này hoàn toàn có thể ảnh hưởng tới quyết định chọn địa điểm du lịch của người dùng. Hệ tư vấn sẽ đóng vai trò ghi nhớ lại sở thích của người dùng theo ngữ cảnh để đưa ra những gợi ý chính xác nhất.

Để tích hợp thông tin ngữ cảnh vào hệ tư vấn truyền thống, người ta đã nghiên cứu ra hai phương pháp: phương pháp lọc ngữ cảnh và phương pháp mô hình hóa ngữ cảnh. Phương pháp lọc ngữ cảnh có thể chia thành hai hướng là lọc trước khi dự đoán hoặc sau khi dự đoán. Hình dưới mô tả những cách để tích hợp yếu tố ngữ cảnh trong hệ tư vấn.



Hình ảnh 1: Tổng quan ba loại phương pháp kết hợp ngữ cảnh

Phương pháp lọc ngữ cảnh sẽ lọc các yếu tố ngữ cảnh không phù hợp trước hoặc sau khi đưa ra gợi ý bằng cách sử dụng phương pháp tư vấn truyền thống. Chẳng hạn đối với hệ thống gợi ý phim có tích hợp yếu tố ngữ cảnh bằng phương pháp lọc ngữ cảnh, bao gồm những yếu tố ngữ cảnh như thời gian trong tuần (ngày trong tuần, cuối tuần) và xem cùng ai (một mình, gia đình, bạn bè):

- Nếu áp dụng phương pháp lọc trước khi gợi ý, hệ thống sẽ lấy ngữ cảnh hiện tại của người dùng để so sánh với từng đánh giá đã có trong ngữ cảnh như thế, từ đó loại bỏ những đánh giá ở ngữ cảnh khác. Ví dụ như người dùng đăng nhập vào cuối tuần và sắp xem phim với bạn bè, hệ thống sẽ loại bỏ tất những đánh giá với điều kiện ngữ cảnh như người dùng vừa đăng nhập. Sau đó bài toán sẽ trở thành bài toán gợi ý truyền thống với hai đối tượng là người dùng và sản phẩm.
- Nếu áp dụng phương pháp lọc sau khi gợi ý, hệ thống sẽ áp dụng kỹ thuật tư vấn truyền thống để đưa ra những gợi ý. Sau đó sẽ tiến hành lọc loại bỏ những gợi ý mà không phù hợp với ngữ cảnh hiện tại của người dùng. Ví dụ hệ thống gợi ý phim biết người dùng chỉ xem phim kinh dị khi xem với bạn bè thì sẽ gợi ý phim kinh dị chỉ khi người dùng đang chuẩn bị xem phim với bạn.

Kỹ thuật kết hợp ngữ cảnh bằng phương pháp mô hình hóa thì khác hoàn toàn với phương pháp lọc ngữ cảnh. Thay vì loại bỏ thông tin dựa vào ngữ cảnh trước hoặc sau khi dự đoán, phương pháp mô hình sẽ tích hợp yếu tố ngữ cảnh vào quá trình xây dựng mô hình dự đoán (hay quá trình huấn luyện dữ liệu). Như vậy những thông tin sẽ không bị mất đi mà sẽ được tính toán cẩn thận để đưa ra tư vấn trong quá trình huấn luyện.

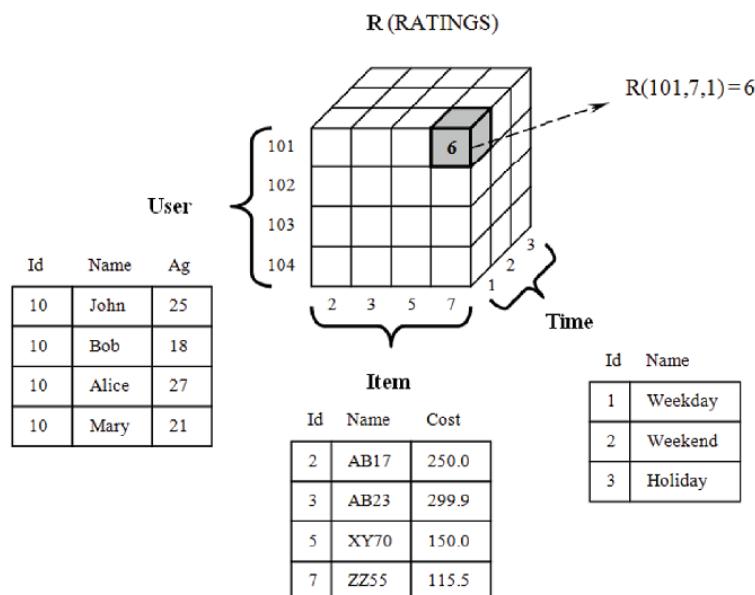
Phương pháp lọc ngữ cảnh thì đơn giản hơn, dễ cài đặt hơn. Tuy nhiên việc loại bỏ dựa vào ngữ cảnh có thể sẽ loại bỏ đi những thông tin quan trọng để có thể đưa ra gợi ý tốt. Hay đối với dữ liệu thưa thì việc loại bỏ thông tin sẽ càng khiến dữ liệu thưa hơn, dẫn tới hệ thống không thể “bắt” được sở thích cá nhân của người dùng. Chính vì vậy phương pháp tích hợp ngữ cảnh bằng cách mô hình hóa sẽ đem lại hiệu quả tốt hơn. Đồ án này sẽ tập trung nghiên cứu kỹ thuật tích hợp yếu tố ngữ cảnh vào hệ tư vấn truyền thông bằng phương pháp mô hình hóa ngữ cảnh.

Những kỹ thuật tích hợp yếu tố ngữ cảnh bằng phương pháp mô hình hóa có thể chia làm hai nhóm là: Mô hình hóa độc lập và mô hình hóa phụ thuộc.

### 1.2.1. Phương pháp mô hình hóa ngữ cảnh độc lập

Phương pháp mô hình hóa ngữ cảnh độc lập là phương pháp xây dựng mô hình ngữ cảnh hoàn toàn độc lập với xây dựng mô hình dự đoán truyền thống, tức là mô hình ngữ cảnh sẽ không phụ thuộc vào những yếu tố như người dùng và sản phẩm.

Yếu tố ngữ cảnh trong phương pháp mô hình hóa ngữ cảnh độc lập trở thành một chiều trong không gian đa chiều bao gồm người dùng, sản phẩm và ngữ cảnh. Từ đó mỗi đánh giá của người dùng sẽ là một giá trị trong không gian nhiều chiều này như hình dưới.



Hình ảnh 2: Phương pháp mô hình hóa ngữ cảnh độc lập

Đối với phương pháp mô hình hóa ngữ cảnh nói chung, thuật toán huấn luyện sẽ phải tìm ra vector đặc trưng biểu diễn cho từng ngữ cảnh cụ thể. Phương pháp mô hình hóa ngữ cảnh độc lập chỉ khác phương pháp mô hình hóa phụ thuộc ở chỗ quá trình tối ưu hàm mục tiêu để tìm ra vector ngữ cảnh hoàn toàn không phụ thuộc vào vector đặc trưng của người dùng hay sản phẩm. Để làm được điều này, người ta thường sử dụng những kỹ thuật phân rã ma trận đa chiều thành những chiều riêng rẽ, và coi mỗi chiều như một đặc trưng của chiều không gian đó. Ví dụ có 4 chiều là người dùng, sản phẩm, thời gian và thời tiết. Thuật toán sẽ tìm ra vector đặc trưng

của người dùng, sản phẩm, thời gian và thời tiết để có thể đưa ra dự đoán. Thuật toán Tensor Factorization (TF) [16] chính là một ví dụ điển hình của phương pháp xây dựng mô hình ngữ cảnh độc lập.

So với phương pháp lọc ngữ cảnh thì phương pháp tích hợp ngữ cảnh độc lập đã có thể bảo tồn những thông tin mà kỹ thuật lọc ngữ cảnh có thể sẽ loại bỏ. Còn so với phương pháp mô hình hóa ngữ cảnh phụ thuộc thì phương pháp mô hình hóa độc lập đơn giản hơn, dễ dàng tích hợp hơn.

Tuy nhiên phương pháp mô hình hóa độc lập cũng có những nhược điểm. Vì quá trình huấn luyện độc lập giữa ngữ cảnh và các đối tượng khác nên những phương pháp này đã bỏ qua sự phụ thuộc giữa yếu tố ngữ cảnh với người dùng và sản phẩm. Ví dụ người dùng A chỉ đánh giá phim tốt vào buổi tối, còn buổi sáng thì đánh giá phim kém, A thích đi du lịch vào mùa đông còn B lại thích đi vào mùa hạ. Những kỹ thuật mô hình hóa độc lập hoàn toàn bỏ qua mối quan hệ giữa người dùng hay sản phẩm với yếu tố ngữ cảnh. Một vấn đề nữa là việc phân rã ma trận đa chiều thành từng chiều riêng rẽ sẽ tiêu tốn rất nhiều tài nguyên nếu có quá nhiều chiều. Chẳng hạn có nhiều loại ngữ cảnh như thời gian, thời tiết, địa điểm, đi cùng với ai, .... Những kỹ thuật tích hợp ngữ cảnh bằng phương pháp phụ thuộc đã được nghiên cứu để giải quyết những vấn đề trên

### 1.2.2. Phương pháp mô hình hóa ngữ cảnh phụ thuộc

Khác với phương pháp mô hình hóa ngữ cảnh độc lập, phương pháp mô hình hóa ngữ cảnh độc lập sẽ huấn luyện để tìm ra vector đặc ngữ cảnh đồng thời với huấn luyện mô hình tư vấn truyền thống. Người ta đã nghiên cứu ra hai hướng xây dựng mô hình hóa ngữ cảnh phụ thuộc. Đó là phương pháp dựa vào độ sai lệch khi có ngữ cảnh [17] và phương pháp dựa vào độ tương quan ngữ cảnh [18]. Nghiên cứu [18] đã chứng minh phương pháp dựa vào độ tương quan ngữ cảnh đem lại kết quả tốt hơn cho bài toán gọi ý top-N. Vì vậy trong giới hạn của đề tài, tác giả sẽ chỉ đề cập tới phương pháp mô hình hóa ngữ cảnh phụ thuộc dựa trên độ tương quan ngữ cảnh.

Ý tưởng của phương pháp mô hình hóa ngữ cảnh phụ thuộc dựa trên độ tương quan là trong những ngữ cảnh giống nhau thì danh sách sản phẩm gợi ý cũng tương tự như nhau. Công thức của dự đoán của phương pháp này là:

$$\tilde{R}_{uic} = \tilde{R}_{ui} \cdot Sim(c_\emptyset, c)$$

Trong đó:

$\tilde{R}_{uic}$  là đánh giá dự đoán giữa người dùng u với sản phẩm i trong ngữ cảnh c.

$\tilde{R}_{ui}$  là đánh giá dự đoán giữa người dùng u với sản phẩm i không xét ngữ cảnh.

$Sim(c_\emptyset, c)$  là độ tương quan giữa ngữ cảnh c và trường hợp không có ngữ cảnh. Độ tương quan này sẽ là bằng 1 nếu  $c = c_\emptyset$

Trong công thức trên, dự đoán  $\tilde{R}_{ui}$  vẫn được tính toán dựa vào những phương pháp tư vấn truyền thống như đã trình bày ở phần trước. Tuy nhiên đó mới chỉ là dự đoán đánh giá của người dùng u với sản phẩm i khi chưa xét yếu tố ngữ cảnh. Giá trị vừa tính toán được sẽ được nhân với độ tương quan ngữ cảnh hiện tại với trường hợp không rõ ngữ cảnh  $Sim(c_\emptyset, c)$ . Có thể thấy rằng trong cùng ngữ cảnh thì kết quả tư vấn sẽ giống nhau do giá trị tương quan là như nhau, ngược lại nếu ngữ cảnh khác nhau thì dù cùng một người dùng cũng sẽ cho kết quả tư vấn khác nhau. Trong trường hợp cần tư vấn khi không rõ ngữ cảnh thì giá trị  $Sim(c_\emptyset, c) = 1$ , do đó kết quả tư vấn sẽ là kết quả của hệ tư vấn truyền thống.

Vấn đề đặt ra lúc này là làm sao để có thể tính độ tương quan ngữ cảnh. Chính xác là làm sao có thể biểu diễn các vector ngữ cảnh một cách phù hợp. Và quá trình tìm ra những vector biểu diễn ngữ cảnh chính là quá trình huấn luyện tìm ra mô hình ngữ phù hợp. Nghiên cứu [18] đã đề xuất hai phương pháp để tính độ tương quan, đó là: Độ tương quan ngữ cảnh độc lập và độ tương quan ngữ cảnh Latent

### 1.2.2.1. Độ tương quan ngữ cảnh độc lập

Độ tương quan ngữ cảnh độc lập là phép ước lượng độ tương quan giữa hai ngữ cảnh cụ thể, trong đó từng điều kiện trong ngữ cảnh này sẽ được so sánh với từng điều kiện tương ứng trong ngữ cảnh kia một cách độc lập, sau đó sẽ ước lượng độ tương quan giữa hai ngữ cảnh theo công thức sau:

$$Sim(c_k, c_l) = \prod_{i=1}^D similarity(c_{ki}, c_{li})$$

Trong đó:

$Sim(c_k, c_l)$  là độ tương quan giữa ngữ cảnh  $c_k$  và ngữ cảnh  $c_l$ .

$D$  là số chiều của ngữ cảnh.

$similarity(c_{ki}, c_{li})$  là độ tương quan giữa chiều thứ  $i$  giữa  $c_k$  và ngữ cảnh  $c_l$ .

Ví dụ ta có ngữ cảnh bao gồm những điều kiện về thời gian và thời tiết. Và ta cần tính toán độ tương quan giữa ngữ cảnh c1(cuối tuần, lạnh) và c2(ngày thường, nóng). Bảng dưới mô tả độ tương quan giữa những giá trị trong từng điều kiện tương ứng. Do độ tương quan ngữ cảnh độc lập chỉ so sánh những giá trị trong cùng một trường điều kiện (thời gian chỉ so sánh với thời gian, thời tiết chỉ so sánh với thời tiết) nên không tồn tại giá trị giữa những giá trị về thời gian và những giá trị về thời tiết. Những giá trị độ tương quan giữa những điều kiện ngữ cảnh trong cùng một trường sẽ được tìm ra trong quá trình huấn luyện tham số (những giá trị này chính là một trong những tham số cần tìm). Giả sử sau quá trình huấn luyện, ta có bảng giá trị như sau:

	Ngày thường	Cuối tuần	Lạnh	Nóng
Ngày thường	1	0.43	n/a	n/a

Cuối tuần	0.43	1	n/a	n/a
Lạnh	n/a	n/a	1	0.78
Nóng	n/a	n/a	0.78	1

Bảng 1: Ví dụ về độ tương quan ngữ cảnh độc lập

Độ tương quan giữa c1(cuối tuần, lạnh) và c2(ngày thường, nóng) trong ví dụ trên sẽ được tính toàn như sau:

$$\begin{aligned} \text{Sim}(c_1, c_2) &= \text{similarity}(\text{cuối tuần}, \text{ngày thường}) \times \text{similarity}(\text{lạnh}, \text{nóng}) \\ &= 0.43 \times 0.78 \\ &= 0.3354 \end{aligned}$$

Độ tương quan độc lập có một nhược điểm là nó bỏ qua sự liên quan giữa những trường ngữ cảnh. Ví dụ như vào thời gian cuối tuần thì thường đi chơi cùng gia đình còn ngày thường thì thường đi chơi với bạn bè. Như vậy trường ngữ cảnh thời gian trong tuần và trường ngữ cảnh đi cùng với ai có thể có mối quan hệ nào đó.

### 1.2.2.2. Độ tương quan ngữ cảnh Latent

Độ tương quan ngữ cảnh độc lập chỉ xem xét độ tương quan giữa những giá trị trong từng trường ngữ cảnh cụ thể sau đó tập hợp lại. Tuy nhiên dữ liệu trong thực tế lại rất thưa. Có những ngữ cảnh chỉ xuất hiện trong tập kiểm tra chứ không có trong huấn luyện thì độ tương quan ngữ cảnh độc lập sẽ không thể đo chính xác. Ví dụ ta cần tính độ tương quan chiều thời gian giữa cuối tuần và ngày nghỉ lễ  $\text{similarity}(\text{cuối tuần}, \text{nghỉ lễ})$ . Cặp tương quan này có trong tập kiểm tra nhưng không có trong tập huấn luyện nên độ tương quan ngữ cảnh độc lập không thể ước lượng chúng. Tuy nhiên trong tập huấn luyện lại có tính toán  $\text{similarity}(\text{cuối tuần}, \text{ngày thường})$  và  $\text{similarity}(\text{ngày thường}, \text{nghỉ lễ})$ . Rõ ràng có thể có một cách nào đó để ước lượng  $\text{similarity}(\text{cuối tuần}, \text{nghỉ lễ})$ . Và đó là ý tưởng hình thành độ tương quan ngữ cảnh Latent.

Ý tưởng của độ tương quan ngữ cảnh Latent cũng là tính toán tương quan những giá trị của từng chiều không gian một cách độc lập rồi tổng hợp lại giống như độ tương quan ngữ cảnh độc lập. Tuy nhiên những giá trị của từng chiều ngữ cảnh sẽ được nhúng vào không gian Latent và biểu diễn chúng dưới dạng một vector. Và độ tương quan những giá trị của từng chiều không gian sẽ được ước lượng bằng tích của hai vector Latent:

$$\text{similarity}(c_{ki}, c_{li}) = V_{c_{ki}} \cdot V_{c_{li}}$$

Trong đó:

$\text{similarity}(c_{ki}, c_{li})$  là độ tương quan giữa chiều thứ i của ngữ cảnh  $c_k$  và  $c_l$ .

$V_{c_{ki}}$  là vector biểu diễn chiều thứ i của ngữ cảnh  $c_k$ .

$V_{c_{li}}$  là vector biểu diễn chiều thứ  $i$  của ngữ cảnh  $c_l$ .

$c_{ki}$  và  $c_{li}$  dù không được có trong quá trình huấn luyện nhưng  $V_{c_{ki}}$  và  $V_{c_{li}}$  được học trong quá trình huấn luyện nên cũng phần nào thể hiện mối quan hệ giữa  $c_{ki}$  và  $c_{li}$ . Như vậy dù không được trực tiếp học trong quá trình huấn luyện thì ta vẫn có thể tính độ tương quan giữa chiều thứ  $i$  của ngữ cảnh  $c_k$  và  $c_l$ .

Ví dụ sau quá trình huấn luyện, ta tìm được biểu diễn vector của các điều kiện ngữ cảnh như sau:

Ngày thường	0,1	0,2	0,3
Cuối tuần	0,5	0,2	0,7
Lạnh	0,3	0,4	0,6
Nóng	0,6	0,9	0,6

Bảng 2: Ví dụ về độ tương quan Latent

Giả sử ta cần tính độ tương quan giữa hai ngữ cảnh  $c_1$ (cuối tuần, lạnh) và  $c_2$ (ngày thường, nóng). Khi đó độ tương quan giữa từng điều kiện ngữ cảnh sẽ được tính như sau:

$$\begin{aligned} \text{Sim}(c_1, c_2) &= \text{similarity}(\text{cuối tuần}, \text{ngày thường}) \times \text{similarity}(\text{lạnh}, \text{nóng}) \\ &= (0.5 \times 0.1 + 0.2 \times 0.2 + 0.7 \times 0.3) \times (0.3 \times 0.6 + 0.4 \times 0.9 + 0.6 \times 0.6) \\ &= 0.3 \times 0.9 \\ &= 0.27 \end{aligned}$$

Độ tương quan ngữ cảnh Latent đã có thể ước lượng độ tương quan giữa những giá trị ngữ cảnh trong cùng một trường dù chúng chưa từng xuất hiện trong tập huấn luyện. Tuy nhiên phương pháp này lại rất tiêu tốn tài nguyên nếu có nhiều trường ngữ cảnh và mỗi trường ngữ cảnh lại có nhiều giá trị do phải tính toán để tìm ra nhiều vector Latent biểu diễn cho từng giá trị này.

### 1.3. Kết luận chương

Trong chương này, tác giả đã lần lượt giới thiệu tổng quan về hệ tư vấn truyền thống với những phương pháp lọc nội dung, lọc cộng tác, hệ tư vấn lai và những ưu/nhược điểm của chúng. Phương pháp lọc cộng tác có có ưu điểm hơn lọc nội dung trong bài toán tư vấn những sản phẩm mà không được mô tả bằng đặc trưng nội dung như những sản phẩm đa phương tiện (phim, bài hát, ảnh, địa điểm du lịch, ...). Chính vì vậy trong khuôn khổ đồ án này, tác giả chỉ tập trung nghiên cứu xây dựng hệ tư vấn đa phương tiện dựa trên phương pháp lọc cộng tác.

Chương này cũng đã giới thiệu về khái niệm ngữ cảnh vai trò của yếu tố ngữ cảnh trong lĩnh vực tư vấn sản phẩm, cũng như những phương pháp tích hợp ngữ cảnh vào hệ tư vấn truyền thống. Chính vì yếu tố ngữ cảnh rất quan trọng nên nó xứng đáng được nghiên cứu một cách tỉ mỉ và nghiêm túc nhằm đem lại trải nghiệm tốt nhất cho người dùng. Tác giả cũng đã đề cập tới những phương pháp tích hợp yếu tố ngữ cảnh trong hệ tư vấn là phương pháp mô hình hóa ngữ cảnh độc lập và mô hình hóa ngữ cảnh phụ thuộc. Phương pháp mô hình hóa ngữ cảnh phụ thuộc có ưu điểm

hơn mô hình hóa ngữ cảnh độc lập ở chỗ quá trình mô hình hóa ngữ cảnh được xây dựng cùng lúc với các đối tượng khác như người dùng hoặc sản phẩm. Vì vậy sẽ tạo xử lý được vấn đề mối quan hệ giữa những đối tượng này và ngữ cảnh. Hai phương pháp mô hình hóa ngữ cảnh phụ thuộc đã được trình bày là mô hình hóa dựa trên độ tương quan ngữ cảnh độc lập và mô hình hóa dựa trên độ tương quan ngữ cảnh Latent.

Phương pháp mô hình hóa dựa trên độ tương quan Latent có cải tiến hơn ở chỗ nó có thể ước lượng được độ tương quan giữa hai ngữ cảnh mà chưa có trong tập dữ liệu, nhưng lại gặp vấn đề tiêu tốn nhiều tài nguyên do mỗi trường ngữ cảnh cần một vector để biểu diễn. Phương pháp mô hình hóa dựa trên độ tương quan Latent cũng chưa giải quyết được vấn đề sự liên quan giữa những điều kiện khác trường ngữ cảnh. Quá trình dự đoán thực tế vẫn chỉ là so sánh độ tương quan trong cùng trường ngữ cảnh (cuối tuần so với trung tuần, lạnh so với nóng), sau đó nhân những trọng số này với nhau để có được độ tương quan giữa hai ngữ cảnh. Nhưng trên thực tế có thể có những trường ngữ cảnh có liên quan tới nhau. Ví dụ có hai trường ngữ cảnh là bạn đồng hành và thời gian trong tuần. Vào cuối tuần không phải đi làm thì mới đi ăn với gia đình còn trung tuần thì chỉ đi ăn với đồng nghiệp. Rõ ràng có mối liên hệ giữa những trường ngữ cảnh trong trường hợp này. Hai phương pháp đã trình bày đã không tập trung vào vấn đề này. Trong nội dung chương 2, tác giả sẽ trình bày phương pháp mô hình hóa ngữ cảnh có thể giải quyết vấn đề trên, đó là kỹ thuật tích hợp ngữ cảnh vào hệ tư vấn đa phương tiện dựa trên lọc cộng tác bằng phương pháp mô hình hóa dựa trên độ tương quan ngữ cảnh đa chiều.

## **CHƯƠNG 2: NGHIÊN CỨU PHƯƠNG PHÁP MÔ HÌNH HÓA NGỮ CẢNH CSLIM TRONG XÂY DỰNG HỆ TƯ VẤN ĐA PHƯƠNG TIỆN DỰA TRÊN NGỮ CẢNH**

Trong chương trước, tác giả đã trình bày về vai trò quan trọng của yếu tố ngữ cảnh trong hệ tư vấn. Đồng thời giới thiệu những phương pháp tích hợp yếu tố ngữ cảnh vào hệ tư vấn truyền thống là phương pháp lọc ngữ cảnh (lọc trước và lọc sau tư vấn truyền thống), phương pháp mô hình hóa độc lập và mô hình hóa phụ thuộc (bao gồm phương pháp dựa trên độ tương quan độc lập và dựa trên độ tương quan Latent). Tác giả cũng đã trình bày những ưu/nhược điểm của những phương pháp này.

Mục tiêu chương 2 trình bày về kỹ thuật tích hợp ngữ cảnh vào hệ tư vấn đa phương tiện dựa trên lọc cộng tác bằng phương pháp mô hình hóa dựa trên độ tương quan ngữ cảnh đa chiều. Nhưng trước khi đi vào chi tiết kỹ thuật tích hợp ngữ cảnh, tác giả sẽ trình bày về một thuật toán tư vấn truyền thống để kết hợp với kỹ thuật tích hợp ngữ cảnh. Đó là thuật toán phương pháp tuyến tính thưa (Sparse Linear Method – SLIM).

### **2.1. Phương pháp tuyến tính thưa SLIM**

Phương pháp tuyến tính thưa (Sparse Linear Method – SLIM [11]) là thuật toán gọi ý dựa trên mô hình. Ý tưởng của SLIM là tìm ra mối quan hệ, cụ thể là độ tương quan, giữa những sản phẩm trong tập dữ liệu thông qua các bước huấn luyện của thuật toán SGD đã trình bày trong phần 1.1.2.2.

Công thức để dự đoán đánh giá của SLIM là:

$$\tilde{r}_{ui} = \sum_{v \in I \setminus \{i\}} r_{uv} \cdot w_{iv} \quad (2.1)$$

Trong đó:

$\tilde{r}_{ui}$  là đánh giá dự đoán của người dùng u với sản phẩm i.

$r_{uv}$  là đánh giá thực tế của người dùng u với sản phẩm v.

$w_{iv}$  là trọng số giữa sản phẩm i và sản phẩm v. Có thể hiểu tương tự như độ tương quan.

I là tập sản phẩm hàng xóm của i dựa vào độ tương quan Pearson.

Công thức trên có thể hiểu là dự đoán của người dùng u và sản phẩm i sẽ phụ thuộc vào những sản phẩm đã được đánh giá bởi người dùng u. Sự phụ thuộc nhiều hay ít sẽ được thể hiện bằng ma trận trọng số giữa những sản phẩm. Sản phẩm càng

tương đồng nhau (trọng số cao) thì sẽ ảnh hưởng tới dự đoán càng nhiều và ngược lại. Vấn đề của thuật toán là phải tìm ra trận trọng số sản phẩm thông qua bước huấn luyện xây dựng mô hình thay vì sử dụng công thức heuristic đánh giá độ tương quan như đối với phương pháp dựa trên bộ nhớ.

Công thức trên còn có thể viết lại như sau:

$$\tilde{R} = R \cdot W \quad (2.2)$$

Trong đó:

$\tilde{R}$  là ma trận dự đoán những giá trị còn thiếu của  $R$ .

$R$  là ma trận đánh giá của những người dùng với những sản phẩm.

$W$  là ma trận trọng số giữa những sản phẩm, thể hiện sự tương đồng giữa những sản phẩm.

$$\tilde{R} = \begin{array}{c|cccc} & i_1 & i_2 & \dots & i_m \\ \hline u_1 & 4 & 0 & \dots & 3 \\ u_2 & ? & 2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ u_n & 5 & 4 & \dots & 1 \end{array} \times \begin{array}{c|cccc} & i_1 & i_2 & \dots & i_m \\ \hline i_1 & 0 & 0.1 & \dots & 0.5 \\ i_2 & 0.1 & 0 & \dots & 0.4 \\ \dots & \dots & \dots & \dots & \dots \\ i_m & 0.5 & 0.4 & \dots & 0 \end{array}$$

Hình ảnh 3: Ví dụ về phương pháp dự đoán đánh giá của SLIM

Quá trình huấn luyện của SLIM chính là quá trình tìm ra ma trận  $W$  phù hợp. Theo như [11], SLIM được huấn luyện bằng thuật toán SGD với hàm mục tiêu sau:

$$\text{Loss} = \frac{1}{2} \|R - \tilde{R}\|_F^2 + \frac{\beta}{2} \|W\|_F^2 + \alpha \|W\|_1 \quad (2.3)$$

Trong đó:

$\tilde{R}$  là ma trận dự đoán những giá trị còn thiếu của  $R$ .

$R$  là ma trận đánh giá của những người dùng với những sản phẩm.

$W$  là ma trận trọng số giữa những sản phẩm, thể hiện sự tương đồng giữa những sản phẩm.

$\|\cdot\|_F^2$ ,  $\|\cdot\|_1$  lần lượt là phép chuẩn hóa  $l_F$  và  $l_1$ .

$$\|W\|_F^2 = (\sqrt{\sum_{i=1}^m \sum_{v=1}^m w_{iv}^2}) \quad \text{và } \|W\|_1 = \sum_{i=1}^m \sum_{v=1}^m w_{iv}$$

$\alpha, \beta$  là hai siêu tham số chuẩn hóa.

Mục tiêu của thuật toán là tìm ra ma trận  $W$  dương sao cho hàm mục tiêu đạt giá trị nhỏ nhất.

Công thức 2.3 có thể khai triển ra như sau:

$$\begin{aligned}
 Loss &= \frac{1}{2} \|R - \tilde{R}\|_F^2 + \frac{\beta}{2} \|W\|_F^2 + \alpha \|W\|_1 \\
 &= \frac{1}{2} \left( \sqrt{\sum_{u=1}^n \sum_{i=1}^m (r_{ui} - \tilde{r}_{ui})^2} \right)^2 + \frac{\beta}{2} \left( \sqrt{\sum_{i=1}^m \sum_{v=1}^m w_{iv}^2} \right)^2 + \alpha \left( \sum_{i=1}^m \sum_{v=1}^m w_{iv} \right) \\
 &= \frac{1}{2} \left( \sum_{u=1}^n \sum_{i=1}^m (r_{ui} - \tilde{r}_{ui})^2 \right) + \frac{\beta}{2} \left( \sum_{i=1}^m \sum_{v=1}^m w_{iv}^2 \right) + \alpha \left( \sum_{i=1}^m \sum_{v=1}^m w_{iv} \right) \quad (2.4)
 \end{aligned}$$

Trong đó:

$n$  là số lượng người dùng.

$m$  là số lượng sản phẩm.

$\tilde{r}_{ui}$  là giá trị dự đoán bởi thuật toán giữa người dùng  $u$  và sản phẩm  $i$ , được tính theo công thức (2.1).

$r_{ui}$  là giá trị đánh giá thực tế trong bộ dữ liệu giữa người dùng  $u$  và sản phẩm  $i$ .

$w_{iv}$  là độ tương đồng giữa sản phẩm  $i$  và sản phẩm  $v$ .

Có thể thấy  $\|R - \tilde{R}\|_F^2$  thể hiện độ sai lệch giữa đánh giá dự đoán và đánh giá thực tế. Phân còn lại của công thức là kỹ thuật bình thường hóa tham số  $l_1$  và  $l_F$  [12] nhằm tránh xảy ra trường hợp quá vừa dữ liệu (overfitting) [13]. Như vậy điều ta mong muốn đó là tìm ra bộ tham số  $W$  (cũng chính là sự tương đồng giữa những sản phẩm) sao cho độ sai lệch là nhỏ nhất những không được quá nhỏ để tránh overfitting.

