ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA HỆ THỐNG THÔNG TIN

------✶------

Khóa luận tốt nghiệp

Tên đề tài :

ỨNG DỤNG HỖ TRỢ KHÁCH DU LỊCH TRÊN ANDROID

GVHD: Th.S Huỳnh Hữu Việt

Lớp HTTT02

Nhóm sinh viên thực hiện:

Nguyễn Minh Hiếu 07520117

Lê Trọng Hiếu 07520119

Tp.HCM, ngày 06 tháng 02 năm 2012

LỜI CẢM ƠN

Để có thể hoàn thành khóa luận này, trước hết nhóm tác giả xin chân thành cảm ơn tập thể quý thầy cô trường Đại Học Công Nghệ Thông Tin - ĐHQG TP.HCM, quý thầy cô khoa Hệ Thống Thông Tin đã giảng dạy và trang bị cho nhóm tác giả những kiến thức từ cơ bản đến nâng cao làm cơ sở giải quyết những khó khăn trong suốt quá trình thực hiện. Tiếp đến, nhóm tác giả đặc biệt gửi lời cảm ơn chân thành sâu sắc đến Th.S Huỳnh Hữu Việt – khoa Hệ Thống Thông Tin, người đã không tiếc công sức, thời gian tận tình hướng dẫn, giúp đỡ và tạo mọi điều kiện thuận lợi giúp nhóm tác giả từng bước hoàn thành khóa luận.

Qua thời gian hơn bốn tháng, nhóm tác giả đã học hỏi thêm được rất nhiều kiến thức cũng như kinh nghiệm. Trong quá trình thực hiện khóa luận, chắc chắn không thể tránh khỏi những thiếu sót. Rất mong nhận được sự góp ý từ quý thầy cô để nhóm tác giả có thể hoàn thiện mình tốt hơn nữa phục vụ cho công việc trong tương lai.

Một lần nữa nhóm tác giả xin chân thành cảm ơn!

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

---oOo---

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

---oOo---

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

MỤC LỤC

DANH MỤC CÁC HÌNH VẼ

CHƯƠNG 1.
GIỚI THIỆU ĐỀ TÀI

---oOo---

Mở đầu, nhóm tác giả muốn giới thiệu một cách sơ lược nhất về đề tài sẽ thực hiện ở khóa luận này. Vì sao chọn đề tài? Bài toán đặt ra là gì? Những khái niệm ban đầu về hướng giải quyết bài toán như thế nào? Qua đó, người đọc sẽ có cái nhìn rất cơ bản về những gì sẽ được triển khai.

1.1 Đặt vấn đề:

Nước ta có nhiều danh lam thắng cảnh thu hút khách du lịch từ nhiều nơi trên thế giới. Với tiềm năng du lịch đa dạng và phong phú, ngành du lịch Việt Nam trong những năm qua ngày càng phát triển mạnh đem lại các lợi ích to lớn về kinh tế - xã hội, thúc đẩy các ngành sản xuất và dịch vụ phát triển. Dự đoán trong tương lai gần, du lịch sẽ là ngành kinh tế mũi nhọn, đóng góp lớn vào GDP của đất nước.

Ngày nay với nền công nghệ thông tin phát triển mạnh, nhiều ứng dụng và dịch vụ ra đời nhằm quảng bá hình ảnh du lịch của đất nước đến bạn bè thế giới. Những năm gần đây, điện thoại thông minh (smartphone) ngày càng trở nên phổ biến. Với lợi thế nhỏ gọn, tiện dụng và thông minh, điện thoại thông minh hỗ trợ rất nhiều cho con người trong các hoạt động hằng ngày của họ.

Thực tế cho thấy khi đi du lịch, những điều kiện ngữ cảnh xung quanh người dùng (thời tiết, nhiệt độ, tâm trạng ...) có ảnh hưởng đến quyết định lựa chọn điểm đến. Vì vậy nhóm tác giả quyết định chọn đề tài ứng dụng hệ thống khuyến nghị theo ngữ cảnh (Context-aware Recommender System) gợi ý địa điểm du lịch trên điện thoại thông minh hỗ trợ khách du lịch nhằm đem lại sự thoải mái, tiện dụng tối đa cho người dùng. Thay vì phải mang bên mình nhiều thứ cho chuyến đi như sách hướng dẫn, bản đồ, lịch trình, nhắc nhở, hay phải mất thời gian qua nhiều trang web, hỏi ý kiến bạn bè, người thân để có thông tin cần thiết ... tất cả sẽ được đem vào chiếc điện thoại nhỏ gọn.

Khi ứng dụng hoàn thành, khách du lịch sẽ cảm thấy thích thú hơn, dễ dàng hơn trong chuyến hành trình vì những thông tin du lịch giờ đây sẽ luôn có ở bên mình, khi cần tìm, sẽ có được thông tin nhanh hơn. Từ đó, nhóm tác giả hi vọng sẽ góp phần thúc đẩy ngành du lịch nước nhà phát triển tốt đẹp hơn.

1.2 Mục tiêu đề tài:

Xây dựng một hệ thống khuyến nghị các địa điểm du lịch cho khách du lịch khi đến với thành phố Hồ Chí Minh. Hệ thống khuyến nghị này sẽ quan tâm đến những điều kiện ngữ cảnh của người dùng và đưa ra những lời khuyến nghị dựa trên những điều kiện ngữ cảnh đó.

Xây dựng một ứng dụng trên điện thoại thông minh kết nối với hệ thống khuyến nghị trên. Ứng dụng này sẽ được triển khai đến người dùng cuối để những thông tin du lịch, thông tin khuyến nghị đến với họ một cách nhanh nhất.

1.3 Phạm vi khóa luận:

Do thời gian giới hạn nên nhóm tác giả chỉ tập trung vào một phạm vi nhất định. Cụ thể như sau:

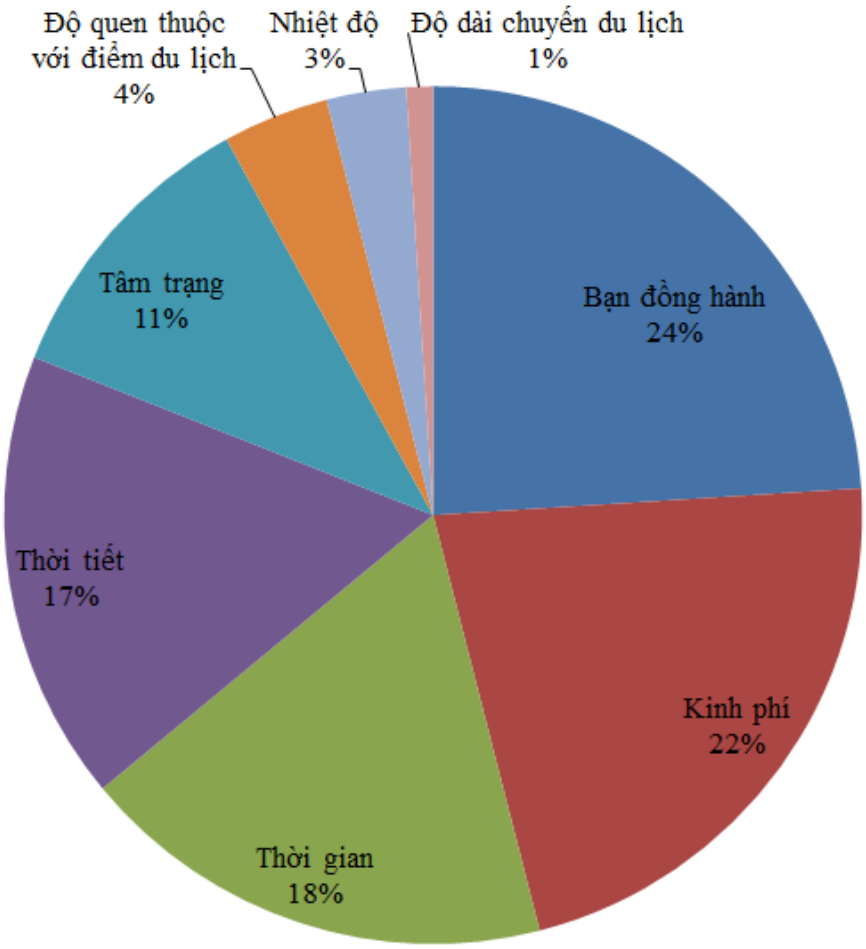
- *Đối tượng khuyến nghị:* các địa điểm du lịch nổi bật trong phạm vi thành phố Hồ Chí Minh (thông tin về các địa điểm này được sưu tầm và tổng hợp từ nhiều nguồn khác nhau trên Internet). Các điểm du lịch sẽ được gợi ý đến cho người dùng một cách thích hợp nhất.
- *Đối tượng được khuyến nghị:* người dùng điện thoại thông minh, có tài khoản trên hệ thống và có để lại những đánh giá về các điểm du lịch.
- *Các điều kiện ngữ cảnh được quan tâm:*

Nhóm tác giả đã tiến hành một cuộc khảo sát 100 người ngẫu nhiên trên Internet dùng Google Form. Với câu hỏi được đặt ra là:

"Theo bạn khi đi du lịch, yếu tố nào gây ảnh hưởng đến quyết định chọn lựa điểm du lịch của bạn nhất? (Vui lòng chọn ra một yếu tố bạn cho là quan trọng nhất).

- *Thời tiết (trời trong xanh, trời nắng, trời mưa ...)*
- *Nhiệt độ (nóng, ấm áp, lạnh ...)*
- *Bạn đồng hành (người thân, bạn bè, đồng nghiệp, người yêu, trẻ em ...)*
- *Thời gian (sáng, trưa, chiều, đầu tuần, cuối tuần ...)*
- *Tâm trạng (vui, buồn, hào hứng, lười biếng ...)*
- *Kinh phí (tiết kiệm, chuyến đi đảm bảo chất lượng, chi tiêu sang trọng ...)*
- *Độ dài chuyến du lịch (1 ngày, 3 ngày, 1 tuần ...)*
- *Độ quen thuộc với điểm du lịch (đã từng đi, người mới đến lần đầu ...)".*

Với 100 người trả lời khảo sát, kết quả như sau:



Hình 1.1: Kết quả khảo sát 100 người ngẫu nhiên.

Dựa vào đó, nhóm tác giả chọn ra bốn điều kiện ngữ cảnh có số người quan tâm nhiều nhất để sử dụng trong khóa luận này, đó là:

- ✓ Thời tiết: trời nắng, trời âm u, trời trong xanh, trời mưa.
- ✓ Bạn đồng hành: đi một mình, đi với bạn bè hoặc đồng nghiệp, đi với gia đình, đi với người yêu, đi với trẻ em.
- ✓ Kinh phí du lịch: chi tiêu tiết kiệm, bảo đảm cho chất lượng, chi tiêu sang trọng.
- ✓ Thời gian: thời gian đi du lịch (ngày tháng năm, mùa, buổi ...)

1.4 Phương pháp nghiên cứu:

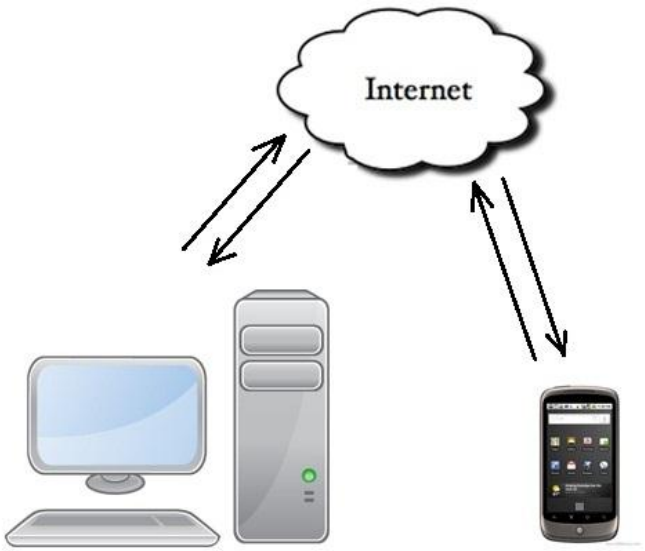
Để thực hiện được đề tài khóa luận này, nhóm tác giả sẽ:

- ➢ Khảo sát, tìm hiểu các phương pháp khuyến nghị đã có. Từ đó chọn ra hướng đi thích hợp nhất để nghiên cứu và phát triển.
- ➢ Khảo sát các ứng dụng đã có trên điện thoại thông minh về du lịch.

- Đưa đến cho người dùng sử dụng thực tế. Sau đó tiến hành kiểm thử kết quả và đánh giá.

1.5 Định hướng giải quyết:

Qua yêu cầu bài toán, để phát triển được ứng dụng trên điện thoại nói trên, về cơ bản, hệ thống cần phải có hai phần chính. Một là máy chủ (server) dùng lưu trữ cơ sở dữ liệu cần thiết. Hai là máy khách (client) dùng để hiển thị dữ liệu lấy từ máy chủ về cho người sử dụng. Ở đây, tất nhiên những chiến điện thoại thông minh sẽ đóng vai trò là các máy khách. Từ đó, người đọc có thể hình dung kiến trúc cơ bản trên thông qua hình vẽ sau đây:



Hình 1.2: Giao tiếp giữa máy chủ và điện thoại qua mạng Internet.

Những phương pháp cụ thể để giải quyết bài toán, nhóm tác giả sẽ lần lượt đi vào chi tiết ở những chương tiếp theo.

1.6 Kết quả dự kiến đạt được:

Xây dựng thành công một hệ thống khuyến nghị du lịch dựa trên ngữ cảnh với một thuật toán thích hợp. Song song đó là xây dựng thành công một ứng dụng trên điện thoại thông minh sử dụng hệ khuyến nghị trên. Cụ thể, hệ thống sẽ hoạt động như sau: người dùng ứng dụng trên điện thoại sẽ đăng ký một tài khoản trên hệ thống, tiến hành đánh giá những địa điểm du lịch. Hệ thống sẽ ghi nhận những đánh giá đó để làm cơ sở khuyến nghị. Khi cần khuyến nghị, người dùng cung cấp những điều kiện ngữ cảnh, hệ thống sẽ dựa vào những điều kiện ngữ cảnh đó để gợi ý những điểm du lịch, trả về điện thoại của người dùng.

CHƯƠNG 2.
KHẢO SÁT CÁC PHƯƠNG PHÁP KHUYẾN NGHỊ

---oOo---

Chương 1 trên đây là những hình dung ban đầu. Thông thường, một bài toán trong lĩnh vực công nghệ thông tin để giải quyết được cần phải có hai phần: phần kỹ thuật xây dựng ứng dụng và phần phương pháp giải quyết bài toán. Ở chương 2, nhóm tác giả sẽ đi sâu vào tìm hiểu những phương pháp khuyến nghị đã có, từ đó lựa chọn ra hướng đi thích hợp nhất với đề tài của nhóm để nghiên cứu, triển khai.

2.1. Mô hình khuyến nghị truyền thống (hai chiều):

Trong lĩnh vực xây dựng các hệ thống khuyến nghị trong quá khứ, các chuyên gia đã làm việc và nghiên cứu khá nhiều. Hầu hết công việc chủ yếu tập trung phát triển những phương pháp gợi ý những đối tượng ưa thích đến cho người dùng. Ví dụ như những trang web gợi ý những bộ phim, gợi ý những quyển sách mà người dùng có thể yêu thích.

Những hệ thống khuyến nghị truyền thống ban đầu, người ta quan tâm đến hai thực thể là người dùng và đối tượng cần được khuyến nghị đến cho người dùng (ở đây nhóm tác giả gọi tắt là đối tượng) [1]. Quá trình khuyến nghị bắt đầu bằng một tập hợp các chỉ số đánh giá của người dùng đối với các đối tượng. Các chỉ số này được cung cấp một cách trực tiếp, tường minh từ người dùng hoặc được suy ra bởi hệ thống dựa vào một số thuật toán nào đó.

Ví dụ có một danh sách các chỉ số đánh giá như sau:

- Người A đánh giá điểm du lịch "ĐLA" với chỉ số là 4 (trên tổng mức 5).
- Người A đánh giá điểm du lịch "ĐLB" với chỉ số là 5 (trên tổng mức 5).
- Người A đánh giá điểm du lịch "ĐLC" với chỉ số là 3 (trên tổng mức 5).
- Người B đánh giá điểm du lịch "ĐLA" với chỉ số là 4 (trên tổng mức 5).
- Người B đánh giá điểm du lịch "ĐLB" với chỉ số là 4 (trên tổng mức 5).

Từ đó sẽ xây dựng một hàm *R* dùng để dự đoán các chỉ số đánh giá còn chưa biết. Ví dụ cần dự đoán người B sẽ đánh giá điểm du lịch "ĐLC" với chỉ số là bao nhiêu.

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

Trong đó *Users* là các người dùng, *Items* là các đối tượng và *Ratings* là các chỉ số đánh giá. Sau khi hàm *R* dự đoán được các chỉ số đánh giá trong toàn bộ không

gian khuyến nghị (*Users* x *Items*), hệ thống khuyến nghị sẽ có thể chọn ra k đối tượng i có chỉ số đánh giá dự đoán cao nhất và gợi ý chúng đến người dùng u.

$$\forall u \in Users, i(u) = arg \max_{i \in Items} R(u,i)$$

Trong thực tế, không nhất thiết phải dự đoán dựa trên toàn bộ không gian (*Users* x *Items*), vì chi phí cho việc tính toán trên một miền lớn như vậy rất tốn kém. Thay vào đó, người ta nghiên cứu để tìm ra những giải pháp hữu hiệu nhằm thu nhỏ không gian dự đoán để giảm thiểu chi phí tính toán. Nhìn chung, những hệ thống khuyến nghị thường được phân thành ba loại: khuyến nghị dựa trên nội dung (content-based), khuyến nghị bằng cách đánh giá độ tương đồng (collaborative filtering) và khuyến nghị lai (hybrid) [2]. Nhóm tác giả sẽ lần lượt nói qua một cách vắn tắt ngay sau đây.

2.1.1. Hệ thống khuyến nghị dựa trên nội dung:

Trong những phương pháp khuyến nghị dựa trên nội dung, chỉ số đánh giá dự đoán *R(u,i)* của người dùng u đối với đối tượng i thường được ước lượng dựa vào những chỉ số dự đoán *R(u,i')* của người dùng u đó đối với những đối tượng $i' \in Items$ (tập hợp các đối tượng) tương tự với đối tượng i. Sự tương tự giữa hai đối tượng i và i' được tính toán tùy theo nội dung của chúng.

Ví dụ trong hệ thống khuyến nghị phim dựa trên nội dung, để gợi ý những bộ phim cho người dùng u, hệ thống cố gắng tìm hiểu những sở thích của người dùng bằng cách phân tích những điểm tương đồng về mặt nội dung của những bộ phim mà người dùng u đã từng đánh giá trong quá khứ. Khi đó, chỉ những bộ phim nào có độ tương tự cao, phù hợp với sở thích của người dùng mới được hệ thống gợi ý.

Nói một cách hình thức, gọi *Content(i)* là một tập hợp các thuộc tính nói lên đặc điểm của đối tượng i. Phần đặc tính của đối tượng i này thường được tính toán bằng cách trích ra một phần trong nội dung của nó và dùng phần trích xuất đó để xác định những yếu tố cần thiết phục vụ cho mục đích khuyến nghị. Có nhiều hệ thống khuyến nghị được thiết kế để gợi ý những đối tượng dựa vào văn bản (text-based). Chúng được dùng để gợi ý trong những trang web, những bảng thông tin mà phần nội dung là những thông tin ở dạng văn bản, được mô tả bằng những từ khóa (keywords). Để xác đinh mức độ quan trọng của những từ khóa dùng gợi ý, cần xác định một độ đo. Độ đo đó là những trọng số được tính toán dựa vào những thông tin thu thập được.

Hàm dự đoán chỉ số đánh giá *R(u,i)* trong những hệ thống khuyến nghị dựa trên nội dung thường được định nghĩa như sau:

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

Trong đó, *ContentBasedProfile(u)* và *Content(i))* được định nghĩa như là những vector $\overrightarrow{w(u)}$, $\overrightarrow{w(\imath)}$ mô tả những đặc tính của người dùng *u* và đối tượng *i* [3]. Và chỉ số đánh giá được dự đoán cho người dùng *u* đối với đối tượng *i* được tính toán dựa trên độ tương tự giữa hai vector, ví dụ như độ tương tự Cosine.

Ngoài những phương pháp truyền thống hầu hết dựa trên việc trích xuất thông tin, những kỹ thuật khác cũng được sử dụng, như phân lớp Bayesian, hay những kỹ thuật máy học gồm có gom cụm, cây quyết định, mạng Neuron nhân tạo ... Những kỹ thuật này khác so với kỹ thuật trích xuất thông tin ở chỗ chúng tính toán những chỉ số dự đoán dựa trên những mô hình (model) được học từ tập dữ liệu bằng cách dùng những phương pháp học thống kê hoặc những kỹ thuật máy học [4].

Ví dụ dựa vào một tập những trang web đã được người dùng đánh giá và xếp thành hai loại là "phù hợp" và "không phù hợp", có thể dùng thuật toán phân lớp Bayesian để phân lớp cho những trang web chưa được đánh giá xếp loại.

Những hệ thống khuyến nghị dựa trên nội dung có những giới hạn nhất định. Đặc biệt là giới hạn về khả năng phân tích phần nội dung. Chúng chỉ hoạt động tốt trên những miền nội dung nơi mà thông tin có thể được trích xuất tự động (thông tin dạng văn bản) hoặc được cung cấp một cách thủ công (ví dụ như thông tin của các bộ phim được mô tả cũng ở dạng văn bản). Còn đối với những thông tin dạng đa phương tiện (hình ảnh, âm thanh …) thì thật sự rất khó khăn [5]. Thông thường, những hệ thống khuyến nghị gợi ý những đối tượng tương tự với những đối tượng mà người dùng đã đánh giá trước đó. Tuy nhiên trong một số trường hợp đặc biệt, đối tượng không nên được gợi ý vì chúng có độ tương tự gần như tuyệt đối, nói cách khác là chúng quá tương tự với những thứ người dùng vừa mới xem. Ví dụ như nhiều mục tin tức khác nhau cùng nói về một sự kiện người dùng vừa xem qua ở mục tin tức nào đó, khi đó người dùng sẽ không quan tâm đến những mục tin tức cùng sự kiện kia, hệ thống cũng không nên gợi ý. Thêm một đòi hỏi nữa, là người dùng phải có đánh giá cho một số lượng các đối tượng trước khi hệ thống có thể hiểu được sở thích và gợi ý cho họ những đối tượng khác. Như vậy, hệ thống sẽ gặp vấn đề đối với những người dùng mới, họ chưa cung cấp hoặc cung cấp rất ít những

chỉ số dự đoán, hệ thống không đủ dữ liệu ban đầu của người dùng đó để có thể đưa ra những lời gợi ý chính xác dành cho họ.[1]

2.1.2. Hệ thống khuyến nghị bằng cách đánh giá độ tương đồng:

Theo truyền thống, những hệ thống khuyến nghị bằng cách đánh giá độ tương đồng sẽ dự đoán những chỉ số đánh giá đối với một đối tượng cho một người dùng nào đó dựa trên những chỉ số đánh giá trước đây của những người dùng khác đối với đối tượng đang xem xét. Nói đơn giản, chỉ số *r(u,i)* dự đoán đánh giá của người dùng *u* đối với đối tượng *i* dựa trên những chỉ số *r(u',i)* của những người dùng *u'* (có tính chất tương tự với người dùng *u*) đối với đối tượng *i*.