Để có thể tìm ra bộ tham số  $W$ , SLIM sẽ sử dụng một thuật toán tối ưu tham số để tìm ra  $W$ . Cụ thể là áp dụng thuật toán SGD đã trình bày ở chương trước trước. Để áp dụng SGD, đầu tiên ta cần tìm công thức cập nhật tham số  $W$ .

Ta có:

$$\frac{\partial \tilde{r}_{ui}}{\partial w_{iv}} = \frac{\partial (\sum_{v \in I \setminus \{i\}} r_{uv} \cdot w_{iv})}{\partial w_{iv}} = r_{uv} \quad (2.5)$$

Như vậy, ta tính được  $\frac{\partial Loss}{\partial w_{iv}}$ :

$$\begin{aligned}
 \frac{\partial Loss}{\partial w_{iv}} &= \frac{\partial \left( \frac{1}{2} \left( \sum_{u=1}^n \sum_{i=1}^m (r_{ui} - \tilde{r}_{ui})^2 \right) \right)}{\partial w_{iv}} + \frac{\partial \left( \frac{\beta}{2} \left( \sum_{i=1}^m \sum_{v=1}^m w_{iv}^2 \right) \right)}{\partial w_{iv}} \\
 &\quad + \frac{\partial (\alpha \left( \sum_{i=1}^m \sum_{v=1}^m w_{iv} \right))}{\partial w_{iv}}
 \end{aligned}$$

$$= \frac{\partial(\frac{1}{2}(\sum_{u=1}^n \sum_{i=1}^m (r_{ui} - \tilde{r}_{ui})^2))}{\partial w_{iv}} + \beta \cdot w_{iv} + \alpha$$

Ta có  $\frac{\partial y}{\partial x} = \frac{\partial y}{\partial z} \cdot \frac{\partial z}{\partial x}$  theo luật xích của phương trình vi phân. Áp dụng luật xích vào (2.6) và kết hợp với (2.5) ta có:

$$\begin{aligned} \frac{\partial(\frac{1}{2}(\sum_{u=1}^n \sum_{i=1}^m (r_{ui} - \tilde{r}_{ui})^2))}{\partial w_{iv}} &= \frac{\partial(\frac{1}{2}(\sum_{u=1}^n \sum_{i=1}^m (r_{ui} - \tilde{r}_{ui})^2))}{\partial \tilde{r}_{ui}} \cdot \frac{\partial \tilde{r}_{ui}}{\partial w_{iv}} \\ &= (r_{ui} - \tilde{r}_{ui}) \cdot r_{uv} \end{aligned}$$

Từ (2.4) và (2.6), cuối cùng ta có được giá trị gradient để cập nhật tham số  $w_{iv}$  là:

$$\frac{\partial Loss}{\partial w_{iv}} = (r_{ui} - \tilde{r}_{ui}) \cdot r_{uv} + \beta \cdot w_{iv} + \alpha$$

Và theo công thức ước lượng tham số của SGD, ta có thể cập nhật  $w_{iv}$ :

$$w_{iv} = w_{iv} - \mu \cdot ((r_{ui} - \tilde{r}_{ui}) \cdot r_{uv} + \beta \cdot w_{iv} + \alpha) \quad (2.7)$$

Trong đó:

$\tilde{r}_{ui}$  là giá trị dự đoán bởi thuật toán giữa người dùng u và sản phẩm i.

$r_{ui}$  và  $r_{uv}$  lần lượt là giá trị đánh giá thực tế trong bộ dữ liệu giữa người dùng u với sản phẩm i và sản phẩm v.

$w_{iv}$  là độ tương đồng giữa sản phẩm i và sản phẩm v.

$\alpha, \beta$  là hai siêu tham số chuẩn hóa  $l_1$  và  $l_F$ .

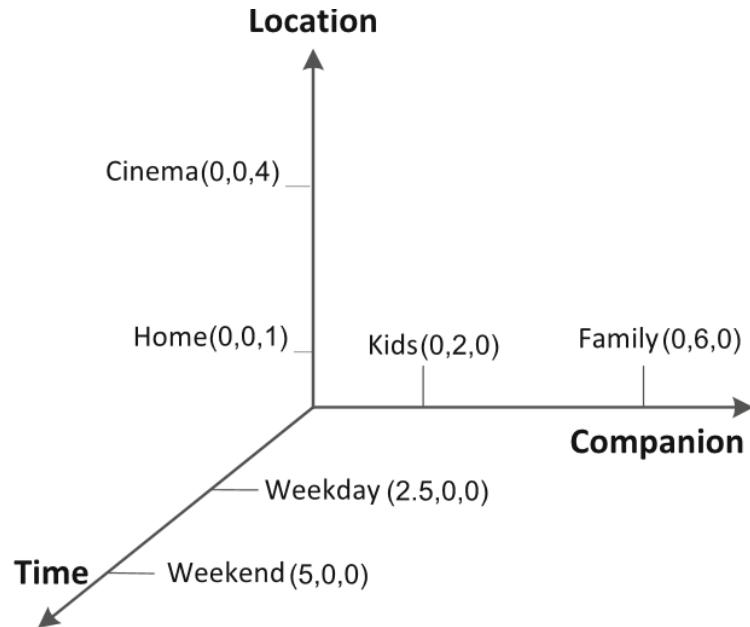
$\mu$  là siêu tham số hệ số học của thuật toán SGD.

Có thể thấy rõ ràng thuật toán SLIM chính là phương pháp dựa trên mô hình. Các tham số về độ tương đồng giữa những sản phẩm hoàn toàn không dựa vào những công thức heuristic như Item-based. Trong lĩnh vực nghiên cứu về hệ tư vấn ngày nay thì những kỹ thuật xây dựng độ tương đồng dựa trên mô hình được nghiên cứu mạnh mẽ hơn do kết quả nó mang lại tốt hơn trên tập dữ liệu nhiều nhiễu. Vậy nên trong đồ án này, tác giả tập trung nghiên cứu phương pháp tư vấn dựa trên mô hình, cụ thể là sử dụng thuật toán SLIM, để kết hợp với kỹ thuật tích hợp ngữ cảnh dựa trên độ tương quan đa chiều.

## 2.2. Mô hình hóa ngữ cảnh dựa trên độ tương quan đa chiều

Đối với độ tương quan ngữ cảnh đa chiều, mỗi trường của ngữ cảnh sẽ được xem như một chiều của vector. Và một ngữ cảnh cụ thể gồm n trường sẽ được biểu

diễn dưới dạng một vector ngữ cảnh n chiều hay là một điểm trong không gian n chiều.



Hình ảnh 4: Không gian ngữ cảnh đa chiều

Mỗi ngữ cảnh cụ thể lúc này có thể được coi là một điểm trong không gian đa chiều. Và độ tương quan ngữ cảnh đa chiều sẽ được tính bằng khoảng cách giữa hai điểm ngữ cảnh trong không gian đa chiều. Những vector biểu diễn ngữ cảnh cụ thể sẽ được học trong quá trình huấn luyện nhằm tối ưu kết quả tư vấn.

Để tính khoảng cách giữa hai điểm ngữ cảnh, ta có thể sử dụng công thức khoảng cách Euclid. Tuy nhiên ta mong muốn độ tương quan bằng 1 khi 2 điểm ngữ cảnh trùng nhau. Vì vậy công thức tính độ tương quan ngữ cảnh đa chiều sẽ như sau:

$$Dist = \sqrt{\sum_{i=1}^D (c_{ki} - c_{li})^2} \quad (2.8)$$

$$Sim(c_k, c_l) = 1 - Dist \quad (2.9)$$

Trong đó:

$Dist$  là khoảng cách Euclid giữa  $c_k$  và  $c_l$ .

$Sim(c_{ki}, c_{li})$  là độ tương quan giữa  $c_k$  và  $c_l$ .

$D$  là tổng số giá trị ngữ cảnh.

$c_k$  và  $c_l$  là hai vector biểu diễn ngữ cảnh k và ngữ cảnh l.

$c_{ki}$  và  $c_{li}$  là chiều thứ i của vector  $c_k$  và  $c_l$ . Để chắc chắn rằng

$Sim(c_k, c_l) \in [0; 1]$  thì  $c_{ki}, c_{li} \in [0; \frac{1}{\sqrt{D}}]$ .

Nghiên cứu [18] đã chứng minh rằng việc sử dụng độ tương quan ngữ cảnh đa chiều đem lại hiệu quả tốt hơn so với độ tương quan ngữ cảnh độc lập và độ tương quan ngữ cảnh Latent cho bài toán tư vấn top-N kết hợp ngữ cảnh. Vì vậy trong phần tiếp theo, tác giả sẽ trình bày một thuật toán kết hợp phương pháp tư vấn truyền thống với yếu tố ngữ cảnh bằng phương pháp mô hình hóa dựa trên độ tương quan ngữ cảnh đa chiều. Đó là thuật toán Contextual Sparse Linear Method Multidimensional Context Similarity (CSLIM\_MCS).

### 2.3. Thuật toán tích hợp ngữ cảnh đa chiều CSLIM\_MCS

CSLIM\_MCS [18] là một thuật toán kết hợp giữa phương pháp tư vấn truyền thống và kỹ thuật tích hợp ngữ cảnh có phụ thuộc đã trình bày ở những chương trước. Cụ thể thuật toán tư vấn truyền thống ở đây là thuật toán SLIM [11] còn phương pháp tích hợp ngữ cảnh là phương pháp tích hợp mô hình hóa dựa vào độ tương quan ngữ cảnh đa chiều.

Chi tiết về thuật SLIM đã được trình bày ở phần trước. Công thức dự đoán của SLIM là:

$$\tilde{R} = R \cdot W$$

Khi áp dụng kỹ thuật tích hợp ngữ cảnh dựa trên độ tương quan ngữ cảnh đa chiều thì công thức dự đoán của CSLIM\_MCS sẽ như sau:

$$\begin{aligned}\tilde{R}_c &= R \cdot W \cdot \text{Sim}(c_\emptyset, c) \\ \text{Sim}(c_\emptyset, c) &= 1 - \text{Dist}_c \\ \text{Dist}_c &= \sqrt{\sum_{i=1}^D (c_i - c_{\emptyset i})^2}\end{aligned}\tag{2.10}$$

Trong đó:

$\tilde{R}_c$  là dự đoán đánh giá trong ngữ cảnh  $c$ .

$R$  là dự đoán đánh giá sẵn có trong tập dữ liệu.

$W$  là bộ tham số cần tìm, thể hiện trọng số giữa những sản phẩm.

$\text{Sim}(c_\emptyset, c)$  là độ tương quan ngữ cảnh đa chiều giữa ngữ cảnh  $c$  và trường hợp không rõ ngữ cảnh.

$c_\emptyset$  là vector biểu diễn trường hợp không rõ ngữ cảnh. Còn  $c$  là vector biểu diễn ngữ cảnh đang xét.

$\text{Dist}_c$  là khoảng cách euclid giữa điểm ngữ cảnh  $c_\emptyset$  và điểm ngữ cảnh  $c$ .

$D$  là số chiều của vector ngữ cảnh.

Những tham số cần tìm của CSLIM\_MCS là  $W$  và những vector biểu diễn  $c$  có trong tập dữ liệu, bao gồm cả  $c_\emptyset$ . Những bộ tham số này sẽ được tìm thấy trong quá trình huấn luyện dữ liệu bằng cách sử dụng thuật toán Stochastic Gradient Descend (SGD) giống như đối với thuật toán SLIM đã trình. Để áp dụng SGD, ta cần tìm công thức cập nhật cho các tham số.

### 2.3.1. Cập nhật tham số trọng số sản phẩm W

$W$  là tham số trọng số thể hiện độ tương đồng giữa những sản phẩm trong thuật toán SLIM cũng như thuật toán CSLIM\_MCS.  $W$  thay vì được ước lượng bằng những công thức heuristic như thuật toán Item-based đã trình bày ở chương trước thì sẽ được tìm ra trong quá trình huấn luyện bằng thuật toán SGD. Trước tiên ta cần tìm công thức cập nhật tham số theo thuật toán SGD. Để tìm được công thức cập nhật tham số, cần tính độ dốc (Gradient) của hàm lỗi bằng cách lấy đạo hàm hàm lỗi.

Công thức (2.10) còn có thể viết như sau:

$$\tilde{r}_{uic} = \sum_{v \in I \setminus \{i\}} r_{uv} \cdot w_{iv} \cdot \text{Sim}(c_\emptyset, c) \quad (2.11)$$

Trong đó:

$\tilde{r}_{ui}$  là đánh giá dự đoán của người dùng  $u$  với sản phẩm  $i$ .

$r_{uv}$  là đánh giá thực tế của người dùng  $u$  với sản phẩm  $v$ .

$w_{iv}$  là trọng số giữa sản phẩm  $i$  và sản phẩm  $v$ . Có thể hiểu tương tự như độ tương quan.

$I$  là tập sản phẩm hàng xóm.

$\text{Sim}(c_\emptyset, c)$  là độ tương quan ngữ cảnh đa chiều giữa ngữ cảnh  $c$  và trường hợp không rõ ngữ cảnh.

Hàm đánh giá độ lỗi của CSLIM\_MCS cũng giống SLIM. Tuy nhiên công thức cập nhật tham số thì thay đổi do công thức dự đoán đánh giá khác.

Trong trường hợp này, ta có:

$$\frac{\partial \tilde{r}_{uic}}{\partial w_{iv}} = \frac{\partial (\sum_{v \in I \setminus \{i\}} r_{uv} \cdot w_{iv} \cdot \text{Sim}(c_\emptyset, c))}{\partial w_{iv}} = r_{uv} \cdot \text{Sim}(c_\emptyset, c) \quad (2.12)$$

Ta có thể tính độ dốc với tham số  $W$  bằng cách lấy đạo hàm hàm lỗi theo  $W$ :

$$\begin{aligned} \frac{\partial \text{Loss}}{\partial w_{iv}} &= \frac{\partial (\frac{1}{2} (\sum_{u=1}^n \sum_{i=1}^m (r_{uic} - \tilde{r}_{uic})^2))}{\partial w_{iv}} + \frac{\partial (\frac{\beta}{2} (\sum_{i=1}^m \sum_{v=1}^m w_{iv}^2))}{\partial w_{iv}} \\ &\quad + \frac{\partial (\alpha (\sum_{i=1}^m \sum_{v=1}^m w_{iv}))}{\partial w_{iv}} \end{aligned}$$

$$\begin{aligned}
&= \frac{\partial(\frac{1}{2}(\sum_{u=1}^n \sum_{i=1}^m (r_{ui} - \tilde{r}_{uic})^2))}{\partial \tilde{r}_{uic}} \cdot \frac{\partial \tilde{r}_{uic}}{\partial w_{iv}} + \beta \cdot w_{iv} + \alpha \\
&= (r_{uic} - \tilde{r}_{uic}) \cdot r_{uv} \cdot \text{Sim}(c_\emptyset, c) + \beta \cdot w_{iv} + \alpha
\end{aligned}$$

Do đó công thức tính độ dốc  $\frac{\partial Loss}{\partial w_{iv}}$  của CSLIM\_MCS sẽ được thay đổi là:

$$\frac{\partial Loss}{\partial w_{iv}} = (r_{uic} - \tilde{r}_{uic}) \cdot r_{uv} \cdot \text{Sim}(c_\emptyset, c) + \beta \cdot w_{iv} + \alpha$$

Và công thức cập nhật tham số  $W$  là:

$$w_{iv} = w_{iv} - \mu \cdot ((r_{uic} - \tilde{r}_{uic}) \cdot r_{uv} \cdot \text{Sim}(c_\emptyset, c) + \beta \cdot w_{iv} + \alpha) \quad (2.13)$$

Trong đó:

$w_{iv}$  là tham số cần cập nhật, là trọng số giữa sản phẩm i và sản phẩm v.

$r_{uic}$  là đánh giá thực tế của người dùng u với sản phẩm i trong ngữ cảnh c.

$\tilde{r}_{uic}$  là đánh giá dự đoán giữa người dùng u với sản phẩm i trong ngữ cảnh c.

$r_{uv}$  là đánh giá giữa người dùng u với sản phẩm v.

$\text{Sim}(c_\emptyset, c)$  là độ tương quan ngữ cảnh đa chiều giữa ngữ cảnh c và trường hợp không rõ ngữ cảnh.

$\mu$  là siêu tham số hệ số học.

$\alpha, \beta$  là hai siêu tham số chuẩn hóa 11/12.

Như vậy ta đã có công thức cập nhật tham số  $W$ . Bước tiếp theo sẽ cần tìm công thức cập nhật vector ngữ cảnh.

### 2.3.2. Cập nhật tham số ngữ cảnh

Với phương pháp tích hợp dựa trên độ tương quan ngữ cảnh đa chiều, với mỗi điều kiện của mỗi trường ngữ cảnh, ta cần tìm ra biểu diễn của chúng thông qua bước huấn luyện. Quá trình này sử dụng thuật toán SGD để tìm ra biểu diễn những điều kiện ngữ cảnh sao cho tối thiểu hóa hàm lỗi. Các cập nhật vector ngữ cảnh sẽ được trình bày dưới đây.

Ta có:

$$\frac{\partial \tilde{r}_{uic}}{\partial c_{\emptyset k}} = \frac{\partial (\sum_{v \in I \setminus \{i\}} r_{uv} \cdot w_{iv} \cdot \text{Sim}(c_\emptyset, c))}{\partial c_{\emptyset k}}$$

$$\begin{aligned}
&= \frac{\partial \left( \sum_{v \in I \setminus \{i\}} r_{uv} \cdot w_{iv} \cdot \left( 1 - \sqrt{\sum_{k=1}^D (c_k - c_{\emptyset k})^2} \right) \right)}{\partial c_{\emptyset k}} \\
&= \sum_{v \in I \setminus \{i\}} r_{uv} \cdot w_{iv} \cdot \frac{c_k - c_{\emptyset k}}{\sqrt{\sum_{k=1}^D (c_k - c_{\emptyset k})^2}}
\end{aligned}$$

Tương tự ta có:

$$\begin{aligned}
\frac{\partial \tilde{r}_{uic}}{\partial c_k} &= \frac{\partial \left( \sum_{v \in I \setminus \{i\}} r_{uv} \cdot w_{iv} \cdot \text{Sim}(c_{\emptyset}, c) \right)}{\partial c_k} \\
&= \frac{\partial \left( \sum_{v \in I \setminus \{i\}} r_{uv} \cdot w_{iv} \cdot \left( 1 - \sqrt{\sum_{k=1}^D (c_k - c_{\emptyset k})^2} \right) \right)}{\partial c_k} \\
&= - \sum_{v \in I \setminus \{i\}} r_{uv} \cdot w_{iv} \cdot \frac{c_k - c_{\emptyset k}}{\sqrt{\sum_{k=1}^D (c_k - c_{\emptyset k})^2}}
\end{aligned}$$

Ta cũng có:

$$\begin{aligned}
\frac{\partial \text{Loss}}{\partial c_k} &= \frac{\partial \left( \frac{1}{2} (\sum_{u=1}^n \sum_{i=1}^m (r_{uic} - \tilde{r}_{uic})^2) \right)}{\partial c_k} + \frac{\partial \left( \frac{\beta}{2} (\sum_{i=1}^m \sum_{v=1}^n w_{iv}^2) \right)}{\partial c_k} \\
&\quad + \frac{\partial (\alpha (\sum_{i=1}^m \sum_{v=1}^n w_{iv}))}{\partial c_k} \\
&= \frac{\partial \left( \frac{1}{2} (\sum_{u=1}^n \sum_{i=1}^m (r_{uic} - \tilde{r}_{uic})^2) \right)}{\partial \tilde{r}_{uic}} \cdot \frac{\partial \tilde{r}_{uic}}{\partial c_k} \\
&= (r_{uic} - \tilde{r}_{uic}) \cdot \left( \sum_{v \in I \setminus \{i\}} r_{uv} \cdot w_{iv} \right) \cdot \frac{c_k - c_{\emptyset k}}{\sqrt{\sum_{k=1}^D (c_k - c_{\emptyset k})^2}}
\end{aligned}$$

Tương tự ta có:

$$\frac{\partial \text{Loss}}{\partial c_{\emptyset k}} = -(r_{uic} - \tilde{r}_{uic}) \cdot \left( \sum_{v \in I \setminus \{i\}} r_{uv} \cdot w_{iv} \right) \cdot \frac{c_k - c_{\emptyset k}}{\sqrt{\sum_{k=1}^D (c_k - c_{\emptyset k})^2}}$$

Và công thức cập nhật những vector ngũ cành là:

$$c_k = c_k - \mu \cdot \left( (r_{uic} - \tilde{r}_{uic}) \cdot \left( \sum_{v \in I \setminus \{i\}} r_{uv} \cdot w_{iv} \right) \cdot \frac{c_k - c_{\emptyset k}}{\sqrt{\sum_{k=1}^D (c_k - c_{\emptyset k})^2}} \right)$$

$$c_{\emptyset k} = c_{\emptyset k} + \mu \cdot \left( (r_{uic} - \tilde{r}_{uic}) \cdot \left( \sum_{v \in I \setminus \{i\}} r_{uv} \cdot w_{iv} \right) \cdot \frac{c_k - c_{\emptyset k}}{\sqrt{\sum_{k=1}^D (c_k - c_{\emptyset k})^2}} \right)$$

Trong đó:

$c_k, c_{\emptyset k}$  lần lượt là chiều thứ k của vector ngũ cảnh c và vector không rõ ngũ cảnh.

$r_{uic}, \tilde{r}_{uic}$  lần lượt là đánh giá thực tế và đánh giá dự đoán của người dùng u với sản phẩm i trong ngũ cảnh c.

$r_{uv}$  là đánh giá thực tế của người dùng u với sản phẩm v.

$w_{iv}$  là trọng số giữa sản phẩm i và sản phẩm v.

$I$  là tổng số sản phẩm hàng xóm.

$D$  là tổng số giá trị ngũ cảnh.

### 2.3.3. Thuật toán huấn luyện và dự đoán của CSLIM\_MCS

Thuật toán CSLIM\_MCS tách biệt hai quá trình huấn luyện và dự đoán đánh giá sản phẩm cho người dùng.

#### Thuật toán huấn luyện CSLIM\_MCS:

##### Đầu vào:

- Ma trận đánh giá  $R$  giữa người dùng và sản phẩm trong ngũ cảnh nhất định.
- Tham số  $k$  hàng xóm để tính toán dựa trên k sản phẩm gần nhất dựa trên độ tương quan Pearson.
- Tham số chuẩn hóa  $\alpha, \beta$  và hệ số học  $\mu$ . Hệ số học sẽ quyết định tốc độ học của thuật toán. Tuy nhiên nếu hệ số học quá cao thì sẽ không thể làm độ lỗi giảm do thuật toán bị dừng sớm vì tốc độ học quá nhanh. Nếu hệ số học quá thấp thì quá huấn luyện rất chậm.
- Tham số số lần lặp huấn luyện  $t$ .

##### Đầu ra:

- Ma trận  $W$  thể hiện trọng số giữa những sản phẩm.
- Vector ngũ cảnh c biểu diễn cho từng điều kiện ngũ cảnh trong tập dữ liệu.

### Các bước tiến hành:

#### Bước 1: Khởi tạo các tham số:

- Tính toán  $k$  hàng xóm dựa vào độ tương quan Pearson đồng thời gán giá trị Pearson tính toán được cho  $W$  để việc huấn luyện nhanh hơn.  
 $w_{ij} = \text{Pearson}(i, j)$
- Đặt  $upbound \leftarrow \frac{1}{\sqrt{D}}$  và  $lowbound \leftarrow 10^{-100}$  với  $D$  là tổng số giá trị ngũ cành.  $lowbound$  khác 0 để tránh trường hợp chia cho 0. Hai giá trị này nhằm giới hạn các giá trị biểu diễn của  $c$  để độ tương quan giữa hai ngũ cành thuộc  $[0; 1]$ .
- Khởi tạo ngẫu nhiên tất cả  $c_k$  với  $k \in D$ ,  $c$  là vector ngũ cành cần tìm.  $c_k \in [lowbound; upbound]$ .

#### Bước 2: Quá trình huấn luyện tìm tham số:

- Lặp  $t$  lần:

- Với từng giá trị đánh giá của người dùng  $u$  với sản phẩm  $i$  trong ngũ cành  $c$ , ta lân lượt tính toán:

$$\tilde{r}_{ui} = \sum_{v \in I \setminus \{i\}} r_{uv} \cdot w_{iv}$$

$$Dist_c = \sqrt{\sum_{i=1}^D (c_i - c_{\emptyset i})^2}$$

$$Sim(c_{\emptyset}, c) = 1 - Dist_c$$

$$\tilde{r}_{uic} = \tilde{r}_{ui} \cdot Sim(c_{\emptyset}, c)$$

- Lặp từng chiều của vector ngũ cành đang xét  $c$  để cập nhật tham số:

$$c_k = c_k - \mu \cdot \left( (r_{uic} - \tilde{r}_{uic}) \cdot \tilde{r}_{ui} \cdot \frac{c_k - c_{\emptyset k}}{Dist} \right)$$

$$c_k = \begin{cases} upbound & (\text{nếu } c_k > upbound) \\ lowbound & (\text{nếu } c_k < lowbound) \\ c_k & (\text{nếu } lowbound \leq c_k \leq upbound) \end{cases}$$

$$c_{\emptyset k} = c_{\emptyset k} + \mu \cdot \left( (r_{uic} - \tilde{r}_{uic}) \cdot \tilde{r}_{ui} \cdot \frac{c_k - c_{\emptyset k}}{Dist} \right)$$

$$c_{\emptyset k} = \begin{cases} upbound & (\text{nếu } c_{\emptyset k} > upbound) \\ lowbound & (\text{nếu } c_{\emptyset k} < lowbound) \\ c_{\emptyset k} & (\text{nếu } lowbound \leq c_{\emptyset k} \leq upbound) \end{cases}$$

- Lặp từng cặp sản phẩm để cập nhật trọng số:

$$w_{iv} = w_{iv} - \mu \cdot ((r_{uic} - \tilde{r}_{uic}) \cdot r_{uv} \cdot Sim(c_{\emptyset}, c) + \beta \cdot w_{iv} + \alpha)$$

Bảng 3: Thuật toán huấn luyện CSLIM\_MCS

Thuật toán dự đoán đánh giá của người dùng như sau:

### Thuật toán dự đoán CSLIM\_MCS:

#### Đầu vào:

- Ma trận đánh giá  $R$  giữa người dùng và sản phẩm trong ngữ cảnh nhất định.
- Ma trận trọng số  $W$  giữa những sản phẩm.
- Vector ngữ cảnh  $c$ .
- Người dùng  $u$ , sản phẩm  $i$ .

#### Đầu ra:

- Đánh giá của người dùng  $u$  với sản phẩm  $i$  trong ngữ cảnh  $c$ .

#### Các bước tiến hành:

- Ta lần lượt tính toán các giá trị sau:

$$\tilde{r}_{ui} = \sum_{v \in I \setminus \{i\}} r_{uv} \cdot w_{iv}$$

$$Dist_c = \sqrt{\sum_{i=1}^D (c_i - c_{\phi i})^2}$$

$$Sim(c_\phi, c) = 1 - Dist_c$$

- Và kết quả dự đoán cần tính là:

$$\tilde{r}_{uic} = \tilde{r}_{ui} \cdot Sim(c_\phi, c)$$

Bảng 4: Thuật toán dự đoán đánh giá CSLIM\_MCS

## 2.4. Kiểm nghiệm thuật toán

### 2.4.1. Bộ dữ liệu kiểm nghiệm

Kết quả kiểm nghiệm dưới đây sử dụng bộ dữ liệu InCarMusic [19] bao gồm thông tin đánh giá của người dùng về những bài hát trong những điều kiện thuộc những ngữ cảnh khác nhau:

- *Phong cách lái xe*: Thư giãn, lái xe thể thao, không rõ ngữ cảnh.
- *Khu vực đang chạy*: Thành thị, vùng núi, nông thôn, đường bờ biển, không rõ ngữ cảnh.
- *Tâm trạng*: Buồn, lười, hạnh phúc, năng động, không rõ ngữ cảnh.
- *Thời gian trong ngày*: Đêm, sáng, chiều, ban ngày, không rõ ngữ cảnh.
- *Loại đường*: Nội thành, ngoằn ngoèo, cao tốc, không rõ ngữ cảnh.
- *Trạng thái*: Buồn ngủ, tỉnh táo, không rõ ngữ cảnh.
- *Điều kiện giao thông*: Tắc đường, nhiều ô tô, đường vắng, không rõ ngữ cảnh.
- *Thời tiết*: Nắng, tuyêt rơi, mưa, nhiều mây, không rõ ngữ cảnh.

Thông tin chi tiết về bộ dữ liệu được trình bày trong bảng dưới đây:

Dataset	#User	#Items	#Ratings	Rating scale
InCarMusic	42	139	3938	1-5

Bảng 5: Thông tin về bộ dữ liệu InCarMusic

Bộ dữ liệu này bao gồm những đánh giá của người dùng từ 1 tới 5 điểm. Để phục vụ cho bài toán top N gợi ý, ta cần tập đầu vào là ma trận nhị phân. Do vậy, những chỉ số đánh giá mà bé hơn 3 sẽ được coi như người dùng không thích bài hát đó trong ngữ cảnh đó. Ngược lại nếu lớn hơn 3 sẽ là thích. Như vậy ma trận đánh giá của người dùng sẽ được chuyển về ma trận nhị phân:

$$R_{i,j}^* = \begin{cases} 1 & \text{nếu } R_{i,j} \geq 3 \\ 0 & \text{nếu } R_{i,j} < 3 \end{cases}$$

$R_{i,j}$  : Đánh giá của người dùng i với bài hát j.

### 2.4.2. Phương pháp đánh giá

Kết quả kiểm nghiệm này sử dụng phương pháp đánh giá chéo (k-fold cross validation) với top 10 gợi ý và sử dụng Precision@10, MAP@10, Recall@10 làm tiêu chí đánh giá.

Phương pháp đánh giá chéo sẽ chia tập dữ liệu thành k phần bằng nhau và chạy vòng lặp k lần. Tại mỗi bước lặp sẽ chọn một phần của tập dữ liệu làm tập kiểm nghiệm và các tập còn lại sẽ là tập huấn luyện. Các kết quả cuối cùng sẽ là trung bình kết quả của những bước lặp.



Hình ảnh 5: Phương pháp 3-fold cross validate

Trong đồ án này sẽ sử dụng phương pháp đánh giá chéo với  $k=5$  để kiểm nghiệm thuật toán. Việc kiểm nghiệm này nhằm đánh giá vai trò của yếu tố ngữ cảnh trong hệ tư vấn. Vậy nên thuật toán được chọn để so sánh với CSLIM\_MCS sẽ là SLIM.

### 2.4.3. Kết quả kiểm nghiệm

Những kết quả đánh giá dưới đây sử dụng một vài tham số được chọn dựa vào kinh nghiệm sao cho đạt kết quả tốt nhất:

- Độ tương quan sử dụng: Pearson correlation.
- Số lượng hàng xóm: 65.

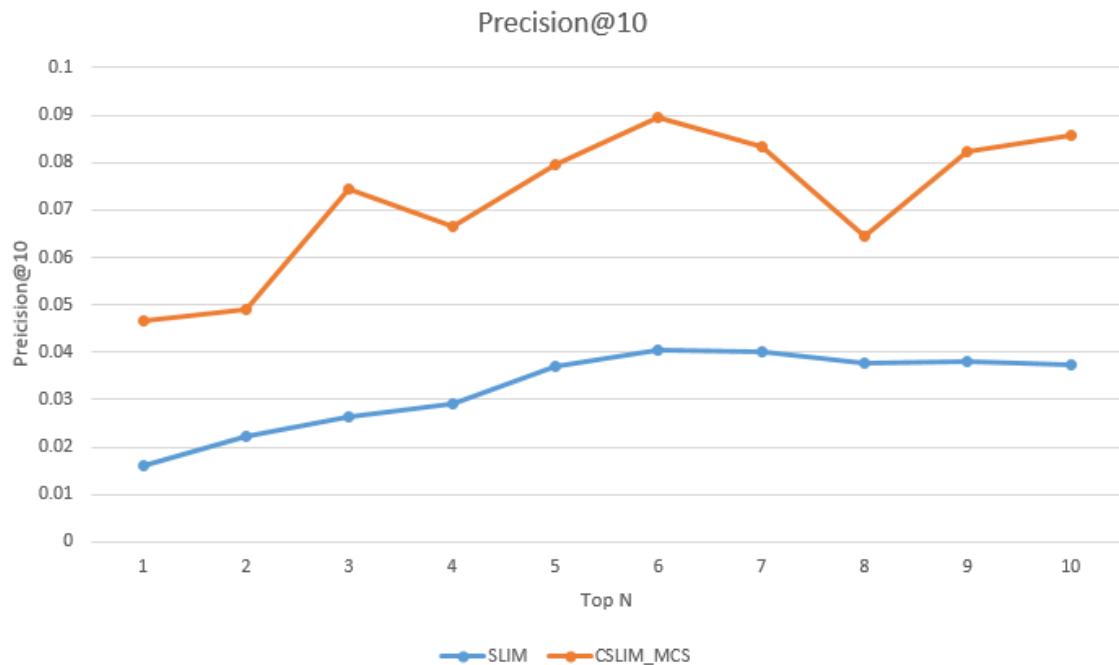
- Hệ số học: 0,001.
- Số lần lặp tối đa: 100.

Hai tham số quan trọng ảnh hưởng nhiều tới kết quả kiểm nghiệm là hai tham số chuẩn hóa  $\alpha$  và  $\beta$ . Sự ảnh hưởng của hai tham số tới các chỉ số được thể hiện trong bảng sau.

$\alpha$	$\beta$	SLIM			CSLIM_MCS		
		Prec@10	MAP@10	Recall@10	Prec@10	MAP@10	Recall@10
1	1	0.04135	0.183057	0.270466	0.05503	0.228592	0.340108
1	2	0.04333	0.183095	0.278044	0.05211	0.228418	0.334473
1	3	0.03871	0.154741	0.240836	0.06526	0.302242	0.375528
1	4	0.03677	0.124885	0.209953	0.06339	0.290966	0.351752
1	5	0.03793	0.137744	0.215869	<b>0.08572</b>	<b>0.312477</b>	<b>0.402636</b>
2	1	0.03123	0.121885	0.174329	0.02907	0.113552	0.207072
2	2	0.03048	0.111793	0.174120	0.03617	0.165616	0.243544
2	3	0.02983	0.107959	0.162861	0.04526	0.187728	0.279440
2	4	0.02989	0.082971	0.136990	0.03902	0.133472	0.213527
2	5	0.03038	0.093426	0.154468	0.03787	0.137161	0.185996
3	1	0.03254	0.115398	0.167384	0.01766	0.055476	0.132145
3	2	0.02804	0.095534	0.154410	0.02127	0.073685	0.160354
3	3	0.03635	0.111968	0.178196	0.02463	0.093356	0.171271
3	4	0.03790	0.088203	0.169820	0.02301	0.092953	0.146934
3	5	0.04245	0.081944	0.185026	0.02474	0.114435	0.178667
4	1	0.03927	0.105607	0.187209	0.01410	0.037469	0.106606
4	2	0.04883	0.072802	0.191290	0.01611	0.049697	0.116928
4	3	0.04958	0.071039	0.197974	0.01728	0.056075	0.122503
4	4	0.05579	0.096488	0.232262	0.02246	0.101127	0.164832
4	5	0.06029	0.098818	0.268198	0.01817	0.061305	0.121580
5	1	0.07236	0.127636	0.332883	0.01035	0.019268	0.075580
5	2	0.06531	0.103405	0.304035	0.01389	0.054963	0.097423
5	3	0.06246	0.122882	0.309221	0.01224	0.045432	0.085919
5	4	0.06343	0.101773	0.317189	0.01492	0.053977	0.106820
5	5	0.06245	0.143918	0.332366	0.016560	0.062265	0.116595

Bảng 6: Kết quả kiểm nghiệm và so sánh SLIM với CSLIM\_MCS

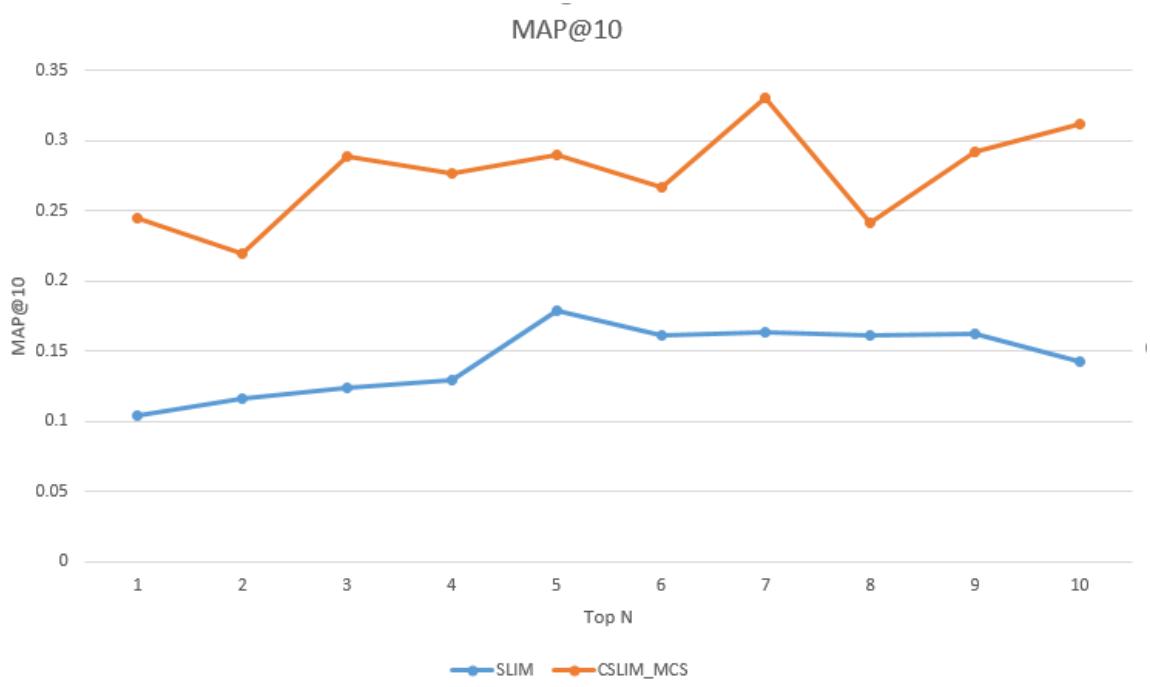
Qua kết quả trên, ta có thể thấy SLIM đạt kết quả tốt nhất với  $\alpha = 1$  và  $\beta = 2$ . Còn CSLIM\_MCS đạt kết quả tốt nhất khi  $\alpha = 1$  và  $\beta = 5$ . Có thể thấy rõ ràng CSLIM\_MCS có kết quả tốt hơn hoàn toàn ở cả 3 chỉ số đánh giá. Để rõ ràng hơn, ta sẽ chọn các tham số  $\alpha, \beta$  mà hai thuật toán đạt kết quả tốt nhất và thay đổi số lượng gợi ý. Bằng cách này, ta có thể xác định thuật toán sử dụng thêm ngữ cảnh CSLIM\_MCS có thực sự tốt hơn SLIM hay không. Kết quả kiểm nghiệm được trình bày trong 3 biểu đồ dưới, tương ứng với 3 chỉ số Precision, MAP và Recall:



Hình ảnh 6: Biểu đồ so sánh chỉ số Prec@10

TopN	1	2	3	4	5	6	7	8	9	10
SLIM	0.016	0.022	0.026	0.029	0.037	0.041	0.040	0.038	0.038	0.037
CSLIM_MCS	0.047	0.049	0.075	0.066	0.080	0.090	0.083	0.064	0.082	0.086

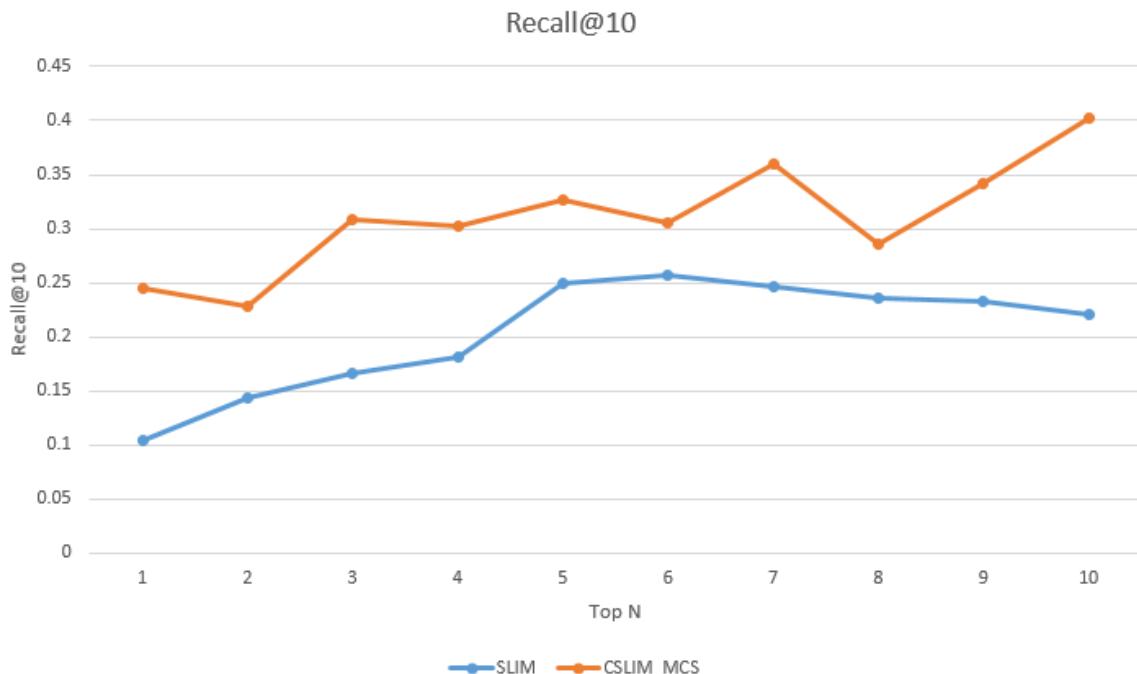
Bảng 7: Bảng kết quả so sánh dựa trên chỉ số Prec@10



Hình ảnh 7: Biểu đồ so sánh dựa trên chỉ số MAP@10

TopN	1	2	3	4	5	6	7	8	9	10
SLIM	0.104	0.116	0.124	0.129	0.179	0.161	0.164	0.161	0.163	0.143
CSLIM_MCS	0.245	0.220	0.289	0.277	0.290	0.266	0.331	0.242	0.292	0.312

Bảng 8: Bảng so sánh dựa trên chỉ số MAP@10



Hình ảnh 8: Biểu đồ so sánh dựa trên chỉ số Recall@10

TopN	1	2	3	4	5	6	7	8	9	10
SLIM	0.104	0.143	0.166	0.182	0.250	0.258	0.246	0.236	0.232	0.220
CSLIM_MCS	0.245	0.229	0.308	0.302	0.327	0.306	0.360	0.286	0.341	0.403

Bảng 9: Bảng so sánh dựa trên chỉ số Recall@10

Kết quả kiểm nghiệm cho thấy thuật toán CSLIM\_MCS cho hiệu quả gọi ý tốt hơn hẳn so với thuật toán SLIM. Điều này có nghĩa việc tích hợp yếu tố ngữ cảnh trong trường hợp này có thể cải thiện đáng kể chất lượng của hệ tư vấn.

## 2.5. Kết luận chương

Trong chương này tác giả đã trình bày về kỹ thuật tích hợp ngữ cảnh bằng phương pháp mô hình hóa dựa trên độ tương quan ngữ cảnh đa chiều vào hệ tư vấn truyền thống (thuật toán SLIM), để xây dựng thuật toán CSLIM\_MCS. Kết quả kiểm nghiệm thuật toán CSLIM\_MCS cho kết quả tốt hơn so với thuật toán SLIM chứng tỏ tính hiệu quả của việc tích hợp yếu tố ngữ cảnh cho bài toán tư vấn các sản phẩm đa phương tiện.

Trong chương 3, tác giả sẽ ứng dụng thuật toán CSLIM\_MCS để xây dựng hệ tư vấn địa điểm du lịch tại thành phố Hồ Chí Minh dựa trên ngữ cảnh.

## CHƯƠNG 3: PHÁT TRIỂN HỆ THỐNG TƯ VẤN ĐỊA DIỂM DU LỊCH TẠI THÀNH PHỐ HỒ CHÍ MINH DỰA TRÊN NGỮ CẢNH

Mục tiêu chính của chương này trình bày thiết kế và xây dựng hệ tư vấn địa điểm du lịch dựa trên ngữ cảnh sử dụng phương pháp CSLIM\_MCS đã được nghiên cứu trong chương 2. Hệ thống cho phép người dùng tra cứu và đưa ra những tư vấn về địa điểm du lịch tại thành phố Hồ Chí Minh. Đồng thời xây dựng thành công ứng dụng di động trên điện thoại chạy hệ điều hành Android sử dụng dịch vụ cung cấp bởi hệ tư vấn đã xây dựng.

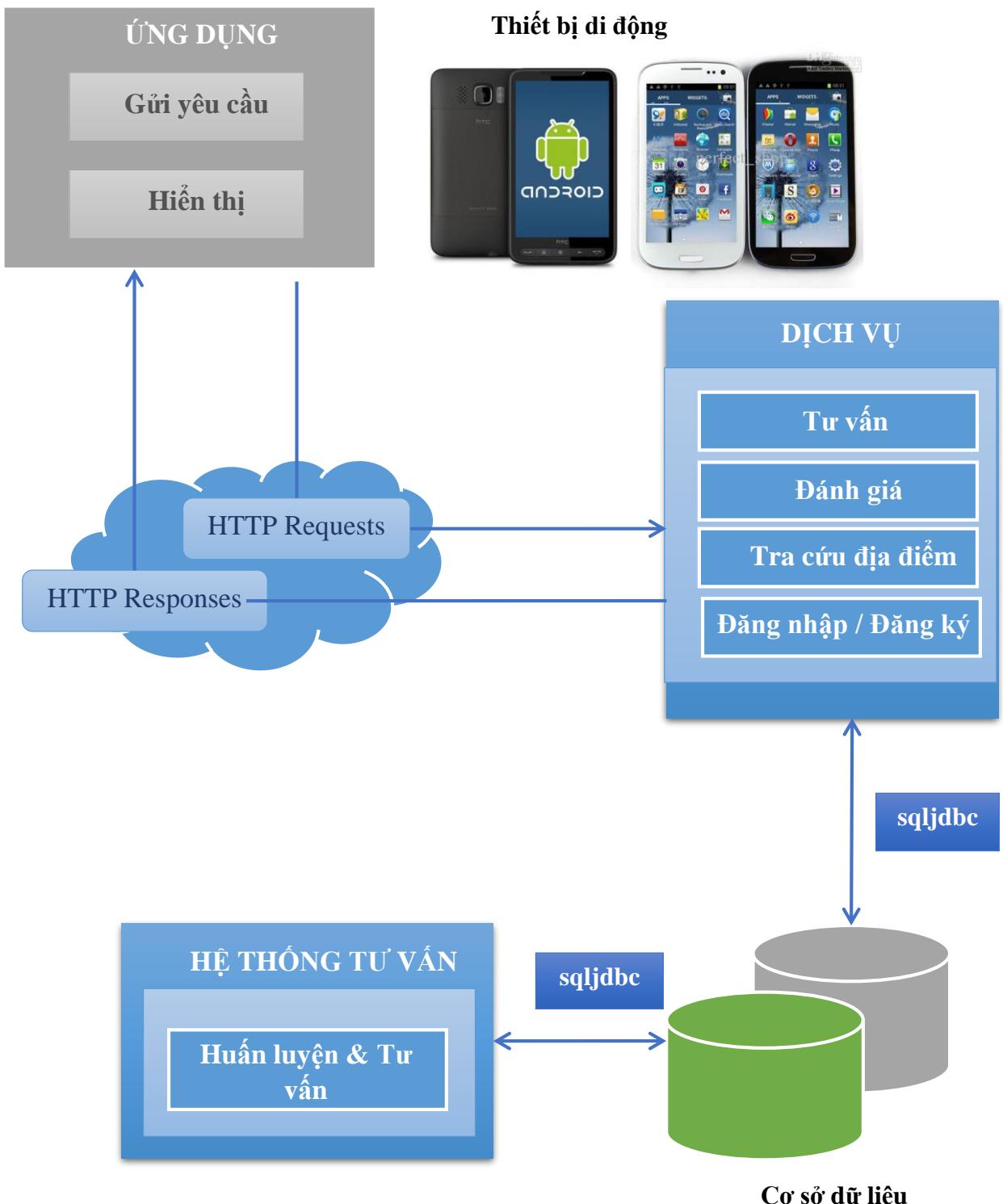
### 3.1. Yêu cầu hoạt động của hệ thống

Người dùng ứng dụng trên điện thoại di động chạy trên hệ điều hành Android có thể đăng nhập vào hệ thống, chọn điều kiện ngữ cảnh và tiến hành đánh giá sản phẩm. Hệ thống ghi nhận những đánh giá đó và lấy làm cơ sở để đưa ra những tư vấn địa điểm du lịch trả về điện thoại để hiển thị thông tin tới người dùng.

### 3.2. Mô hình tổng quát hệ thống

Kiến trúc của hệ thống được mô tả trong hình dưới đây. Thiết kế bao gồm hai phần chính:

- *Ứng dụng client:* Ứng dụng di động được viết cho nền tảng Android, hỗ trợ người dùng những chức năng như đăng nhập/ đăng ký, tra cứu địa điểm du lịch, đánh giá và xem những gợi ý địa điểm du lịch từ hệ thống.
- *Ứng dụng server:* Hệ thống server bao gồm:
  - *Dịch vụ web* phụ trách việc truyền nhận thông tin giữa client và server.
  - *Cơ sở dữ liệu* lưu trữ thông tin về người dùng, địa điểm du lịch, đánh giá và những gợi ý địa điểm du lịch cho người dùng.
  - *Hệ thống tư vấn* bao gồm hai chức năng chính là huấn luyện dữ liệu để đưa ra những tư vấn về địa điểm du lịch và lưu những tư vấn này vào cơ sở dữ liệu.



Hình ảnh 9: Mô hình tổng quan hệ thống

### 3.3. Công nghệ áp dụng

Tương ứng với mô hình tổng quan ở trên, công nghệ áp dụng cho từng phần sẽ bao gồm:

- *Ứng dụng client:* Ứng dụng client chạy trên hệ điều hành Android.

- *Dịch vụ web:* Sử dụng dịch vụ web REST.
- *Hệ thống tư vấn:* Là ứng dụng console chạy trên máy tính, sử dụng thư viện CARSKit triển khai thuật toán CSLIM\_MCS để đưa ra những gợi ý cho người dùng.

### 3.3.1. Hệ điều hành Android

Android là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005. Hiện nay phiên bản mới nhất của hệ điều hành này là Android 7.0 hay Android N (Nougat).



*Hình ảnh 10: Android Nougat*

Android có mã nguồn mở và Google phát hành mã nguồn theo giấy phép Apache. Chính mã nguồn mở cùng với một giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên được điều chỉnh và phân phối Android một cách tự do. Từ đó các hãng điện thoại di động có thể tự do can thiệp khá sâu vào hệ điều hành và tạo nên những thay đổi cho phù hợp với phần cứng của mình.

Android được viết bằng ngôn ngữ lập trình Java, một ngôn ngữ lập trình phổ biến nhất trên thế giới hiện nay. Android là hệ điều hành mã nguồn mở, được viết bằng ngôn ngữ phổ biến, là hệ điều hành có khả năng tinh chỉnh mạnh mẽ, không quá nặng nề và giá rẻ. Chính vì vậy mà Android trở thành lựa chọn hàng đầu cho những công ty công nghệ cao. Hơn nữa những ứng dụng Android còn có thể được phân phối qua kho ứng dụng Google Play, một trong kho ứng dụng lớn nhất trên thế giới. Đây là một kênh phân phối ứng dụng rất uy tín. Ước tính đến tháng 10 năm 2012, có khoảng 700.000 ứng dụng trên Android, và số lượt tải ứng dụng từ Google Play, cửa hàng ứng dụng chính của Android, ước tính khoảng 25 tỷ lượt, thu về nguồn lợi nhuận khổng lồ.

Trong đồ án này, tác giả lựa chọn sử dụng nền tảng Android vì hai lý do:

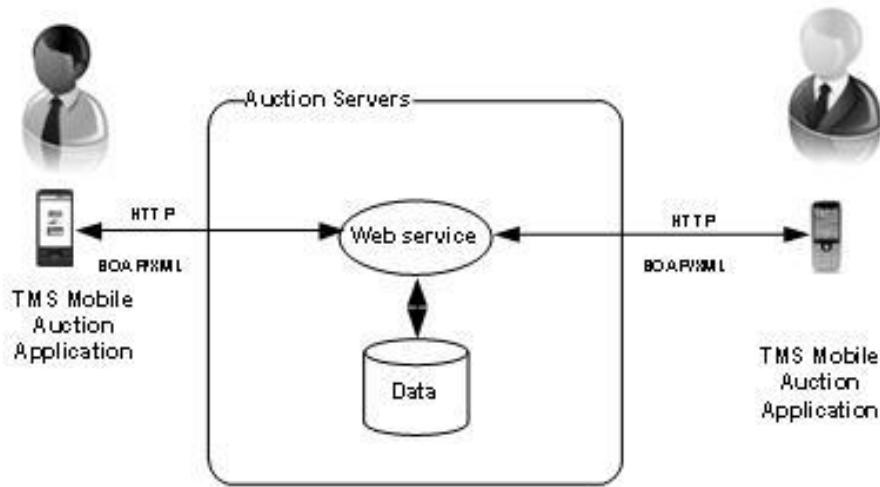
- Android phổ biến bậc nhất trên thế giới so với những nền tảng cho thiết bị di động khác. Vì vậy khi gặp khó khăn trong quá trình phát triển, việc tham khảo cộng đồng lập trình viên Android rất dễ dàng.
- Android là mã nguồn mở nên có rất nhiều những thư viện tiện ích, giúp cho quá trình phát triển ứng dụng Android trở nên đơn giản hơn rất nhiều. Ví dụ như thư viện Picasso hỗ trợ vấn đề tải và cache ảnh, thư viện OKHttp hỗ trợ việc gửi yêu cầu và nhận hồi đáp từ những dịch vụ web qua giao thức HTTP hoặc HTTPS, Simple XML cho phép ép một chuỗi có cấu trúc XML về một đối tượng cụ thể, ...

Ứng dụng “Đi đâu bây giờ ?” sử dụng mô hình Client – Server nên để server và client có thể kết nối với nhau, tác giả sử dụng công nghệ dịch vụ web REST, cho phép ứng dụng Android có thể tương tác với server qua giao thức HTTP.

### 3.3.2. Dịch vụ web REST

#### 3.3.2.1. Dịch vụ web là gì ?

Có nhiều cách định nghĩa khác nhau về dịch vụ web. Theo tổ chức W3C, dịch vụ web là một hệ thống phần mềm được thiết kế để hỗ trợ sự giao tiếp giữa các thiết bị với nhau thông qua những giao thức trên đường truyền mạng.



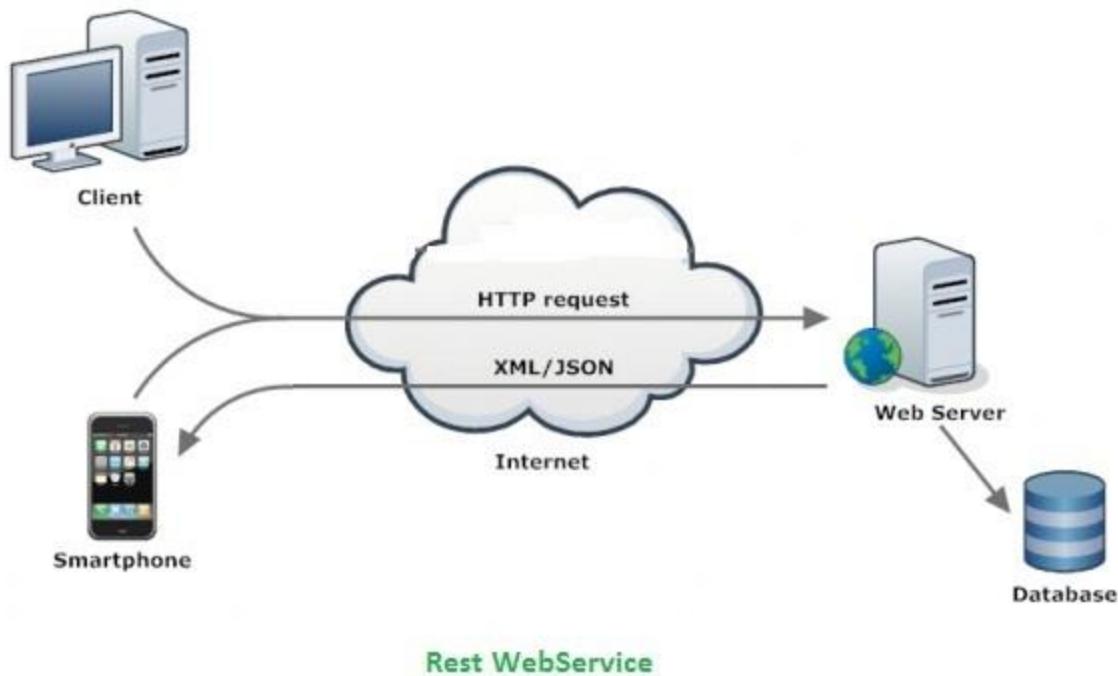
Hình ảnh 11: Mô hình của dịch vụ web

Dịch vụ web giúp thông tin được truyền tải theo cách độc lập nền tảng thiết bị và ngôn ngữ lập trình, chạy trên đa hệ điều hành vì mọi thứ đều được chuẩn hóa về dạng web. Dịch vụ web đặc biệt hữu dụng khi ta xây dựng một ứng dụng với số lượng người dùng lên đến hàng trăm, hàng ngàn người và phân tán ở những địa điểm khác nhau.

#### 3.3.2.2. REST

REST (Representational State Transfer) đã được chọn sử dụng rộng rãi thay cho dịch vụ web truyền thống dựa trên SOAP và WSDL. Bằng chứng quan trọng của

sự thay đổi này chính là việc các công ty dẫn đầu trong lĩnh vực cung cấp dịch vụ mạng 2.0 như Yahoo, Google và Facebook đã phản đối các giao thức dựa trên SOAP hoặc WSDL và ủng hộ phương thức hướng đến tài nguyên và dễ sử dụng đối với các dịch vụ của họ.



*Hình ảnh 12: Dịch vụ web REST*

REST định nghĩa các quy tắc kiến trúc để bạn thiết kế dịch vụ web chú trọng vào tài nguyên hệ thống, bao gồm các trạng thái tài nguyên được định dạng như thế nào và được chuyển tải qua HTTP thông qua số lượng lớn người dùng và được viết bởi những ngôn ngữ khác nhau. Hệ thống REST xoay quanh việc đơn giản hóa việc lấy các biểu diễn của một tài nguyên hệ thống.

Một đặc tính quan trọng của dịch vụ web REST là sử dụng một cách rõ ràng các phương thức HTTP theo cách một giao thức được xác định bởi RFC 2616:

- Để tạo một tài nguyên trên máy chủ, bạn cần sử dụng phương thức **POST**.
- Để truy xuất một tài nguyên, sử dụng **GET**.
- Để thay đổi trạng thái một tài nguyên hoặc để cập nhật nó, sử dụng **PUT**.
- Để huỷ bỏ hoặc xoá một tài nguyên, sử dụng **DELETE**.

REST xác định cấu trúc thư mục giống URI. Loại URI này có thứ bậc, có điểm khởi nguồn tại một đường dẫn đơn giản, và có nhánh đi ra là các nhánh phụ thể hiện các vùng chính của dịch vụ. Theo định nghĩa này, một URI không chỉ là một chuỗi bị cắt không giới hạn, mà còn là một cây với các nhánh chính và nhánh dọc nối với nhau tại các nút. Ví dụ ta có dịch vụ truy xuất bằng URI như sau:

“<http://www.myservice.com/discussion/topics/{topic}>”

Phần gốc, “/discussion”, có một nút /topics bên dưới nó. Phía dưới là một chuỗi tên các chủ đề, như xã hội, kỹ thuật, v.v.., mỗi chủ đề chỉ ra một loại. Trong cấu trúc này, ta dễ dàng có thể liệt kê một vài chủ đề cụ thể bằng cách gõ một vài thứ sau “/topics/” như “/topics/{science}” sẽ liệt kê ra những bài viết với chủ đề về khoa học.

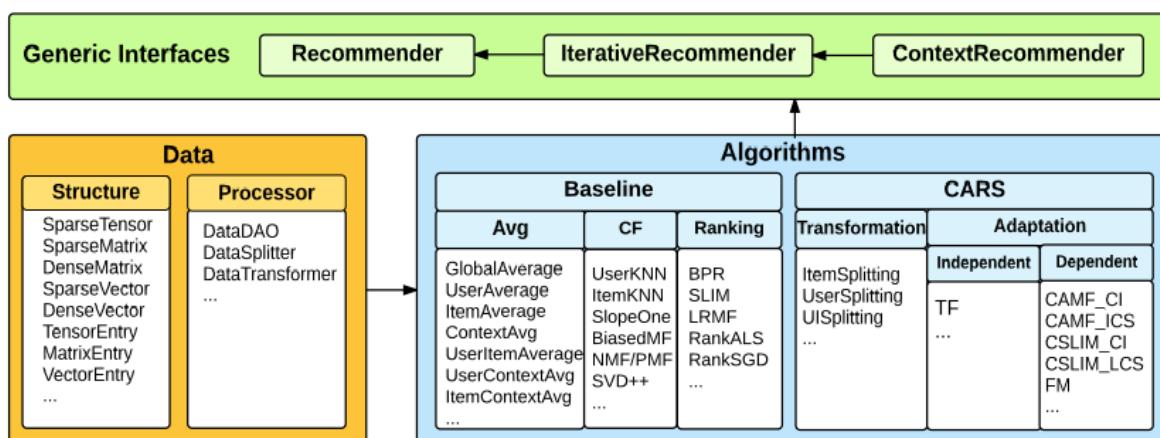
REST có thể trả về cho client tài nguyên dưới dạng XML hoặc JSON. Phía client chỉ cần ép ngược từ XML (JSON) về đối tượng cần sử dụng. Ứng dụng “Đi đâu bây giờ?” lưu trữ tài nguyên trên server và sử dụng dịch vụ web REST để truy xuất những tài nguyên dưới dạng mô tả XML, và sau đó ép về các đối tượng cần sử dụng.

Để xây dựng dịch vụ web REST, tác giả sử dụng thư viện Jersey, một thư viện mã nguồn mở, được sử dụng để phát triển những dịch vụ web REST bằng Java. Phía ứng dụng android sẽ sử dụng thư OkHttpClient để gọi tới dịch vụ web, dịch vụ web sẽ truy xuất thông tin từ trong cơ sở dữ liệu MySQL và trả về dưới dạng mô tả XML. Sau đó ứng dụng sẽ sử dụng thư viện Simple XML để ép về các đối tượng cần sử dụng.