Những thuật toán khuyến nghị đánh giá độ tương đồng có thể được phân thành hai nhóm là phương pháp khuyến nghị dựa trên kinh nghiệm (memory-based hay còn gọi là heuristic-based) và phương pháp khuyến nghị dựa trên mô hình (model-based) [6].

a) Phương pháp khuyến nghị dựa trên kinh nghiệm:

Phương pháp khuyến nghị dựa trên kinh nghiệm dự đoán những chỉ số đánh giá dựa vào tập hợp tất cả những đối tượng đã được đánh giá bởi những người dùng trước đó. Giá trị cần dự đoán *r(u,i)* của người dùng *u* đối với đối tượng *i* thường được tính toán bằng một hàm tổng hợp (aggregation) của những chỉ số đánh giá từ những người dùng khác đối với đối tượng *i* (ở đây xét đến những người dùng tương tự với người dùng *u*, có thể là tất cả người dùng, hoặc chọn ra tập hợp n người dùng tương tự nhất):

$$r(u,i) = \underset{u' \in U}{aggr}\, r(u',i)$$

Trong đó, *U* là tập hợp *n* người dùng tương tự với người dùng *u* (*n* có giá trị nhỏ nhất là 1, lớn nhất là toàn bộ tập hợp những người dùng) và họ đã có đánh giá trên đối tượng *i*. Một vài ví dụ về công thức tính *r(u,i)* như sau:

$$r(u,i) = \frac{1}{N} \sum_{u' \in U} r(u',i)$$

$$r(u,i) = k \sum_{u' \in U} sim(u.u') \text{ x } r(u',i)$$

$$r(u,i) = \overline{r(u)} \sum_{u' \in U} sim(u.u') \text{ x } [r(u',i) - \overline{r(u')}\,]$$

Trong đó, k là tác nhân chuẩn hóa các đánh giá của người dùng. Ví dụ có những người có xu hướng đánh giá mức 4 điểm, 5 điểm là tốt nhất, có những người cũng là đánh giá mức tốt nhất nhưng chỉ chọn thang điểm 3, 4. Vì vậy cần có sự chuẩn hóa. Chỉ số k thường được tính bằng công thức:

$$k = 1/\sum_{u' \in U} |sim(u,u')|$$

$\overline{r(u)}$ là chỉ số đánh giá trung bình của người dùng u đối với các đối tượng được định nghĩa bằng công thức:

$$\overline{r(u)} = (1/|Su|)\sum_{i \in Su} r(u,i)$$

$$Su = \{i \in Items | r(u,i) \neq \emptyset\}$$

sim(u,u') là độ tương tự (similarity) giữa hai người dùng u và *u'*. Những người dùng có độ tương tự càng cao thì khả năng sở thích của họ giống nhau càng tăng, mức độ ảnh hưởng của những chỉ số đánh giá *r(u',i)* trong công thức càng lớn. Điều này giúp chỉ số dự đoán *r(u,i)* cho người dùng u có độ chính xác cao hơn. Có nhiều cách để tính độ tương tự này, trong hầu hết các cách tính toán, người ta dựa vào những chỉ số đánh giá trên những đối tượng mà cả người dùng u cũng như *u'* đã từng đánh giá. Có hai công thức được dùng phổ biến đó là hệ số tương quan Pearson và hệ số tương quan Cosine.

Hệ số tương quan Pearson:

$$sim(x,y) = \frac{\sum_{s \in Sxy}\left[r(x,s) - \overline{r(x)}\right].\left[r(y,s) - \overline{r(y)}\right]}{\sqrt{\sum_{s \in Sxy}\left[r(x,s) - \overline{r(x)}\right]^2 \sum_{s \in Sxy}\left[r(y,s) - \overline{r(y)}\right]^2}}$$

Hệ số tương quan Cosine:

$$sim(x,y) = \cos(\vec{x},\vec{y}) = \frac{\vec{x}.\vec{y}}{\left|\left|\vec{x}\right|\right| \text{ x } \left|\left|\vec{y}\right|\right|} = \frac{\sum_{s \in Sxy} r(x,s)r(y,s)}{\sqrt{\sum_{s \in Sxy} r^2(x,s)}\sqrt{\sum_{s \in Sxy} r^2(y,s)}}$$

Trong đó, *r(x,s)* và *r(y,s)* là những chỉ số đánh giá đối với đối tượng s lần lượt của người dùng x và y, $\overline{r(x)}$, $\overline{r(y)}$ là các chỉ số đánh giá trung bình của người dùng x và y vừa được nhắc đến ở trên, $Sxy = \{s \in Items \ | r(x,s) \neq \emptyset \wedge r(y,s) \neq \emptyset\}$ là tập hợp các đối tượng cùng được đánh giá bởi người dùng x và y, $\vec{x}.\vec{y}$ là tích vô hướng của hai vector $\vec{x}$ và $\vec{y}$. [1]

Các nghiên cứu cũng đã tìm ra nhiều cách để cải tiến, mở rộng cách tính các hệ số tương quan. Hầu hết các hệ số tương quan này đều dùng để tính độ tương tự giữa những người dùng. Ngoài ra, những hệ số này cũng được dùng để tính độ tương tự giữa những đối tượng với nhau. Như vậy, trong phương pháp khuyến nghị dựa trên kinh nghiệm, có thể phân thành hai nhánh nhỏ là phương pháp dựa trên người dùng (user-based) và phương pháp dựa trên đối tượng (item-based). Có nghiên cứu cho rằng phương pháp dựa trên đối tượng trong một số trường hợp có thể cho hiệu năng cao hơn, chất lượng dự đoán cũng cao hơn. [1]

b) Phương pháp khuyến nghị dựa trên mô hình:

Việc thiết kế và mô phỏng các mô hình (máy học và các thuật toán khai phá dữ liệu) cho phép các hệ thống học và nhận ra các mô hình (pattern) phức tạp thông qua bước huấn luyện dữ liệu, và sau đó có thể đưa ra những dự đoán thông minh trong hệ thống khuyến nghị. Các thuật toán khuyến nghị dựa trên mô hình như mô hình Bayes (Bayesian model), mô hình gom cụm (clustering model), mạng phụ thuộc (Dependency Network) đã được đưa ra để giải quyết những thiếu sót của phương pháp khuyến nghị dựa trên kinh nghiệm [7],[8]. Thông thường các thuật toán phân lớp được sử dụng trong các hệ thống khuyến nghị nếu đánh giá của người dùng là tuyệt đối (phân loại), trong khi đó các mô hình hồi qui và phương thức SVD (Singular Value Decomposition) áp dụng cho các đánh giá bằng số liệu. Trong giới hạn đề tài (các đánh giá của người dùng là số liệu) nhóm tác giả tiến hành khảo sát các mô hình hồi qui.[9]

Quay trở lại phương pháp khuyến nghị dựa trên kinh nghiệm đã đề cập ở trên, có thể nhận ra trong nhiều trường hợp hai vector có thể có khoảng cách Euclide xa nhau nhưng chúng lại cũng có thể rất gần nhau với độ đo Cosine hay Pearson. Sử dụng mô hình hồi qui trong hệ thống khuyến nghị sẽ giải quyết tốt hơn những vấn đề trên. Hồi qui là phương pháp sử dụng một phép toán xấp xỉ để dự đoán đánh giá của người dùng. Ta có thể xem tập $X = (X_1, X_2, ..., X_n)$ là tập biến số đại diện cho sự ưa thích của một người dùng đối với những đối tượng khác nhau. Mô hình hồi qui tuyến tính có thể được phát biểu như sau:

$$Y = \alpha X + \beta$$

Trong đó α là một ma trận $n \times k$, $\beta = (\beta_1, \beta_2, ..., \beta_n)$ đại diện cho độ nhiễu trong lựa chọn của người dùng, Y là một ma trận $n \times m$ với Y_{ij} là đánh giá của

người dùng i cho đối tượng j, X là một ma trận $k \times m$ với mỗi cột là một đánh giá của một người dùng trong không gian đánh giá k chiều. Ta có thể dễ dàng thấy được ma trận Y ở đây rất thưa thớt. Để giải quyết vấn đề này, Canny [10] đề xuất một kỹ thuật phân tích (Sparse Factor Analysis) giúp thay thế những ô trống này bằng những giá trị mặc định (giá trị trung bình của một vài ô có giá trị, trung bình theo cột, theo dòng, hoặc tất cả).

Cũng như hệ thống khuyến nghị dựa trên nội dung, hệ thống khuyến nghị bằng cách đánh giá độ tương đồng cũng có những giới hạn, khiếm khuyết nhất định. Vẫn là vấn đề người dùng mới, hệ thống không đủ dữ liệu để có thể đưa ra những lời gợi ý cho người mới. Giờ đây thêm vấn đề đối tượng mới chưa được người dùng nào đánh giá, hệ thống cũng không đủ dữ liệu để tính toán cho ra những lời gợi ý (có thể thấy rõ khó khăn này ở bước tính độ tương tự giữa hai đối tượng trong phương pháp dựa trên đối tượng vừa được nói đến ở phần trên). Và một vấn đề quan trọng khác hệ thống khuyến nghị bằng cách đánh giá độ tương đồng phải đối mặt là độ thưa thớt của dữ liệu đánh giá. Những đối tượng nổi bật được quan tâm sẽ có nhiều chỉ số đánh giá từ nhiều người dùng hơn. Trong khi đó, có một vài đối tượng hiếm khi được người dùng chú ý, dẫn đến không có dữ liệu đánh giá cho các đối tượng này. Ví dụ như trong hệ thống khuyến nghị phim, những phim ăn khách sẽ được nhiều người quan tâm đánh giá, cũng có nhiều phim ít được quan tâm, chỉ nhận được rất ít đánh giá từ người dùng. Điều này dẫn đến khi khuyến nghị, những phim ít được quan tâm cũng sẽ hiếm khi được gợi ý, mặc dù chúng có thể rất phù hợp với sở thích của một số ít đối tượng người dùng nào đó.

2.1.3. Hệ thống khuyến nghị lai:

Để hạn chế những khiếm khuyết trên, người ta đã nghiên cứu tìm cách kết hợp phương pháp dựa trên nội dung với phương pháp đánh giá độ tương đồng với nhau, dẫn đến sự ra đời của hệ thống khuyến nghị lai. Ví dụ như lập trình cho hệ thống học và lưu giữ những thông tin cá nhân người dùng bằng những phương pháp rút trích, phân tích nội dung, sau đó tiến hành so sánh những thông tin cá nhân đó để xác định được những người dùng có tính cách, sở thích tương tự nhau để đưa ra gợi ý bằng phương pháp đánh giá tương đồng. Ví dụ như tùy vào độ tuổi, giới tính, sở thích phim ảnh khác nhau, một người có xu hướng xem những loại phim khác nhau. Khi tìm được những người có độ tuổi, giới tính, sở thích tương tự, thì khả năng họ

có cùng xu hướng xem những thể loại phim giống nhau là rất cao. Một số cách kết hợp khác như triển khai hai hệ thống riêng biệt, một dùng phương pháp dựa trên nội dung, một dùng phương pháp đánh giá độ tương đồng, chỉ số dự đoán sau cùng được đưa ra sau khi phân tích kết hợp chỉ số của hai hệ thống riêng biệt. Hoặc tùy từng thời điểm mà chọn dùng hệ thống này hay hệ thống kia tùy thuộc vào chất lượng thông tin gợi ý của hệ thống nào tốt hơn.

Có những nghiên cứu đưa ra nhận xét rằng hệ thống khuyến nghị lai sẽ cho kết quả gợi ý tốt hơn so với từng hệ thống khuyến nghị riêng lẻ dùng phương pháp dựa trên nội dung hoặc phương pháp đánh giá độ tương đồng. [1],[2]

2.1.4. Đánh giá chung về mô hình khuyến nghị hai chiều:

Nhìn chung, những phương pháp khuyến nghị hai chiều vừa được trình bày trên đây có một vài nét đặc thù và khó khăn chung sau đây (dựa theo khảo sát từ tài liệu [2]):

- *Sự thưa thớt về dữ liệu:* do không phải lúc nào tất cả các đối tượng cũng được người dùng cung cấp chỉ số đánh giá một cách đầy đủ nên ma trận hai chiều (*Users* x *Items*) như đã nhắc đến ở những phần trên thường bị thưa thớt dữ liệu. Điều này ảnh hưởng đến tính chính xác của kết quả dự đoán. Như hai trường hợp "người dùng mới" và "đối tượng mới", hệ thống không đủ dữ liệu ban đầu để đưa ra gợi ý. Cũng không phải tất cả người dùng đều được hệ thống gợi ý vì đôi khi có những người sở thích của họ thuộc loại đặc biệt, hệ thống không tìm được những người tương tự để tiến hành gợi ý. Hoặc hai người tuy có cùng sở thích nhưng có thể họ không cùng đánh giá trên những đối tượng chung, hệ thống cũng không thể nhận ra họ là những người dùng tương tự nhau. Để hạn chế, khắc phục vấn đề này, có những nghiên cứu về những kỹ thuật thu giảm số chiều như Singular Value Decomposition (SVD), Latent Semantic Indexing (LSI), content-boosted CF, Tree Augmented Naïve Bayes Optimized By Extended Logistic Regression (TAN-ELR), Maximum margin Matrix Factorizations (MMMF) …
- *Vấn đề mở rộng dữ liệu:* khi số lượng người dùng và đối tượng tăng cao, đó sẽ trở thành vấn đề nghiêm trọng vì không gian khuyến nghị (*Users* x *Items*) sẽ trở nên khổng lồ từ đó số lượng tính toán trên không gian đó cũng sẽ tăng lên rất nhiều. Trong trường hợp này, những phương pháp thu giảm số chiều như SVD cũng giúp

hạn chế được vấn đề, hay thuật toán Incremental SVD được cải tiến từ SVD. Hay thuật toán Item-based Pearson, gom cụm người dùng …

❖ *Sự tương tự, trùng lặp:* có những đối tượng là hoàn toàn tương tự nhau hoặc rất giống nhau nhưng lại được đặt tên khác nhau. Hệ thống không nhận ra điều này, xem những đối tượng đó như những đối tượng riêng biệt khác nhau. Ví dụ như *"HCM city", "Thành Phố Hồ Chí Minh", "TP.HCM"*. Thuật toán Latent Semantic Indexing (LSI) phần nào giải quyết được vấn đề này.

❖ *Vấn đề những người dùng cá biệt:* thực tế có những người ý kiến của họ không hoàn toàn đồng ý hoặc bất đồng ý với bất cứ nhóm người dùng nào khác. Vậy nên trong hệ thống khuyến nghị, vai trò của những người này sẽ không quan trọng. Để giải quyết vấn đề này, người ta đưa ra một phương pháp khuyến nghị lai giữa phương pháp dựa trên nội dung và phương pháp đánh giá độ tương đồng.

❖ *Vấn đề cố tình gian lận đánh giá:* do ai cũng tham gia đánh giá được, sẽ có những người cố tình đánh giá tốt cho đối tượng của họ, hoặc cố tình đánh giá xấu cho những đối tượng của đối thủ cạnh tranh. Có nghiên cứu cho rằng thuật toán khuyến nghị dựa trên đối tượng (item-based) ít bị ảnh hưởng hơn thuật toán khuyến nghị dựa trên người dùng (user-based). Cũng có những nghiên cứu đưa ra mô hình giúp phát hiện những gian lận đánh giá này.

❖ Và còn một vài vấn đề khác như: sự riêng tư của người dùng, nhiễu dữ liệu, sự giải thích đến cho người dùng tại sao hệ thống lại gợi ý cho họ cái này mà không phải là cái khác … đang được tiếp tục nghiên cứu trên thế giới.

2.2. Mô hình khuyến nghị đa chiều:

Trên đây là cái nhìn tổng quan về các hệ thống khuyến nghị truyền thống. Ta có thể nhận thấy điểm chung ở các hệ thống trên là chúng chỉ quan tâm đến người dùng và đối tượng (chỉ hai chiều) chứ không quan tâm đến những thông tin ngữ cảnh bên ngoài. Có nhiều cách định nghĩa khái niệm thông tin ngữ cảnh. Định nghĩa được dùng nhiều nhất và phù hợp nhất trong tình huống này là của Dey (2001): *"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 đến sự tương tác giữa người dùng và ứng dụng, bao gồm cả bản thân người dùng và ứng dụng đó."* Ví dụ dễ hiểu hơn, thời gian, nơi chốn, thời tiết, tâm trạng … là những thông tin ngữ cảnh. Chúng có thể ảnh hưởng đến các chỉ số đánh giá của người

dùng đối với đối tượng, từ đó kéo theo sự ảnh hưởng của những gợi ý trong hệ thống khuyến nghị. Từ nhu cầu thực tế đó, dẫn đến sự ra đời của các hệ thống khuyến nghị đa chiều (ngoài hai chiều là người dùng và đối tượng, mở rộng thêm các chiều khác được quan tâm như các chiều biểu thị các thông tin của điều kiện ngữ cảnh).

Trong hệ thống khuyến nghị hai chiều, ta có hàm R dùng để dự đoán các chỉ số đánh giá chưa biết như sau:

$$R: Users \text{ x } Items \rightarrow Ratings$$

Giờ đây, với hệ thống khuyến nghị đa chiều, hàm R được bổ sung thêm thông tin ngữ cảnh và trở thành:

$$R: Users \text{ x } Items \text{ x } Contexts \rightarrow Ratings$$

Ví dụ trong hệ thống khuyến nghị du lịch, ta có người dùng là những người cần được hệ thống gợi ý những điểm du lịch, đối tượng là những điểm du lịch, và ngữ cảnh là thời gian đi (buổi sáng/buổi tối, các tháng, các mùa), bạn đồng hành (đi một mình, đi với bạn trai bạn gái, đi với gia đình, trẻ nhỏ …). Khi đó những chỉ số đánh giá cho một điểm du lịch bởi một người sẽ phụ thuộc vào những ngữ cảnh đó. Ví dụ vào buổi tối mùa thu đi cùng gia đình thì địa điểm *ABC* là hấp dẫn nhất.

Để dễ hình dung khái niệm đa chiều, người ta thường dùng mô hình dữ liệu đa chiều dựa trên OLAP [1]. Giả sử ta có các chiều là $D_1, D_2, D_3, ..., D_n$. Trong đó có hai chiều như đã biết là chiều *"người dùng"* và chiều *"đối tượng"*. Còn lại là các chiều *"ngữ cảnh"*. Mỗi chiều là một tập con của một tập hợp tích Descartes gồm nhiều thuộc tính. $D_i \subseteq A_{i1} \text{ x } A_{i2} \text{ x } ... \text{ x } A_{ik}$, trong đó mỗi thuộc tính A_{ik} định nghĩa một miền giá trị. Thêm nữa, một hoặc nhiều thuộc tính tạo thành một khóa để phân biệt duy nhất. Trong một số trường hợp, một chiều có để được định nghĩa bằng một thuộc tính đơn lẻ (khi đó $k = 1$ trong A_{ik}).

Ví dụ ta có không gian khuyến nghị ba chiều là:

Người dùng x Đối tượng x Thời gian

Trong đó:

Người dùng ⊆ Tên người dùng x Địa chỉ x Thu nhập x Tuổi.

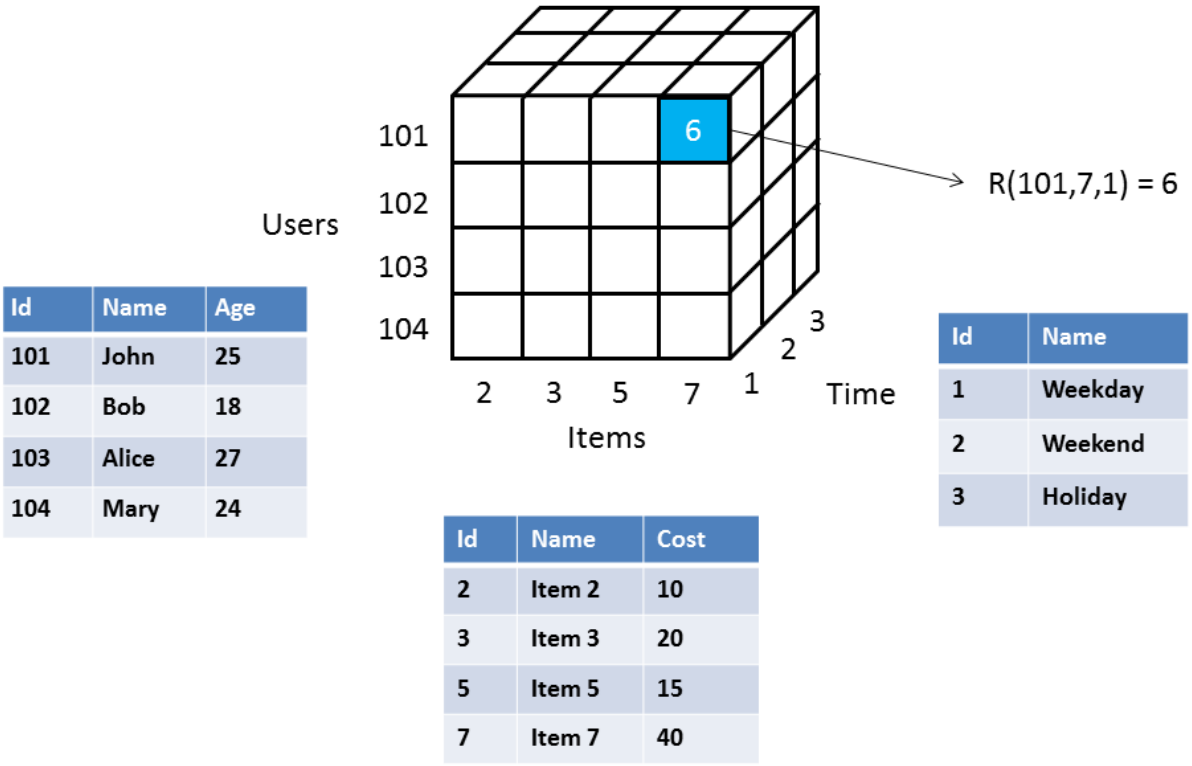
Đối tượng ⊆ Tên đối tượng x Thể loại x Giá cả

Thời gian ⊆ Năm x Tháng x Ngày

Khi đó hàm *R: Users x Items x Contexts → Ratings* sẽ trở thành

R: Users x Items x Times → Ratings

có ý nghĩa rằng một người dùng *u* ∈ *Users* thích đối tượng *i* ∈ *Items* vào thời điểm *t* ∈ *Times* với mức độ thích thể hiện bằng một chỉ số đánh giá nào đó.