### 3.3.3. Thư viện CARSKit

Để có thể đưa ra những gợi ý thông minh bằng những kỹ thuật đã trình bày ở chương 2, tác giả sử dụng thư viện CARSKit (Context-aware Recommender System Kit). CARSKit là một thư viện mã nguồn mở Java được cấp giấy phép bản quyền GNU, hỗ trợ trong việc thực thi những thuật toán của hệ tư vấn nói chung và hệ tư vấn kết hợp ngữ cảnh nói riêng.

CARSKit có thiết kế kiến trúc rất linh hoạt, có thể tích hợp để sử dụng những thuật toán đã được cài đặt sẵn hoặc mở rộng thuật toán hay tự cài đặt một thuật toán mới. Hình ảnh dưới đây là toàn bộ kiến trúc của CARSKit.



Hình ảnh 13: Kiến trúc của CARSKit

CARSKit chia làm 3 phần chính:

- **Generic Interfaces:** Bao gồm những lớp trừu tượng để lập trình viên từ đó triển khai các thuật toán. Đây cũng là giao diện để lập trình viên gán bộ dữ

liệu và các siêu tham số cho thuật toán. Những hàm đánh giá thuật toán (bao gồm đánh giá cho bài toán dự đoán đánh giá và bài toán dự đoán xếp hạng) được cài đặt sẵn trong lớp *Recommender*.

- *Data*: Khối này bao gồm những cấu trúc dữ liệu phục vụ cho việc lưu trữ, xử lý tính toán trên ma trận cỡ lớn và những thao tác lưu trữ, tiền xử lý dữ liệu đầu vào như lưu trữ ma trận đánh giá có ngữ cảnh (phục vụ cho tư vấn truyền thống) hoặc không có ngữ cảnh (phục vụ cho tư vấn kết hợp ngữ cảnh), chia tập dữ liệu để thực hiện kỹ thuật đánh giá chéo, ...
- *Algorithms*: Khối này là những thuật toán đã được cài đặt sẵn, bao gồm cả những thuật toán tư vấn truyền thống (SLIM, MF, User-based, Item-based, ...) và những thuật toán tư vấn kết hợp ngữ cảnh (CSLIM\_LCS, CLSIM\_MCS, ...).

Hệ thống tư vấn địa điểm du lịch kết hợp ngữ cảnh cho ứng dụng “Đi đâu bây giờ ?” sử dụng thuật toán CSLIM\_MCS của bộ thư viện CARSKit để đưa ra những gợi ý về địa điểm du lịch cho người dùng dựa vào những điều kiện ngữ cảnh do người dùng cung cấp.

### **3.4. Phân tích thiết kế hệ thống tư vấn địa điểm du lịch tại thành phố Hồ Chí Minh**

Trong phần này, tác giả sẽ phân tích những yêu cầu của ứng dụng, từ đó thiết kế tương tác cho ứng dụng và phân tích thiết kế cho hệ thống tư vấn địa điểm du lịch tại thành phố Hồ Chí Minh. Trước tiên, ta cần phân tích những yêu cầu của ứng dụng.

#### **3.4.1. Phân tích yêu cầu**

Trước tiên ta cần phân tích ứng dụng có những chức năng gì. Từ đó có thể xây dựng những chức năng của server đáp ứng những yêu cầu của client.

##### **3.4.1.1. Phía client**

Ứng dụng client bao gồm những chức năng sau:

- Đăng nhập/Đăng ký tài khoản.
- Gọi ý địa điểm du lịch:
  - Gọi ý top 10 điểm đến có đánh giá trung bình cao nhất.
  - Gọi ý cho người dùng dựa theo ngữ cảnh người dùng chọn.
  - Gọi ý những địa điểm cùng danh mục với địa điểm người dùng đang xem.
- Tìm kiếm địa điểm:
  - Tìm kiếm dựa vào tên địa điểm du lịch.
  - Lọc kết quả tìm kiếm theo danh mục địa điểm du lịch.
- Tương tác với một địa điểm cụ thể:
  - Xem thông tin chi tiết địa điểm du lịch đó.
  - Đánh giá (có kèm thông tin ngữ cảnh) địa điểm du lịch.
  - Liên hệ với ban quản lý địa điểm du lịch.

- Bản đồ chỉ đường từ vị trí hiện tại tới địa điểm đó.
- Lưu/Xóa địa điểm đó trong danh sách yêu thích.
- Quản lý danh sách yêu thích:
  - Hiển thị danh sách yêu thích.
  - Xóa những địa điểm trong danh sách yêu thích.

#### **3.4.1.2. Phía server**

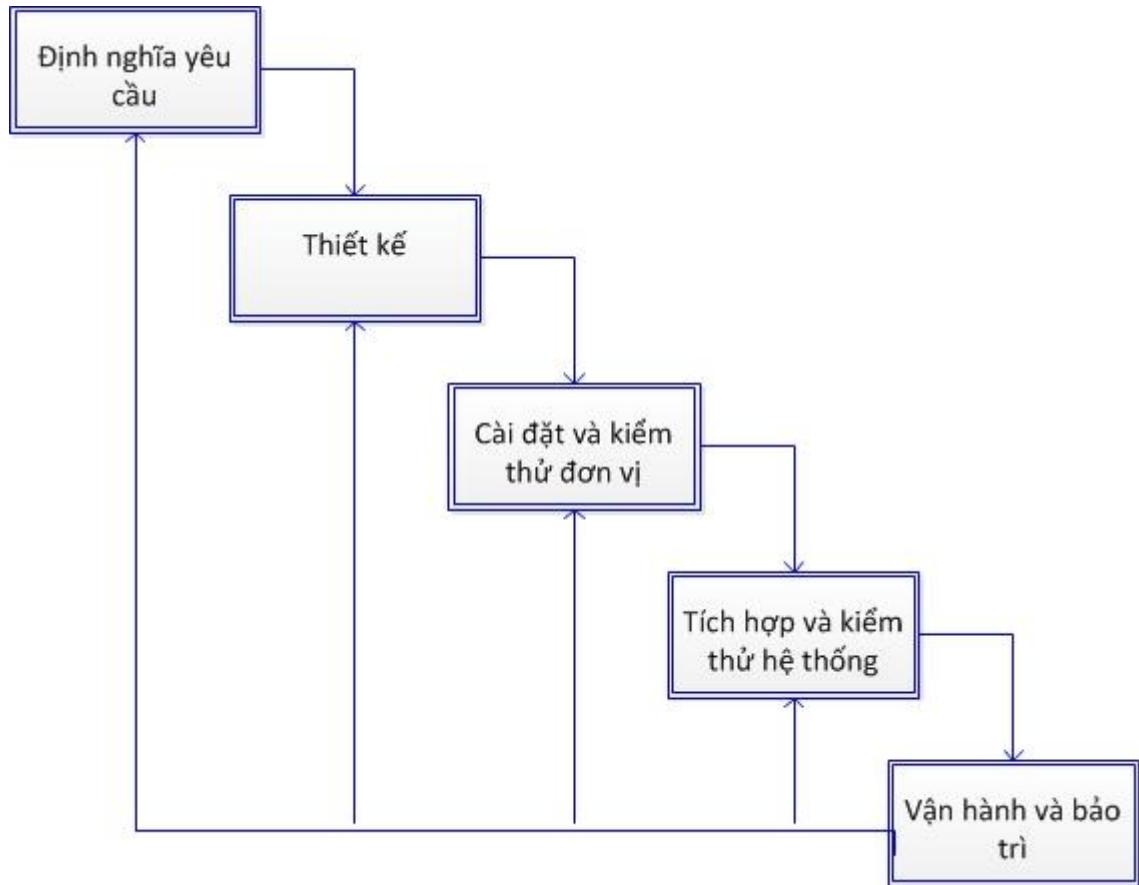
Phía server cần được xây dựng để có khả năng đáp ứng yêu cầu từ client:

- Quản lý người dùng:
  - Lưu trữ thông tin người dùng.
  - Đăng ký người dùng mới.
  - Đăng nhập.
- Gợi ý:
  - Đưa ra top 10 địa điểm có đánh giá trung bình cao nhất.
  - Đưa ra 10 gợi ý địa điểm cho người dùng dựa vào ngữ cảnh từ phía client (sử dụng CARSKit để dự đoán trước và lưu vào cơ sở dữ liệu, khi cần đưa ra gợi ý thì chỉ cần truy vấn).
  - Đưa ra những địa điểm cùng danh mục với địa điểm người dùng đang xem.
- Tìm kiếm:
  - Đưa ra danh sách kết quả tìm kiếm địa điểm có tên chưa chuỗi từ khóa từ phía client.
- Quản lý địa điểm du lịch:
  - Lưu trữ thông tin địa điểm du lịch.
  - Đưa ra thông tin chi tiết địa điểm du lịch.
- Quản lý đánh giá của người dùng với địa điểm:
  - Lưu trữ đánh giá của người dùng với địa điểm trong một ngữ cảnh.
  - Thêm đánh giá của người dùng với địa điểm trong một ngữ cảnh.
- Huấn luyện dữ liệu và lưu trữ những gợi ý.

Như vậy các yêu cầu của hệ thống đã khá rõ ràng. Bước tiếp theo ta sẽ tiến hành thiết kế tương tác, phục vụ cho quá trình xây dựng giao diện sản phẩm.

#### **3.4.2. Mô hình phát triển phần mềm**

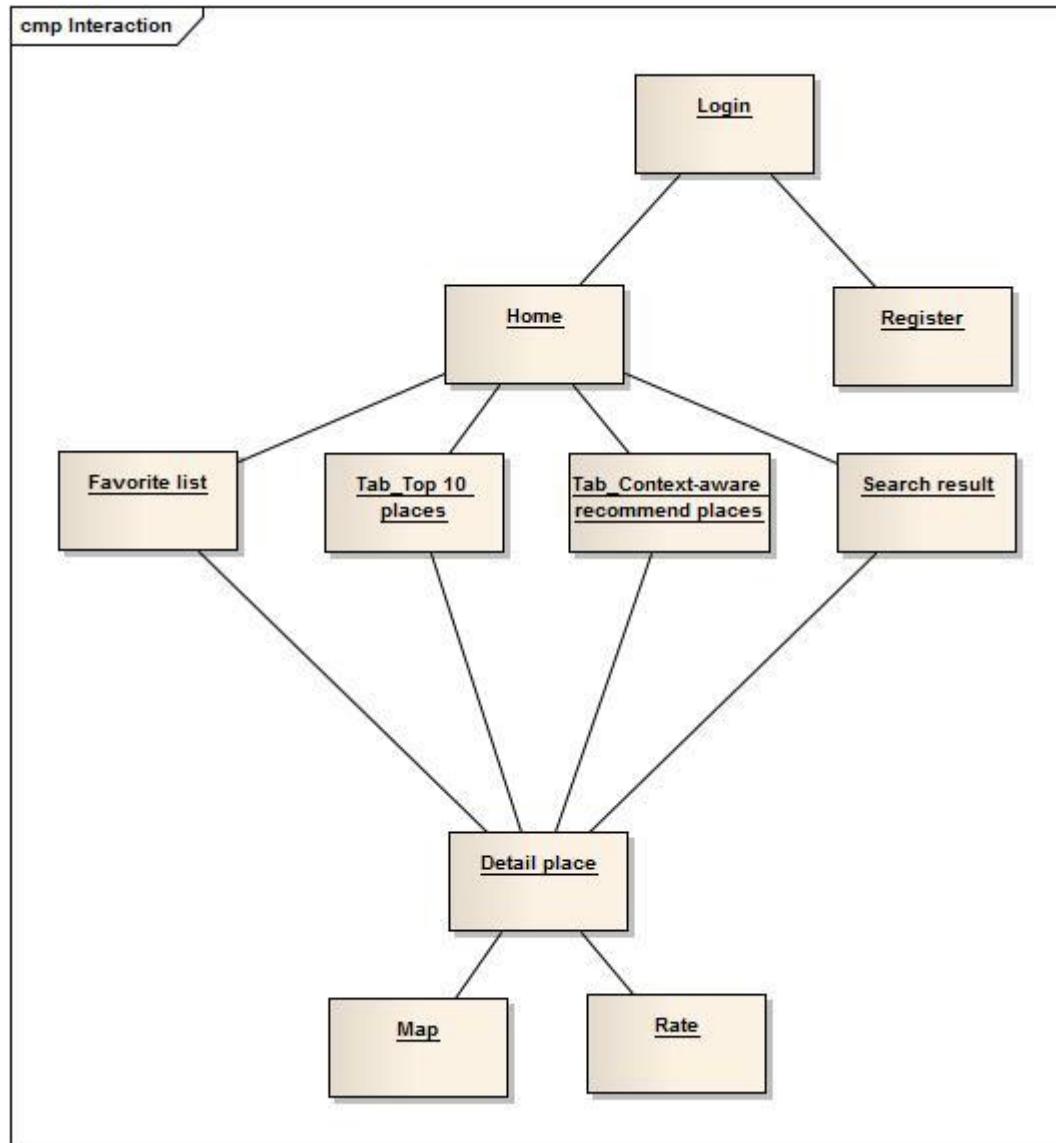
Đồ án sử dụng mô hình thác nước vì project không quá lớn và các chức năng của ứng dụng rõ ràng, không quá phức tạp. Mặt khác, khóa luận đã xác định đầy đủ các yêu cầu cần thiết với ứng dụng này.



Hình ảnh 14: Phát triển ứng dụng theo mô hình thác nước

### 3.4.3. Hệ thống giao diện ứng dụng gợi ý địa điểm du lịch

Dưới đây là cây phân cấp giao diện của ứng dụng. Cây phân cấp giao diện thể hiện mối quan hệ giữa các giao diện. Cây này cũng thể hiện những thao tác của người dùng khi muốn tới một giao diện nào đó.



Hình ảnh 15: Cây phân cấp giao diện ứng dụng

Có tất cả 8 giao diện, trong đó:

- *Favorite list*: Hiển thị danh sách địa điểm đã yêu thích.
- *Home*: Bao gồm hai tab:
  - *Top 10 places*: Hiển thị danh sách 10 địa điểm có đánh giá trung bình cao nhất.
  - *Context-aware recommend places*: Hiển thị danh sách địa điểm gợi ý dựa trên ngữ cảnh.
- *Search result*: Hiển thị danh sách kết quả tìm kiếm.
- *Login*: Màn hình đăng nhập.
- *Register*: Màn hình đăng ký.
- *Detail place*: Hiển thị thông tin chi tiết của một địa điểm du lịch.
- *Map*: Hiển thị chỉ dẫn đường tới địa điểm được chọn.
- *Rate*: Hiển thị mẫu đánh giá địa điểm du lịch.

Màn hình đầu tiên khi người dùng truy cập ứng dụng là màn hình **đăng nhập** (*Login*). Sau đó người dùng có thể chuyển qua giao diện **đăng ký** (*Register*) bằng cách chọn đăng ký hoặc chuyển qua giao diện **màn hình chính** (*Home*) bằng cách bỏ qua đăng nhập hoặc đăng nhập đúng tài khoản.

Khi ở giao diện **đăng ký**, nếu quá trình đăng ký thành công thì sẽ tự động chuyển qua giao diện **màn hình chính**, nếu quá trình đăng ký thất bại, người dùng nếu nhấn phím cứng quay lại thì sẽ chuyển về màn hình **đăng nhập**.

Tại giao diện **màn hình chính**, người dùng có thể chọn một trong hai tab là top 10 địa điểm (*Top 10 places*) và gợi ý dựa vào ngữ cảnh (*Context-aware recommend places*) để hiển thị ra danh sách địa điểm. Người dùng cũng có thể chuyển qua giao diện **danh sách yêu thích** (*Favorite list*) bằng cách sử dụng side menu hoặc **kết quả tìm kiếm** (*Search result*) bằng cách tìm kiếm địa điểm.

Trong những giao diện có chứa danh sách địa điểm (bao gồm **màn hình chính**, **danh sách yêu thích** và **kết quả tìm kiếm**), người dùng có thể tới giao diện **chi tiết địa điểm** (*Detail place*) bằng cách chọn một địa điểm trong danh sách.

Để đánh giá một địa điểm, tại giao diện **chi tiết địa điểm**, người dùng chọn đánh giá và sẽ chuyển tới giao diện **đánh giá địa điểm du lịch** (*Rate*). Tại giao diện **chi tiết địa điểm** người dùng cũng có thể tìm chỉ dẫn đường đi trên bản đồ bằng cách chọn bản đồ. Ứng dụng sẽ chuyển tới giao diện **bản đồ chỉ dẫn** (*Map*).

### 3.4.4. Phân tích thiết kế hệ thống cho ứng dụng gợi ý địa điểm du lịch tại thành phố Hồ Chí Minh

#### 3.4.4.1. Phía client

##### 3.4.4.1.1. Biểu đồ use case

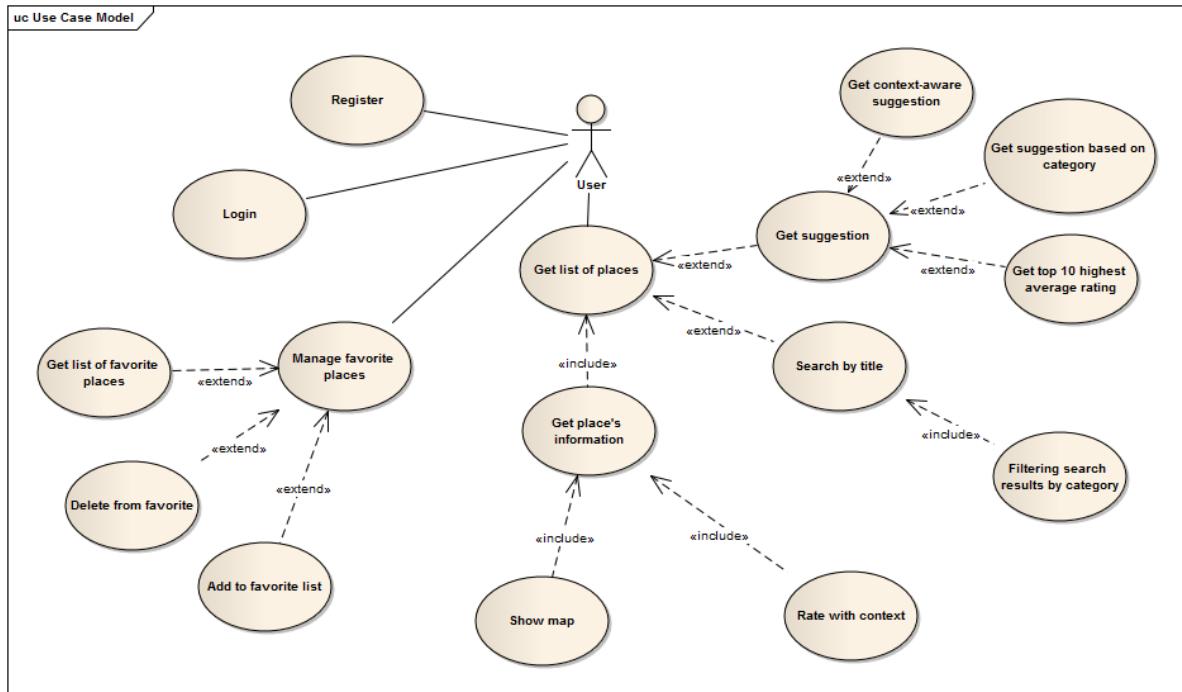
Ở phía ứng dụng client, tác nhân duy nhất ảnh hưởng tới ứng dụng là: *Người dùng (User)*, là người sử dụng ứng dụng tư vấn địa điểm du lịch tại thành phố Hồ Chí Minh. Bảng dưới đây mô tả khái quát những ca sử dụng mà người dùng có thể tương tác với ứng dụng:

STT	Ca sử dụng	Mô tả ca sử dụng
1	Đăng nhập (Login)	Người dùng tiến hành đăng nhập vào hệ thống.
2	Đăng ký (Register)	Người dùng đăng ký thành viên để có thể đăng nhập.
3	Xem danh sách yêu thích (Get list of favorite places)	Người dùng xem danh sách địa điểm du lịch đã lưu.
4	Xóa địa điểm khỏi danh sách yêu thích (Delete from favorite)	Người dùng loại bỏ một địa điểm khỏi danh sách yêu thích.

5	Thêm địa điểm vào danh sách yêu thích (Add to favorite)	Người dùng thêm địa điểm đang xem vào danh sách yêu thích.
6	Xem thông tin chi tiết một địa điểm (Get place's information)	Người dùng xem thông tin chi tiết của địa điểm du lịch được chọn.
7	Xem bản đồ (Show map)	Người dùng xem địa điểm du lịch trên bản đồ.
8	Liên hệ ban quản lý địa điểm (Call manager)	Người dùng gọi điện cho cơ quan quản lý địa điểm du lịch.
9	Đánh giá địa điểm du lịch kèm ngữ cảnh (Rate with context)	Người dùng điện và gửi đánh giá địa điểm du lịch.
10	Tìm kiếm địa điểm du lịch dựa theo tên (Search by title)	Người dùng nhập tên (hoặc một phần của tên) địa điểm du lịch để tìm kiếm địa điểm.
11	Lọc kết quả tìm kiếm theo danh mục (Filtering search results by category)	Người dùng chọn những danh mục địa điểm nào sẽ được hiển thị trên màn hình.
12	Xem danh sách top 10 địa điểm (Get top 10 highest average rating)	Người dùng xem danh sách 10 địa điểm có đánh giá trung bình cao nhất.
13	Xem danh sách địa điểm cùng danh mục (Get suggestion based on category)	Người dùng xem danh sách địa điểm có cùng danh mục với địa điểm đang xem.
14	Xem những địa điểm được gợi ý dựa vào ngữ cảnh (Get context-aware suggestion)	Người dùng xem danh sách những địa điểm được gợi ý dựa vào những thông tin ngữ cảnh cung cấp.

Bảng 10: Danh sách ca sử dụng phía client

Biểu đồ use case dưới đây sẽ chỉ ra những ca sử dụng mà tác nhân *người dùng* có thể tương tác với ứng dụng.



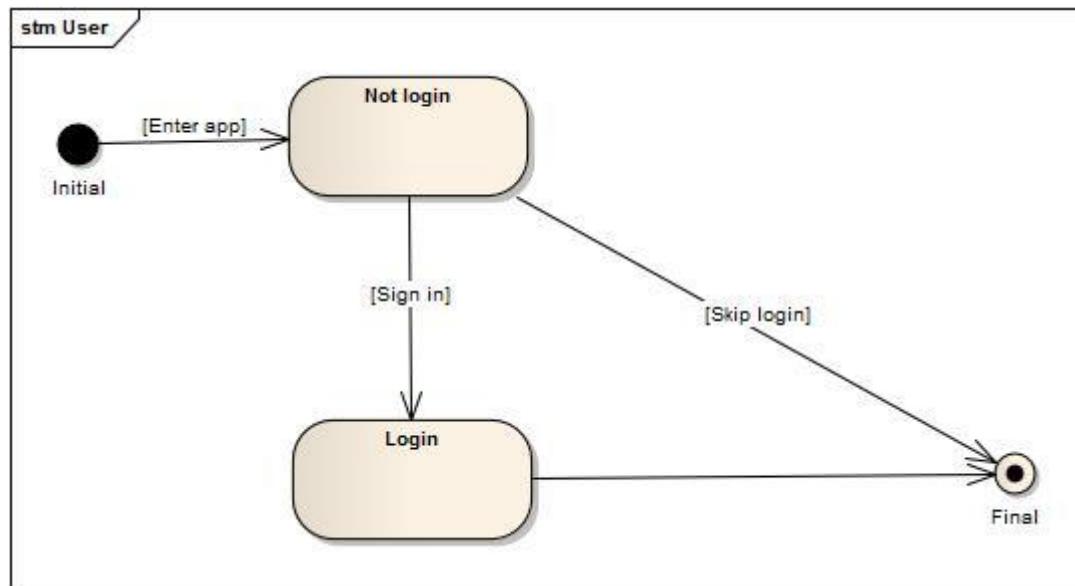
Hình ảnh 16: Biểu đồ use case phía client

Người dùng có thể đăng ký để đăng nhập vào hệ thống. Việc đăng nhập sẽ giúp quá trình đưa ra gợi ý dựa vào ngữ cảnh tốt hơn.

Người dùng sẽ xem danh sách địa điểm du lịch dựa vào những gợi ý (bao gồm top 10 địa điểm có đánh giá trung bình cao nhất, những bộ địa điểm cùng danh mục với địa điểm đang xem và gợi ý dựa vào ngữ cảnh) hoặc tìm kiếm. Sau khi chọn một địa điểm, người dùng sẽ được xem thông tin chi tiết. Người dùng có thể đánh giá địa điểm du lịch, xem chỉ dẫn đường đi, liên hệ ban quản lý địa điểm du lịch đó hoặc thêm/xóa trong danh sách yêu thích.

#### 3.4.4.1.2. Biểu đồ trạng thái

Biểu đồ trạng thái sẽ thể hiện sự chuyển hóa giữa những trạng thái mà có ảnh hưởng tới ứng dụng của một đối tượng. Ở phía client chỉ có đối tượng User có trạng thái.

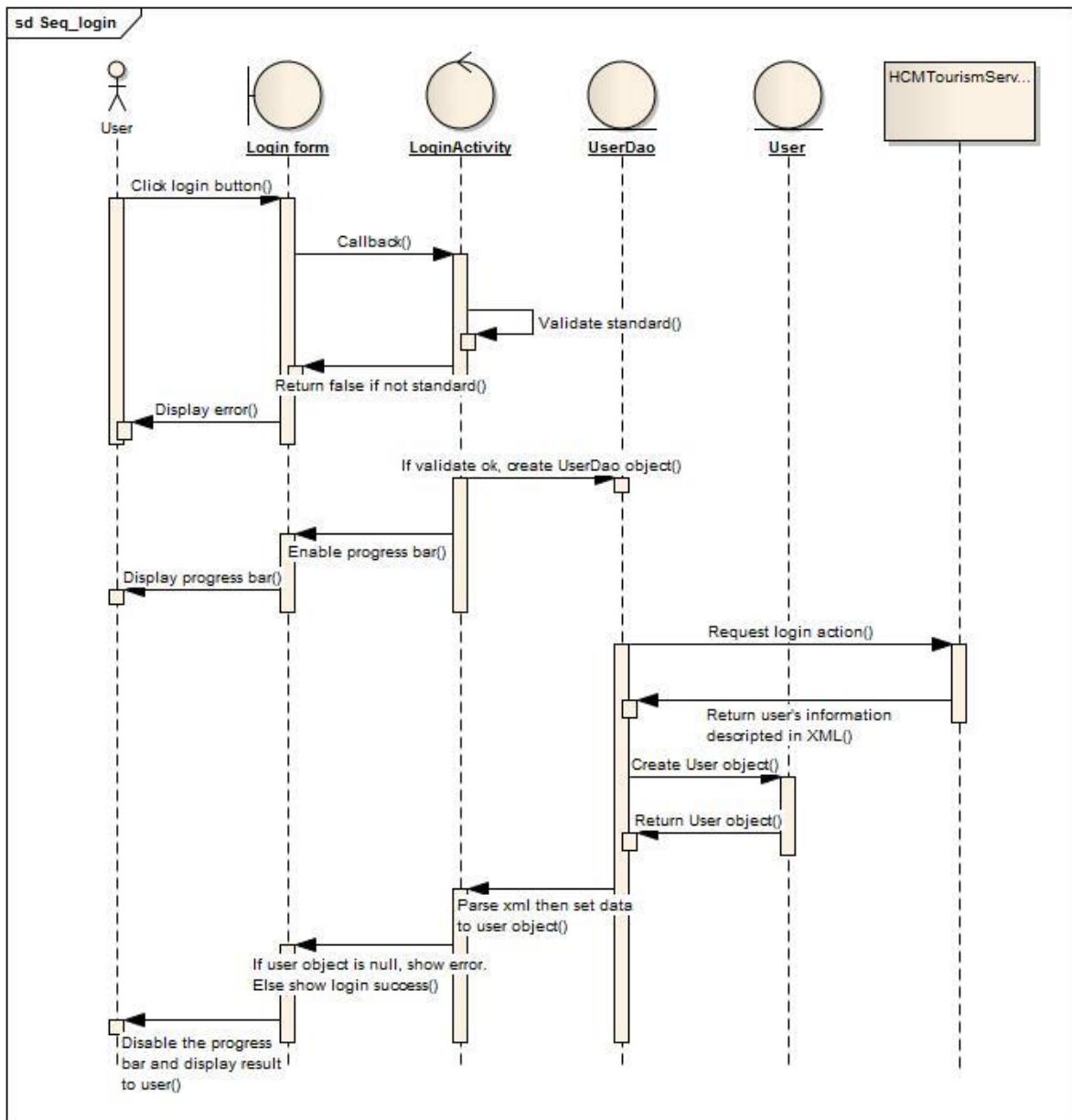


Hình ảnh 17: Biểu đồ trạng thái của User

Khi sử dụng chức năng gọi ý dựa vào ngữ cảnh, người dùng cần đăng nhập vào hệ thống để có thể xác định được thông tin người dùng, phục vụ cho quá trình đưa ra gợi ý. Vì vậy đối tượng User sẽ có hai trạng thái là đăng nhập hoặc chưa đăng nhập.