Id	Name	Age
101	John	25
102	Bob	18
103	Alice	27
104	Mary	24

Id	Name	Cost
2	Item 2	10
3	Item 3	20
5	Item 5	15
7	Item 7	40

Id	Name
1	Weekday
2	Weekend
3	Holiday

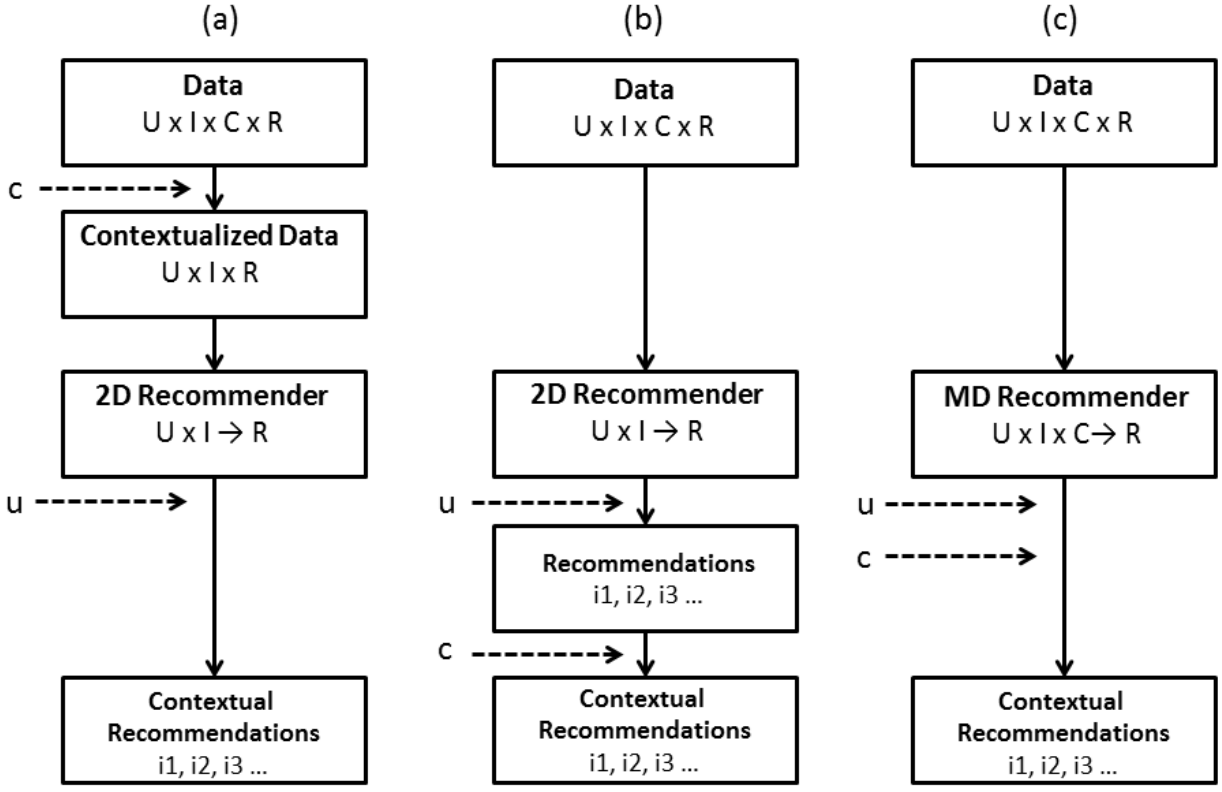
Hình 2.1: Ví dụ không gian khuyến nghị ba chiều.

Không gian khuyến nghị ba chiều có thể được mô tả trong khối lập phương ở hình trên. Ô tô đậm cho biết chỉ số *R(101,7,1) = 6* có ý nghĩa là người dùng có mã số *101* đánh giá đối tượng có mã số *7* trong điều kiện thời gian có mã số *1* với chỉ số đánh giá là *6*. Trong khối lập phương trên, không phải ô nào cũng có giá trị do người dùng chưa tiến hành đánh giá. Mục tiêu của hệ thống khuyến nghị là dự đoán giá trị tại những ô còn thiếu đó, từ đó đưa ra lời gợi ý đến với người dùng. Cũng cần xem xét rằng không phải tất cả các điều kiện ngữ cảnh đều cần thiết cho mục đích gợi ý. Ví dụ trường hợp gợi ý người dùng mua sách. Xét các điều kiện ngữ cảnh sau: mục đích mua sách (để giải trí, để học …), thời gian sẽ đọc quyển sách đó (đầu tuần, cuối tuần …), nơi sẽ đọc sách (ở trường, ở nhà, trên máy bay …), giá cổ phiếu ở thị trường chứng khoán vào thời điểm mua sách. Ở đây, có thể thấy giá cổ phiếu ít ảnh hưởng đến quyết định mua sách của người dùng hơn là mục đích mua sách.

Trên thế giới đã có những nghiên cứu về vấn đề xây dựng hệ thống khuyến nghị có điều kiện ngữ cảnh kèm theo. Hiện tại chúng được phân thành hai nhóm:

một là dùng điều kiện ngữ cảnh rồi tiến hành truy vấn, tìm kiếm những nội dung phù hợp cho việc gợi ý, hai là gợi ý dựa vào suy luận và dự đoán những sở thích của người dùng (có liên quan đến điều kiện ngữ cảnh, sở thích bị điều kiện ngữ cảnh tác động). Nhóm một được sử dụng rộng rãi trong các hệ thống khuyến nghị du lịch. Những hệ thống này thường dùng những điều kiện ngữ cảnh hiện tại được cung cấp trực tiếp từ người dùng (sở thích, tâm trạng …) hoặc từ môi trường (thời gian, thời tiết, vị trí hiện tại …) sau đó truy vấn, tìm kiếm những nguồn tài nguyên thích hợp nhất để đưa ra gợi ý. Ví dụ như hệ thống sẽ chỉ gợi ý những nhà hàng gần nhất với vị trí hiện tại của người dùng. Còn với nhóm hai, người ta dùng kỹ thuật để mô hình hóa và học những thói quen, sở thích của người dùng bằng cách theo dõi sự tương tác giữa người dùng với hệ thống hoặc suy ra từ những đánh giá của người dùng đối với những đối tượng trước đây. Nghĩa là người dùng không cần cung cấp thông tin ngữ cảnh một cách trực tiếp mà hệ thống sẽ tự động thu thập, phân tích và suy luận rồi đưa ra lời gợi ý phù hợp nhất với người dùng.

Vậy những thông tin ngữ cảnh sẽ được sử dụng trong quá trình đưa ra gợi ý vào lúc nào? Hình dưới đây mô tả ba trường hợp:



Hình 2.2: Thông tin ngữ cảnh trong hệ thống khuyến nghị. [1]

Trường hợp a: điều kiện ngữ cảnh được sử dụng ở pha chọn lọc không gian dữ liệu hệ thống dùng để gợi ý. Dữ liệu ban đầu gồm người dùng, đối tượng, ngữ cảnh và những chỉ số đánh giá tương ứng. Điều kiện ngữ cảnh hiện tại *c* của người dùng *u* được dùng để chọn ra tập dữ liệu có liên quan đến đúng ngữ cảnh đó. Như vậy, sau khi chọn, ta có thể bỏ qua điều kiện ngữ cảnh *c*. Bài toán trở về bài toán khuyến nghị hai chiều. Từ đây, ta có thể dùng bất cứ thuật toán khuyến nghị hai chiều nào để tiến hành dự đoán các chỉ số đánh giá.

Ví dụ cho trường hợp này là hướng tiếp cận dựa trên việc thu giảm số chiều (reduction-based, ở đây là chiều ngữ cảnh). Lợi ích của việc này là có thể tái sử dụng tất cả những kỹ thuật khuyến nghị hai chiều sau khi chiều ngữ cảnh được thu giảm. Để dễ hình dung, ta tiếp tục ví dụ gợi ý phim với không gian gợi ý (*Users x Items x Times*) gồm người dùng, đối tượng và thời gian. Quá trình thu giảm số chiều được minh họa như sau:

$$\forall(u,i,t) \in U \text{ x } I \text{ x } T, R^{D}_{User \text{ x } Item \text{ x } Time}(u,i,t) = R^{D[Time=t](User,Item,Rating)}_{User \text{ x } Item}(u,i)$$

Ở đây, cần gợi ý cho người dùng *u* những bộ phim *i* mà người đó có thể thích xem vào thời điểm *t*. Trước hết, hệ thống tiến hành chọn ra tập những chỉ số đánh giá có thời điểm đánh giá là *t*. Khi đó, số chiều D của không gian khuyến nghị sẽ được thu giảm chỉ còn người dùng, đối tượng với các chỉ số đánh giá tương ứng. Từ đây, sẽ tiến hành các thuật toán khuyến nghị hai chiều *(Users x Items)* để đưa ra gợi ý.

Trường hợp b: điều kiện ngữ cảnh được sử dụng ở pha sau khi đã có lời gợi ý. Ban đầu, vẫn là không gian dữ liệu có kèm thông tin ngữ cảnh. Người ta phớt lờ đi xem như những dữ liệu đánh giá đó không bị ảnh hưởng bởi những thông tin ngữ cảnh. Sau đó dùng những thuật toán khuyến nghị hai chiều để tiến hành dự đoán những chỉ số đánh giá và gợi ý trên toàn bộ tập dữ liệu ban đầu. Sau khi có kết quả gợi ý cho người dùng *u*, họ lọc ra những lời gợi ý phù hợp với điều kiện ngữ cảnh *c* rồi mới chuyển đến người dùng.

Ví dụ tương tự như với trường hợp a. Hệ thống dự đoán những bộ phim người dùng có thể thích mà không quan tâm ngữ cảnh. Sau khi có danh sách các bộ phim đó, hệ thống sẽ dùng những điều kiện ngữ cảnh để lọc lại danh sách phim sao cho phù hợp. Ví dụ nếu hệ thống biết rằng người dùng *u* chỉ thích xem phim kinh dị vào những ngày cuối tuần thì trong danh sách phim trên, hệ thống sẽ chỉ chọn ra những

phim nào thuộc thể loại kinh dị và được những người dùng quan tâm xem lúc cuối tuần.

Trường hợp c: những điều kiện ngữ cảnh được dùng một cách trực tiếp trong kỹ thuật mô hình hóa ở ngay pha dự đoán chỉ số đánh giá. Sau đó đưa ra những lời gợi ý. Những thuật toán được dùng ở đây nhìn chung phức tạp hơn nhiều so với những thuật toán khuyến nghị hai chiều truyền thống. Chúng cũng được phân thành hai nhóm là nhóm dựa trên kinh nghiệm, và nhóm dựa trên mô hình. Người ta nghiên cứu và tính toán những trọng số giữa những chỉ số dự đoán trong không gian đa chiều thay vì chỉ tính toán độ tương tự giữa người dùng với người dùng, đối tượng với đối tượng. Hoặc xây dựng những mô hình tính toán hồi quy tuyến tính phục vụ cho việc dự đoán những chỉ số đánh giá còn thiếu [1].

Do tính chất phức tạp cũng như những giới hạn trong quá trình thực hiện khóa luận, nhóm tác giả sẽ không đi vào chi tiết những kỹ thuật này. Phần khảo sát sơ bộ những hệ thống, những kỹ thuật khuyến nghị dừng lại tại đây. Chương tiếp theo sẽ là phần trình bày chi tiết thuật toán được nhóm tác giả nghiên cứu và triển khai trong hệ thống khuyến nghị du lịch ở khóa luận này.

CHƯƠNG 3.
TÌM HIỂU PHƯƠNG PHÁP THU GIẢM SỐ CHIỀU KẾT HỢP MÔ HÌNH HỒI QUY TUYẾN TÍNH

---oOo---

Sau khi tiến hành khảo sát các phương pháp khuyến nghị ở những phần trên, ở phần này nhóm tác giả sẽ trình bày chi tiết về phương pháp sẽ được hiện thực trong khóa luận này. Chương này gồm có 3 phần: phần 1 và 2 là phương pháp thu giảm số chiều (reduction-based), phần 3 sẽ trình bày về mô hình hồi qui được sử dụng kết hợp để dự đoán đánh giá người dùng.

3.1. Phương pháp thu giảm số chiều:

Ý tưởng chính của phương pháp được nhóm tác giả lựa chọn là đưa bài toán đa chiều về bài toán hai chiều *(Users* x *Items)* đã có lời giải. Do đó một trong những ưu điểm của phương pháp này là có thể kết hợp với bất kỳ một phương pháp khuyến nghị hai chiều nào đã trình bày ở chương 2 sau khi bước giảm chiều hoàn tất. Để hiểu phương pháp này, chúng ta quay lại với bài toán khuyến nghị hai chiều với ví dụ sau:

$$R^{D}_{User \times Item} : U \times I \rightarrow Rating$$

Với hàm dự đoán đánh giá trên, ta có *D* là tập chứa bộ dữ liệu <*user, item, rating*> cho mỗi đánh giá của từng người dùng với một đối tượng cụ thể, từ đó có thể dự đoán bất kỳ một đánh giá nào. Tương tự, khi thêm một chiều mới là thời gian thì công thức dự đoán sẽ là:

$$R^{D}_{User \times Item \times Time} : U \times I \times T \rightarrow Rating$$

Ở đây *D* là tập chứa bộ dữ liệu <*user, item, time, rating*> cho mỗi đánh giá cụ thể. Hàm dự đoán đánh giá trong không gian ba chiều trên có thể biểu diễn thông qua hàm dự đoán hai chiều như sau:

$$\forall (u, i, t) \in U \times I \times T,$$

$$R^{D}_{User \times Item \times Time}(u, i, t) = R^{D[Time=t](User,Item,Rating)}_{User \times Item}(u, c)$$

Trong đó *D[Time = t](User, Item, Rating)* là tập dữ liệu đánh giá trích xuất từ tập *D* bằng cách chọn ra những dữ liệu mà chiều *Time* có giá trị là *t* và sau đó chỉ giữ lại trường *User*, *Item*, và *Rating*. Nói cách khác, nếu ta biểu diễn tập dữ liệu *D*

trong cơ sở dữ liệu quan hệ, thì *D[Time = t](User, Item, Rating)* là tập dữ liệu chọn lọc từ *D* bằng hai phép toán quan hệ: phép chọn trước và phép chiếu sau.

Cần lưu ý trong nhiều trường hợp thì tập *D[Time = t](User, Item, Rating)* có thể không có đủ dữ liệu đánh giá để dự đoán bằng thuật toán dự đoán hai chiều. Do đó, phương pháp tổng quát để thu giảm số trong hệ thống khuyến nghị đa chiều có thể sẽ không sử dụng chính xác ngữ cảnh *t* để dự đoán đánh giá *r(u, i, t)* mà thay vào đó là tập ngữ cảnh S_t *(contextual segment)* chứa các ngữ cảnh cao hơn hoặc bằng với *t*. Ví dụ, nếu muốn dự đoán số điểm mà John sẽ đánh giá cho nhà thờ Đức Bà khi đi vào ngày chủ nhật, ta có thể sẽ không lựa chọn *t = "Chủ nhật"* để tính toán mà thay vào đó là *t = "Cuối tuần"* tùy thuộc vào sự đa dạng của dữ liệu. Một cách tổng quát, ta có công thức:

$$R^{D}_{User\times Item\times Time}(u,i,t) = R^{D[Time\in S_t](User,Item,AGGR(Rating))}_{User\times Item}(u,i)$$

Trong công thức trên ta sử dụng hàm *AGGR(Rating)* bởi vì một người dùng có thể có nhiều đánh giá cho một đối tượng ứng với những ngữ cảnh khác nhau, ở đây là trong những thời điểm khác nhau. Do đó, cần phải kết hợp các đánh giá này lại bằng một hàm kết tập (thường là tính trung bình cộng) khi thu giảm số chiều trong không gian khuyến nghị đa chiều.

Hướng tiếp cận thu giảm ba chiều ở trên về hai chiều có thể được mở rộng thành một phương thức tổng quát để thu giảm không gian khuyến nghị n chiều thành m chiều (với m < n). Tuy nhiên trong đề tài này nhóm tác giả sử dụng *m = 2* để dễ dàng áp dụng các thuật toán khuyến nghị hai chiều sẵn có.

Ta qui ước gọi hai chiều cơ bản là người dùng và đối tượng, các chiều còn lại gọi là chiều ngữ cảnh như thời gian, bạn đồng hành, thời tiết, tâm trạng, … Các tập dữ liệu đánh giá được trích xuất theo các giá trị của ngữ cảnh là một phân khúc dữ liệu (segment), ví dụ ta có phân khúc dữ liệu *"Cuối tuần"* trích xuất từ tập dữ liệu đánh giá *D* sẽ chứa tất cả các đánh giá cho những địa điểm được đi vào cuối tuần:

$$Weekend = \{d \in D | (d.Time.weekend = true)\}$$

Tương tự phân khúc dữ liệu *"Cuối tuần – Bạn bè"* sẽ chứa các đánh giá cho những địa điểm được đi vào cuối tuần với bạn bè:

$$Weekend - Friends = d \in D \mid (d.Time.weekend = true) \cap (d.Companion = Friends)\}$$

Ta minh họa cách thức hoạt động của phương pháp thu giảm số chiều bằng ví dụ sau: giả sử ta muốn dự đoán đánh giá của John cho nhà thờ Đức Bà vào buổi sáng. Để tính $R^{D}_{User\times Item\times Time}(John, Duc\ Ba\ church, Morning)$, đầu tiên thuật toán sẽ loại bỏ chiều thời gian bằng cách chọn ra tập dữ liệu đánh giá mà thời gian là vào buổi sáng từ tập dữ liệu gốc *D*. Như vậy bài toán bây giờ được chuẩn hóa về bài toán hai chiều *(Users x Items)* với tập dữ liệu là các đánh giá vào buổi sáng. Sau đó, có thể áp dụng bất kỳ thuật toán dự đoán hai chiều nào như đã trình bày ở chương 2 (phương pháp nhóm tác giả lựa chọn là sử dụng mô hình hồi qui sẽ được trình bày chi tiết ở những phần sau). Qua đó cho thấy ý tưởng của phương pháp thu giảm số chiều này khá đơn giản: ví dụ nếu ta muốn dự đoán một đánh giá vào buổi sáng của một người dùng cho một đối tượng cụ thể, chỉ cần quan tâm tới những đánh giá đã có trước đó với thời gian vào buổi sáng.

Vấn đề tiếp theo cần được xem xét là liệu mô hình cục bộ này (mô hình khuyến nghị trên miền dữ liệu đã giới hạn theo ngữ cảnh) có tốt hơn mô hình toàn cục (mô hình khuyến nghị mà tất cả các ngữ cảnh được mặc nhiên bỏ qua không xét đến) hay không? Hãy cùng xem xét ví dụ sau:

Ta có không gian khuyến nghị ba chiều $User \times Item \times X$, với X là chiều ngữ cảnh có hai giá trị là h và t. Gọi các giá trị đánh giá khi $X = t$ là n_t và khi $X = h$ là n_h. Giả định rằng tất cả người dùng đều có cùng sở thích như nhau (cùng thích những đối tượng với mức độ như nhau nghĩa là n_t hoặc n_h tùy ngữ cảnh) và $n_t \neq n_h$.

Với giả định trên phương pháp thu giảm số chiều luôn dự đoán các đánh giá chưa biết một cách chính xác. Có thể dễ dàng thấy điều này khi ta áp dụng phương pháp đánh giá hai chiều truyền thống, ví dụ nếu ta dùng phương pháp khuyến nghị dựa trên kinh nghiệm như sau:

$$r_{u,i} = k \sum_{u' \in U} sim(u, u') \times r_{u',i}$$

Theo đó, nếu ta sử dụng các thông tin ngữ cảnh về đối tượng đã được đánh giá, trong trường hợp phân khúc dữ liệu $X = t$, tất cả các đánh giá $r_{u',i}$ đều bằng n_t, và do đó $r_{u,i} = n_t$ bất chấp độ đo tương quan giữa các người dùng. Tương tự với phân khúc dữ liệu $X = h$, ta luôn có $r_{u,i} = n_h$. Do đó tất cả dự đoán đều chính xác với thực tế khi áp dụng phương pháp thu giảm số chiều (mô hình cục bộ).

Ngược lại, trong mô hình toàn cục, khi áp dụng thuật toán khuyến nghị hai chiều sẽ sử dụng trộn lẫn các đánh giá n_t và n_h sẽ không chính xác. Tùy thuộc vào sự phân phối của các đánh giá đã có và trên một đánh giá cần dự đoán, sai số có thể chạy từ 0 (khi chỉ những dữ liệu đánh giá đúng được lựa chọn để tính toán) đến $|n_t - n_h|$ (khi chỉ những dữ liệu đánh giá sai được lựa chọn để tính toán).

Lý do mà phương pháp khuyến nghị thu giảm số chiều chính xác hơn phương pháp khuyến nghị truyền thống là trong ví dụ trên chiều X chia miền dữ liệu ra làm hai phần riêng biệt (nhóm các đánh giá cho $X = t$ là n_t và nhóm các đánh giá cho $X = h$ là n_h với $n_t \neq n_h$). Tuy nhiên nếu $n_t = n_h$ thì chiều X sẽ không còn giá trị trong mục đích khuyến nghị (việc đánh giá trong hai trường hợp $X = t$ và $X = h$ là như nhau, nghĩa là không ảnh hưởng tới quyết định của người dùng). Trong trường hợp này, phương pháp khuyến nghị thu giảm số chiều có độ chính xác không hơn phương pháp khuyến nghị truyền thống vì việc thu giảm số chiều ở đây là không có ý nghĩa. Trong trường hợp tổng quát, $X \in \{X_1, X_2, \dots, X_n\}$ thì khả năng $X_1 = X_2 = \dots = X_n$ là rất thấp, do đó phương pháp khuyến nghị thu giảm số chiều sẽ tốt hơn phương pháp khuyến nghị truyền thống trong đại đa số các trường hợp.

Bên cạnh đó ta có thể nhận thấy khả năng xảy ra trường hợp việc dự đoán trên một vài phân khúc dữ liệu bằng phương pháp thu giảm số chiều sẽ có kết quả thấp hơn phương pháp truyền thống hoặc thậm chí không thể dự đoán vì số lượng dữ liệu đánh giá trong tập phân khúc dữ liệu sau khi thu giảm là quá ít (dữ liệu thưa thớt). Một trong những giải pháp cho vấn đề này là kết hợp phương pháp khuyến nghị thu giảm số chiều với phương pháp khuyến nghị truyền thống sẽ được trình bày ở phần tiếp theo.

3.2. Kết hợp phương pháp khuyến nghị thu giảm số chiều và phương pháp khuyến nghị truyền thống:

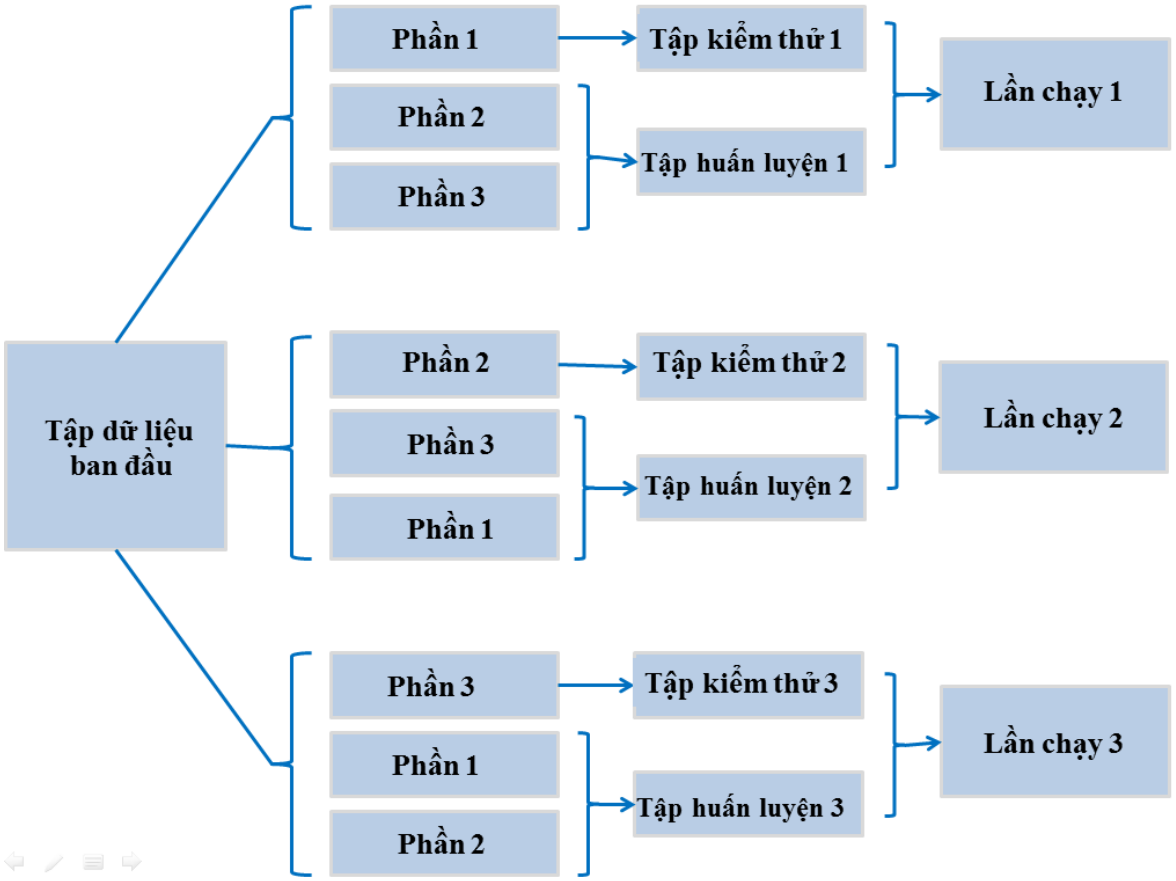
Trước khi đi vào phương pháp kết hợp này nhóm tác giả sẽ trình bày một vài khái niệm tiên quyết.

Để kết hợp hai phương pháp, đầu tiên cần những thước đo độ hiệu quả để so sánh phương pháp nào tốt hơn phương pháp nào trên những phân khúc dữ liệu khác nhau. Một vài thước đo được sử dụng rộng rãi như: độ sai lệch trung bình tuyệt đối (MAE – Mean Absolute Error), độ sai lệch trung bình bình phương (MSE – Mean Squared Error), sự tương quan giữa dự đoán và thực tế (F-measure).

Trong phần này, nhóm tác giả sử dụng một hàm đo độ hiệu quả trừu tượng là $\mu_{A,X}(Y)$ sử dụng cho một thuật toán khuyến nghị A với tập dữ liệu huấn luyện X và đánh giá cho tập dữ liệu Y (có sẵn), với $X \cap Y = \emptyset$. Với mỗi $d \in Y$, gọi $d.R$ là giá trị đánh giá thực cho giá trị d (đang cần được dự đoán), $d.R_{A,X}$ là đánh giá được dự đoán bằng thuật toán A trên tập dữ liệu huấn luyện X cho d. Hàm $\mu_{A,X}(Y)$ được định nghĩa như một hàm thống kê trên hai tập dữ liệu đánh giá $\{d.R | d \in Y\}$ và $\{d.R_{A,X} | d \in Y\}$. Ví dụ độ sai lệch trung bình tuyệt đối được định nghĩa như sau:

$$\mu_{A,X}(Y) = \frac{1}{|Y|} \sum_{d \in Y} |d.R_{A,X} - d.R|$$

Như đã đề cập ở trên, trong $\mu_{A,X}(Y)$ thì $X \cap Y = \emptyset$, nói cách khác tập dữ liệu huấn luyện và tập dữ liệu đánh giá phải riêng biệt với nhau. Trong các nghiên cứu, kỹ thuật *N-fold Cross Validation* thường được sử dụng. Dữ liệu gốc sẽ được chia thành n phần bằng nhau, và quá trình huấn luyện / đánh giá được thực hiện lặp lại n lần. Tại mỗi lần huấn luyện / đánh giá, một phần dữ liệu dùng để đánh giá và *(n-1)* phần còn lại dùng để huấn luyện. Ví dụ sau đây cho $n = 3$, dữ liệu ban đầu chia làm ba phần, và có ba lần thực hiện quá trình huấn luyện / đánh giá. Lần thứ nhất, phần một làm tập đánh giá, hai phần còn lại làm tập huấn luyện. Lần thứ hai, phần hai làm tập đánh giá. Lần thứ ba, đến lượt phần ba làm tập đánh giá.



Hình 3.1: Kỹ thuật Cross Validation với n = 3.

Sau ba pha chạy như trên, ở mỗi pha sẽ tính toán lỗi (dựa trên số phần tử trong tập dữ liệu đánh giá có được phân lớp chính xác hay không). Từ đó cho ra chỉ số đánh giá độ chính xác trung bình cho tập dữ liệu huấn luyện ban đầu.

Áp dụng kỹ thuật Cross Validation trên tập dữ liệu đánh giá đã có T ta sẽ có các tập dữ liệu huấn luyện và đánh giá *Xi* và *Yi (i = 1, 2, ...)*, với $X_i \cap Y_i = \emptyset$ và $X_i \cup Y_i = T$. Ta có thể dự đoán đánh giá cho $d.R$ bằng cách tính trung bình cộng các giá trị dự đoán trên các tập huấn luyện như sau:

$$d.R_{A,T} = \frac{1}{|C|}\sum_{i \in C} d.R_{A,X_i}, \text{ với } C = \{i | d \in Y_i\}$$

Để đơn giản, ta ký hiệu độ đo hiệu quả của thuật toán A trên tập dữ liệu T là $\mu_{A,T}(T)$. Lưu ý ký hiệu này không có nghĩa là tập dữ liệu huấn luyện và kiểm thử là như nhau mà chúng phải đảm bảo điều kiện như đã trình bày ở trên.

Bây giờ, nhóm tác giả sẽ tiếp tục với phương pháp kết hợp giữa phương pháp thu giảm số chiều và phương pháp khuyến nghị truyền thống để giải quyết vấn đề đã nêu ở phần trước. Phương pháp này gồm có hai pha: đầu tiên là sử dụng các đánh giá của người dùng đã biết để chọn ra những phân khúc dữ liệu vượt trội so với

phương pháp truyền thống (áp dụng thuật toán trên toàn bộ dữ liệu). Sau đó, để dự đoán một đánh giá, cần chọn ra phân khúc dữ liệu phù hợp nhất với điều kiện ngữ cảnh và áp dụng thuật toán khuyến nghị hai chiều trên phân khúc dữ liệu này [11]. Cụ thể như sau:

- Pha thứ nhất được chạy *"offline"* gồm có ba bước:
  - Bước 1: Tìm tất cả các phân khúc dữ liệu với số lượng đánh giá lớn hơn một ngưỡng định sẵn.
  - Bước 2: Từ tập phân khúc dữ liệu *S* trên, chạy thuật toán *A* trên *S* và tính toán độ hiệu quả $\mu_{A,S}(S)$ với kỹ thuật lấy mẫu Cross Validation. Ta cũng chạy thuật toán *A* trên toàn bộ tập dữ liệu *T* và tính được độ hiệu quả $\mu_{A,T}(S)$ trên tập dữ liệu đánh giá *S*. Sau đó so sánh hai kết quả này để quyết định xem phương pháp nào hiệu quả hơn với cùng tập dữ liệu đánh giá *S* và chỉ giữ lại những phân khúc dữ liệu khi áp dụng phương pháp thu giảm số chiều đạt hiệu quả cao hơn này.
  - Bước 3: Loại bỏ những phân khúc dữ liệu có được sau bước 2 nếu trong tập phân khúc dữ liệu này cũng tồn tại một phân khúc dữ liệu *Q* tổng quát hơn nó và có độ hiệu quả cũng cao hơn.

Đầu vào:

T	Tập dữ liệu đánh giá trong không gian khuyến nghị đa chiều.
$R_{A,T}$	Hàm dự đoán đánh giá dựa trên thuật toán *A* và tập dữ liệu huấn luyện *T*.
μ	Hàm đo độ hiệu quả
N	Số đánh giá có sẵn nhỏ nhất cần có của một phân khúc dữ liệu hợp lệ

Đầu ra:

$\overline{SEGM(T)}$	Tập phân khúc dữ liệu mà bằng phương pháp thu giảm số chiều dựa trên thuật toán *A* có hiệu quả cao hơn thuật toán *A* thuần túy.

Thuật toán:

1. Tìm tập *SEGM(T)* là tập các phân khúc dữ liệu có số lượng đánh giá đã có lớn hơn hoặc bằng *N*.

2. Với mỗi phân khúc dữ liệu $S \in SEGM(T)$ tính $\mu_{A,S}(S)$ và $\mu_{A,T}(S)$, và chỉ giữ lại những phân khúc dữ liệu $S \in SEGM(T)$ mà $\mu_{A,S}(S)$ tốt hơn $\mu_{A,T}(S)$.
3. Với những phân khúc dữ liệu còn lại trong *SEGM(T)* sau bước trên, loại bỏ tất cả những phân khúc dữ liệu S nếu tồn tại một phân khúc dữ liệu Q sao cho $S \subset Q$ và $\mu_{A,Q}(Q)$ tốt hơn $\mu_{A,S}(S)$. Tập phân khúc dữ liệu được giữ lại sau cùng là $\overline{SEGM(T)}$ cần tìm.

- Pha thứ hai là bước chạy trực tuyến (online) để dự đoán một đánh giá của một người dùng cho một đối tượng với điều kiện ngữ cảnh cho trước. Gọi d là giá trị cần dự đoán, đầu tiên ta duyệt qua các phân khúc dữ liệu ứng viên $\overline{SEGM(T)} = \{S_1, S_2, \dots S_k\}$ theo thứ tự giảm dần độ hiệu quả, sau đó chọn ra phân khúc dữ liệu đầu tiên mà d thuộc về (theo ngữ cảnh). Nếu d không nằm trong bất cứ phân khúc dữ liệu ứng viên nào, ta sẽ sử dụng thuật toán A trên dữ liệu toàn cục T để dự đoán đánh giá cho $R_{A,T}(d)$. Ngược lại nếu tìm được phân khúc dữ liệu S_j phù hợp thì kết quả trả về sẽ là $R_{A,Sj}(d)$.

Đầu vào:

$\overline{SEGM(T)} = \{S_1, S_2, \dots S_k\}$	Tập phân khúc dữ liệu S_1 đến S_k được sắp xếp theo thứ tự giảm dần độ hiệu quả μ, nghĩa là $\mu_{A,S_1}(S_1) > \cdots > \mu_{A,S_k}(S_k)$.
d	Giá trị đánh giá d cần dự đoán.

Đầu ra:

$d.R$	Giá trị đánh giá được dự đoán cho d.

Thuật toán:

1. $j = 0$
2. $j = min_{i=1..k}\{i | d \in S_i\}$
3. Nếu $j = 0$ thì $d.R = R_{A,T}(d)$ //d không thuộc bất kỳ phân khúc dữ liệu S_i nào. Ngược lại thì $d.R = R_{A,S_j}(d)$

Với phương pháp kết hợp này, kỹ thuật thu giảm số chiều chỉ được sử dụng khi phân khúc dữ liệu sau khi thu giảm cho kết quả tốt hơn phương pháp hai chiều truyền thống. Ngược lại, nếu kết quả xấu hơn, sẽ quay về sử dụng phương pháp truyền thống. Vì vậy cách tiếp cận kết hợp kỹ thuật thu giảm số chiều với phương pháp truyền thống được mong đợi sẽ cho kết quả chính xác hơn hoặc bằng so với phương pháp truyền thống.

3.3. Mô hình hồi qui trong hệ thống khuyến nghị hai chiều

Trong phần này nhóm tác giả sẽ trình bày về thuật toán khuyến nghị sử dụng mô hình hồi qui áp dụng trong khóa luận, được tham khảo từ tài liệu [9].

Đánh giá của người dùng a cho đối tượng i dựa trên những đánh giá trước đó trên tập đối tượng I_a của chính người dùng đó có thể được tính gần đúng như sau:

$$p_{a,i} = \sum_{j \in I_a} w_{j,i} \times f_{j,i}(r_{aj}) \quad (1)$$

Trong đó $f_{j,i}$ là hàm dự đoán đánh giá cho đối tượng i dựa trên những đánh giá cho đối tượng j, và $w_{j,i}$ là trọng số tương ứng, $i, j = 1, \dots I, i \neq j$.

$f_{j,i}$ có thể được mô hình bằng hàm hồi qui tuyến tính như sau [9]:

$$f_{j,i}(x) = x\alpha_{j,i} + \beta_{j,i} \quad (2)$$

Với $\alpha_{j,i} = \sum_{u \in U_i \cap U_j} (r_{ui} - r_{*i})(r_{uj} - r_{*j}) \Big/ \sum_{u \in U_i \cap U_j} (r_{ui} - r_{uj})^2$

và $\beta_{j,i} = r_{*i} - \alpha_{j,i} r_{*j}$

Ta có thể tính độ sai lệch trung bình bình phương của hàm $f_{j,i}$ trên như sau [9]:

$$MSE_{j,i} = E_{U_i \cap U_j} \left\{ \left(r_{uj} - f_{ji}(r_{uj}) \right)^2 \right\} = \frac{1}{|U_i \cap U_j|} \sum_{u \in U_i \cap U_j} \left(r_{uj} - f_{ji}(r_{uj}) \right)^2 \quad (3)$$

Bài toán còn lại là tìm cách thức để tính toán trọng số $w_{j,i}$. Nếu giả định rằng $f_{j,i}$ và $f_{k,i}$ là những dự đoán khách quan cho đối tượng i thì độ tương quan $p^i_{j,k}$ giữa những dự đoán này có thể được tính như sau [9]:

$$p^{i}_{j,k} = \frac{\{\{C_i\}_{j,j} + \{C_i\}_{k,k} - E_{U_i \cap U_j \cap U_k}[(f_{j,i} - f_{k,i})^2]\}}{2\sqrt{\{C_i\}_{j,j}\{C_i\}_{k,k}}} \quad (4)$$

Trong đó $\{C_i\}_{j,j} = MSE_{j,i}$. Sau khi tính độ tương quan $p^{i}_{j,k}$ giữa các bộ đối tượng (i, j, k) như công thức trên, ta sẽ tính trung bình để có độ tương quan trung bình p_{AV}, từ đó có thể tính trọng số $w_{j,i}$ như sau:

$$w_{j,i} = \frac{1/\{C_i^*\}_{j,j}}{\sum_{k \in I_a} 1/\{C_i^*\}_{k,k}} \quad (5)$$

Với $j \in I_a$ và $\{C_i^*\}_{j,j} = MSE_{j,i} - p_{AV} \times \min_k(MSE_{k,i})$

Và $w_{j,i} = 0$ khi $j \notin I_a$.

Cuối cùng, ta có thuật toán dự đoán đánh giá như sau:

Pha huấn luyện

Với tập dữ liệu đánh giá D cho trước

1. Tinh toán *I(I-1)* dự đoán $f_{j,i}$ với $i, j = 1 \ldots I, i \neq j$ sử dụng công thức (2) và tính độ sai lệch $MSE_{j,i}$ tương ứng sử dụng công thức (3)
2. Đánh giá độ tương quan $p^{i}_{j,k}$ giữa các bộ đối tượng *(i, j, k)* bằng công thức (4) và sau đó lấy trung bình độ tương quan p_{AV}.

Pha dự đoán

Với mỗi người dùng a đang cần khuyến nghị với các đánh giá $r_{a,j}$ trước đây, $j \in I_a$

Với mỗi đối tượng $i \in I \backslash I_a$:

1. Tính gần đúng trọng số $w_{j,i}, j \in I_a$ sử dụng công thức (5).
2. Dự đoán đánh giá $p_{a,i}$ cho đối tượng i bằng công thức (1).

CHƯƠNG 4.
KHẢO SÁT CÁC KỸ THUẬT XÂY DỰNG ỨNG DỤNG

---oOo---

Ở chương 3, nhóm tác giả đã trình bày phần phương pháp được lựa chọn để giải quyết bài toán. Sang chương 4, nhóm tác giả sẽ đi sâu vào tìm hiểu những kỹ thuật quan trọng cần thiết cho việc hiện thực hóa hệ thống. Đầu tiên, nhóm tác giả sẽ tìm hiểu về cơ sở dữ liệu phục vụ cho việc cài đặt thuật toán khuyến nghị, tiếp đến sẽ khảo sát và lựa chọn nền tảng thiết bị di động, và cuối cùng là kỹ thuật truyền nhận thông tin giữa hệ thống và thiết bị di động.

4.1. Cơ sở dữ liệu hỗ trợ cài đặt thuật toán khuyến nghị:

Như đã nói hệ thống nhóm tác giả hướng tới là hệ thống khuyến nghị trong không gian đa chiều *(Users x Items x Contexts)* (người dùng, đối tượng, và các ngữ cảnh). Ngoài ra, G. Adomavicius [11], [12] đã xây dựng hệ thống hỗ trợ khuyến nghị trên các phân cấp của các chiều. Ví dụ hệ thống sẽ sử dụng các tập phân khúc dữ liệu "Mùa" (Xuân / Hạ / Thu / Đông) để tính toán khuyến nghị thay vì tập phân khúc dữ liệu "Giai đoạn trong ngày" (Sáng / Tối) khi không đủ dữ liệu. Ngoài ra hệ thống khuyến nghị còn có thể khai thác thông tin chi tiết của các chiều, chẳng hạn người dùng với các thông tin như tên, tuổi, giới tính, sở thích, … có thể được sử dụng để khuyến nghị tốt hơn. Tất cả các yêu cầu trên của hệ thống có thể được tích hợp bằng cách tiếp cận sử dụng kho dữ liệu (data warehouse) OLAP, bởi vì nó cung cấp khả năng tổ chức mô hình dữ liệu đa chiều cũng như hệ thống phân cấp. Trong hệ thống của mình, nhóm tác giả lựa chọn sử dụng nền tảng cơ sở dữ liệu OLAP của SQL Server vì khả năng triển khai dễ dàng và nhanh chóng trong thời gian giới hạn của khóa luận.

4.2. Hệ điều hành cho điện thoại thông minh:

4.2.1. Tại sao triển khai hệ thống khuyến nghị trên điện thoại?

Những hệ thống khuyến nghị thông thường được triển khai trên ứng dụng để bàn, nhất là ở các trang web trên Internet. Ví dụ như trang Amazon gợi ý những sản phẩm, trang Youtube gợi ý những bản nhạc đến người dùng … và rất nhiều trang khác nữa. Ở đây, sản phẩm cần được xây dựng trong khóa luận này hướng đến đối tượng người dùng là những người đi du lịch, di chuyển thường xuyên chứ không ở cố định một ví trí để lướt web tìm thông tin như đang ở nhà, hoặc các hàng quán

Internet. Trường hợp này, rõ ràng ứng dụng web không phát huy được hiệu quả. Cụ thể, nhóm tác giả sẽ không chọn web mà quyết định chọn điện thoại thông minh để triển khai ứng dụng vì những lý do sau:

- Người dùng luôn đem theo bên mình chiếc điện thoại. Vì vậy khi cần, ngay lập tức có thể sử dụng mọi lúc mọi nơi. Thay vì phải tìm đến nơi có máy vi tính và Internet mới có thể tra cứu thông tin, như thế thật bất tiện trong các chuyến du lịch.

- Trên điện thoại thông minh có hệ thống GPS với bản đồ giúp người dùng xác định vị trí, đường đi, cùng nhiều tiện ích khác. Vị trí người dùng cũng ảnh hưởng đến kết quả tìm kiếm thông tin. Ví dụ như khi muốn tìm những điểm du lịch gần nhất, tùy theo vị trí hiện tại của người dùng, kết quả trả về sẽ khác nhau. Với ứng dụng web trên máy vi tính, yếu tố vị trí người dùng không có ảnh hưởng.

- Khó đoán trước suy nghĩ và hành động của người dùng. Ở thời điểm này, điều kiện này, họ quyết định thế này. Ở thời điểm khác, điều kiện khác, họ thay đổi quyết định. Vì vậy, luôn mang bên mình chiếc điện thoại, người dùng sẽ được gợi ý kịp thời kịp lúc. Đặc biệt là khi đang trong chuyến du lịch, họ có thể "hỏi" chiếc điện thoại của mình, không phải mất nhiều thời gian liên lạc những người quen biết để có được những lời khuyên.

Bên cạnh những lợi thế trên, điện thoại cũng có những nhược điểm. Một trong số đó là giao diện người dùng. Điện thoại đa số có cỡ màn hình từ 3-4 inches, không phải màn hình rộng 19-21 inches như máy tính. Việc bố trí các chức năng, cách hiển thị thông trên một khung nhìn nhỏ cũng phải rất cẩn thận sao cho vừa đầy đủ cũng vừa không gây rối mắt cho người dùng. Thêm nữa phần cứng trên điện thoại không mạnh mẽ như trên máy tính cá nhân (về bộ nhớ, bộ vi xử lý, thời lượng pin …), nên khả năng xử lý cũng có phần hạn chế. Những kỹ thuật xử lý khó khăn phức tạp sẽ được thực hiện trên máy tính nào đó và trả kết quả về hiển thị trên điện thoại.