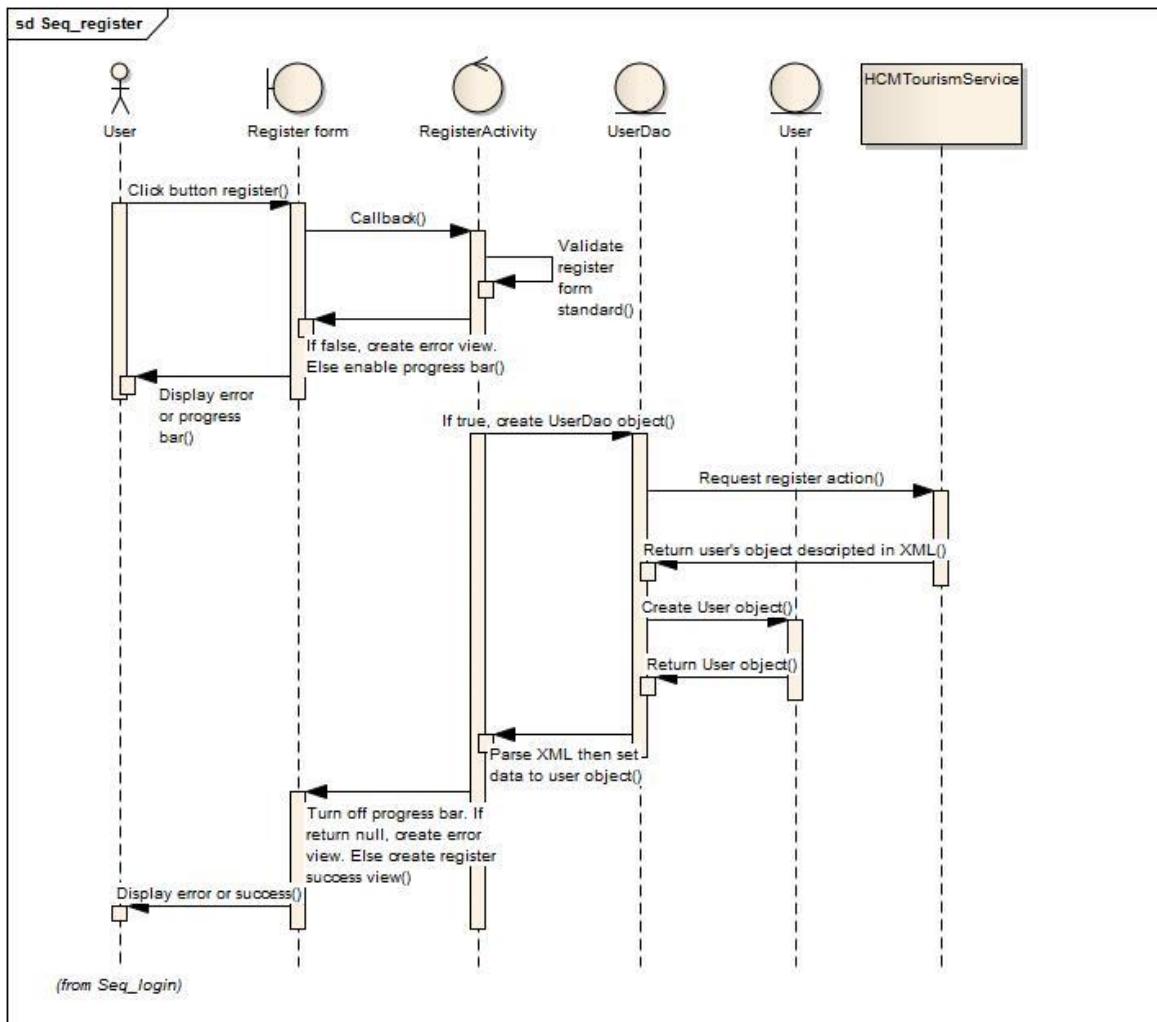
#### 3.4.4.1.3. Biểu đồ tuần tự

Biểu đồ tuần tự phía client thể hiện luồng thực hiện các chức năng phía client.

a) *Chức năng đăng nhập*

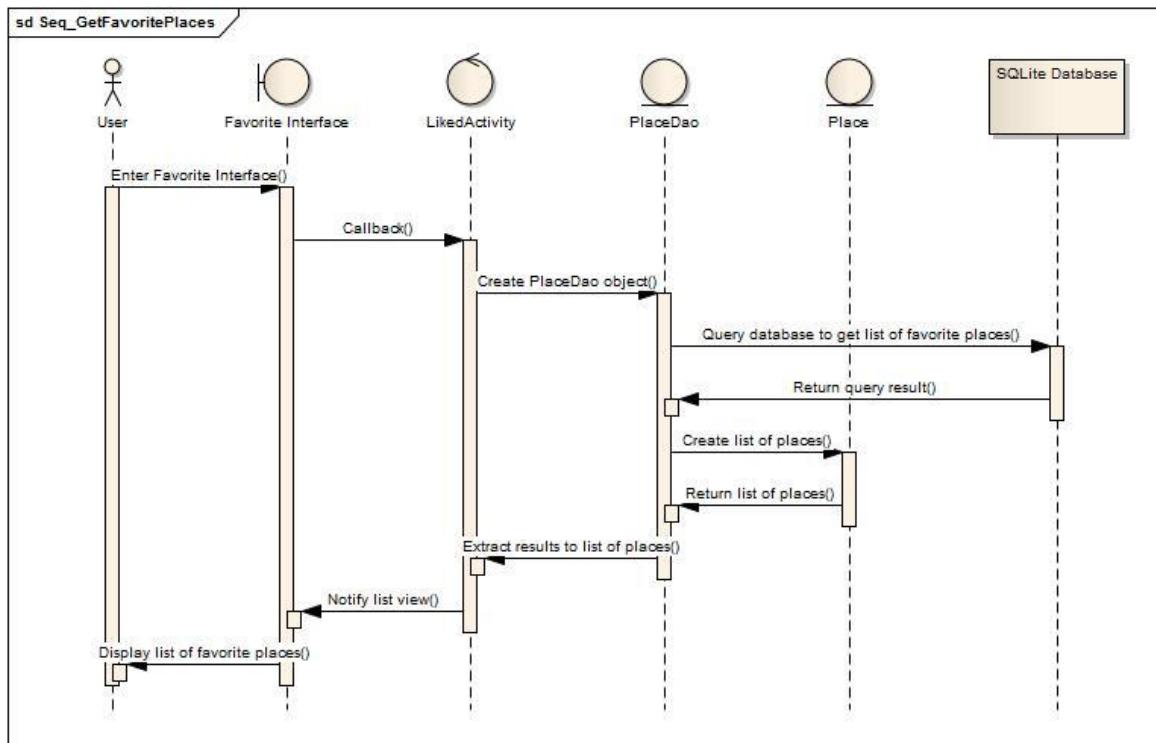
Hình ảnh 18: Biểu đồ tuần tự chức năng đăng nhập

## b) Chức năng đăng ký



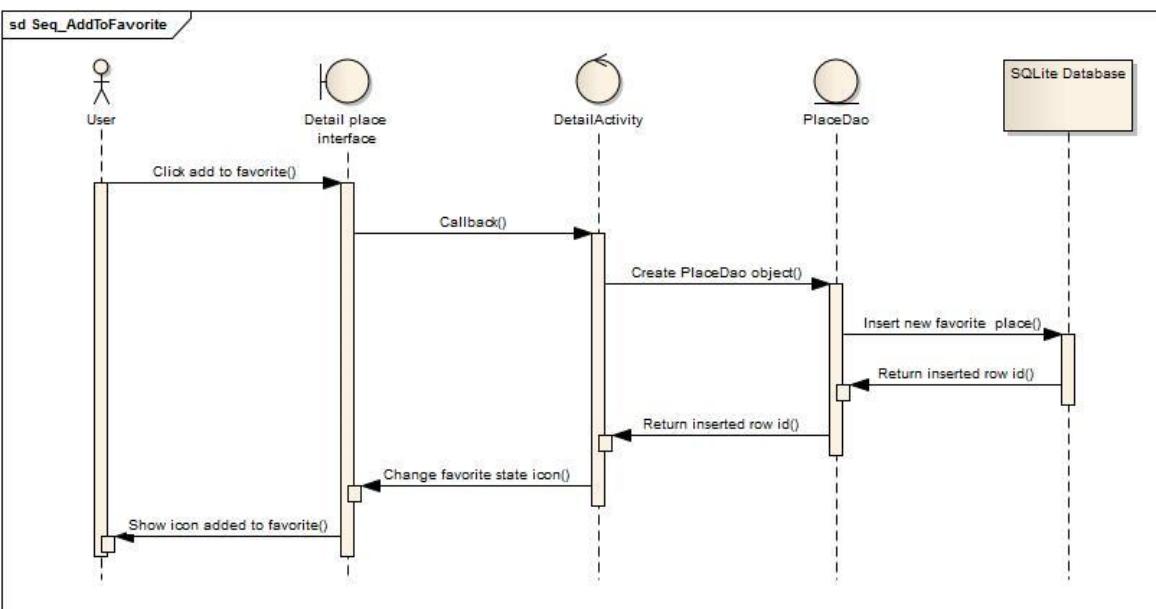
Hình ảnh 19: Biểu đồ tuần tự chức năng đăng ký

## c) Chức năng hiển thị danh sách yêu thích



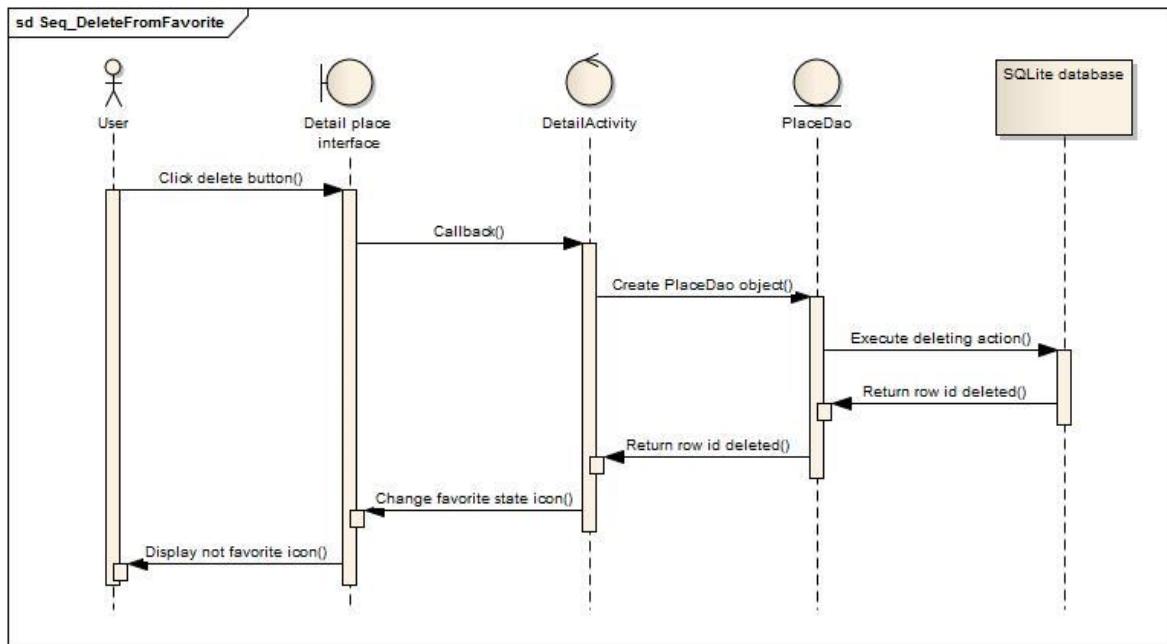
Hình ảnh 20: Biểu đồ tuần tự chức năng đưa ra danh sách yêu thích

## d) Chức năng thêm vào danh sách yêu thích



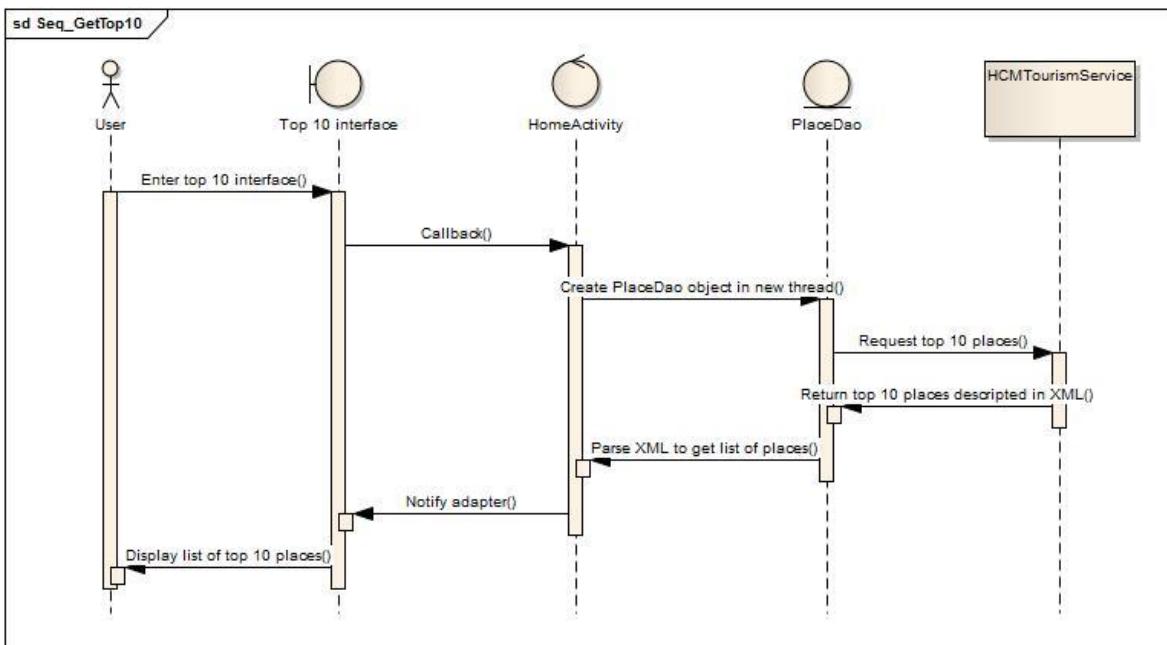
Hình ảnh 21: Biểu đồ tuần tự chức năng thêm vào danh sách yêu thích

## e) Chức năng xóa khỏi danh sách yêu thích



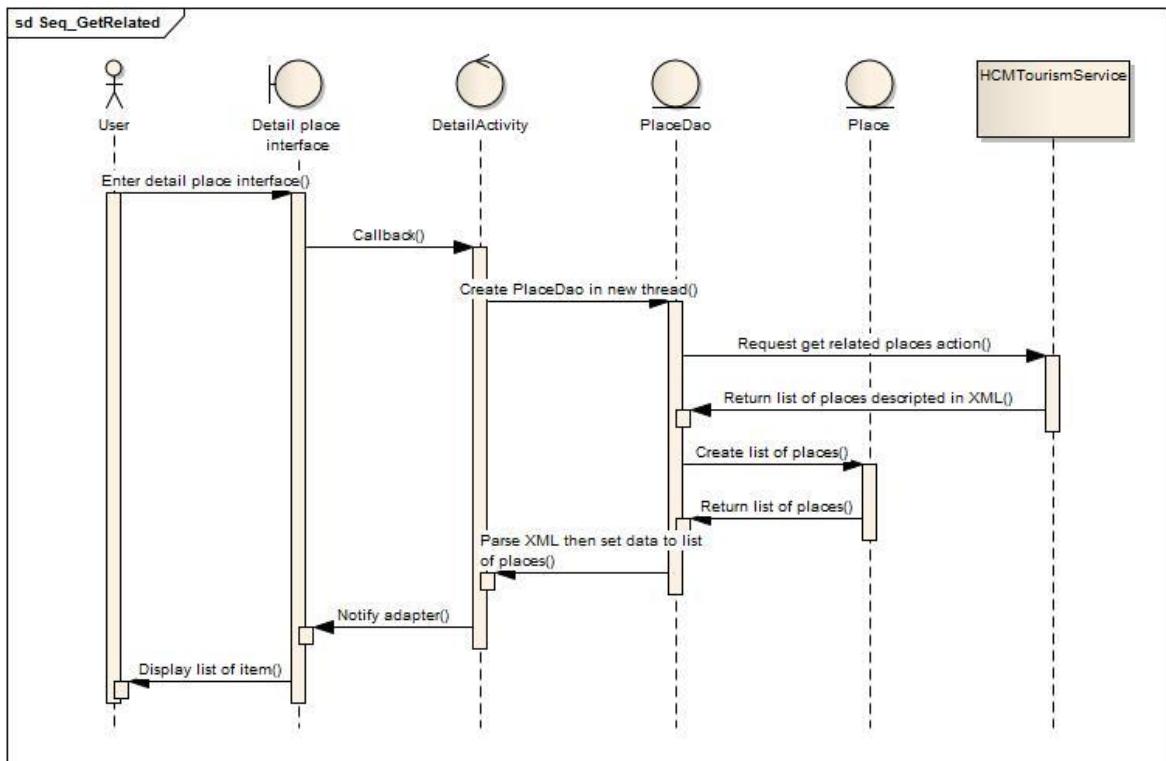
Hình ảnh 22: Biểu đồ tuần tự chức năng xóa khỏi danh sách yêu thích

## f) Chức năng hiển thị top 10 gợi ý



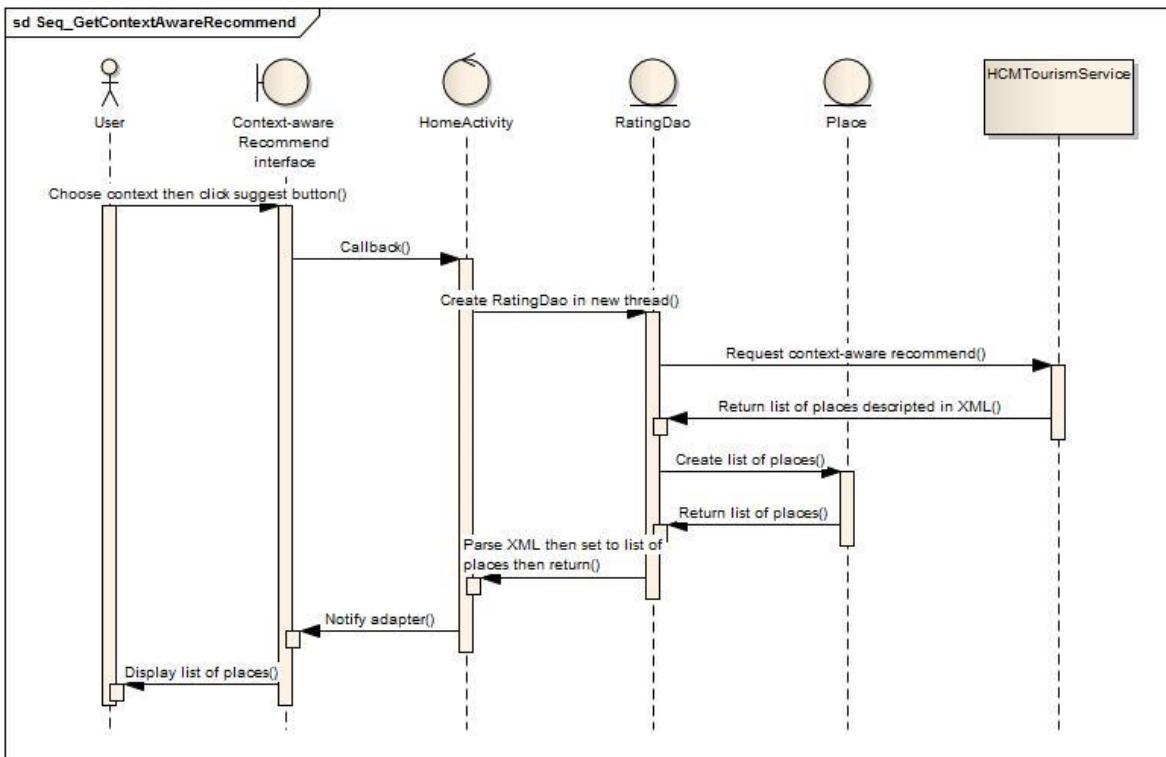
Hình ảnh 23: Biểu đồ tuần tự chức năng đưa ra top 10 gợi ý

## g) Chức năng hiển thị danh sách địa điểm liên quan

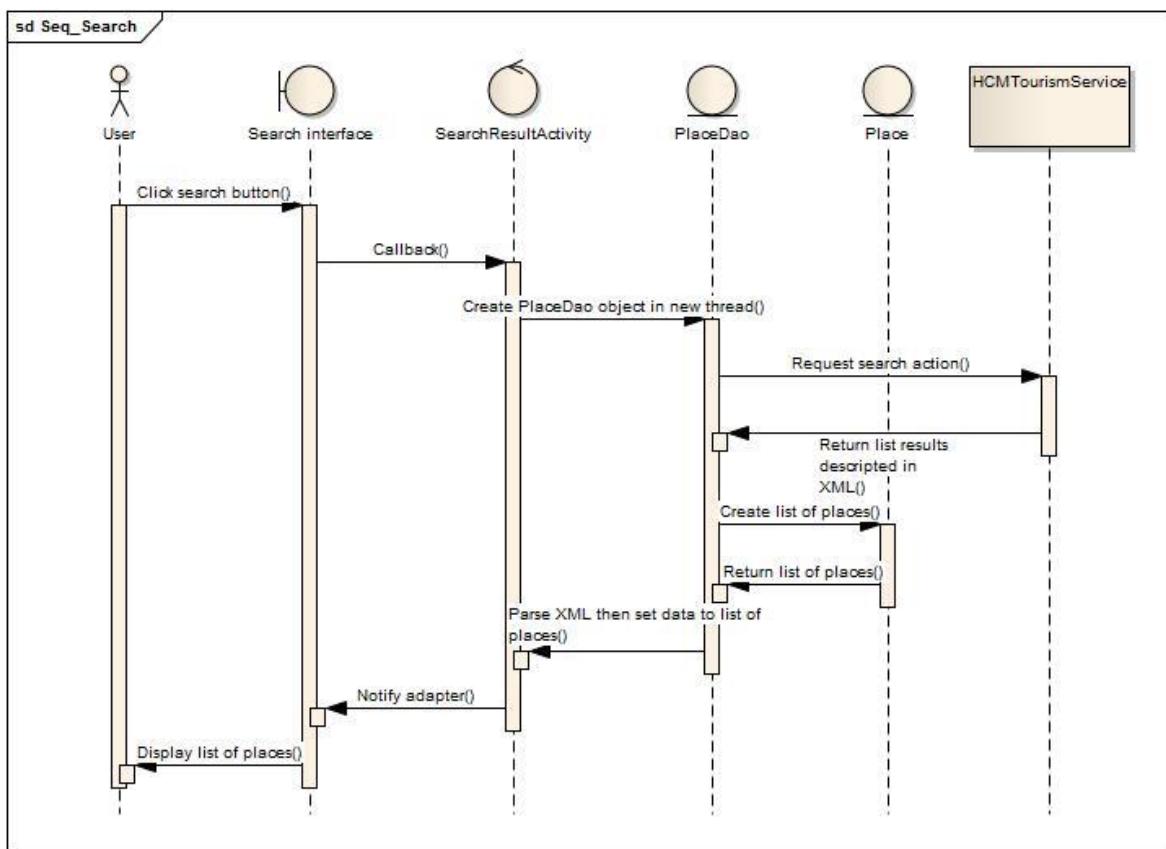


Hình ảnh 24: Biểu đồ tuần tự chức năng gọi ý địa điểm liên quan

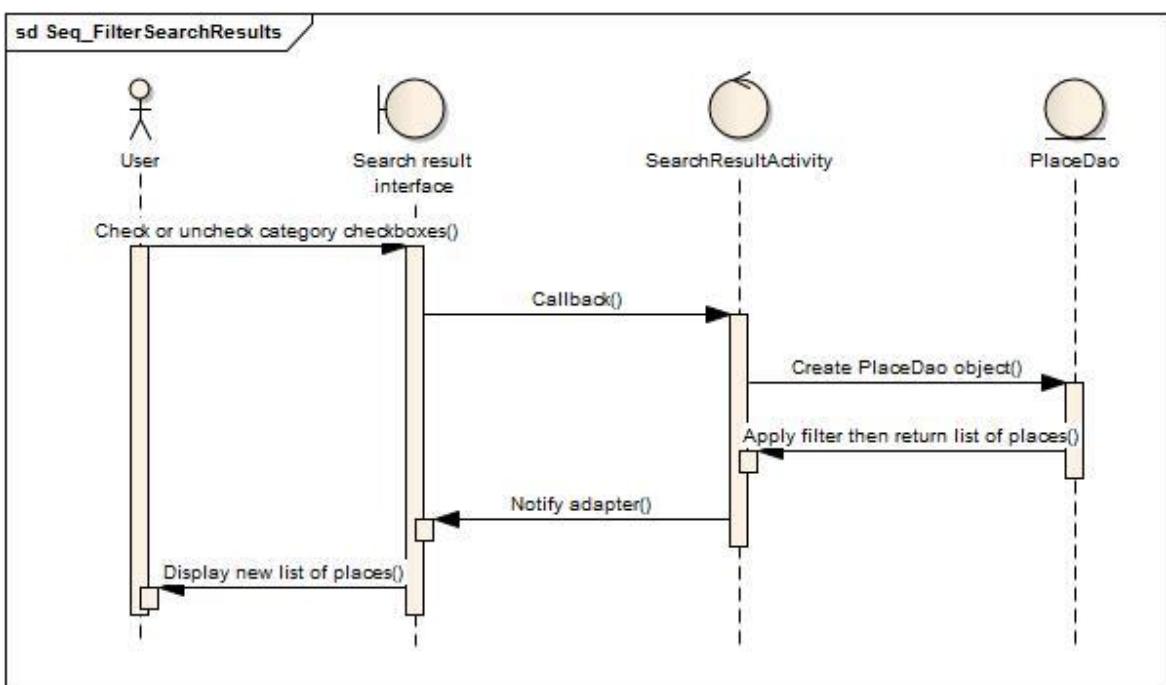
## h) Chức năng hiển thị gợi ý dựa vào ngữ cảnh



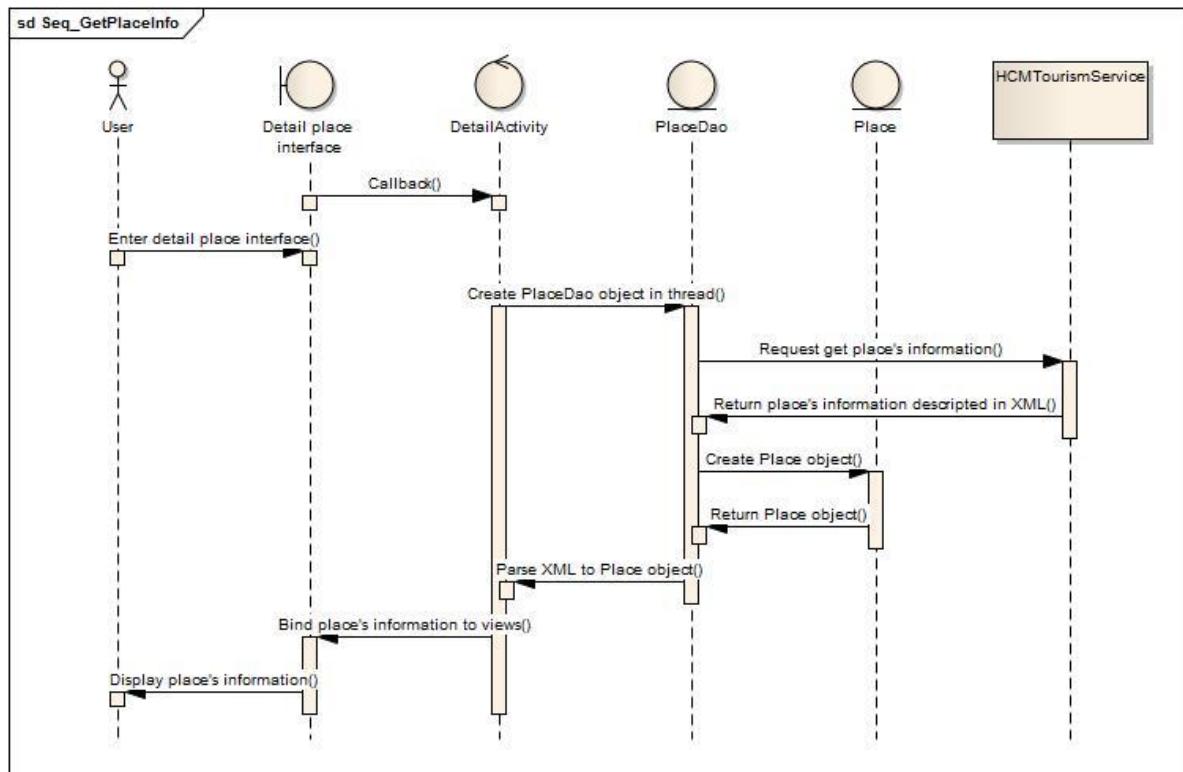
Hình ảnh 25: Biểu đồ tuần tự chức năng gợi ý dựa trên ngữ cảnh

i) *Chức năng tìm kiếm*

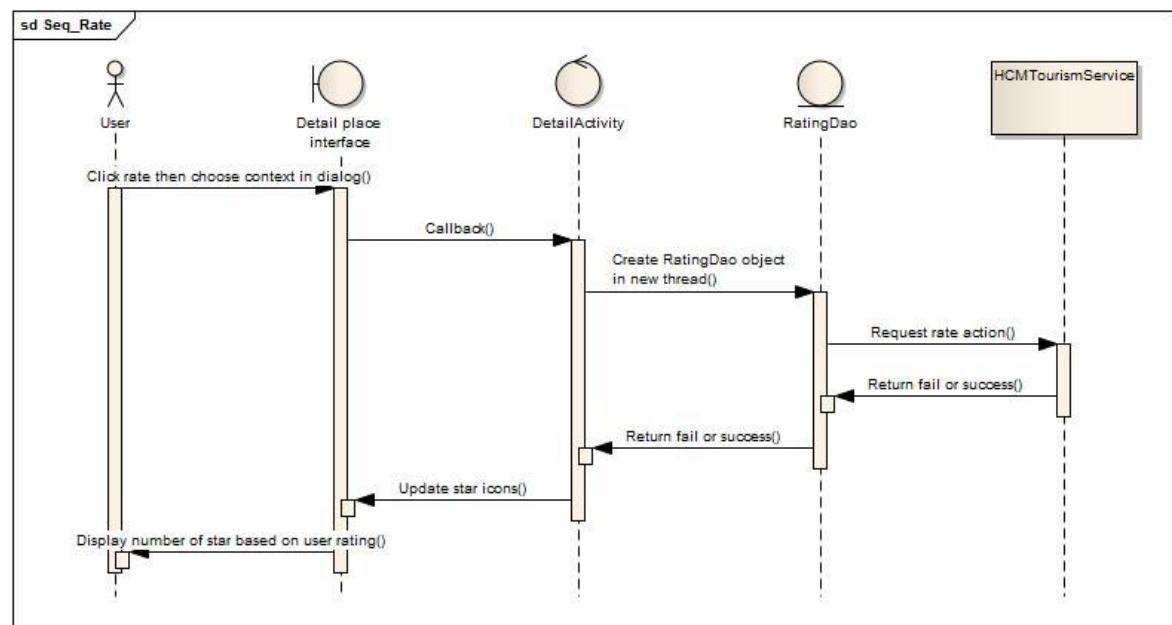
Hình ảnh 26: Biểu đồ tuần tự chức năng tìm kiếm

j) *Chức năng lọc kết quả tìm kiếm*

Hình ảnh 27: Biểu đồ tuần tự chức năng lọc kết quả tìm kiếm

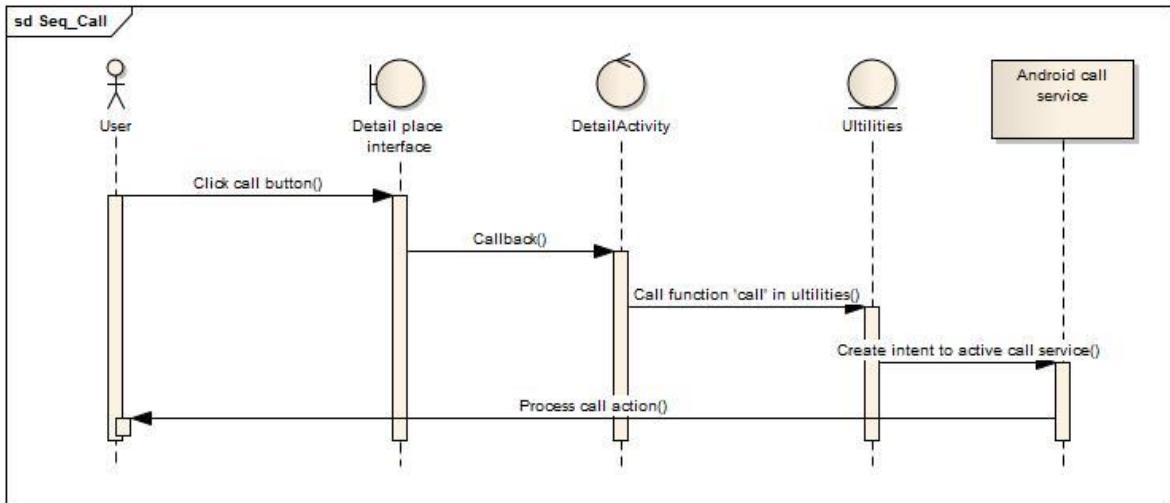
k) *Chức năng hiển thị thông tin chi tiết một địa điểm du lịch*