4.2.2. Chọn lựa giữa ứng dụng thuần và ứng dụng web trên điện thoại?

Khi phát triển một ứng dụng trên điện thoại, có hai hướng làm: một là phát triển ứng dụng thuần tương ứng với từng nền tảng hệ điều hành điện thoại được cung cấp, hai là phát triển ứng dụng trên nền tảng web được tối ưu về giao diện để thể hiện tốt trên các màn hình điện thoại kích cỡ nhỏ. Hai dạng ứng dụng đó tùy trường hợp mà sẽ được các nhà phát triển lựa chọn.

❖ *Ứng dụng thuần (native-app):* là những ứng dụng viết dựa hoàn toàn vào API gốc của hệ điều hành đó. Mỗi hệ điều hành có API đặc trưng. Ứng dụng viết trên Android không thể chạy trên Iphone và ngược lại.

❖ *Ứng dụng nền tảng web (web-app):* là những ứng dụng web (dùng HTML, CSS, Javasript …) chạy trên trình duyệt của điện thoại. Ứng dụng nền tảng web có thể chạy trên bất cứ hệ điều hành nào miễn là có trình duyệt và Internet.

Mỗi loại đều có ưu nhược điểm riêng, nhưng nhóm tác giả đã quyết định chọn ứng dụng thuần vì những lý do sau:

- ✓ Có thể can thiệp sâu vào những tính năng, phần cứng của điện thoại do API được thiết kế riêng biệt cho nền tảng điện thoại đó. Trên web, sẽ có những hạn chế nhất định.
- ✓ Ứng dụng có khả năng chạy nền khi cần thiết.
- ✓ Có thể sử dụng khi không có Internet.
- ✓ Trong tương lai, có thể dễ dàng thương mại hóa trên các cửa hàng ứng dụng (App Store).

4.2.3. Tại sao chọn Android?

Hiện nay trên thế giới, những nền tảng nổi bật nhất và chiếm phần lớn thị phần hệ điều hành cho điện thoại thông minh là iOS (của Apple), Android (của Goole) và Windows Phone (của Microsoft). Hệ điều hành iOS từ khi ra đời đã hướng đến mục tiêu phục vụ cho những nhu cầu thiên về giải trí trên điện thoại. Có thể thấy trên kho ứng dụng của Apple, đa phần là các trò chơi. Để lập trình được những ứng dụng chạy trên iOS, không phải lập trình viên nào cũng có điều kiện. Trước hết, cần phải có hệ điều hành MAC OS. Hệ điều hành này được cài sẵn trên các Macbook với giá bán không hề rẻ. Và IDE dùng để lập trình là Xcode cũng không phải miễn phí. Thời gian gần đây, hệ điều hành Windows Phone của Microsoft bắt đầu được phát triển. Hiện tại, tính năng của Windows Phone nhìn chung không hấp dẫn bằng iOS hay Android. Số lượng ứng dụng, số lượng lập trình viên tham gia phát triển, cũng như số lượng người dùng điện thoại thông minh nền tảng Windows Phone còn khá ít so với các nền tảng khác. Nhìn chung, nổi bật nhất trên thị trường điện thoại thông minh lúc này vẫn là Android. Nhóm tác giả có kèm một bảng so sánh các hệ điều hành điện thoại ở phần phụ lục B cuối quyển báo cáo này.

Android được biết đến như là một hệ điều hành mã nguồn mở trên điện thoại di động. Hiện nay, Android được sử dụng cả trên những thiết bị điện tử khác như: máy tính bảng, tivi, thiết bị giải trí đa phương tiện, ... Android hiện đang được phát triển bởi Google. Trước đây, Android được phát triển dựa trên nền tảng Linux bởi công ty liên hợp Android (sau đó được Google mua lại vào năm 2005). Các nhà phát triển viết ứng dụng cho Android dựa trên ngôn ngữ Java. Sự ra mắt của Android vào năm 2007 gắn với sự thành lập của liên minh thiết bị cầm tay mã nguồn mở nhằm mục đính tạo nên một chuẩn mở cho điện thoại di động trong tương lai. Phiên bản Android đầu tiên dành cho các dòng điện thoại thông minh là 1.5. Hiện nay, bản mới nhất là 4.0 - Ice Cream Sandwich.

Dưới đây là những thành phần cốt lõi của hệ điều hành Android: [13] [14]

- *Applications:* Khi bắt đầu cài đặt Android trên điện thoại di động, các ứng dụng cơ bản như email, SMS, lịch, bản đồ, trình duyệt, quản lý danh bạ … được tích hợp sẵn. Tất cả những ứng dụng khác có thể được xây dựng thêm bằng ngôn ngữ lập trình Java và cài đặt vào điện thoại.
- *Application Framework:* cho phép lập trình viên dễ dàng xây dựng những dụng mạnh mẽ có khả năng tái sử dụng cao. Những thành phần của ứng dụng này có thể được kế thừa để sử dụng hoặc phát triển thêm cho những ứng dụng khác. Những người lập trình có toàn quyền truy xuất, sử dụng những sức mạnh phần cứng của chiếc điện thoại trong lúc lập trình ứng dụng (GPS, bluetooth, WiFi, cảm biến gia tốc, la bàn …)
- *Libraries:* gồm một tập các thư viện C/C++ được viết sẵn hỗ trợ xử lý âm thanh, hình ảnh, hiệu ứng đồ họa 2D, 3D, trình duyệt web, cơ sở dữ liệu SQLite …
- *Android Runtime:* mỗi ứng dụng Android chạy trong một thể hiện của máy ảo Dalvik. Trên Android hỗ trợ chạy đa nhiệm. Máy ảo Dalvik thực thi những file ứng dụng ở dạng .dex (Dalvik Executable) được tối ưu hóa cho bộ nhớ và phần cứng điện thoại.
- *Linux Kernel:* Android dựa trên nhân Linux phiên bản 2.6 cung cấp khả năng bảo mật, quản lý bộ nhớ, quản lý tiến trình …

Tóm lại, nhóm tác giả quyết định chọn nền tảng Android vì những lý do sau đây:

- ✓ Android là hệ điều hành mã nguồn mở do Google xây dựng và phát triển. Nguồn tài liệu tham khảo dồi dào cũng như cộng đồng lập trình viên rất đông trên toàn cầu.
- ✓ Điện thoại sử dụng Android ngày càng chiếm thị phần lớn do giá thành rẻ hơn so với các nền tảng khác. Theo khảo sát mới nhất, trong quý 3/2011, Android dẫn đầu thị trường điện thoại thông minh với tỉ lệ khoảng 43%. Dự kiến sẽ tiếp tục tăng trong thời gian tới.
- ✓ Về hiệu năng, Android đáp ứng tốt không thua kém các hệ điều hành khác. Thêm nữa, phía sau là Google với những nền tảng dịch vụ tuyệt vời.
- ✓ Chi phí đầu tư để lập trình trên Android miễn phí, đơn giản. Ngôn ngữ lập trình Android xuất phát từ Java, một ngôn ngữ rất phổ biến trên thế giới. Các IDE lập trình được cung cấp miễn phí cho lập trình viên.

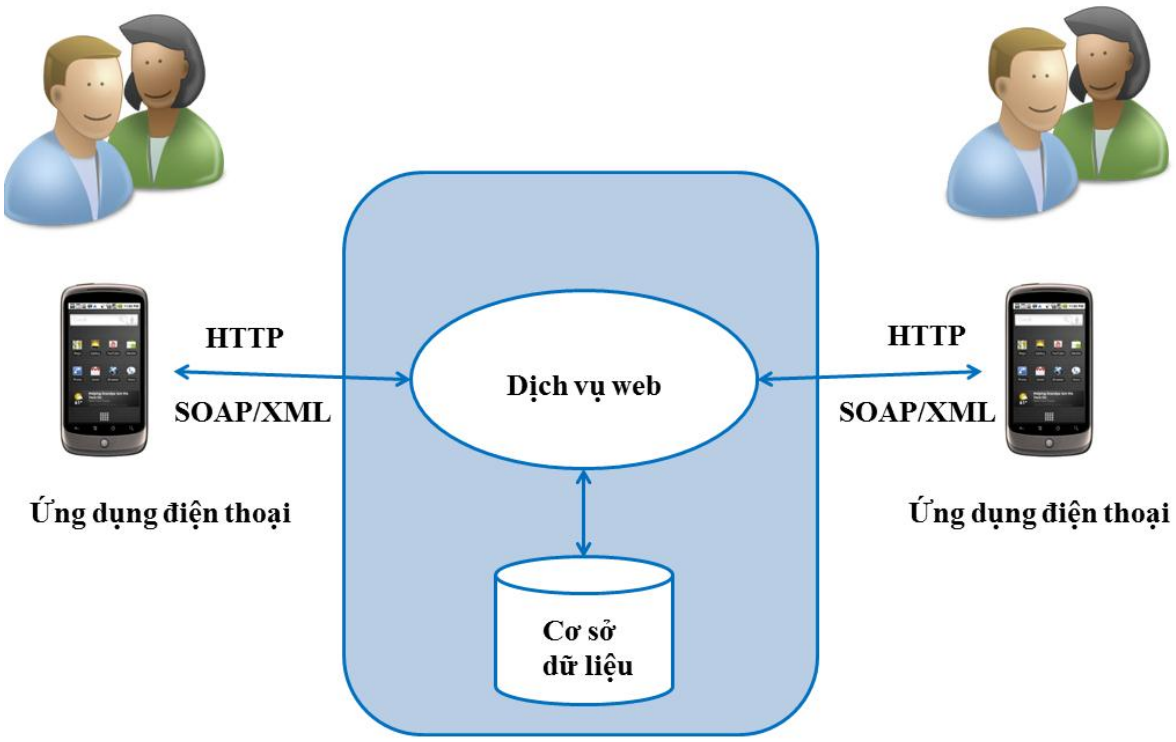
4.3. Dịch vụ web:

Cũng như lập trình web, giữa máy chủ với một số lượng rất lớn các máy khách giao tiếp nhau thông qua giao thức HTTP trên mạng Internet. Trong lập trình điện thoại, những dịch vụ web (web services) được sử dụng nhiều.

4.3.1 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.

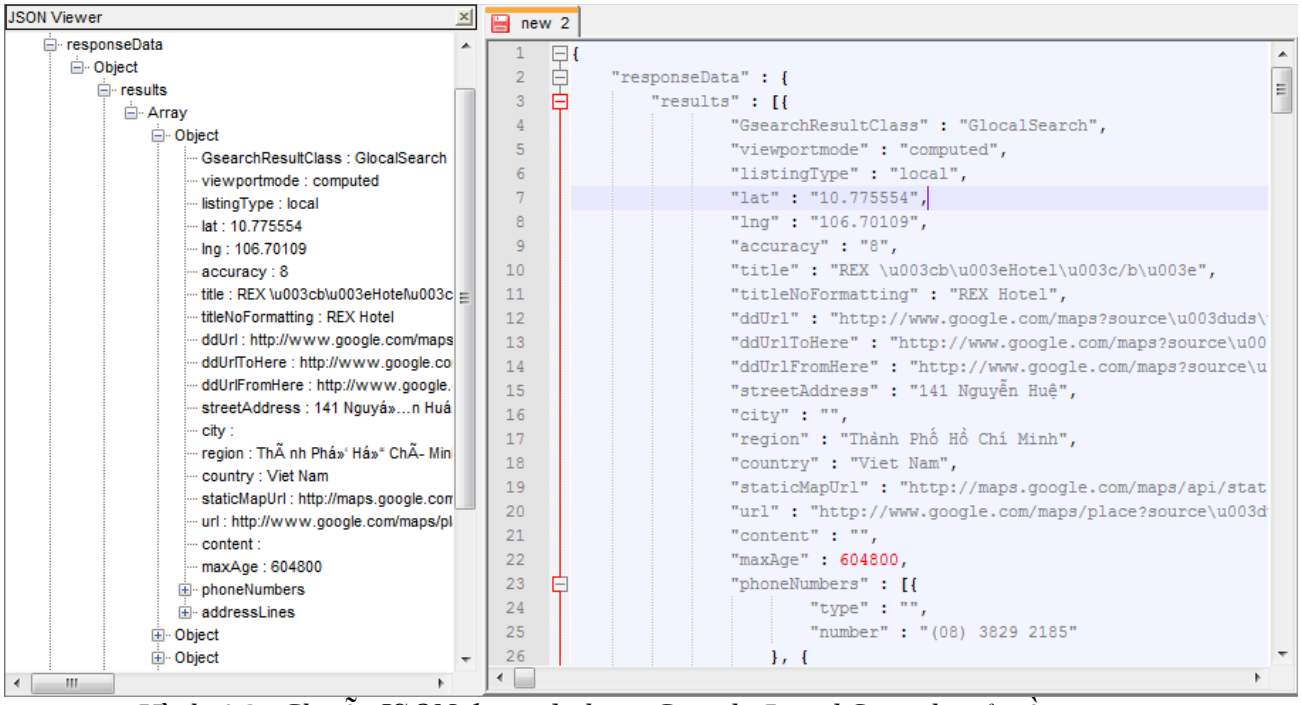
Hình dưới đây minh họa cách truyền tải dữ liệu từ một máy chủ đến nhiều thiết bị điện thoại khác nhau cho nhiều người ở nhiều nơi thông qua dịch vụ web. Không những điện thoại, bất kỳ thiết bị nào khác có hỗ trợ Internet đều có thể truyền nhận dữ liệu.



Hình 4.1: Dịch vụ web cho điện thoại.

Điều này 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 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.

Một ví dụ về dịch vụ web được biết đến nhiều nhất là Google Local Search. Đây là một dạng dịch vụ web được Google xây dựng nhằm mục đích hỗ trợ người dùng tìm kiếm các địa điểm xung quanh mình. Trên điện thoại gửi một yêu cầu lên máy chủ, máy chủ xử lý và trả về kết quả là chuỗi JSON chứa các địa điểm cần tìm. Ví dụ với yêu cầu tìm các nhà hàng xung quanh địa điểm có tọa độ (10.7783,106.6962) ở quận 1, thành phố Hồ Chí Minh, chuỗi JSON trả về là:



Hình 4.2: Chuỗi JSON được dịch vụ Google Local Search trả về.

4.3.2 Windows Communication Foundation (WCF)

Trong hệ thống, do cơ sở dữ liệu cũng như kho dữ liệu nhóm tác giả chọn những công nghệ của Microsoft, nên nhóm tác giả cũng sẽ chọn WCF để xây dựng dịch vụ web. WCF là một phần của .NET Framework cung cấp một mô hình lập trình thống nhất hỗ trợ xây dựng một cách nhanh chóng và dễ dàng các ứng dụng theo hướng dịch vụ (service-oriented) mà giữa chúng truyền thông với nhau thông qua web.

Người ta hoàn toàn có thể xây dựng một web service dùng ASP.NET với dịch vụ web ASMX. Nhưng WCF cung cấp nhiều lợi ích hơn ASMX, có thể kể đến như:

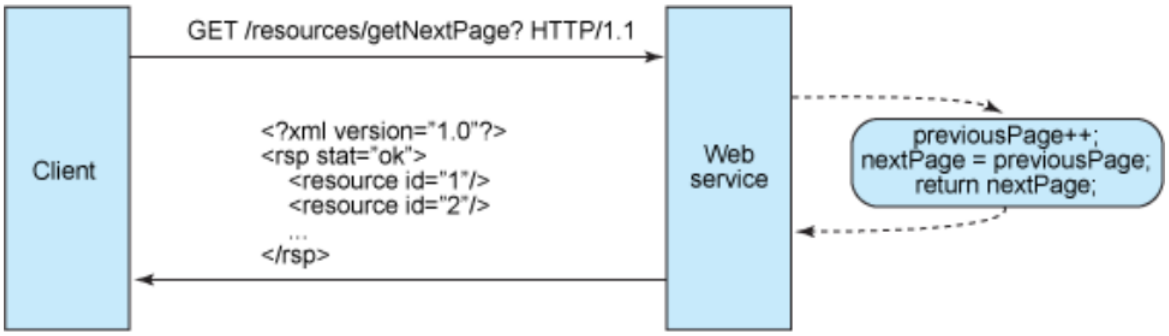
- Hỗ trợ truyền nhận thông tin không chỉ qua giao thức HTTP, mà còn có thể thông qua TCP hoặc những giao thức mạng khác.
- Khả năng chuyển đổi giữa các giao thức truyền nhận thông tin một cách dễ dàng.
- Hỗ trợ việc cung cấp dịch vụ dưới nhiều dạng khác nhau như: WinForms Applications, Console Applications, Windows Services, Web Applications (ASP.NET) trên những phiên bản khác nhau của Internet Information Services (IIS).
- Hỗ trợ khả năng bảo mật, độ tin cậy, các giao tác.

- Hỗ trợ định dạng SOAP (Simple Object Access Protocol – là một giao thức cho phép ứng dụng truyền nhận thông tin thông qua HTTP dưới dạng XML) cũng như REST (Representational State Transfer – là một sự thay thế đơn giản hơn cho SOAP).

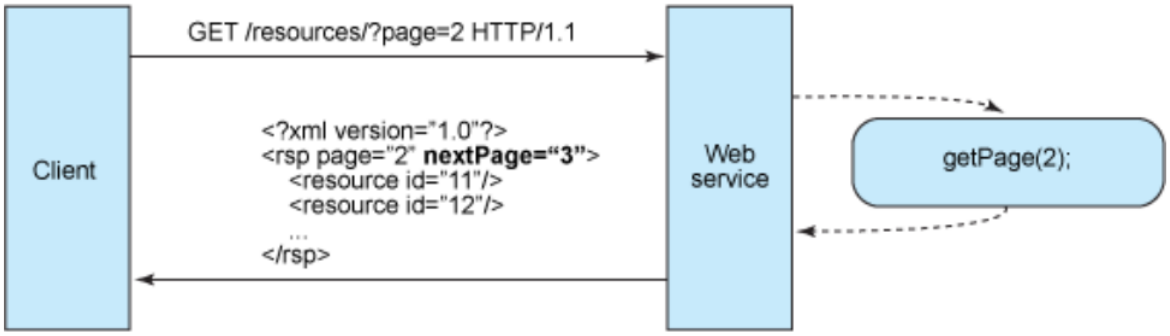
Ở khóa luận này, nhóm tác giả xây dựng một dịch vụ web dựa trên nền tảng REST vì những lợi thế vượt trội so với SOAP:

✓ Sử dụng các phương thức HTTP một cách rõ ràng: dùng POST để tạo một tài nguyên trên máy chủ, dùng GET để truy xuất một tài nguyên, dùng PUT để thay đổi trạng thái một tài nguyên hoặc để cập nhật nó, dùng DELETE để huỷ bỏ hoặc xoá một tài nguyên.

✓ Phi trạng thái (stateless): phía client gửi các yêu cầu hoàn chỉnh và độc lập mà không cần kiểm soát các trạng thái bên trong giữa các lần yêu cầu.



Hình 4.3: Thiết kế có lưu giữ trạng thái (stateful)



Hình 4.4: Thiết kế phi trạng thái (stateless)

✓ Hiển thị cấu trúc thư mục như URIs: ví dụ để tập hợp và duyệt các bài viết theo những đề tài khác nhau, có thể xây dựng đường dẫn cho dịch vụ web với cấu trúc như sau:http://www.myservice.org/discussion/topics/{topic}

✓ Thông tin được truyền đi ở định dạng JSON (JavaScript Object Notation) hoặc XML (Extensible Markup Language).

CHƯƠNG 5
XÂY DỰNG HỆ THỐNG

---oOo---

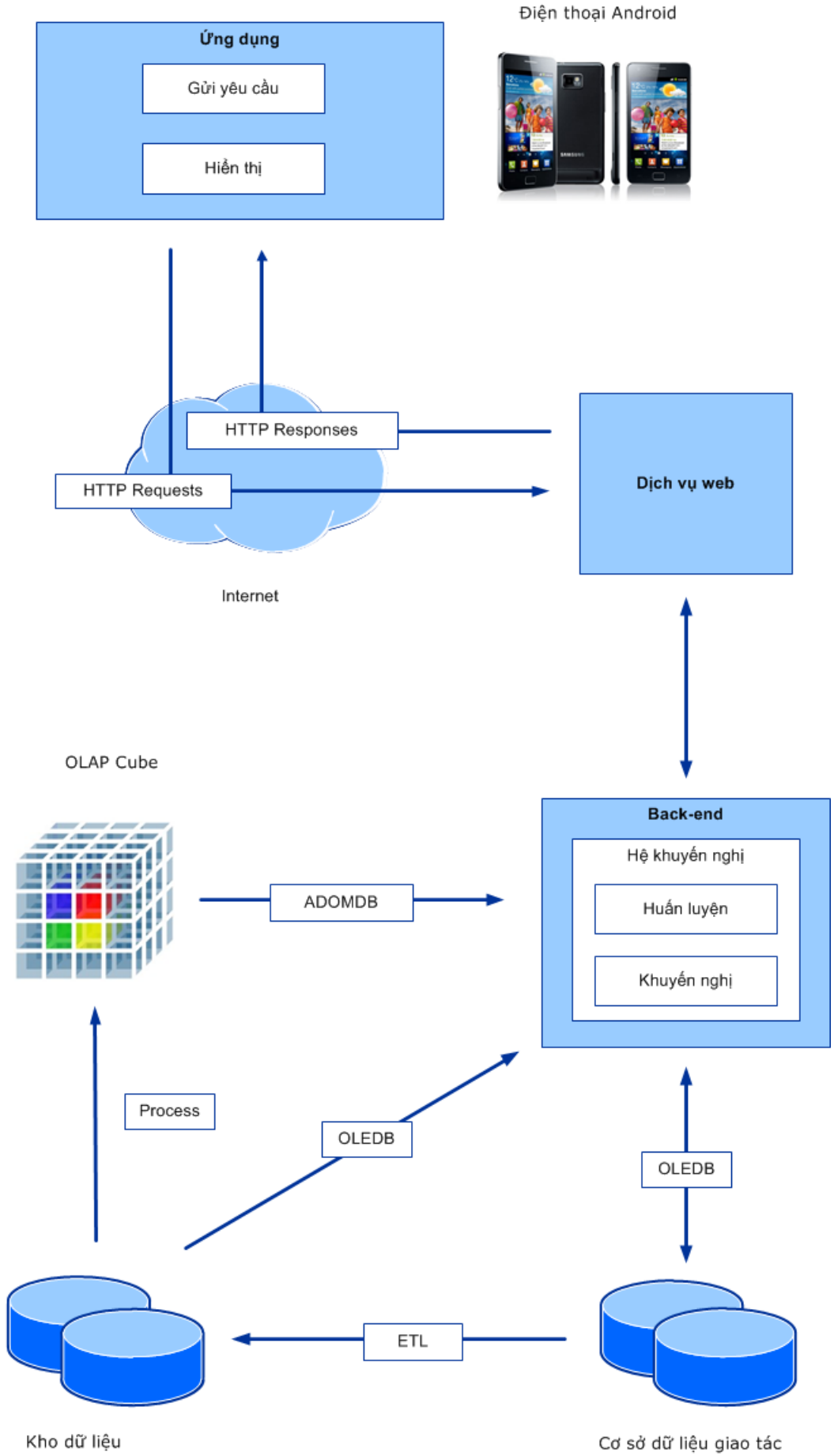
Qua các chương trên, nhóm tác giả đã khảo sát, phân tích cũng như lựa chọn những kỹ thuật, những phương pháp phù hợp để giải quyết bài toán. Ở chương 5 này, nhóm tác giả sẽ cụ thể hóa việc xây dựng hệ thống như thế nào.

5.1. Kiến trúc hệ thống:

Sau khi đã tìm hiểu kỹ hơn về mặt kỹ thuật, bây giờ nhóm tác giả có thể đưa ra một kiến trúc hoàn chỉnh hơn cho hệ thống, gồm có hai phần chính như sau:

- ❖ Phần một: ứng dụng máy khách được viết trên nền tảng hệ điều hành Android hỗ trợ khách du lịch các chức năng như: tìm kiếm thông tin các điểm du lịch, các gợi ý về các điểm du lịch phù hợp với người dùng, bản đồ …
- ❖ Phần hai: ứng dụng máy chủ bao gồm:
  - Cơ sở dữ liệu, kho dữ liệu và OLAP Cube.
  - Hệ thống khuyến nghị người dùng dựa trên ngữ cảnh.
  - Dịch vụ web phụ trách việc truyền nhận thông tin giữa máy khách và máy chủ.

Những mục tiếp theo trong chương này, nhóm tác giả sẽ trình bày chi tiết cách thực hiện từng phần trên.

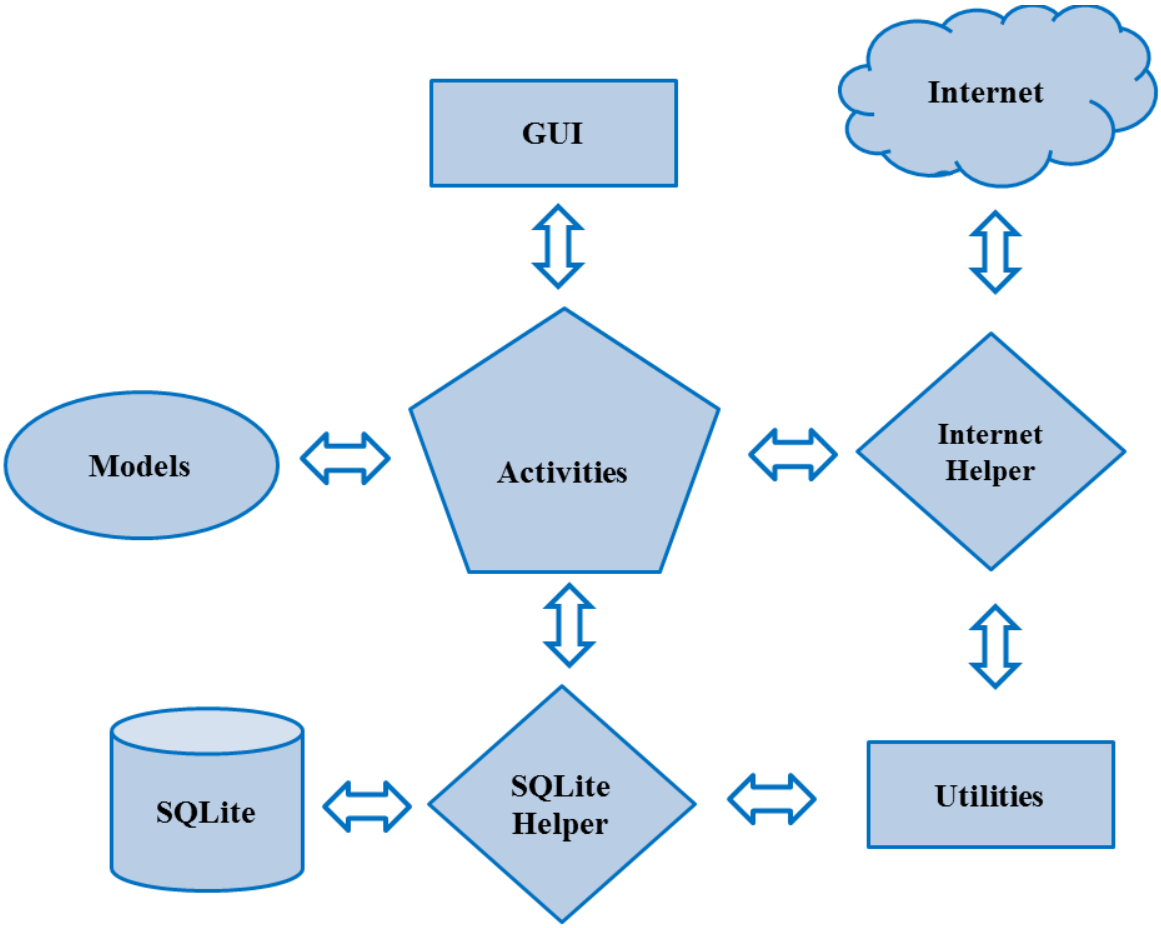


Hình 5.1: Kiến trúc hệ thống.

5.2. Xây dựng ứng dụng trên Android:

5.2.1. Kiến trúc ứng dụng Android:

Dưới đây là hình ảnh minh họa kiến trúc ứng dụng trên Android.



Hình 5.2: Kiến trúc của ứng dụng Android.

Những thành phần quan trọng nhất gồm:

➢ *Activies:* giữ vai trò chủ đạo tương tự như Form trong lập trình Windows Form. Ứng với mỗi màn hình trên điện thoại là một activity phụ trách việc hiển thị, điều khiển các chức năng tương ứng.

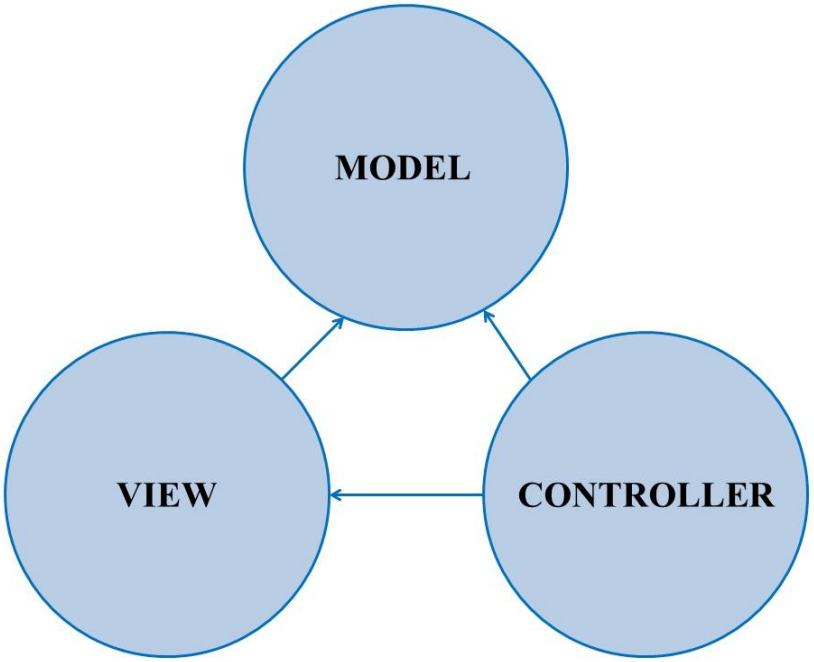
➢ *Graphical user interface (GUI):* giao diện người dùng được thiết kế bởi những file XML và các file ảnh định dạng PNG, JPG, BMP.

➢ *Cơ sở dữ liệu SQLite:* cơ sở dữ liệu cục bộ trên điện thoại được xây dựng nhằm lưu trữ những thông tin cấu hình ngữ cảnh, địa điểm du lịch ưa thích mà người dùng muốn lưu lại, cùng một số thông tin khác. Và SQLite Helper là thành phần trung gian xây dựng những hàm để thực hiện kết nối, truy vấn đến cơ sở dữ liệu.

➢ *Kết nối Internet:* Internet Helper là những hàm hỗ trợ điện thoại kết nối Internet để truy xuất dịch vụ bản đồ Google Maps cũng như truy xuất đến hệ thống khuyến nghị đã được nhóm tác giả đề cập ở phần trên.

➢ *Ultilities:* là những lớp khác được xây dựng để hỗ trợ một số xử lý khác như xử lý chuỗi JSON, các hàm hỗ trợ tính toán, lấy vị trí người dùng, xử lý mảng, chuỗi …

Về mô hình tổ chức code, nhóm tác giả sử dụng mô hình MVC (Model – View - Controller) quen thuộc.



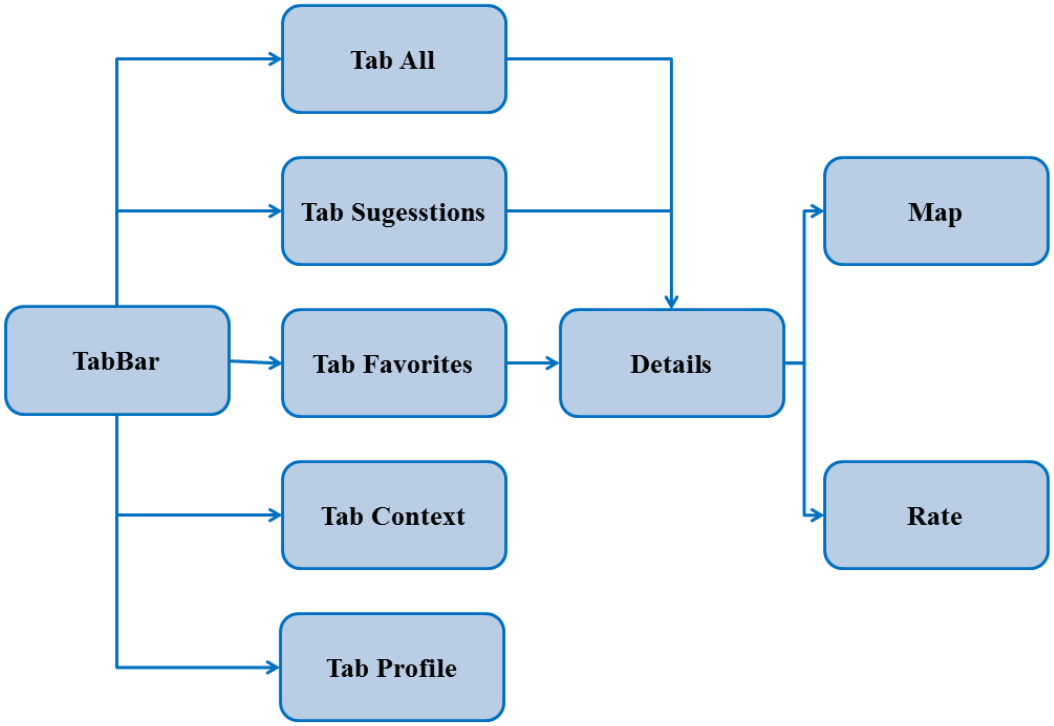
Hình 5.3: Mô hình MVC.

- *Controller:* chính là các Activities, ghi nhận các sự kiện, các yêu cầu từ người dùng, từ đó quyết định View nào và Model nào sẽ được dùng để đáp ứng các yêu cầu đó.

- *View:* là phần giao diện người dùng. Ở đây, chúng là các file XML trong Android dùng để định nghĩa giao diện các màn hình. Ứng với một màn hình, sẽ có một file XML.

- *Model:* là mô hình ánh xạ các bảng trong cơ sở dữ liệu thành các lớp tương ứng. Model cũng bao gồm các lớp chứa các hàm hỗ trợ xử lý các yêu cầu mà Controller ghi nhận (kết nối cơ sở dữ liệu, tính toán, trả kết quả về hiển thị lên View).

5.2.2. Sơ đồ các màn hình:



Hình 5.4: Sơ đồ các màn hình trong ứng dụng.

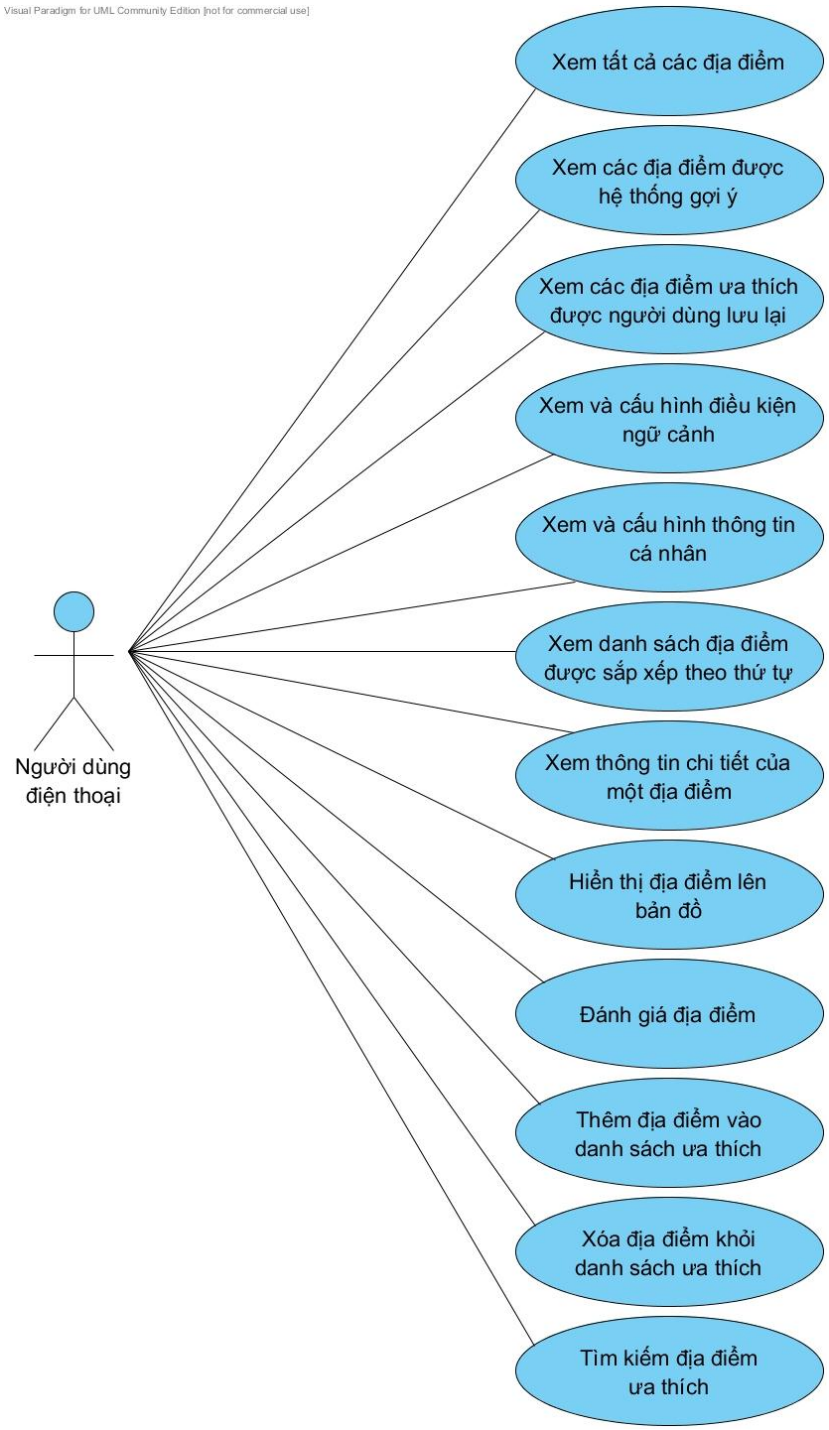
Ứng dụng gồm năm tab:

- Tab All: màn hình danh sách tất cả các địa điểm du lịch xung quanh người dùng.
- Tab Suggestions: màn hình danh sách các địa điểm du lịch được hệ thống khuyến nghị gợi ý cho người dùng tùy vào điều kiện ngữ cảnh họ cung cấp.
- Tab Favorites: màn hình danh sách các địa điểm du lịch ưa thích người dùng muốn lưu lại.
- Tab Context: màn hình phần cấu hình thông tin các điều kiện ngữ cảnh.
- Tab Profile: màn hình phần cấu thông tin người dùng.

Từ màn hình All, Suggestions hoặc Favorites, khi người dùng chọn một địa điểm, sẽ chuyển sang màn hình Details hiển thị thông tin chi tiết của địa điểm đó. Từ màn hình Details, khi người dùng chọn nút Show On Map sẽ chuyển sang màn hình Map hiển thị địa điểm lên bản đồ. Tương tự, khi người dùng chọn nút Rate sẽ chuyển sang màn hình Rate cho phép người dùng đánh giá địa điểm (tùy thuộc vào điều kiện ngữ cảnh họ đã chọn ở tab Context). Mức độ đánh giá có 5 mức: thấp nhất là 1, cao nhất là 5. Cụ thể hơn, nhóm tác giả sẽ chụp những hình minh họa ở chương tiếp theo.

5.2.3. Các use cases trong hệ thống:

Để xây dựng được hệ thống, nhóm tác giả dùng phương pháp UML để phân tích thiết kế hệ thống. Tuy nhiên nếu trình bày chi tiết tất cả quá trình phân tích thiết kế thì sẽ rất dài dòng. Vậy nên, nhóm tác giả sẽ bỏ qua các sơ đồ khác trong phân tích thiết kế UML nhưng vẫn giữ lại sơ đồ use case để người đọc có thể hiểu rõ ràng các chức năng có trong ứng dụng. Phần mô tả chi tiết từng use case có thể xem tại phụ lục A cuối quyển báo cáo.

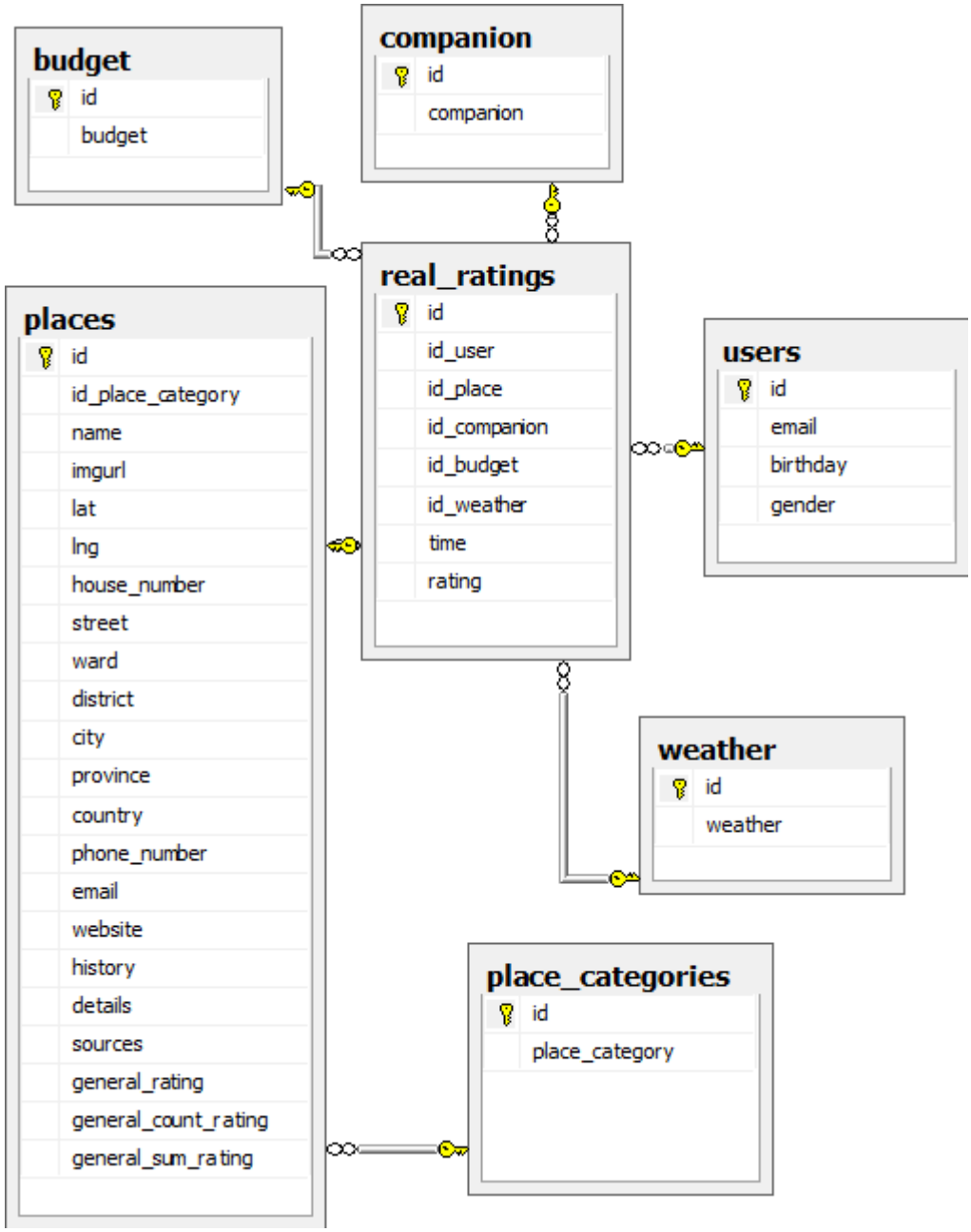


Hình 5.5: Sơ đồ use case.