Hình ảnh 28: Biểu đồ tuần tự chức năng hiển thị thông tin chi tiết địa điểm

l) *Chức năng đánh giá địa điểm*

Hình ảnh 29: Biểu đồ tuần tự chức năng đánh giá địa điểm

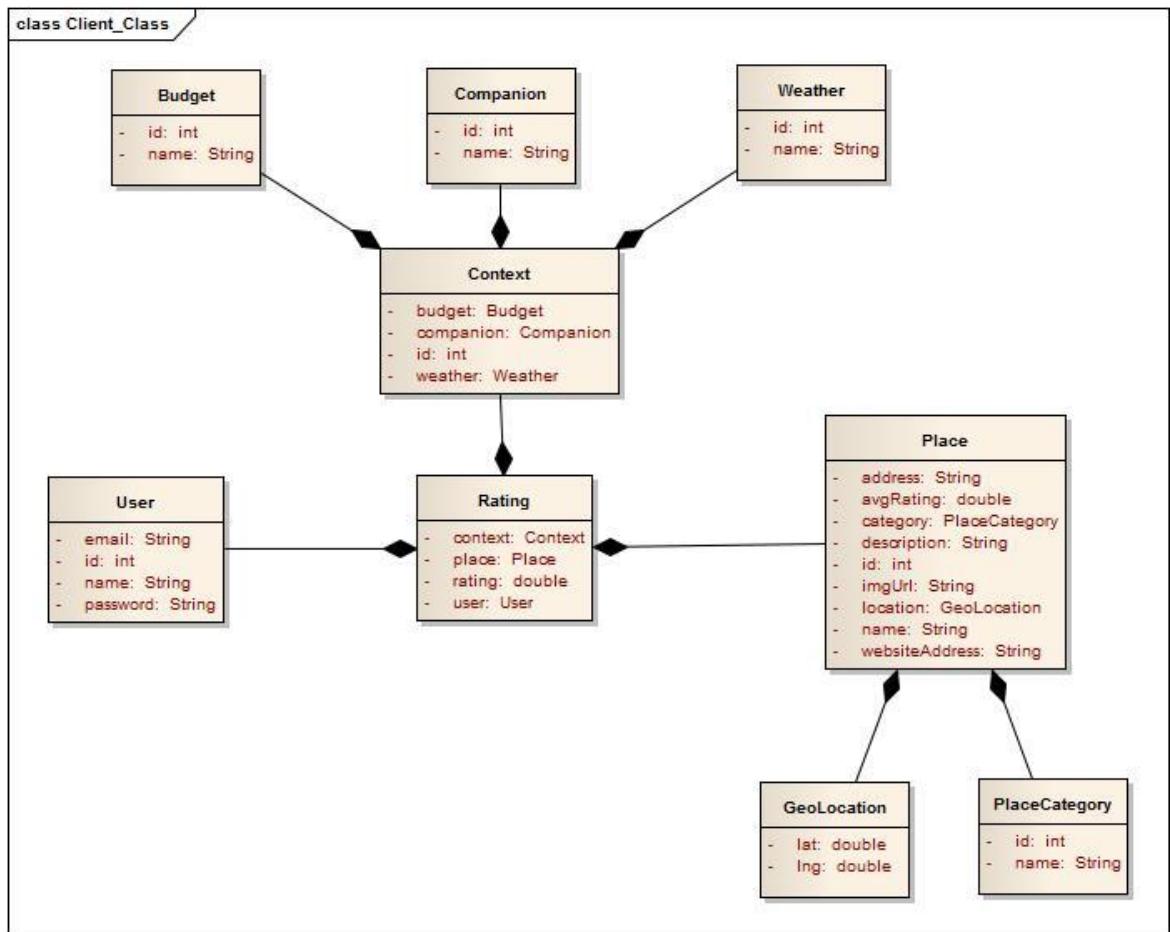
m) *Chức năng liên hệ quản lý địa điểm du lịch*



Hình ảnh 30: Biểu đồ tuần tự chức năng liên hệ quản lý địa điểm

#### 3.4.4.1.4. Biểu đồ lớp

Biểu đồ lớp thể hiện mối quan hệ giữa các lớp đối tượng trong ứng dụng:



Hình ảnh 31: Biểu đồ lớp phía client

### 3.4.4.2. Phía server

#### 3.4.4.2.1. Biểu đồ use case

Các tác nhân ảnh hưởng tới hệ thống phía server là:

STT	Tác nhân	Mô tả
1	Client	Là những thiết bị Android được điều khiển bởi người dùng.
2	Admin	Là người sử dụng ứng dụng phía server, điều khiển việc huấn luyện dữ liệu, đưa ra những tư vấn rồi lưu lại vào cơ sở dữ liệu

Bảng 11: Danh sách các tác nhân phía server

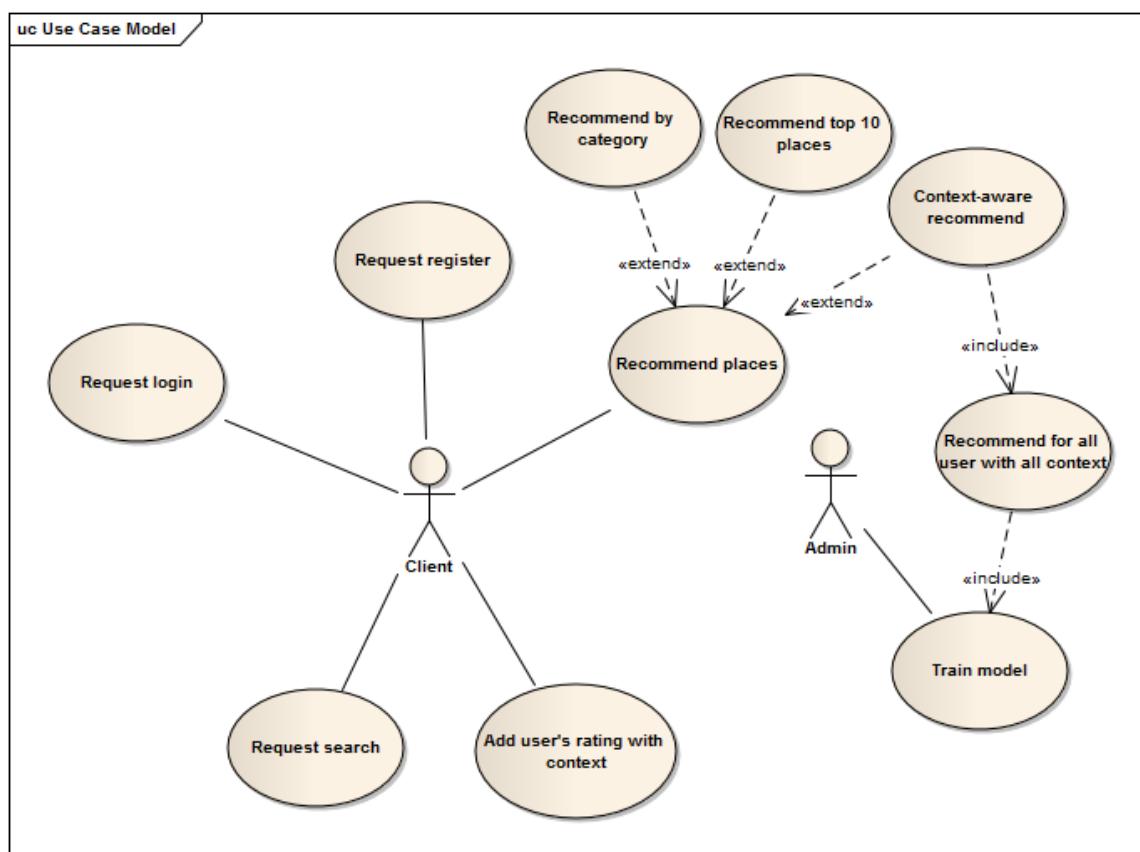
Bảng dưới đây sẽ mô tả những ca sử dụng phía server mà hai tác nhân *Client* và *Admin* có thể tương tác với hệ thống:

STT	Ca sử dụng	Mô tả ca sử dụng
1	Yêu cầu đăng ký (Request register)	Client gửi yêu cầu đăng ký tới server. Server kiểm tra, lưu trữ và trả kết quả đăng ký về cho client.
2	Yêu cầu đăng nhập (Request login)	Client gửi yêu cầu đăng nhập tới server. Server xác nhận thông tin và trả về kết quả cho client.
3	Yêu cầu tìm kiếm (Request search)	Client gửi yêu cầu tìm kiếm địa điểm, server sẽ trả về danh sách địa điểm phù hợp cho client.
4	Thêm đánh giá có ngữ cảnh (Add rating with context)	Client yêu cầu thêm đánh giá, server sẽ lưu lại đánh giá của người dùng.
5	Yêu cầu gợi ý địa điểm cùng danh mục (Recommend by category)	Client yêu cầu danh sách địa điểm cùng danh mục.
6	Yêu cầu gợi ý top 10 địa điểm đánh giá trung bình cao nhất (Recommend top 10 places)	Client yêu cầu danh sách 10 địa điểm có đánh giá trung bình cao nhất.
7	Yêu cầu gợi ý dựa trên ngữ cảnh (Context-aware recommend)	Client yêu cầu danh sách địa điểm gợi ý dựa trên ngữ cảnh.
8	Huấn luyện mô hình dữ liệu (Train model)	Admin điều khiển ứng dụng phía server để lấy dữ liệu và huấn luyện tìm ra tham số mô hình phù hợp, phục vụ cho mục đích tư vấn địa điểm.

9	Đưa ra tư vấn cho tất cả người dùng trong mọi ngữ cảnh có thể (Recommend for all user with all context)	Admin dựa vào tham số mô hình đã huấn luyện để đưa ra tư vấn cho tất cả người dùng trong mọi ngữ cảnh có thể xảy ra.
---	---	--

Bảng 12: Danh sách các ca sử dụng phía server

Biểu đồ use case phía server sẽ chỉ ra những chức năng của dịch vụ web REST cho hệ thống tư vấn địa điểm du lịch tại thành phố Hồ Chí Minh và chức năng đưa ra gợi ý của phần mềm tư vấn địa điểm dựa theo ngữ cảnh.



Hình ảnh 32: Biểu đồ use case phía server

Dịch vụ web REST của hệ thống có những chức năng sau:

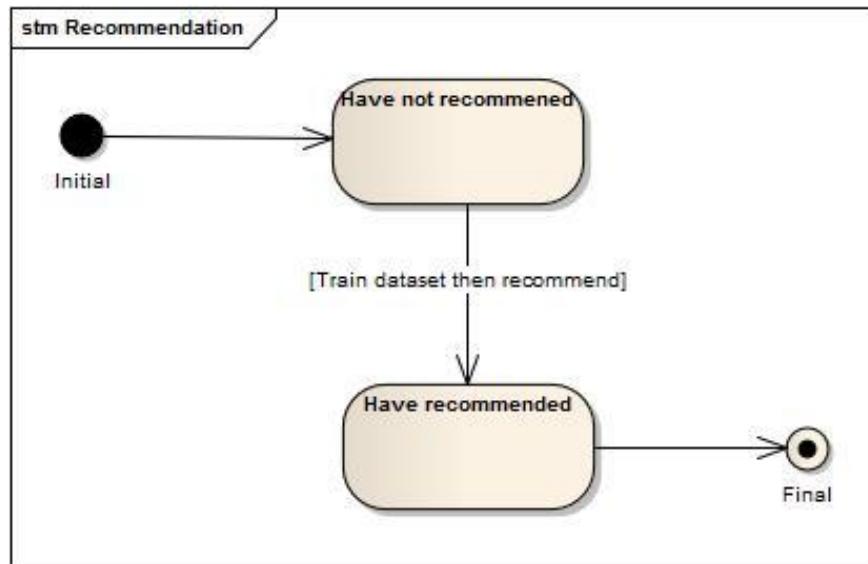
- Đáp ứng nhu cầu đăng nhập, đăng ký từ phía client.
- Trả về danh sách địa điểm từ yêu cầu tìm kiếm của phía client.
- Trả về danh sách gợi ý (bao gồm top 10 địa điểm có đánh giá trung bình cao nhất, danh sách cùng danh mục và gợi ý dựa vào ngữ cảnh).
- Thêm đánh giá có ngữ cảnh do client gửi lên vào cơ sở dữ liệu.

Phần mềm đưa ra gợi ý dựa trên ngữ cảnh sẽ huấn luyện dữ liệu đánh giá trước và đưa ra những kết quả gợi ý cho từng người dùng với từng ngữ cảnh. Sau đó kết quả gợi ý sẽ được lưu trữ trong cơ sở dữ liệu. Khi người dùng cần gợi ý thì chỉ cần truy vấn cơ sở dữ liệu. Phần mềm đưa ra gợi ý dựa trên ngữ cảnh bao gồm những chức năng sau:

- Huấn luyện dữ liệu.
- Đưa ra gợi ý dựa vào ngữ cảnh.

#### **3.4.4.2.2. Biểu đồ trạng thái**

Ở phía server, đối tượng có trạng thái thay đổi chỉ có đối tượng Recommendation.



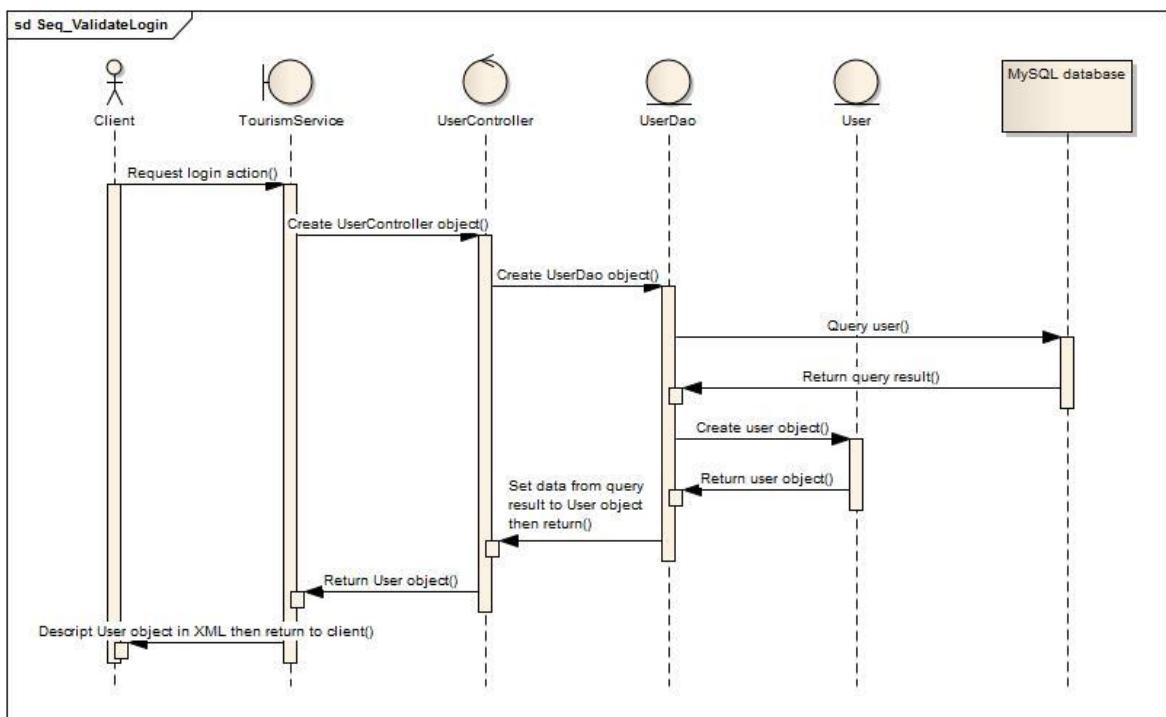
Hình ảnh 33: Biểu đồ trạng thái của đối tượng Recommendation

Khi client gửi lên yêu cầu cần đưa ra gợi ý dựa vào ngữ cảnh cho một người dùng chưa được huấn luyện và đưa ra gợi ý trước, server cần tải lại mô hình và gợi ý cho người dùng.

#### **3.4.4.2.3. Biểu đồ tuần tự**

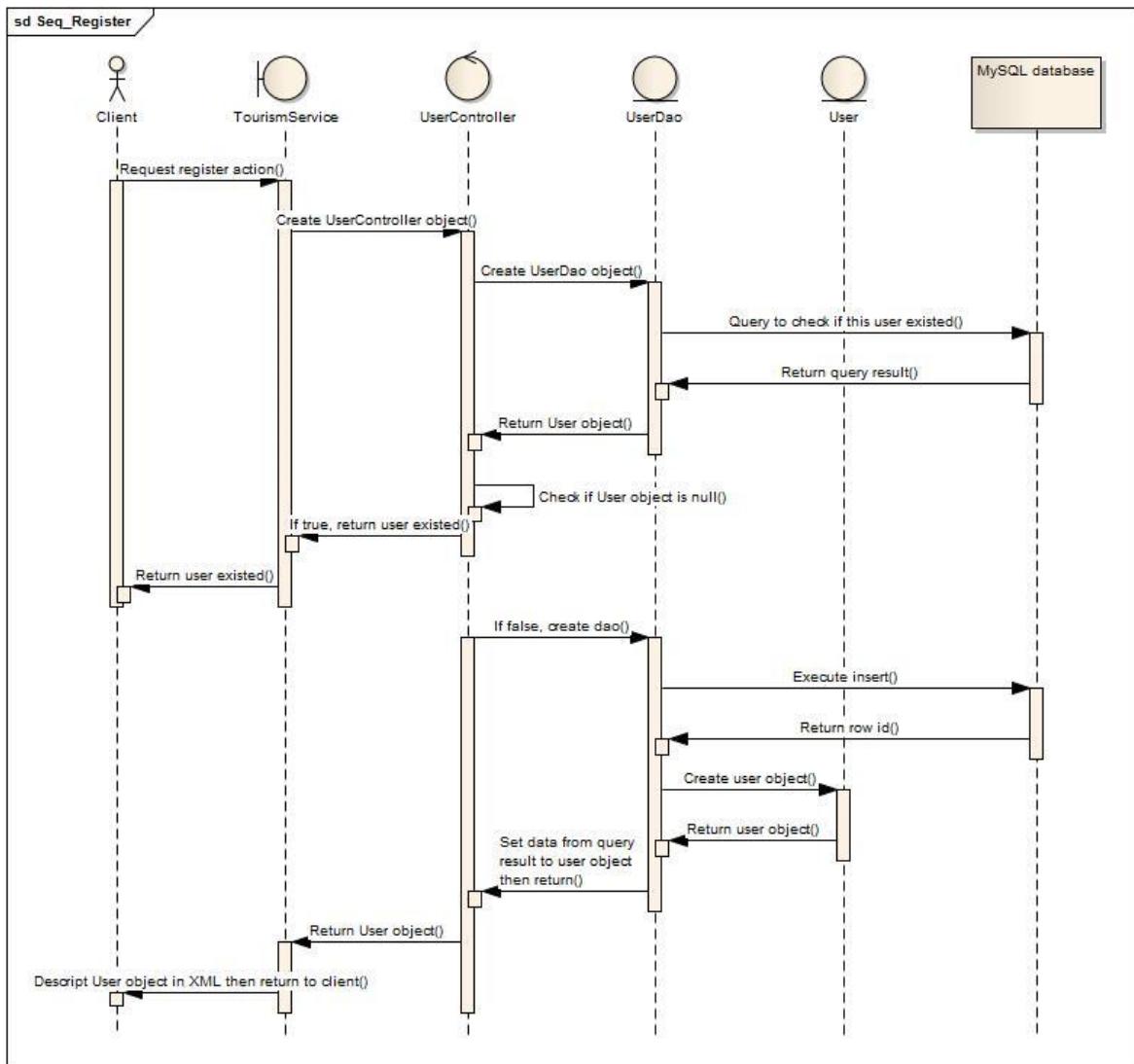
Biểu đồ tuần tự phía server thể hiện luồng thực hiện các chức năng phía server, bao gồm cả những dịch vụ REST và phần mềm gợi ý dựa trên ngữ cảnh.

## a) Chức năng xác nhận đăng nhập



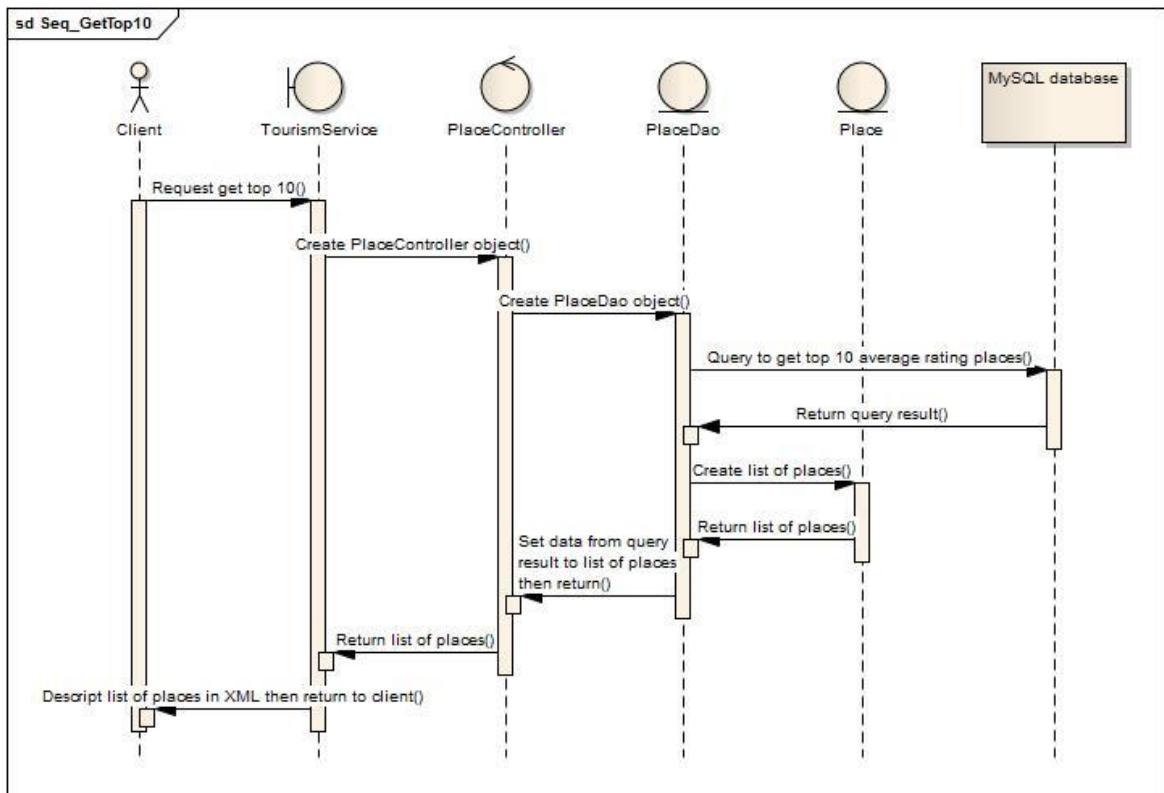
Hình ảnh 34: Biểu đồ tuần tự chức năng đăng nhập

## b) Chức năng đăng ký



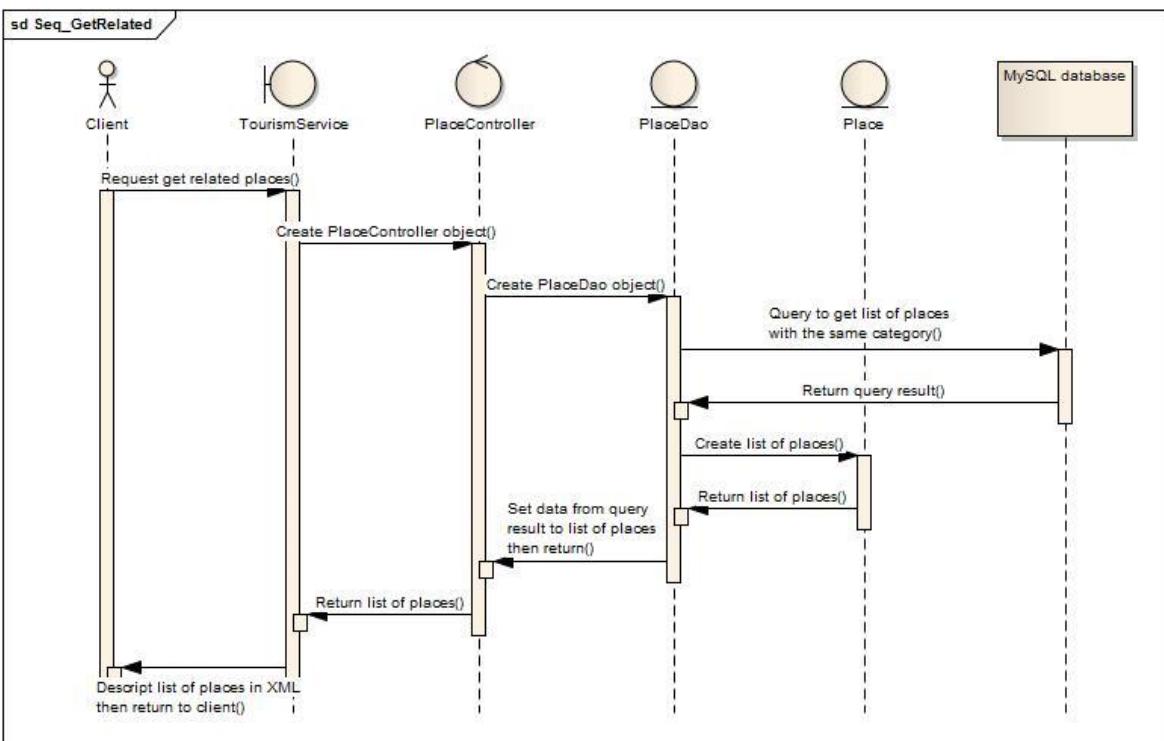
Hình ảnh 35: Biểu đồ tuần tự chức năng đăng ký

c) Chức năng lấy ra top 10 địa điểm có đánh giá trung bình cao nhất



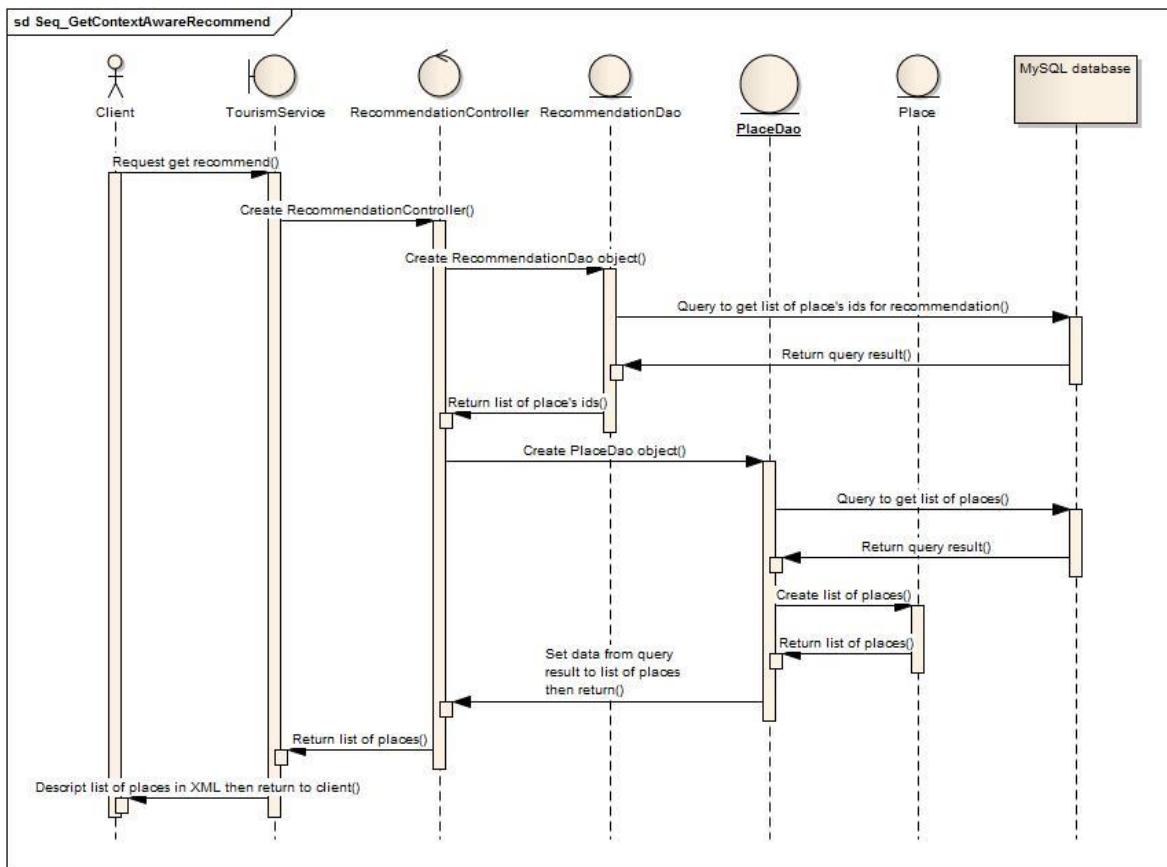
Hình ảnh 36: Biểu đồ tuần tự chức năng gọi ý top 10

d) Chức năng lấy ra địa điểm liên quan

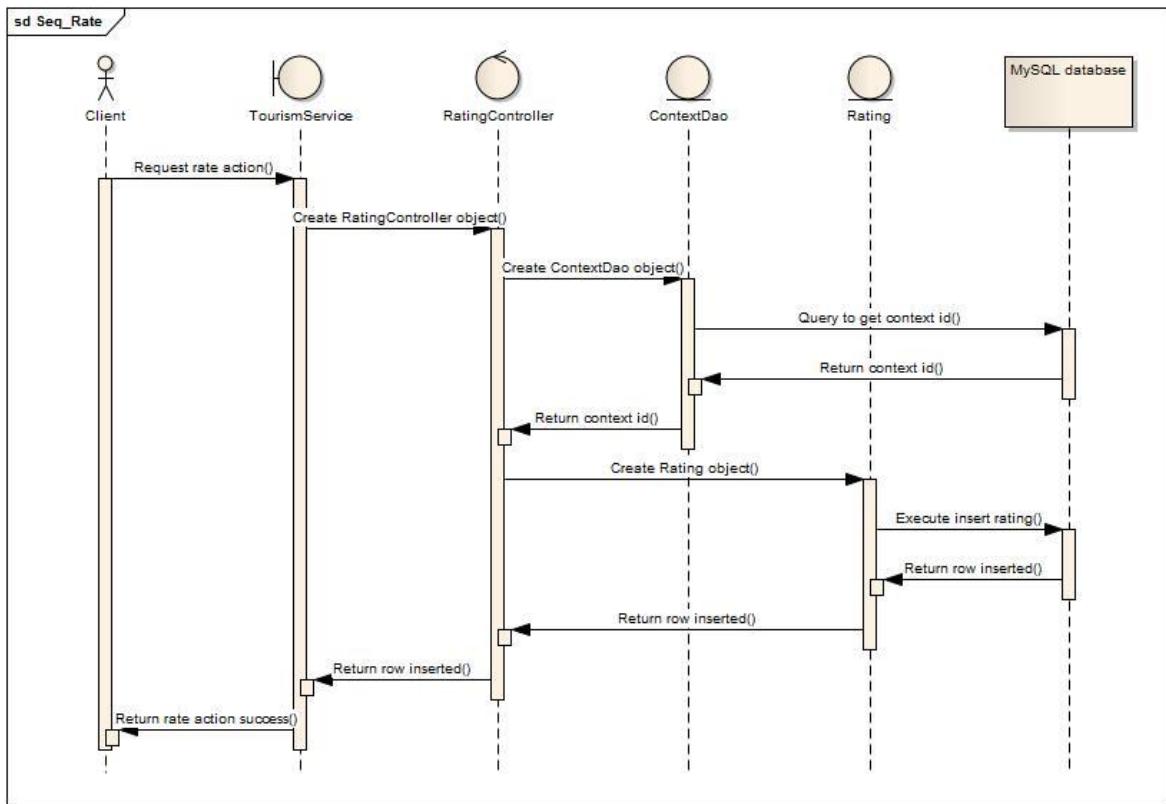


Hình ảnh 37: Biểu đồ tuần tự chức năng địa điểm liên quan

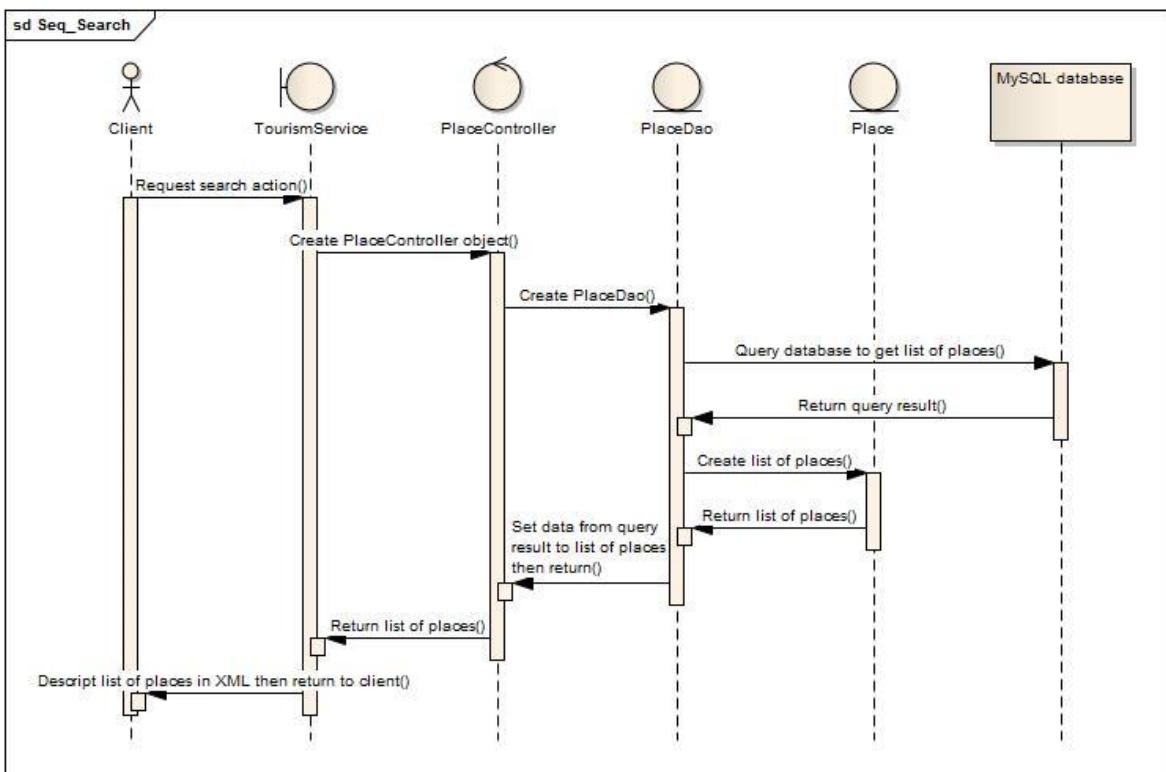
e) Chức năng lấy ra gợi ý dựa vào ngữ cảnh