5.3. Xây dựng ứng dụng trên máy chủ:

5.3.1. Cơ sở dữ liệu, kho dữ liệu và OLAP:

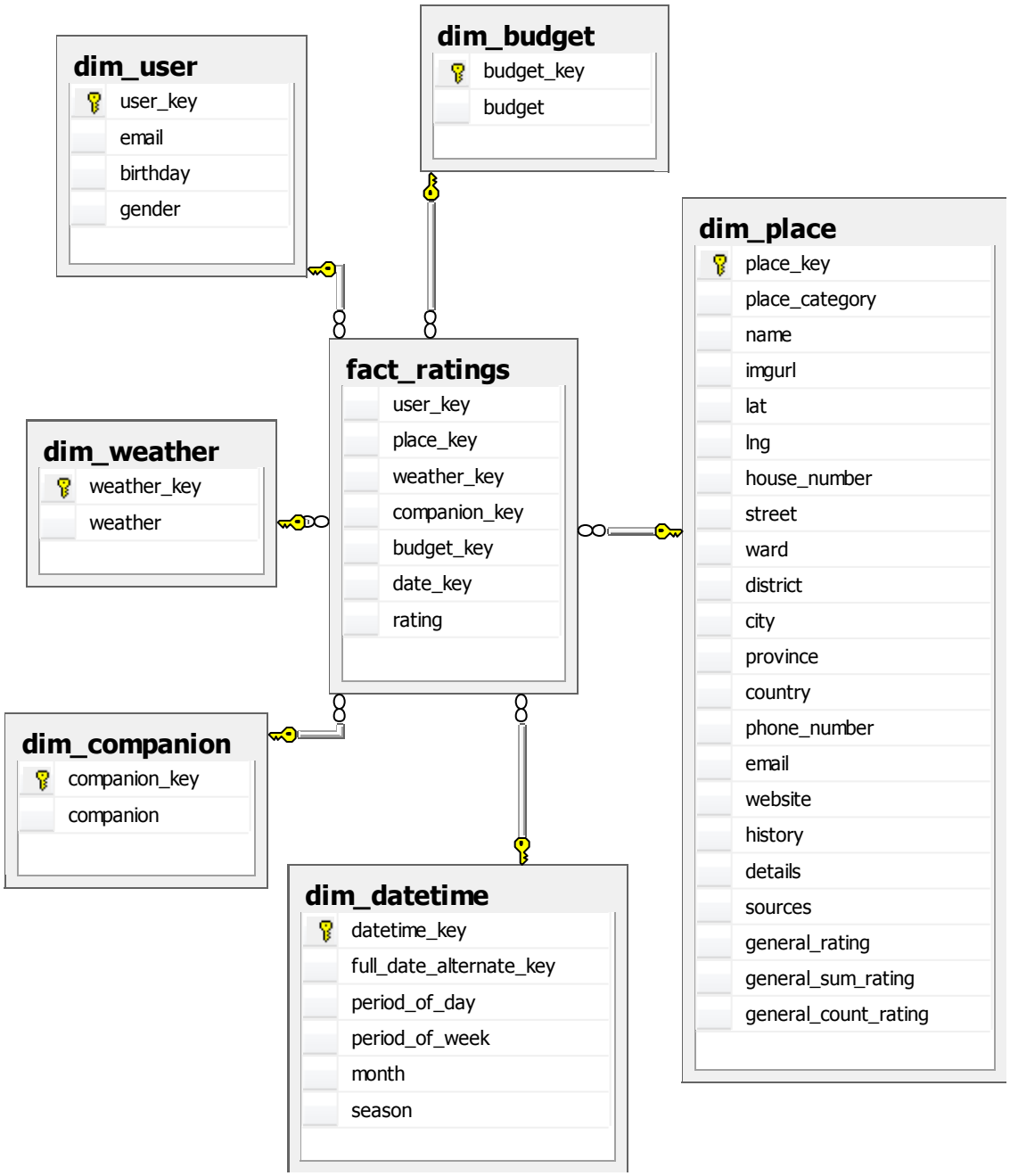
Hình dưới đây mô tả cơ sở dữ liệu phục vụ việc quản lý và lấy đánh giá từ người dùng:



Hình 5.6: Cơ sở dữ liệu quan hệ.

Sau một khoảng thời gian định kỳ nhóm tác giả sẽ cập nhật kho dữ liệu từ cơ sở dữ liệu giao tác trên, điều này nhằm mục đích chuẩn bị cho việc huấn luyện dữ liệu như đã trình bày ở chương 3 và 4, hơn nữa việc huấn luyện được thực hiện trên dữ liệu chỉ được phép đọc (không được phép ghi) và tách biệt với cơ sở dữ liệu giao tác (tương tác liên tục với ứng dụng máy khách) là một việc làm cần thiết.

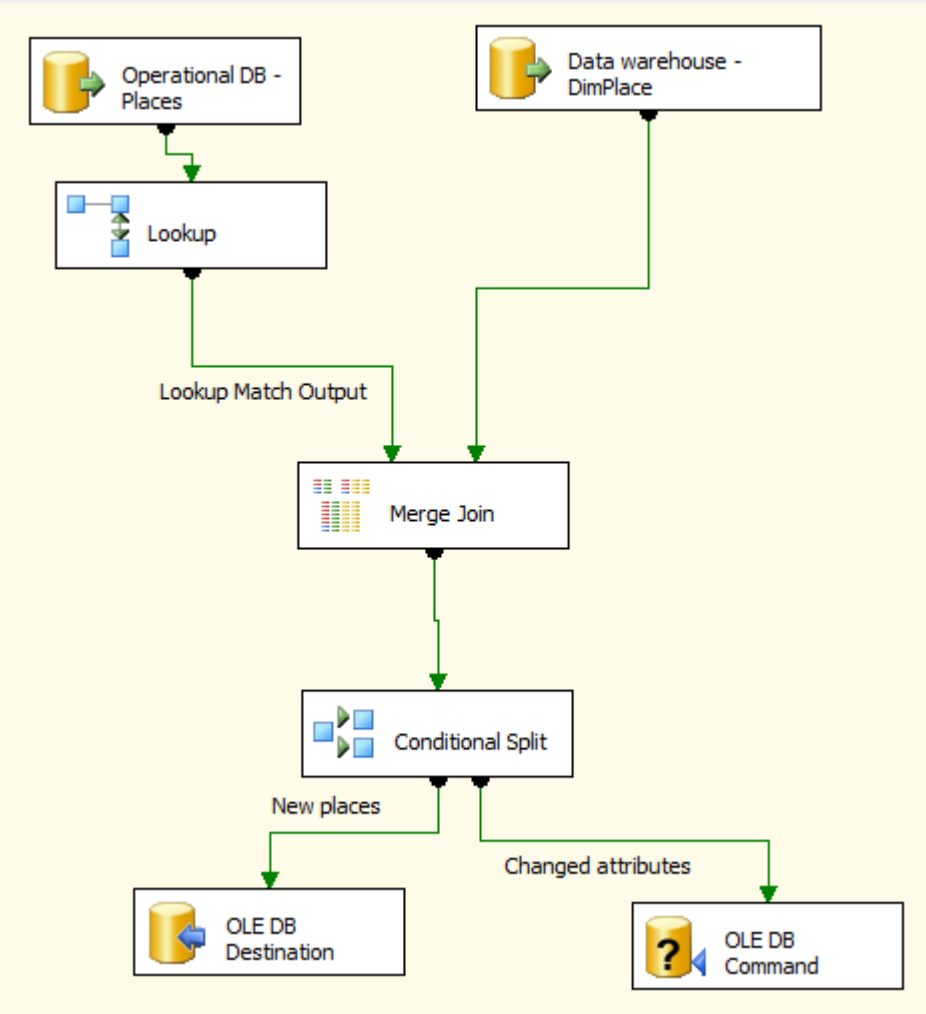
Kho dữ liệu có cấu trúc như sau:



Hình 5.7: Kho dữ liệu.

Quá trình ETL (Extract, Transform, Load) để cập nhật dữ liệu cho kho dữ liệu được xây dựng thông qua SSIS (SQL Server Integration Services) như sau:

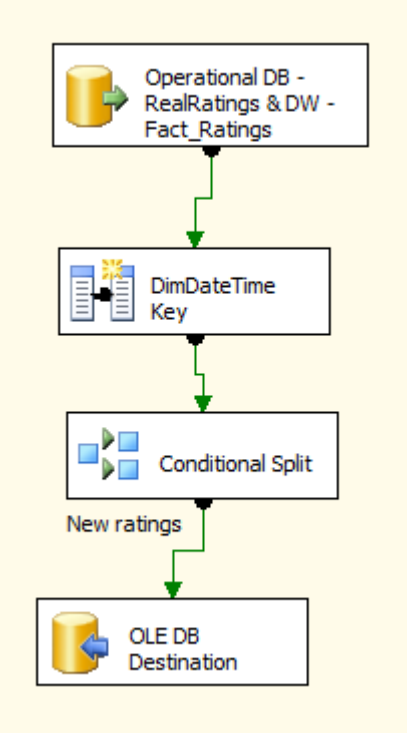
- Chiều địa điểm (dim_place):



Hình 5.8: ETL cho bảng dim_place.

Đầu tiên, toàn bộ dữ liệu các địa điểm đang có từ cơ sở dữ liệu phục vụ giao tác (bảng places) và dữ liệu trên kho dữ liệu (bảng dim_place) được trộn lại với nhau bằng phép toán kết trái, từ đó dữ liệu sẽ được chia làm hai phần là những dữ liệu mới và những dữ liệu có sự thay đổi. Cuối cùng dữ liệu sẽ được thêm mới hoặc sửa đổi vào kho dữ liệu.

- Quá trình trên thực hiện tương tự với các chiều khác như chiều kinh phí (dim_budget), chiều bạn đồng hành (dim_companion), chiều thời tiết (dim_weather), chiều người dùng (dim_user).
- Chiều thời gian (dim_datetime): xây dựng cấu trúc phân cấp (ngày, tuần, tháng, mùa …). Chi tiết hơn, người đọc có thể tham khảo tập tin *"Dim_DateTime.sql"* trong phần mã nguồn đính kèm báo cáo.
- Chiều dữ liệu đánh giá thực tế của người dùng (fact_ratings):



Hình 5.9: ETL cho bảng fact_ratings.

Từ tập dữ liệu trên cơ sở dữ liệu phục vụ giao tác, nhóm tác giả tính toán thêm khóa đại diện cho chiều thời gian (dim_datetime) sau đó so sánh thời gian này với thời gian của lần cuối cùng thực hiện quá trình ETL trước đó (lấy từ kho dữ liệu) để có các đánh giá mới nhất (kể cả các đánh giá được người dùng thay đổi) và thêm mới vào kho dữ liệu.

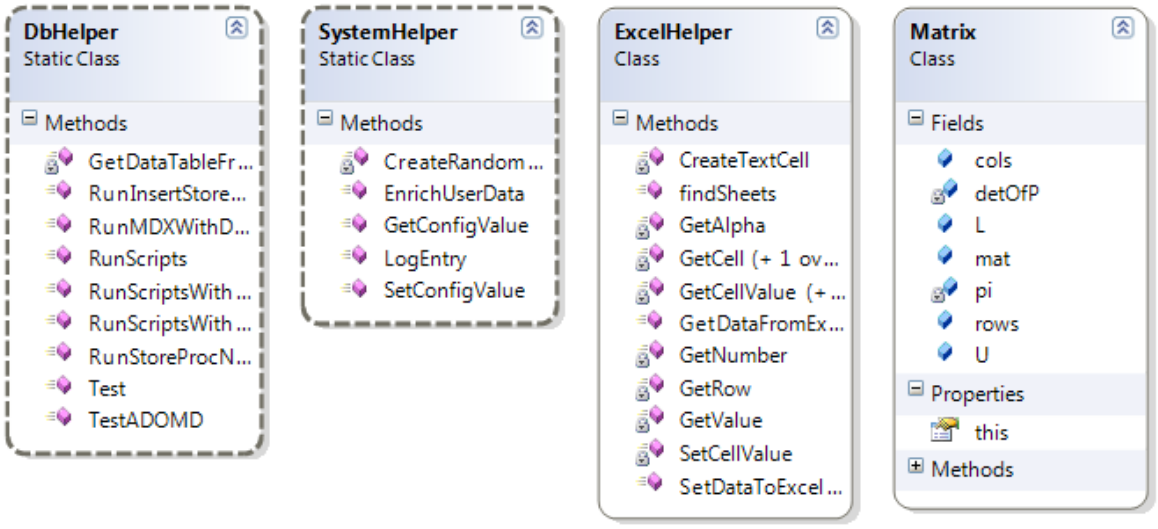
Sau khi đã cập nhật thành công dữ liệu vào kho dữ liệu, dữ liệu tiếp tục được cập nhật vào OLAP Cube.

5.3.2. Hiện thực thuật toán khuyến nghị:

Lý thuyết về thuật toán khuyến nghị đã được trình bày chi tiết ở chương 3 trong báo cáo này. Sau đây, nhóm tác giả sẽ trình bày vắn tắt phần triển khai thực tế như sau:

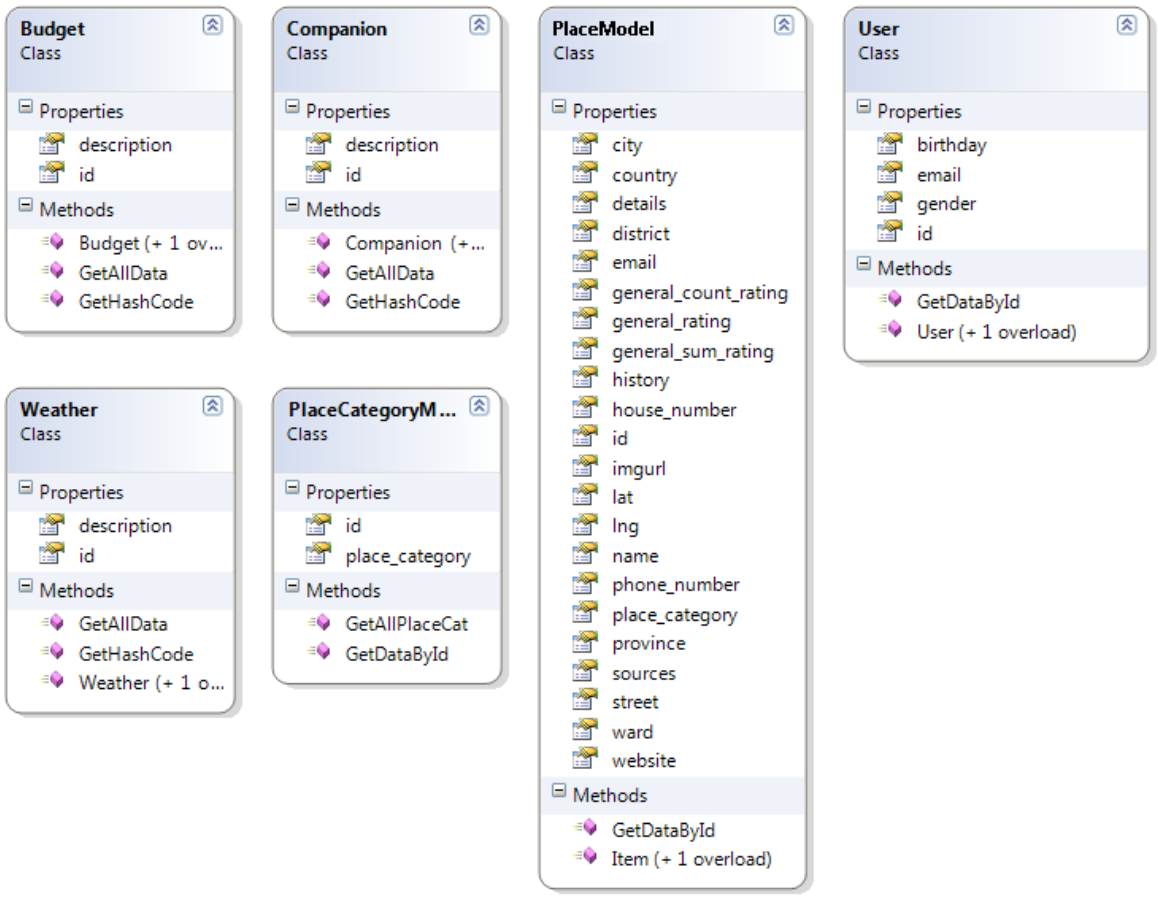
Tổ chức code gồm ba gói chính:

- *Helper:* thư viện cần thiết cho việc cài đặt hệ thống:
  - DbHelper: lớp hỗ trợ việc kết nối và truy vấn dữ liệu.
  - SystemHelper: lớp hỗ trợ chung cho hệ thống như đọc và chỉnh sửa các tùy chỉnh, lưu vết hệ thống.
  - ExcelHelper: lớp hỗ trợ thao tác đọc ghi tập tin excel.
  - Matrix: lớp hỗ trợ thao tác ma trận.



Hình 5.10: Các lớp thuộc gói Helper.

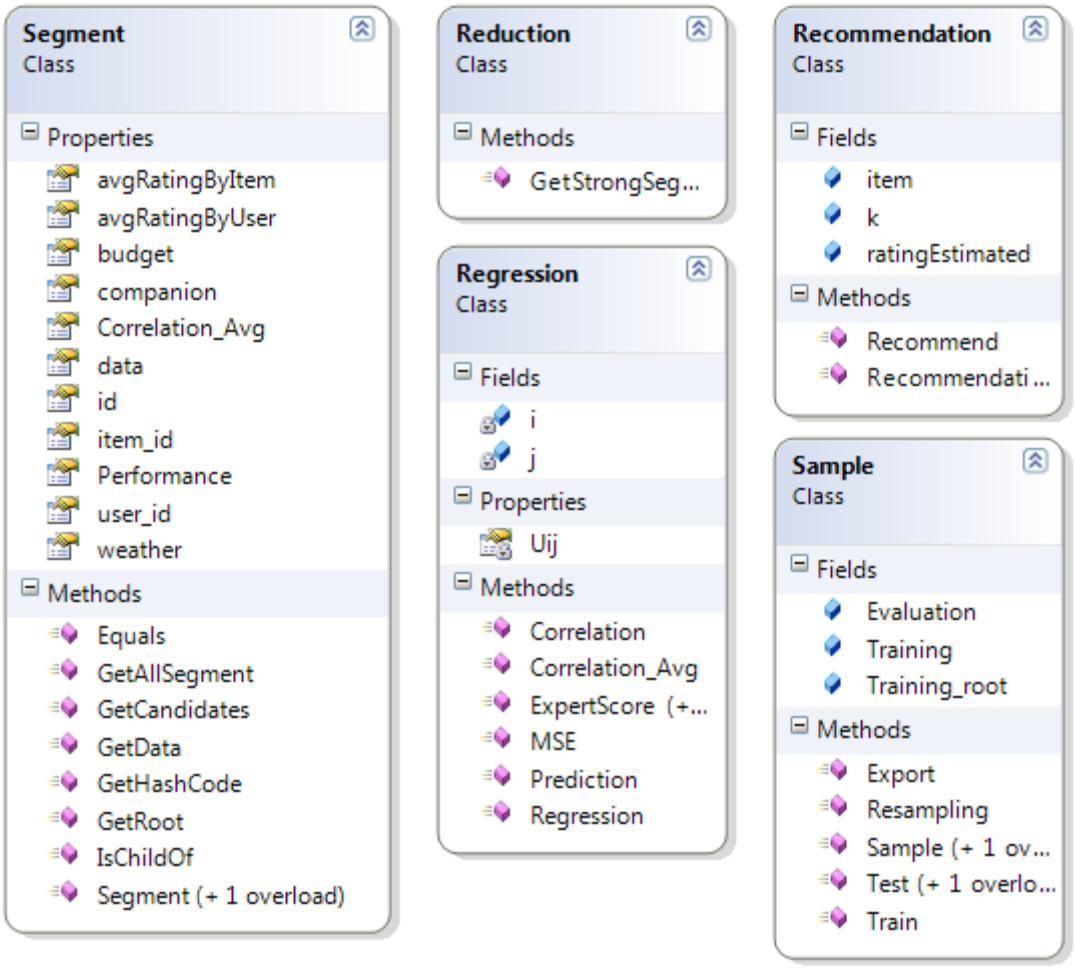
- *Model:* các mô hình ánh xạ các lớp tương ứng với các bảng dưới cơ sở dữ liệu: users, places, place_categories, time, budget, companion, weather.



Hình 5.11: Các lớp thuộc gói Model – hệ khuyến nghị.

- *RS core:* phần nhân của thuật toán.
  - Segment: lớp định nghĩa phân khúc dữ liệu.
  - Sample: lớp định nghĩa một bộ mẫu (tập huấn luyện, tập đánh giá).

- Regression: lớp cài đặt thuật toán hồi qui để dự đoán đánh giá.
- Reduction: lớp cài đặt thuật toán thu giảm số chiều.
- Recommendation: lớp hiện thực việc khuyến nghị.

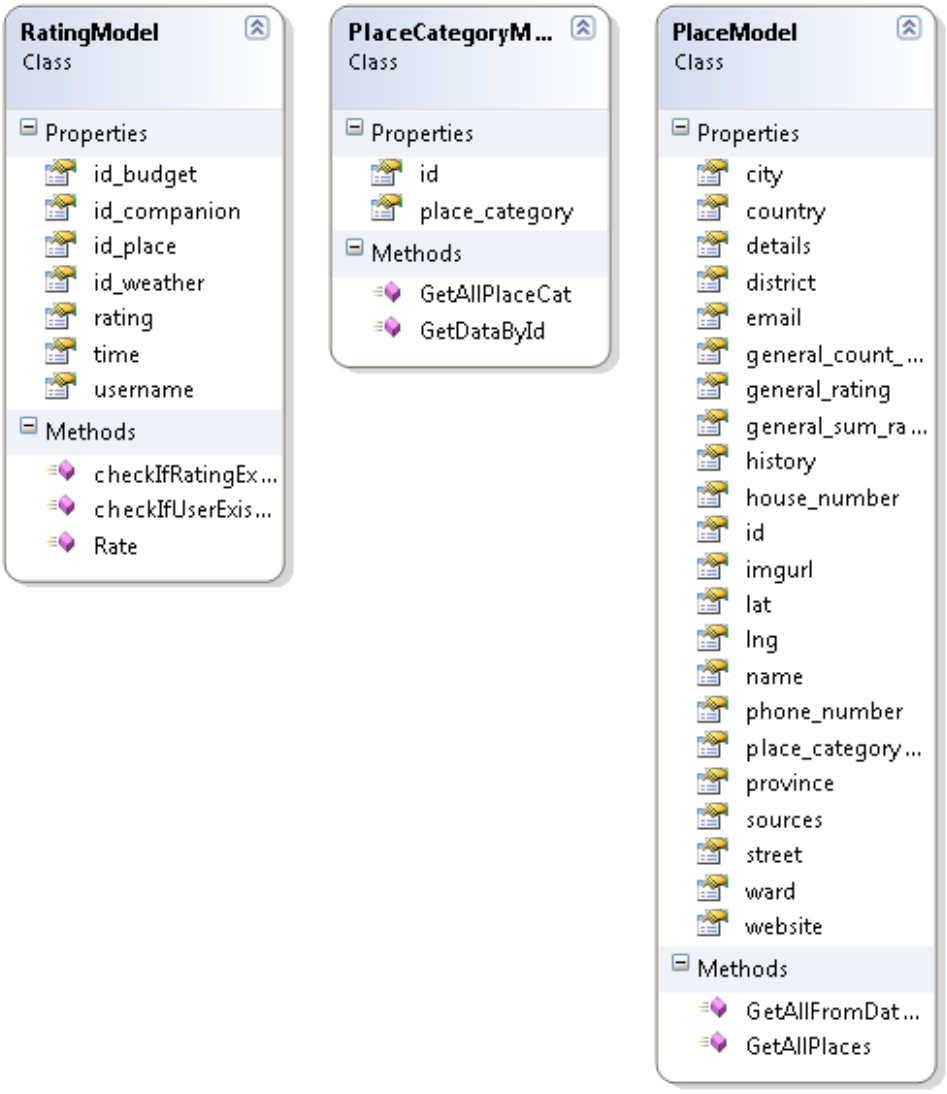


Hình 5.12: Các lớp thuộc gói RS core.