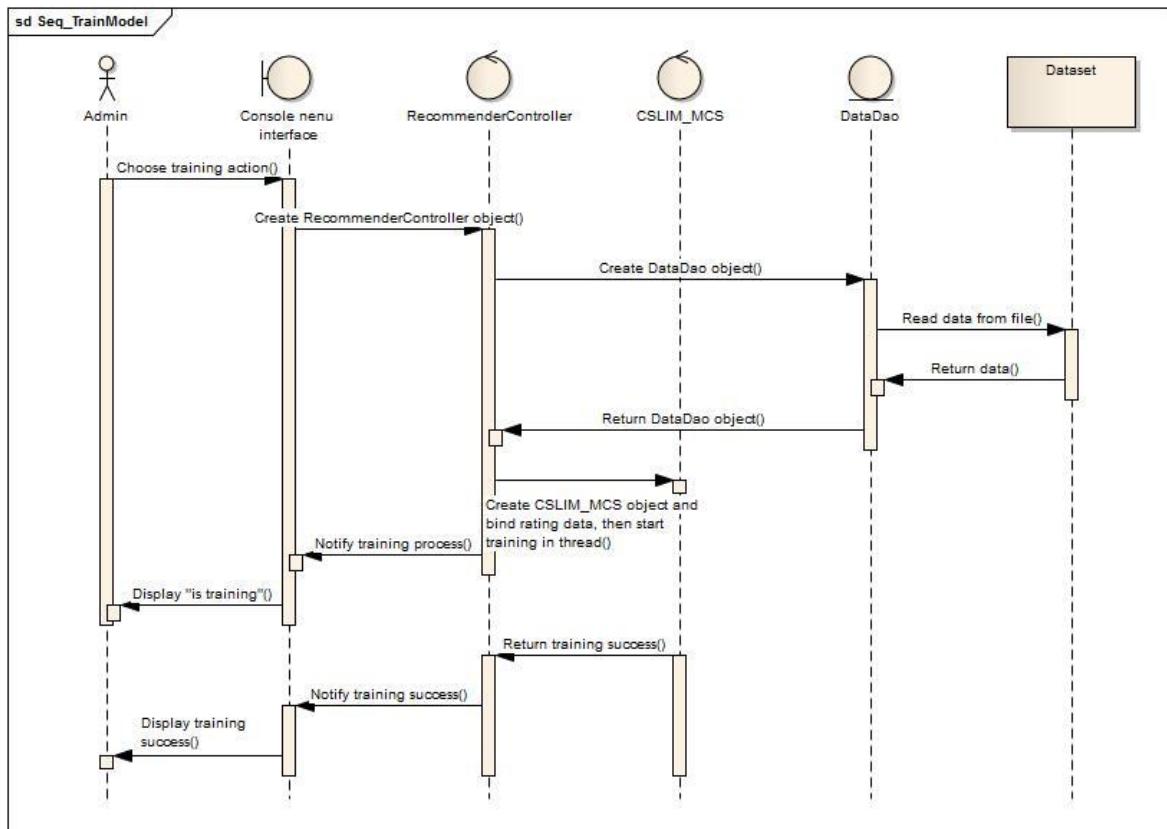
Hình ảnh 38: Biểu đồ tuần tự chức năng gợi ý dựa vào ngữ cảnh

f) *Chức năng thêm đánh giá từ phía client*

Hình ảnh 39: Biểu đồ tuần tự chức năng đánh giá địa điểm du lịch

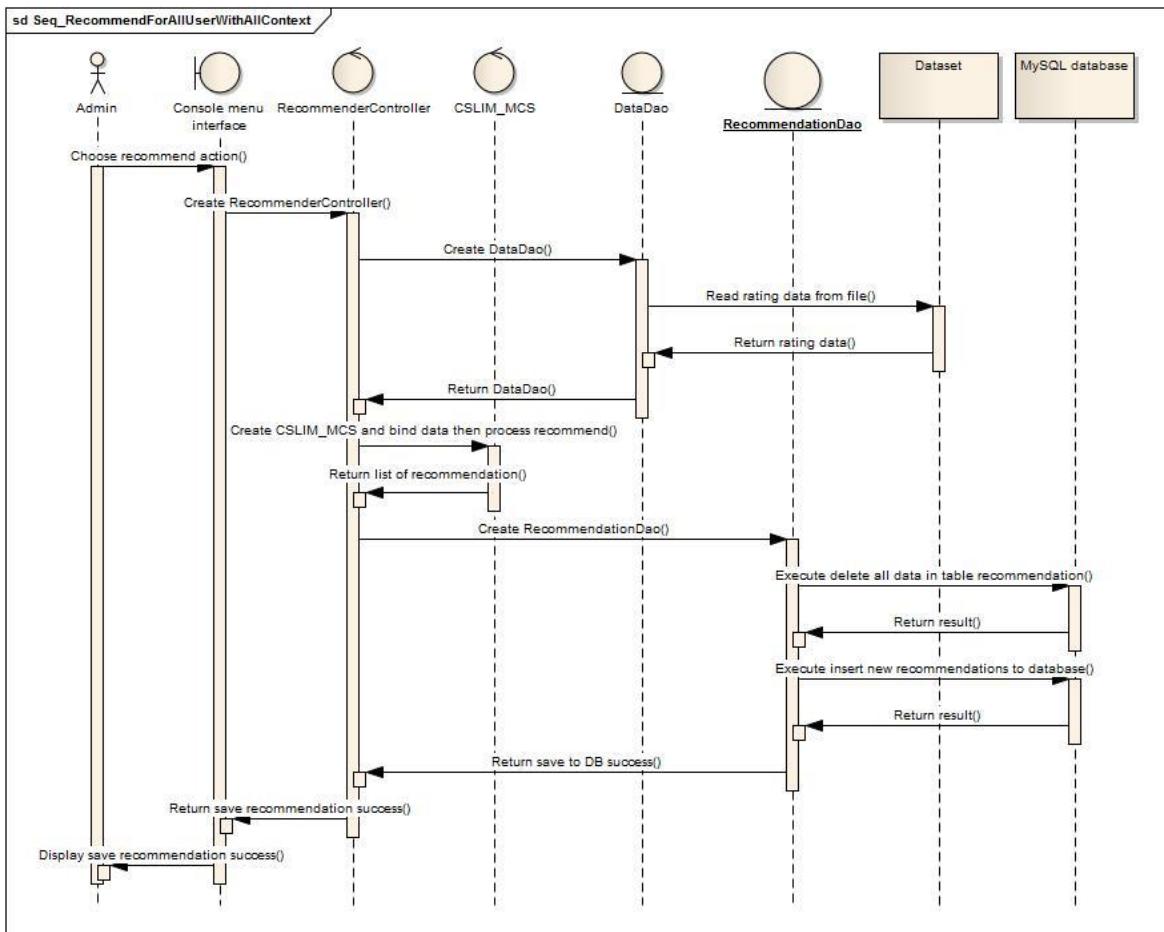
g) *Chức năng tìm kiếm*

Hình ảnh 40: Biểu đồ tuần tự chức năng tìm kiếm

h) *Chức năng huấn luyện dữ liệu*

Hình ảnh 41: Biểu đồ tuần tự chức năng huấn luyện dữ liệu

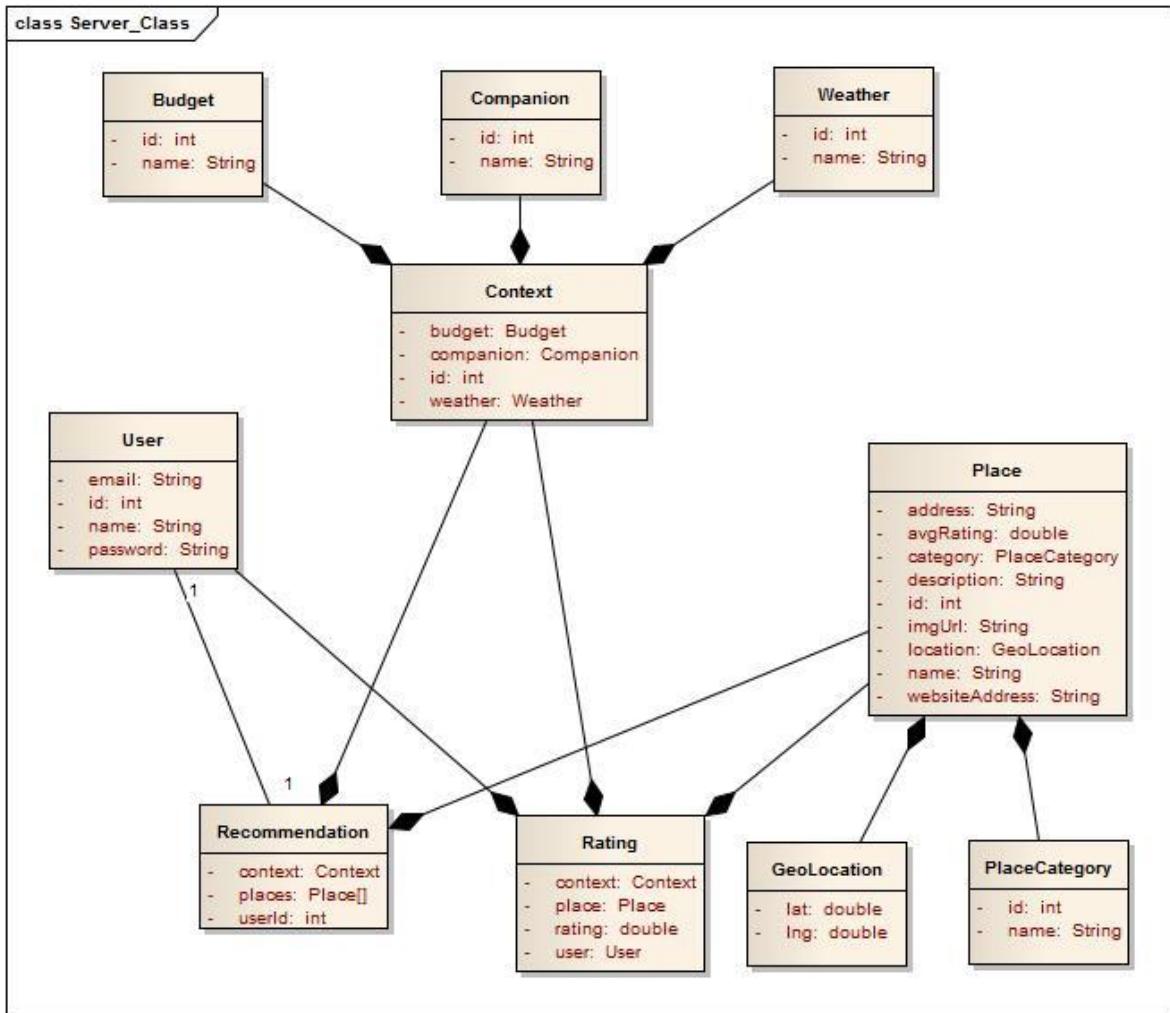
i) *Chức năng đưa ra gợi ý dựa trên ngữ cảnh*



Hình ảnh 42: Biểu đồ tuần tự chức năng đưa ra gợi ý và lưu vào cơ sở dữ liệu

#### 3.4.4.2.4. Biểu đồ lớp

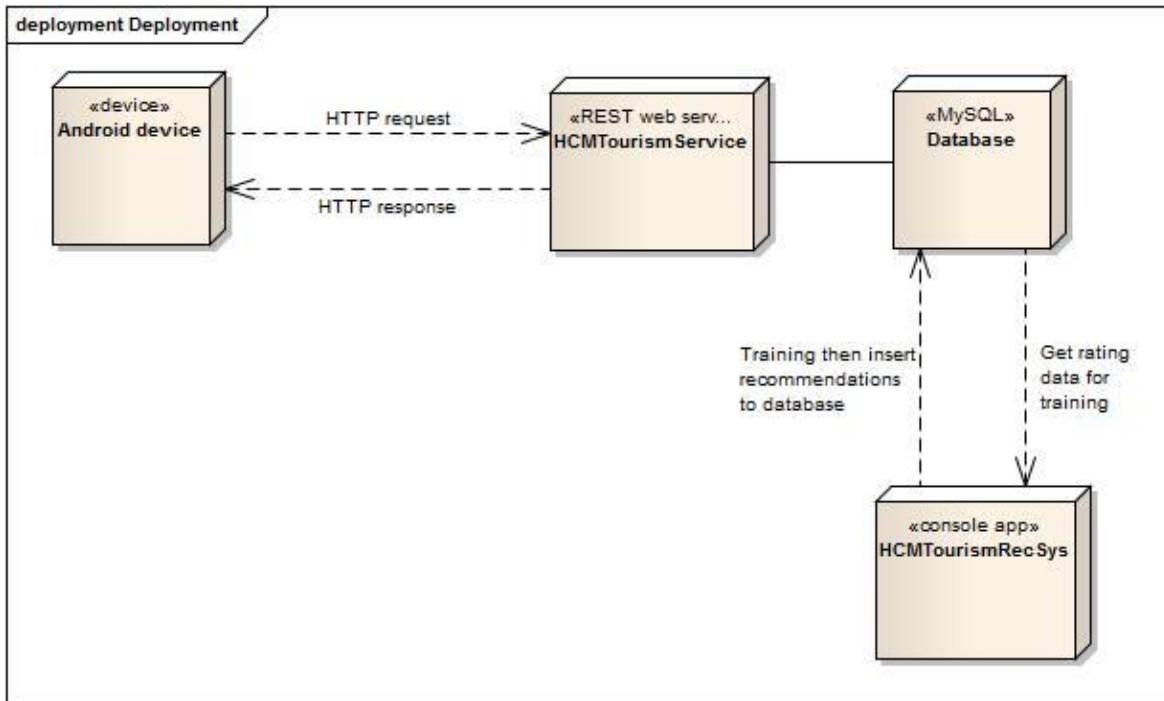
Biểu đồ lớp phía server cũng gần tương tự phía client. Tuy nhiên có thêm lớp Recommendation để tiện cho việc sau khi đưa ra gợi ý sẽ đẩy kết quả vào cơ sở dữ liệu.



Hình ảnh 43: Biểu đồ lớp phía server

#### 3.4.4.3. Biểu đồ triển khai hệ thống

Biểu đồ triển khai hệ thống thể hiện mối liên kết giữa thành phần liên quan trong hệ thống client – server. Dưới đây là biểu đồ triển khai hệ thống của ứng dụng tư vấn địa điểm du lịch tại thành phố Hồ Chí Minh.



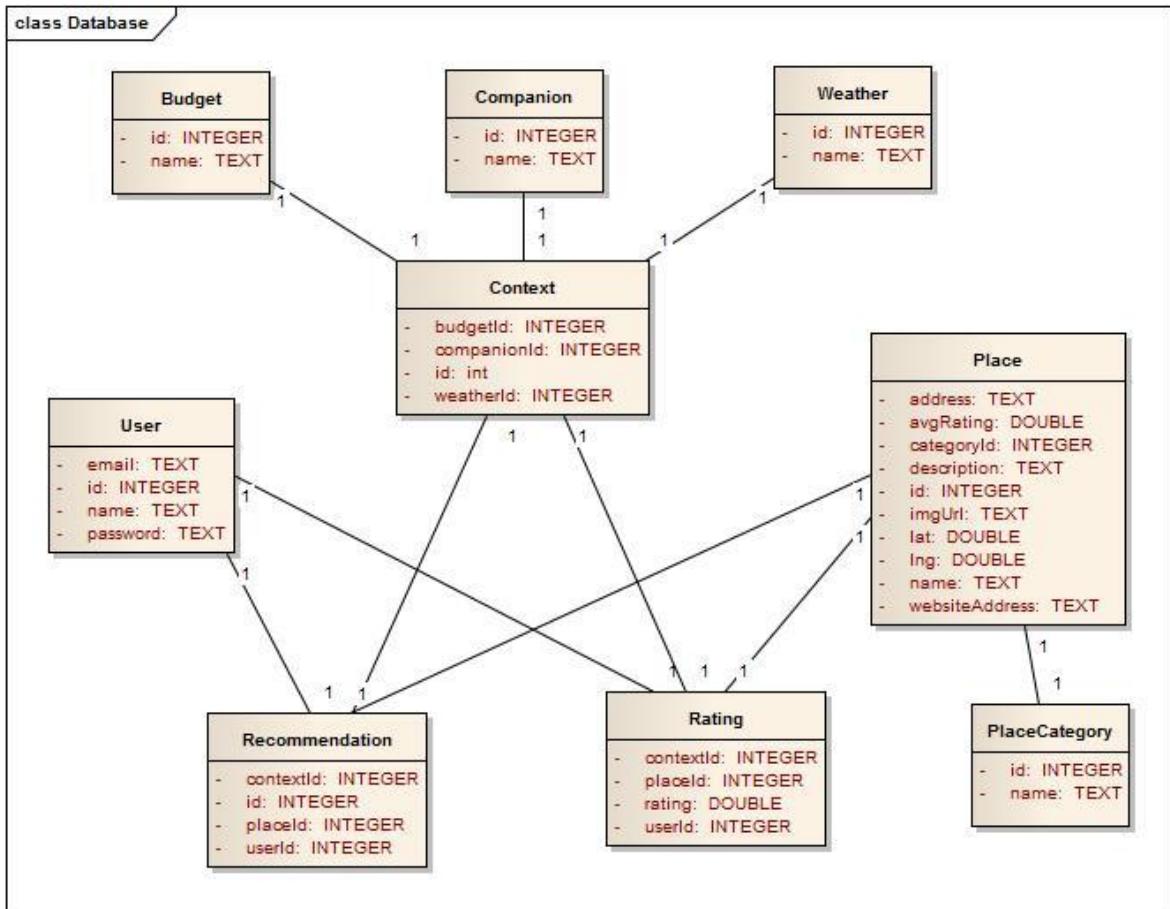
Hình ảnh 44: Biểu đồ triển khai hệ thống

Hệ thống ứng dụng này bao gồm 4 khối:

- *Android device*: Là những thiết bị đầu cuối sử dụng ứng dụng, đóng vai trò là client gửi yêu cầu và nhận phản hồi của dịch vụ web REST để hiển thị thông tin.
- *HCMTourismService*: Là dịch vụ web REST, đóng vai trò trung gian, nhận yêu cầu, truy vấn cơ sở dữ liệu và gửi trả phản hồi kết quả về client.
- *Database*: Là cơ sở dữ liệu, lưu trữ thông tin về người dùng, địa điểm và đánh giá của người dùng. Cơ sở dữ liệu này còn lưu danh sách những địa điểm sẽ gợi ý cho người dùng trong những điều kiện ngữ cảnh cụ thể.
- *HCMTourismRecSys*: Là một ứng dụng console do người quản trị điều khiển, thực hiện lấy thông tin đánh giá của người dùng từ cơ sở dữ liệu rồi huấn luyện và đưa ra những gợi ý dựa trên ngữ cảnh để lưu vào cơ sở dữ liệu.

#### 3.4.4.4. Biểu đồ cơ sở dữ liệu quan hệ

Biểu đồ cơ sở dữ liệu quan hệ thể hiện thiết kế hệ thống các bảng trong cơ sở dữ liệu. Dưới đây là biểu đồ cơ sở dữ liệu quan hệ của hệ thống tư vấn địa điểm du lịch tại thành phố Hồ Chí Minh.



Hình ảnh 45: Biểu đồ cơ sở dữ liệu quan hệ

Cơ sở dữ liệu của hệ thống này bao gồm những bảng sau:

- **Budget**, **Companion**, **Weather**: là ba bảng lưu trữ ba trường ngữ cảnh lần lượt là ngân sách, bạn đồng hành và thời tiết.
- **Context**: Là bản lưu trữ hợp của những điều kiện ngữ cảnh của ngân sách, bạn đồng hành và thời tiết.
- **User**: Lưu trữ thông tin về người dùng.
- **Place**: Lưu trữ thông tin về địa điểm du lịch.
- **PlaceCategory**: Lưu trữ thông tin về danh mục địa điểm du lịch.
- **Rating**: Lưu trữ đánh giá của người dùng trong một ngữ cảnh nào đó.
- **Recommendation**: Lưu trữ những gợi ý về địa điểm cho từng người dùng trong từng ngữ cảnh cụ thể.

## KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 1. Kết luận

Trong đồ án này, tác giả đã nghiên cứu tổng quan về hệ tư vấn truyền thống và những kỹ thuật kết hợp yếu tố ngữ cảnh, đặc biệt nghiên cứu chi tiết về thuật toán phương pháp tuyến tính thừa (SLIM) và kỹ thuật mô hình hóa ngữ cảnh dựa trên độ tương quan ngữ cảnh đa chiều (Multidimensional Context Similarity), để hình thành thuật toán tư vấn kết hợp ngữ cảnh CSLIM\_MCS.

Tác giả cũng đã thiết kế và xây dựng ứng dụng “Đi đâu bây giờ ?” cùng với hệ thống gợi ý địa điểm du lịch tại thành phố Hồ Chí Minh sử dụng thuật toán CSLIM\_MCS. Tuy nhiên hệ thống vẫn còn một số nhược điểm sau:

- *Bộ dữ liệu nghèo nàn*: Dữ liệu phục vụ cho quá trình huấn luyện rất ít, dẫn đến kết quả gợi ý không đa dạng.
- *Gợi ý “tĩnh”*: Việc huấn luyện và đưa ra những gợi ý hoàn toàn phải thực hiện do quản trị viên và đưa toàn bộ những gợi ý cho từng người dùng trong từng ngữ cảnh vào cơ sở dữ liệu. Điều này có thể khiến client xử lý nhanh nhưng sẽ khiến cho hệ thống không linh động và nếu số lượng người dùng và điều ngữ cảnh lớn hơn thì sẽ dẫn đến quá tải về mặt lưu trữ trong cơ sở dữ liệu.
- *Ứng dụng client đơn giản*: Do giới hạn về mặt thời gian nên ứng dụng phía client mới chỉ xây dựng rất đơn giản, ít chức năng.

### 2. Phương hướng phát triển

Hệ thống đã được xây dựng còn nhiều vấn đề như đã trình bày phía trên. Trong thời gian tiếp theo, tác giả có thể phát triển hệ thống ứng dụng như sau:

- *Thu thập thêm dữ liệu*: Tiến hành khảo sát, thu thập thêm dữ liệu để hiệu quả của thuật toán gợi ý được tốt hơn, đa dạng hơn.
- *Mở rộng địa điểm du lịch*: Thêm nhiều địa danh du lịch khác, không chỉ gói gọn trong thành phố Hồ Chí Minh.
- *Chuyển đổi hệ thống học online*: Thay vì để quản trị viên điều khiển việc tính toán đưa ra gợi ý, hệ thống sẽ tự động cập nhật khi có người dùng đưa ra thêm đánh giá. Và cũng sẽ không lưu trước những gợi ý mà quá trình đó chỉ thực thi khi có yêu cầu từ phía người dùng (tách riêng quá trình huấn luyện và dự đoán).
- *Mở rộng chức năng*: Mở rộng thêm các chức năng cho ứng dụng phía client.

## TÀI LIỆU THAM KHẢO

- [1] Francesco Ricci, L.R., Bracha Shapira, Paul B. Kantor, *Recommender Systems Handbook*, 2011.
- [2] Gediminas Adomavicius, Alexander Tuzhilin, *Context-aware recommender systems*, 2008.
- [3] National Institute of Informatics, *An information-theoretic perspective of tf-idf measures*, 2002.
- [4] Rummel, R. J, *Understanding Correlation*, 1976.
- [5] D.Billsus, M.P., *Learning collaborative information filters (in International Conference on Machine Learning, Morgan Kaufmann Publishers)*, 1998.
- [6] M.O'Connor, J.H., *Clustering items for collaborative filtering (in Proceedings of the ACM SIGIR Workshop on Recommender Systems 1999)*, 1999.
- [7] Vidushi Sharma, Sachin Rai, Anurag Dev, *A Comprehensive Study of Artificial Neural Networks*, 2012.
- [8] Bottou, Léon, *Stochastic Learning (in Advanced Lectures on Machine Learning, LNAI, 3176, Springer)*, 2004.
- [9] S. GarcíaEmail authorA. FernándezJ. LuengoF. Herrera, *A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability*, 2009.
- [10] R.Greinemr, X.S., B.Shen, W.Zhou, *Structural extension to logistic regression: discriminative parameter learning of belief net classifiers (Machine Learning, vol.59, no.3, pp.297–322)*, 2005.
- [11] Xia Ning, George Karypis, *Sparse Linear Methods for Top-N Recommender Systems*, 2011.
- [12] Gradshteyn, I. S. and Ryzhik, I. M, *Tables of Integrals, Series, and Products*, 6th ed, 2000.
- [13] Tom Dietterich, *Overfitting and undercomputing in machine learning*, 1995.
- [14] B.M.Sarwar, G.K., J.Konstan, J.Riedl, *Incremental SVD-based algorithms for highly scaleable recommender systems (in Proceedings of the 5th International Conference on Computer and Information Technology 2002)*, 2002.
- [15] S.Deerwester, S.T.D., G.W.Furnas, T.K.Landauer, R.Harshman, *Indexing by latent semantic analysis (Journal of the American Society for Information Science, vol.41, no.6, pp.391–407)*, 1990.

- [16] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, Nuria Oliver, *Multiverse Recommendation: N-dimensional Tensor Factorization for Context-aware Collaborative Filtering*, 2010.
- [17] Yong Zheng, Bamshad Mobasher, Robin Burke, *Deviation-Based Contextual SLIM Recommenders*, 2014.
- [18] Yong Zheng, Bamshad Mobasher, Robin Burke, *Similarity-Based Context-Aware Recommendation*, 2015.
- [19] Linas Baltrunas, Marius Kaminskas, Bernd Ludwig, Omar Moling, Francesco Ricci, Aykan Aydin, Karl-Heinz Lüke, Roland Schwaiger, *InCarMusic: Context-Aware Music Recommendations in a Car*, 2011.
- [20] Nguyễn Minh Hiếu, Lê Trọng Hiếu, *Hệ thống khuyến nghị hỗ trợ khách du lịch dùng điện thoại Android*, 2011.

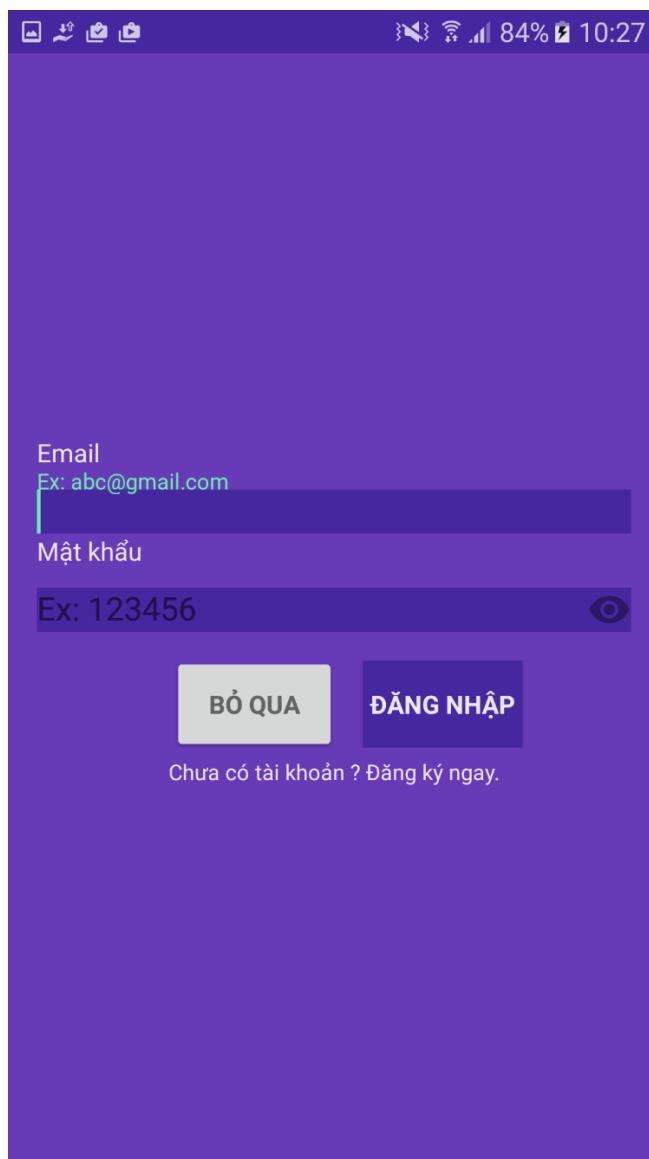
## PHỤ LỤC: CÀI ĐẶT VÀ TRIỂN KHAI

### 1. Môi trường cài đặt

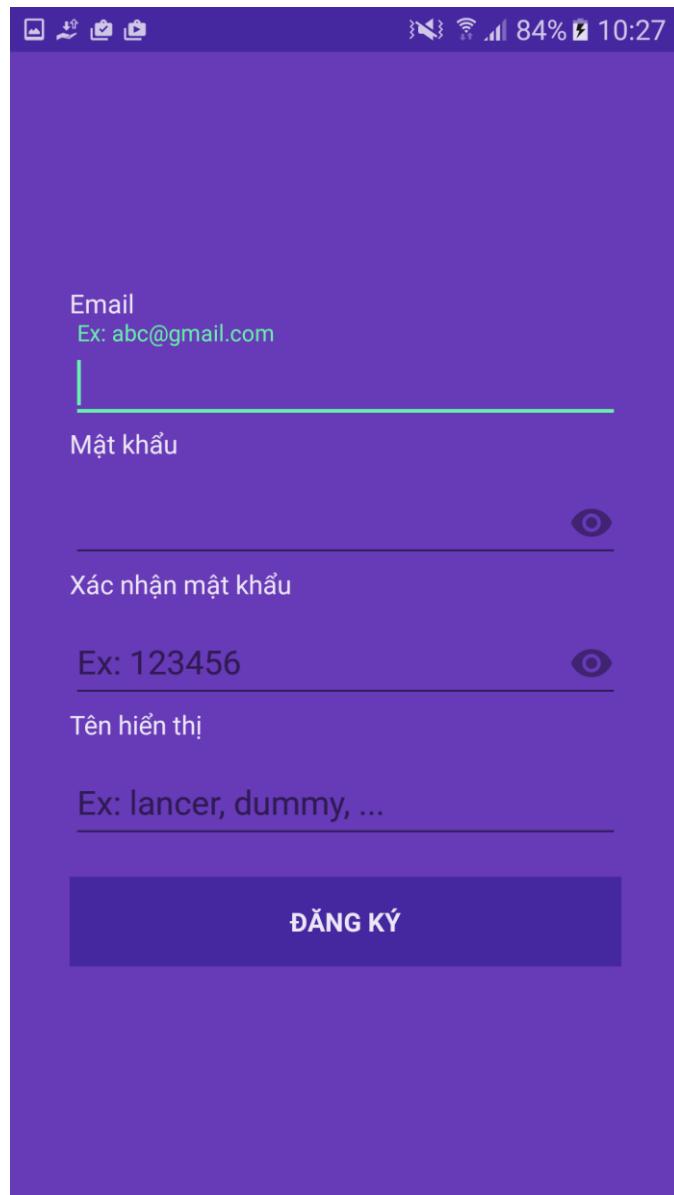
- Thiết bị đầu cuối: Smart phone chạy trên hệ điều hành Android.
- Web server: Dịch vụ web được triển khai trên Tomcat apache.
- Cơ sở dữ liệu: Hệ quản trị cơ sở dữ liệu được sử dụng là MySQL và được triển khai bằngxampp.

### 2. Triển khai ứng dụng

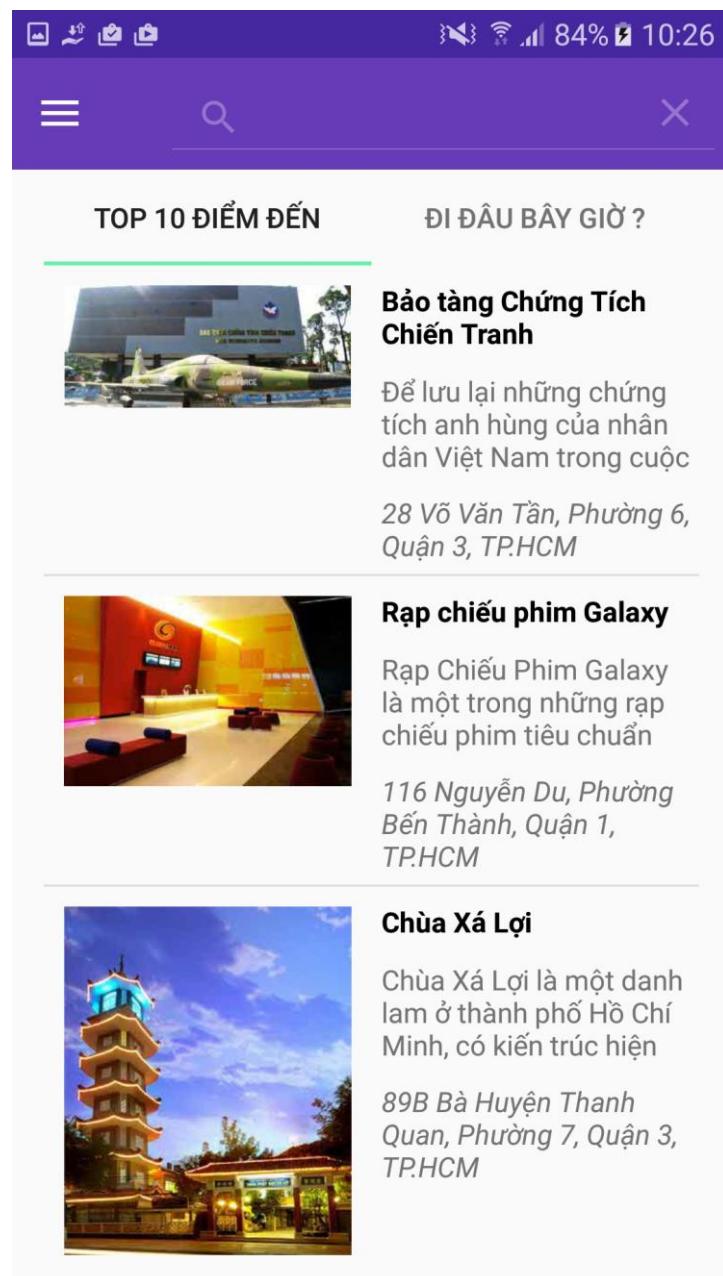
Hệ thống ứng dụng được triển khai với môi trường như trên. Dưới đây là một số hình ảnh của ứng dụng.



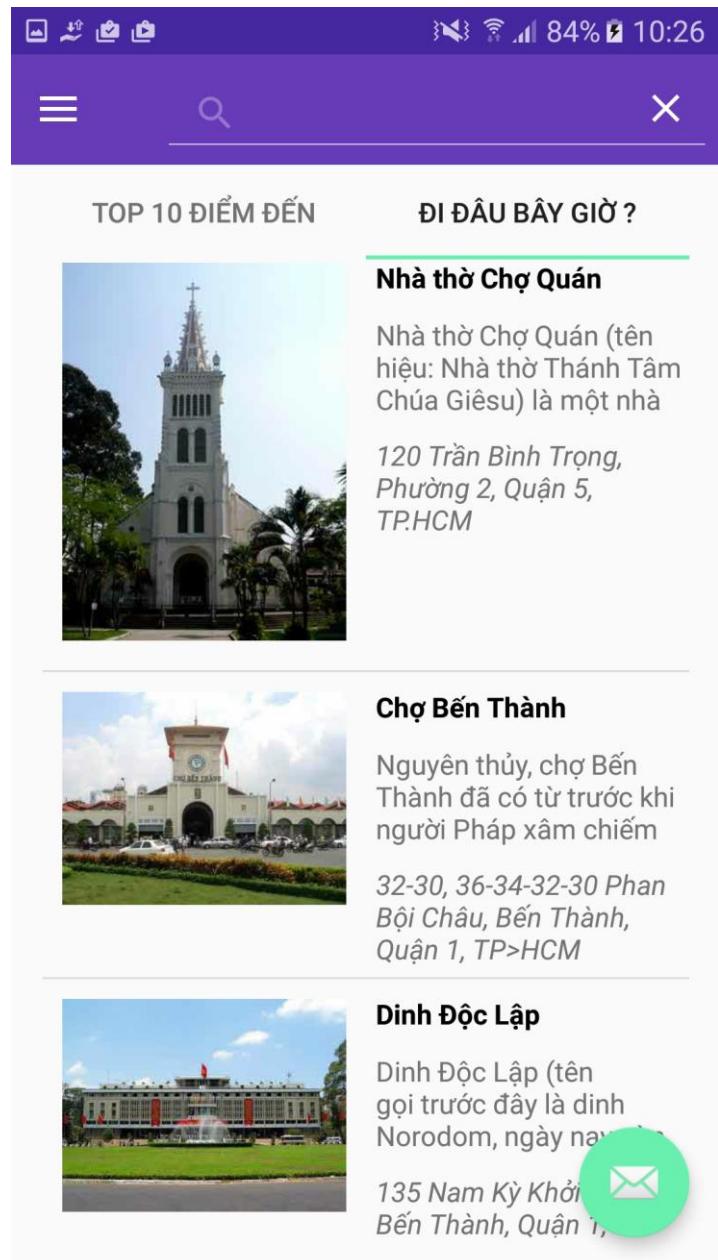
Hình ảnh 46: Giao diện đăng nhập



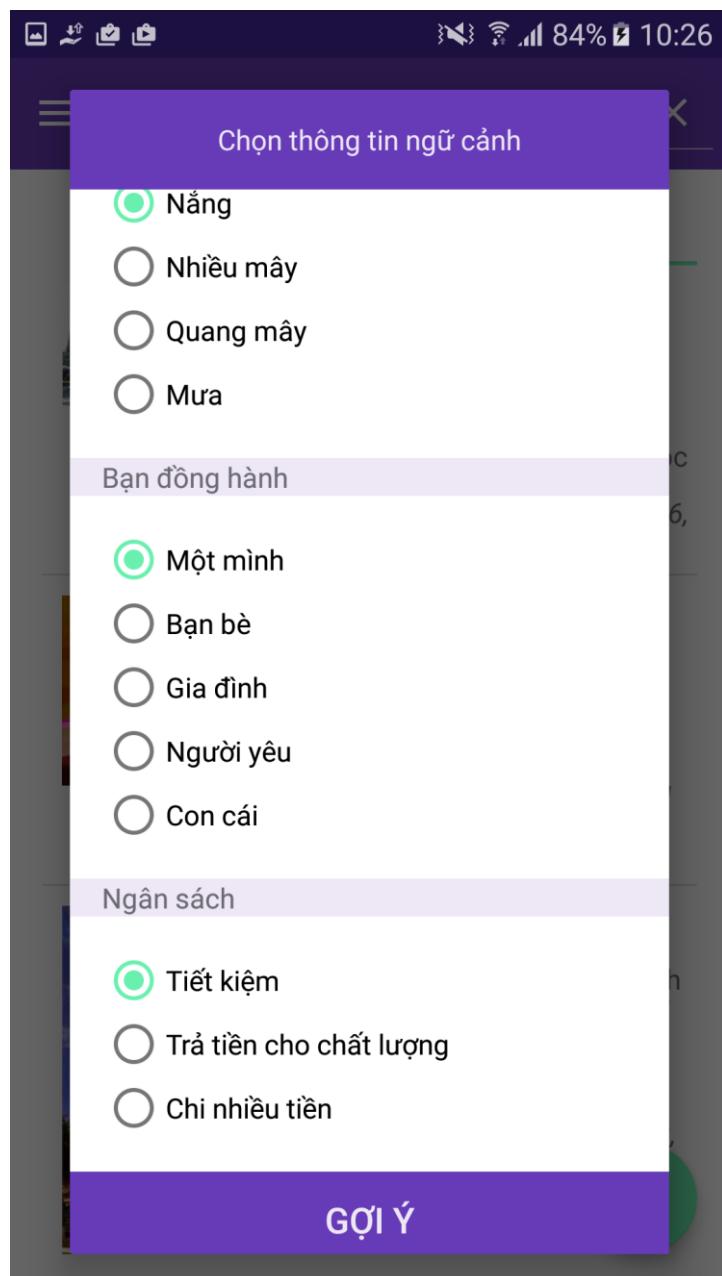
Hình ảnh 47: Giao diện đăng ký



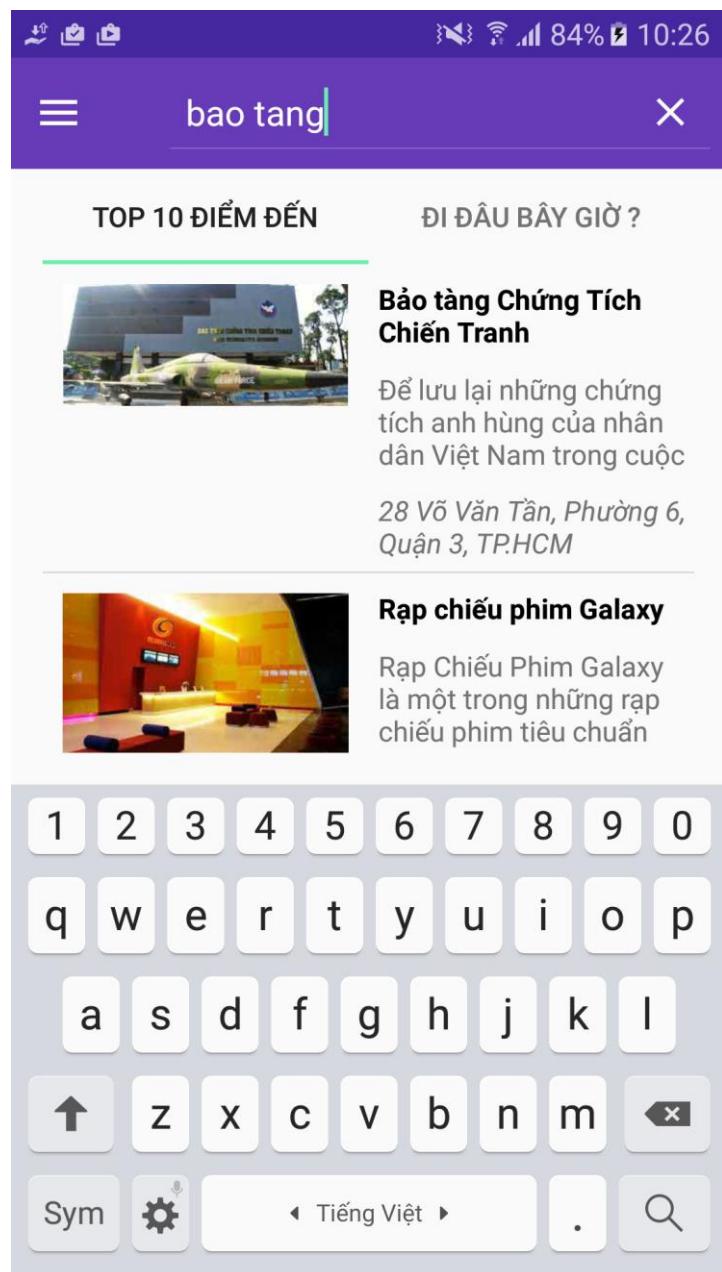
Hình ảnh 48: Giao diện mà hình chính, tab top 10



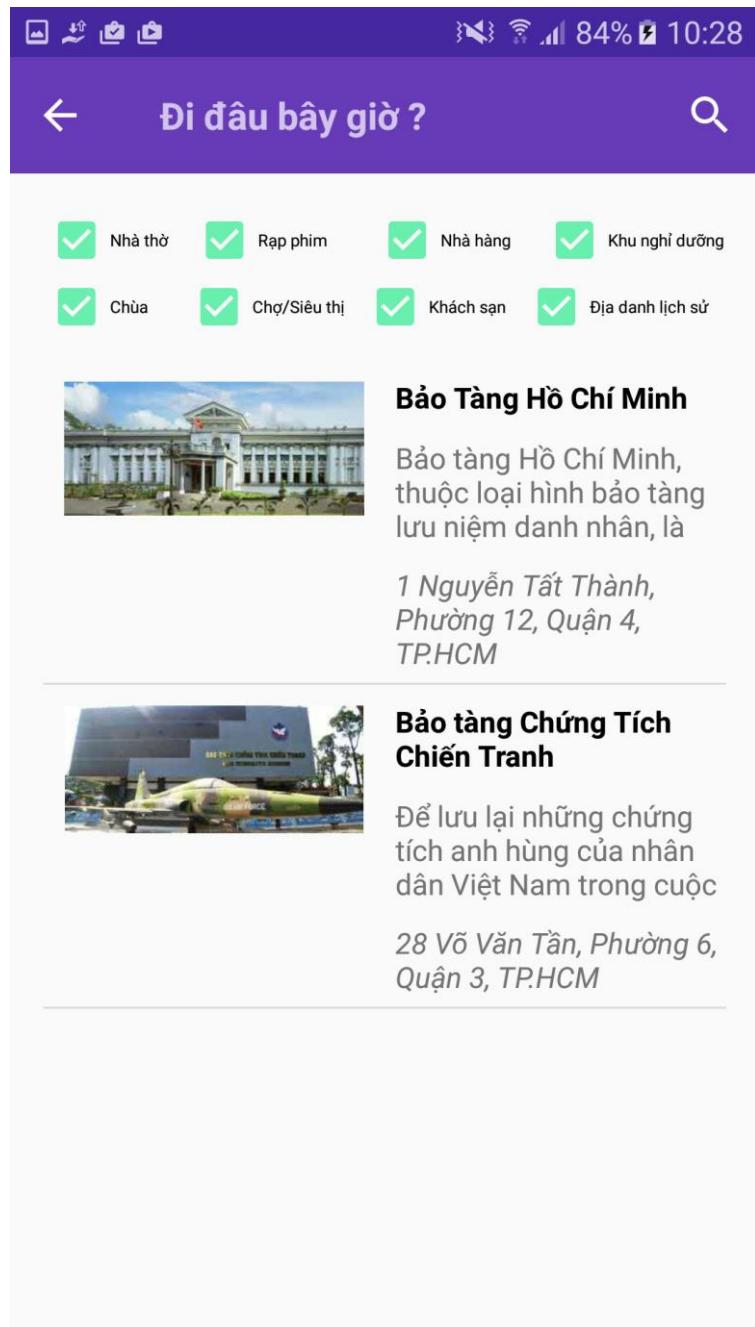
Hình ảnh 49: Giao diện màn hình chính tab gợi ý ngữ cảnh



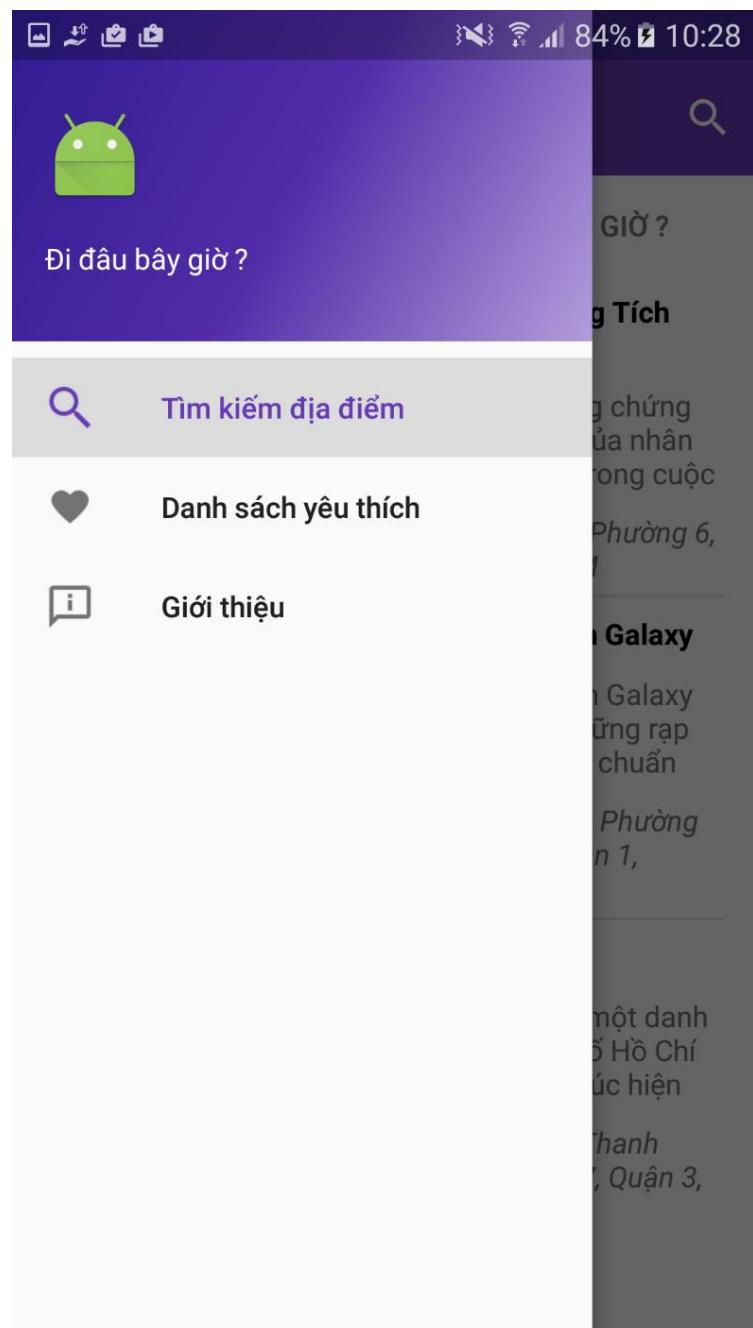
Hình ảnh 50: Dialog chọn ngữ cảnh khi gợi ý



Hình ảnh 51: Tính năng tìm kiếm trên giao diện chính



Hình ảnh 52: Giao diện kết quả tìm kiếm



Hình ảnh 53: Giao diện của sổ điều hướng

**Đi đâu bây giờ ?**

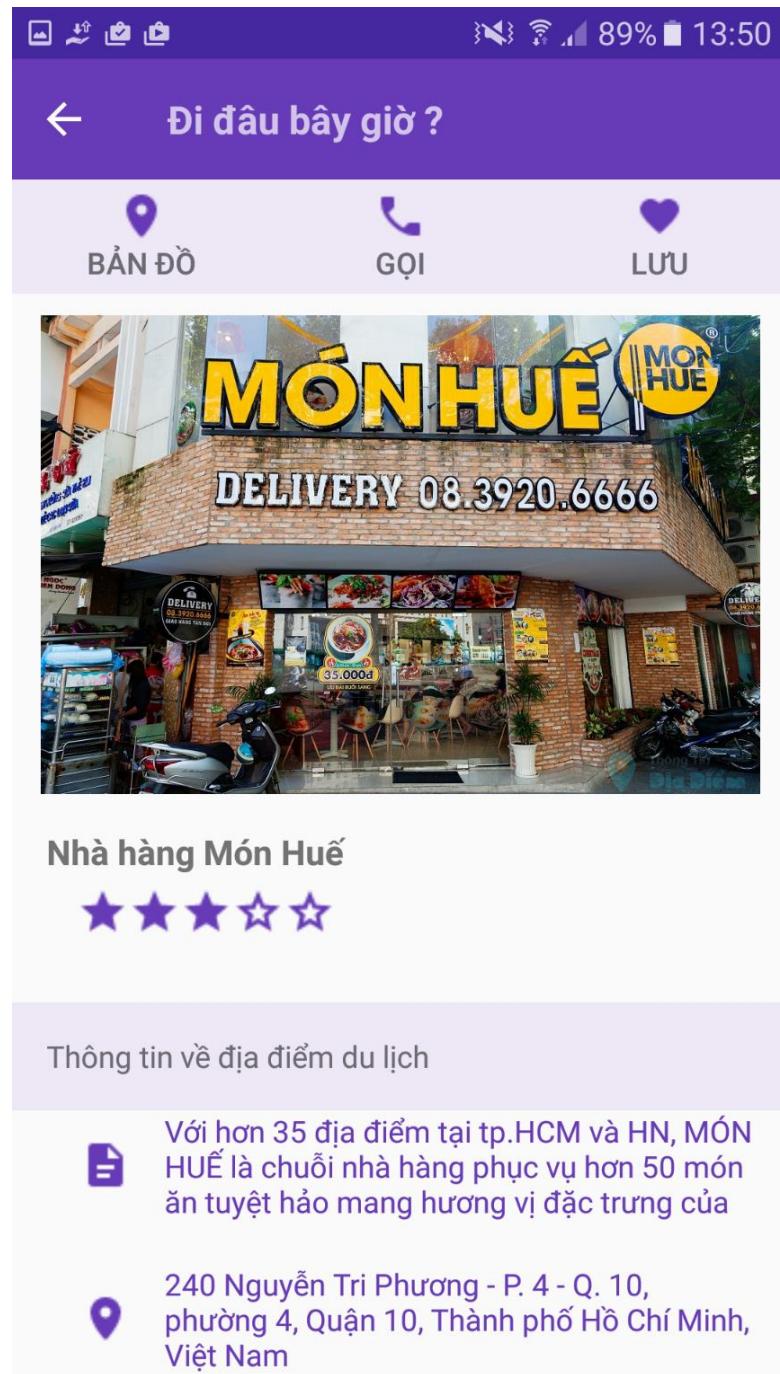
**Chợ Lớn (Chợ Bình Tây)**  
Chợ Bình Tây còn được biết với tên gọi là Chợ Lớn Mới được xây dựng  
57A Tháp Mười, Quận 6, TP.HCM

**Bảo tàng Chứng Tích Chiến Tranh**  
Để lưu lại những chứng tích anh hùng của nhân dân Việt Nam trong cuộc  
28 Võ Văn Tần, Phường 6, Quận 3, TP.HCM

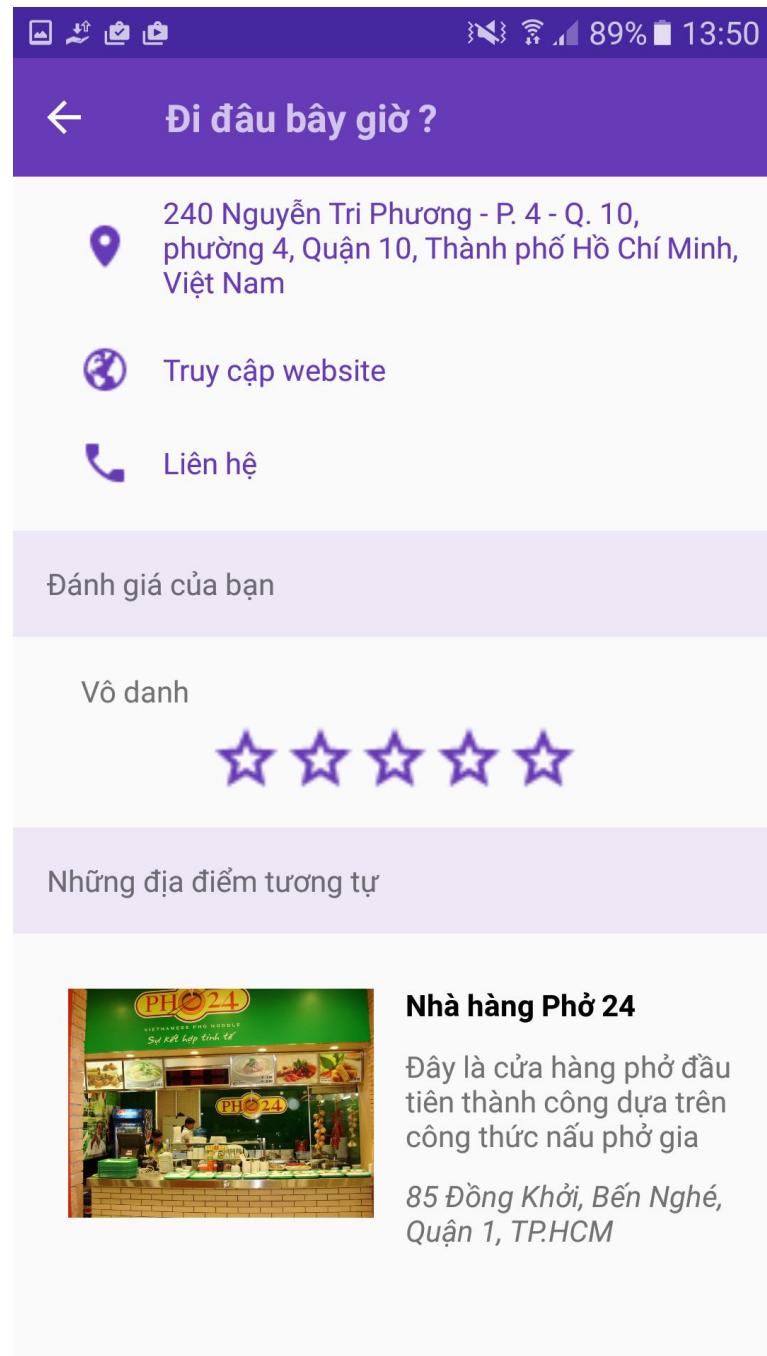
**Nhà hàng Món Huế**  
Với hơn 35 địa điểm tại tp.HCM và HN, MÓN HUẾ là chuỗi nhà hàng  
240 Nguyễn Tri Phương - P. 4 - Q. 10, phường 4, Quận 10, Thành phố Hồ Chí Minh, Việt Nam

**Công viên nước Đàm Sen**  
Đến với Công Viên Nước Đàm Sen

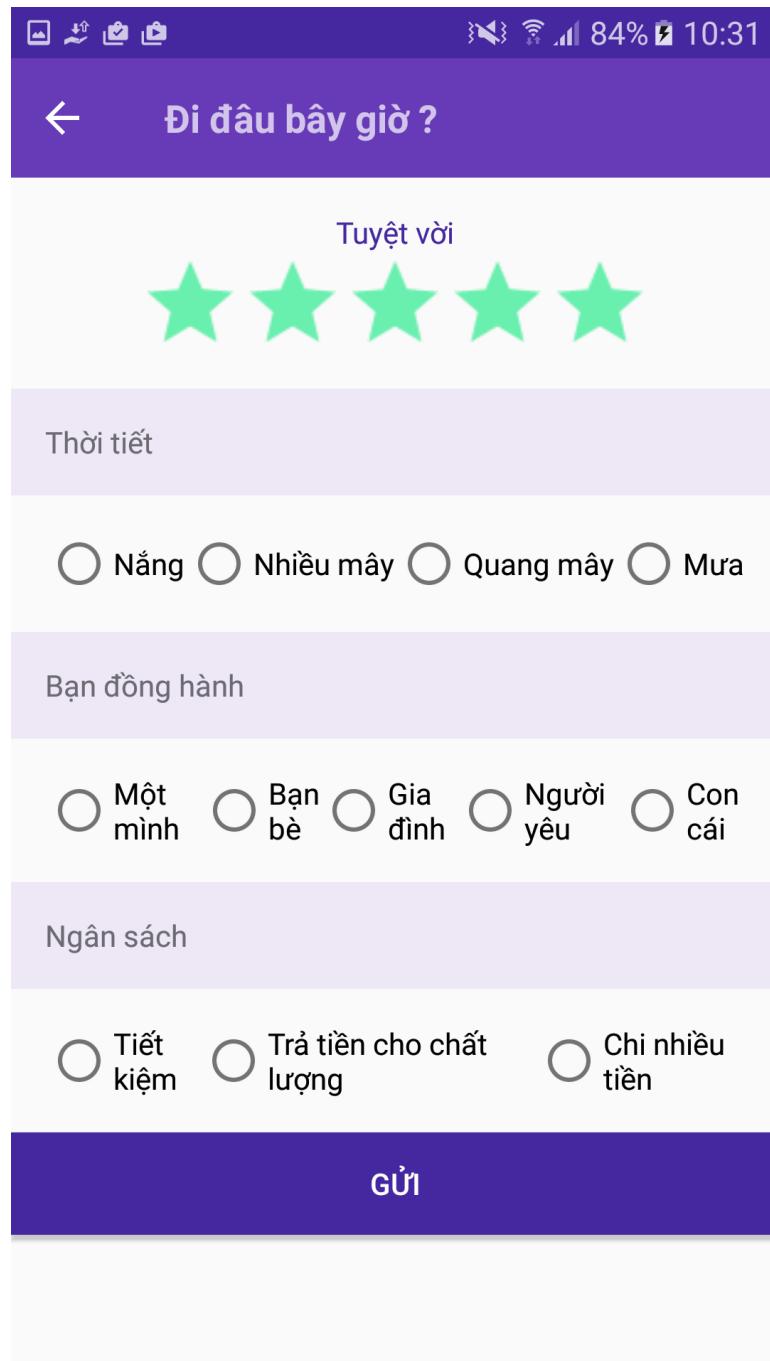
*Hình ảnh 54: Giao diện danh sách yêu thích*



Hình ảnh 55: Giao diện chi tiết sản phẩm (1)



Hình ảnh 56: Giao diện chi tiết sản phẩm (2)



Hình ảnh 57: Giao diện đánh giá sản phẩm