5.3.3. Hiện thực dịch vụ web:

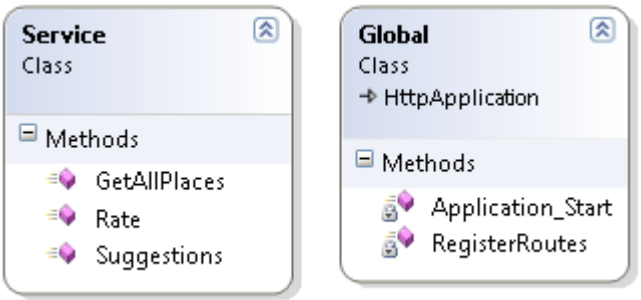
Tương tự, phần dịch vụ web cũng đã được trình bày chi tiết ở mục 4.3 của báo cáo này. Sau đây là phần triển khai sẽ được trình bày ngắn gọn. Microsoft cung cấp sẵn một mô hình mẫu giúp lập trình viên xây dựng dịch vụ web dạng REST nhanh hơn. Đó là "WCF REST Service Template 4.0" dùng ngôn ngữ lập trình C# hoặc Visual Basic. Dựa theo mô hình mẫu đó, các gói chính trong tổ chức code sẽ gồm:

- *Model:* gồm các lớp dùng để ánh xạ những bảng cần thiết dưới cơ sở dữ liệu. bảng real_ratings, bảng place_categories và bảng places.



Hình 5.13: Các lớp thuộc gói Model – dịch vụ web WCF.

- *Service:* cung cấp các lớp, các phương thức dùng triển khai dịch vụ web.
  - Global: lớp quy định cách thực thi các đường dẫn của trang web.
  - Service: lớp chứa ba phương thức giúp đáp ứng các yêu cầu từ điện thoại (lấy toàn bộ danh sách địa điểm, đánh giá một địa điểm, lấy danh sách địa điểm được gợi ý).



Hình 5.14: Các lớp thuộc gói Service – dịch vụ web WCF.

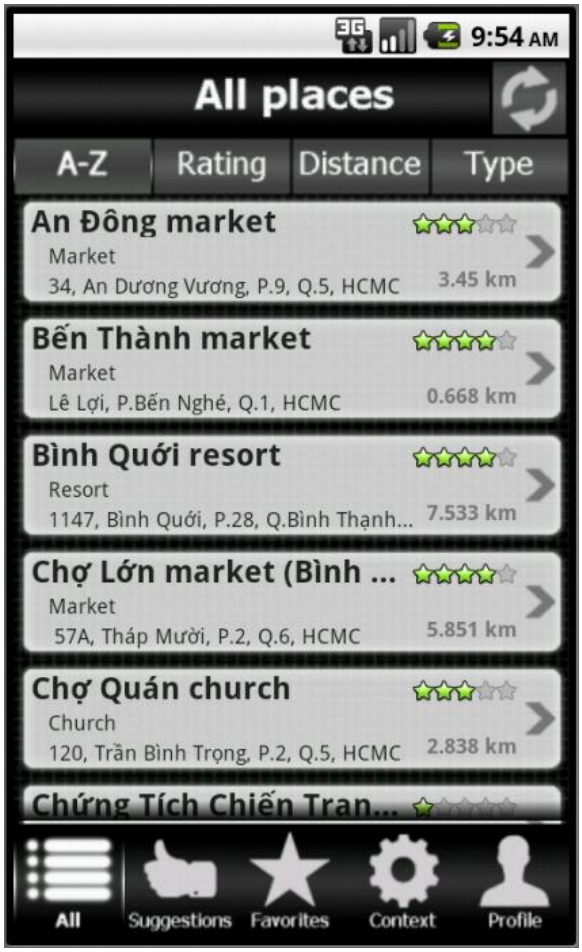
CHƯƠNG 6.
NGHIỆM THU KẾT QUẢ

---oOo---

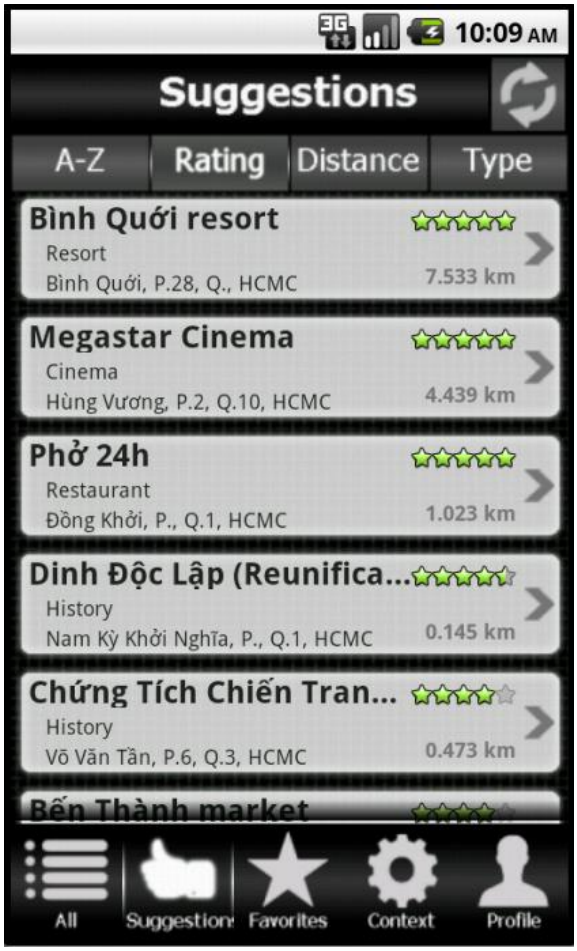
Chương này nhóm tác giả sẽ trình bày những hình ảnh minh họa cho các phần trình bày ở các chương trước cũng như phần thực nghiệm và đánh giá về những kết quả đạt được sau khi tiến hành xây dựng toàn bộ hệ thống.

6.1. Chương trình minh họa:

Sau đây là ảnh chụp các màn hình ứng dụng khi chạy trên máy ảo Android (Android Simulator).



Hình 6.1: Màn hình All places.



Hình 6.2: Màn hình Suggestions.

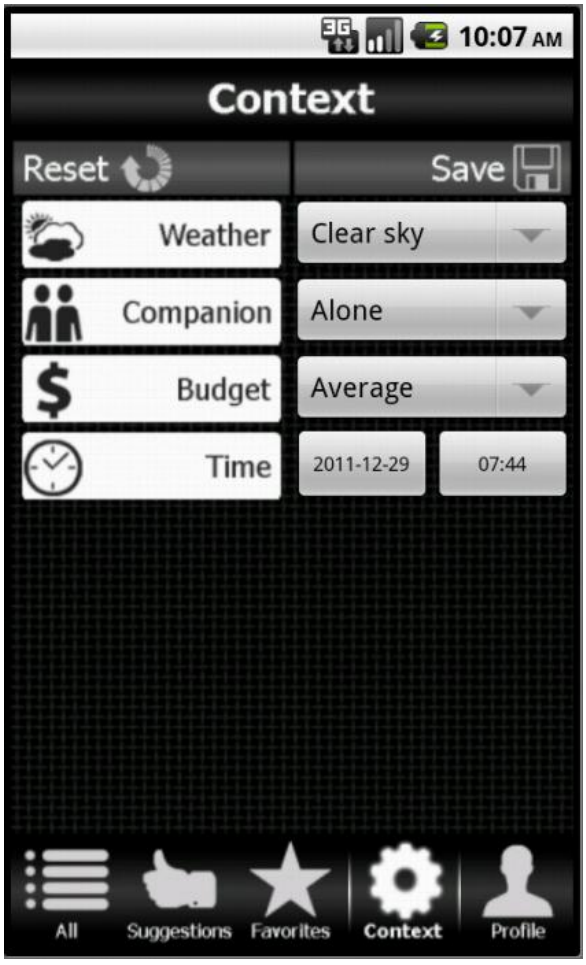
Mô tả hình 6.1 và hình 6.2:

Chương trình gồm 5 tab: All, Suggestions, Favorites, Context, Profile.

Màn hình All places thể hiện tất cả các điểm du lịch có trong hệ thống. Trên cùng là nút *Refresh* để tải lại màn hình. Phía dưới là bốn nút *A-Z*, *Rating*, *Distance*,

Type dùng để sắp xếp danh sách các địa điểm tương ứng theo tên, theo chỉ số đánh giá chung chung (chỉ số này không tính tới điều kiện ngữ cảnh và được lấy từ trang web du lịch uy tín tên LonelyPlanet trên Internet), theo khoảng cách từ vị trí hiện tại của người dùng đến điểm du lịch, và theo thể loại của điểm du lịch (là nhà thờ, nhà hàng, khách sạn, khu nghỉ dưỡng, chợ …). Vị trí hiện tại của người dùng lấy từ hệ thống GPS, nếu không lấy được thì lấy điểm mặc định ở Dinh Độc Lập, Q1, Tp.HCM làm vị trí hiện tại.

Màn hình Suggestions tương tự màn hình All Places. Nhưng đây là những địa điểm du lịch được hệ thống khuyến nghị gợi ý cho người dùng dựa vào những điều kiện ngữ cảnh họ cung cấp ở tab Context. Chỉ số đánh giá ở đây là những chỉ số đánh giá dự đoán do hệ khuyến nghị tính toán và đưa ra, không phải những chỉ số đánh giá chung chung như ở màn hình All places.



Hình 6.3: Màn hình Context.

Hình 6.4: Màn hình Profile.

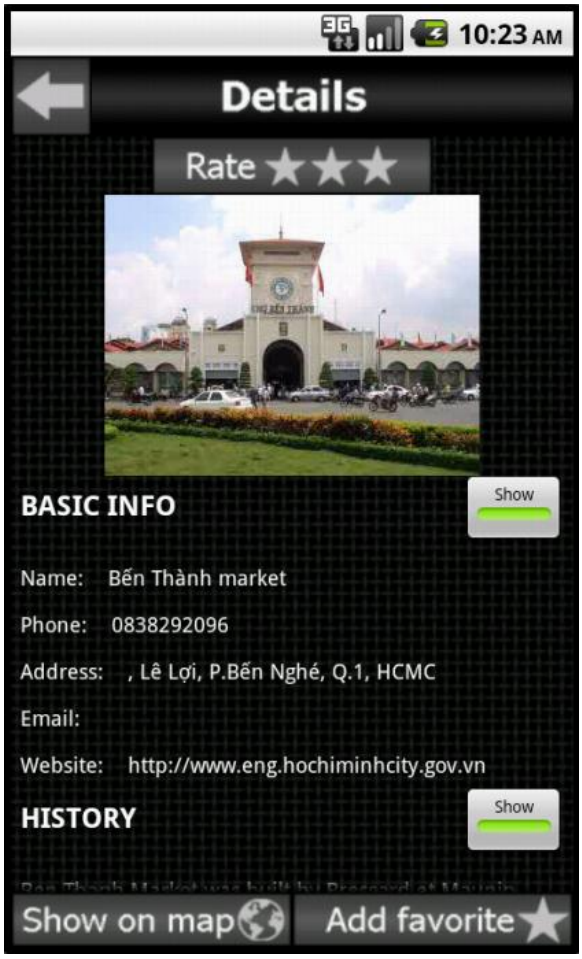
Mô tả hình 6.3 và hình 6.4:

Màn hình Context thể hiện các điều kiện ngữ cảnh của người dùng.

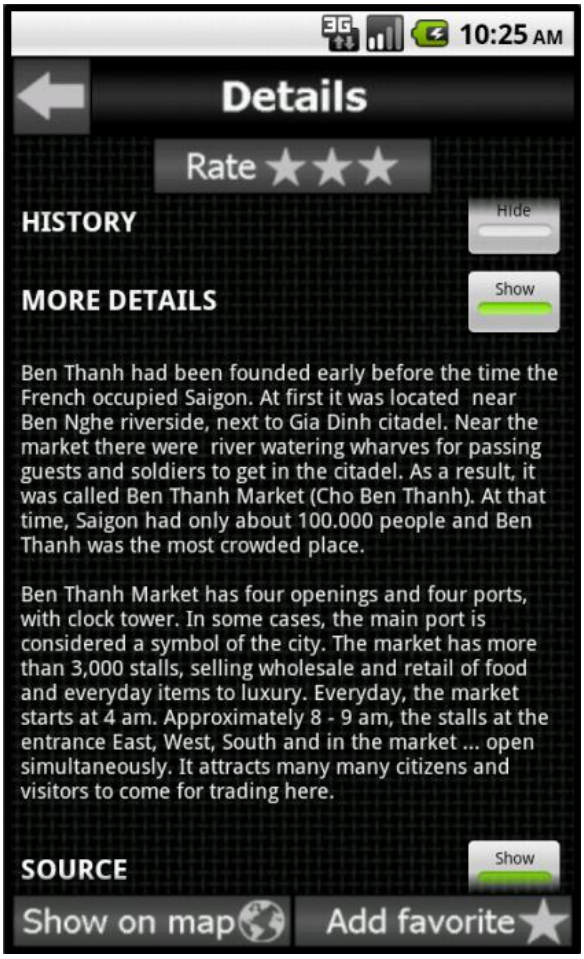
Người dùng chọn các điều kiện ngữ cảnh (thời tiết, bạn đồng hành, kinh phí du lịch, thời điểm du lịch) và nhấn nút *Save*, những thông tin đó được lưu tại cơ sở dữ liệu cục bộ trên điện thoại. Nếu nhấn nút *Reset*, sẽ trả về các giá trị ngữ cảnh mặc định và thời gian hiện tại.

Khi đánh giá một địa điểm, điện thoại sẽ lấy những thông tin ngữ cảnh cùng chỉ số đánh giá gửi lên máy chủ của hệ thống khuyến nghị (nghĩa là chỉ số đánh giá đó sẽ tính trong những điều kiện ngữ cảnh này, với điều kiện khác, người dùng có thể đánh giá khác). Tương tự, khi khuyến nghị, hệ thống khuyến nghị sẽ khuyến nghị các điểm du lịch dựa vào những điều kiện ngữ cảnh được người dùng lưu lại ở tab Context này.

Màn hình Profile tạm thời lưu giữ hai thông tin của người dùng là giới tính và ngày sinh. Hiện tại, hệ thống khuyến nghị chưa sử dụng đến những thông tin này. Nhưng về sau, khi tiếp tục phát triển, những thông tin cá nhân này sẽ cần thiết cho việc khuyến nghị. Như nhóm tác giả đã trình bày ở mục 4.1, hệ thống khuyến nghị còn có thể khai thác thông tin chi tiết của các chiều, chẳng hạn chiều người dùng với các thông tin như tên, tuổi, giới tính, sở thích, … có thể được sử dụng để khuyến nghị tốt hơn.



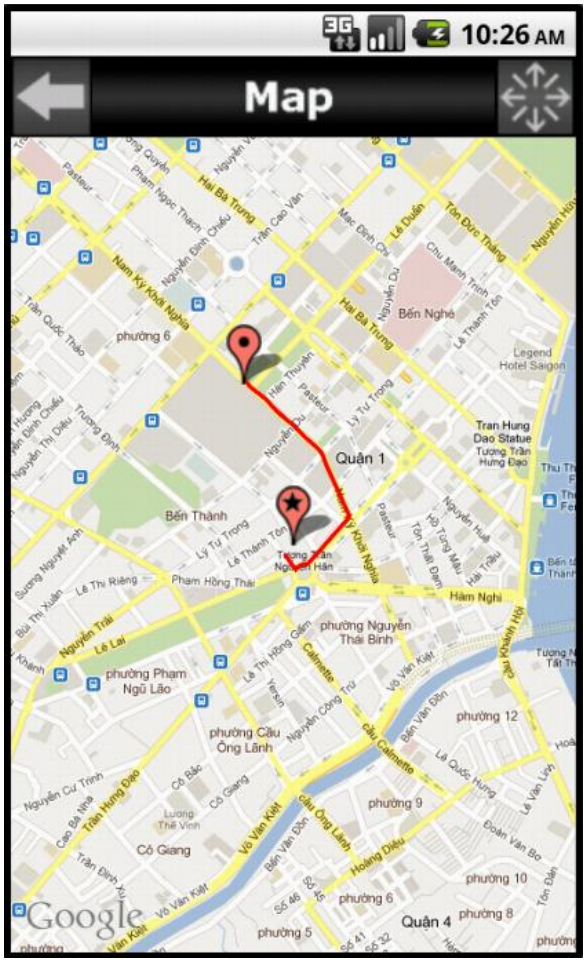
Hình 6.5: Màn hình Details.



Hình 6.6: Màn hình Details (2).

Mô tả hình 6.5 và hình 6.6:

Khi người dùng chọn một điểm du lịch bên tab All places hoặc tab Suggestions, điện thoại sẽ chuyển qua màn hình Details hiển thị các thông tin chi tiết của điểm đó. Nút *Rate* dùng để chuyển qua màn hình Rate cho phép người dùng gửi thông tin đánh giá về điểm du lịch này lên máy chủ của hệ thống khuyến nghị. Nút *Show on map* cho phép hiển thị địa điểm lên bản đồ (Google Maps) và đường đi từ điểm hiện tại đến điểm đó (có hướng dẫn đường đi bằng lời). Nút *Add favorite* cho phép người dùng lưu thông tin địa điểm này xuống cơ sở dữ liệu cục bộ trên điện thoại. Những điểm du lịch ưa thích sẽ được hiển thị ở màn hình Favorites. Khi người dùng nhấn lên số điện thoại, điện thoại sẽ thực hiện cuộc gọi, nhấn lên trang web, sẽ hiển thị nội dung trang web của địa điểm.



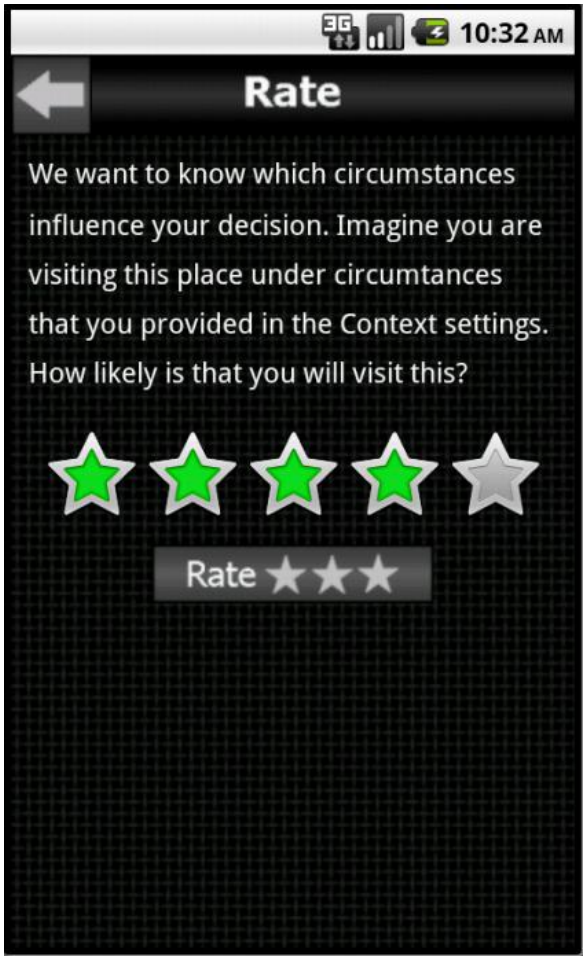
Hình 6.7: Màn hình Map.

Hình 6.8: Màn hình Map – Directions.

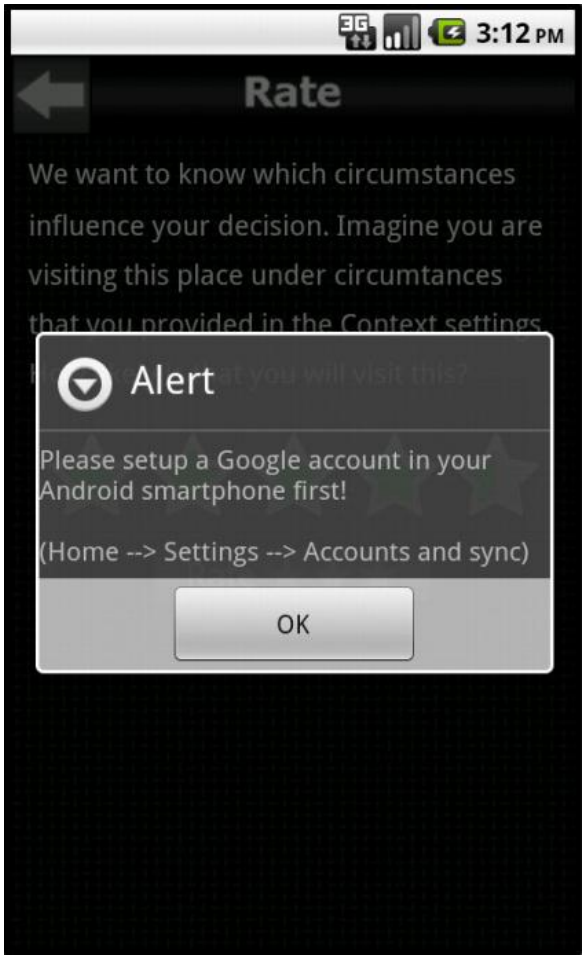
Mô tả hình 6.7 và hình 6.8:

Màn hình Map hiển thị vị trí của người dùng hiện tại và vị trí điểm du lịch lên bản đồ Google Maps. Kèm theo đó là đường đi giữa hai điểm. Nút *Direction* ở góc phải bên trên hiển chị chỉ dẫn đường đi nối giữa vị trí hiện tại của người dùng tới điểm du lịch bằng văn bản. Thông tin chỉ dẫn này được lấy từ dịch vụ web CloudMade.

Khi triển khai ứng dụng Android đến người dùng, để hiển thị được bản đồ Google Maps lên màn hình điện thoại, nhà phát triển ứng dụng bắt buộc phải đăng ký với Google một mã sử dụng gọi là *"Google Maps API key"*. Nhóm tác giả sẽ trình bày cách đăng ký này ở phần phụ lục C cuối quyển báo cáo.



Hình 6.9: Màn hình Rate.

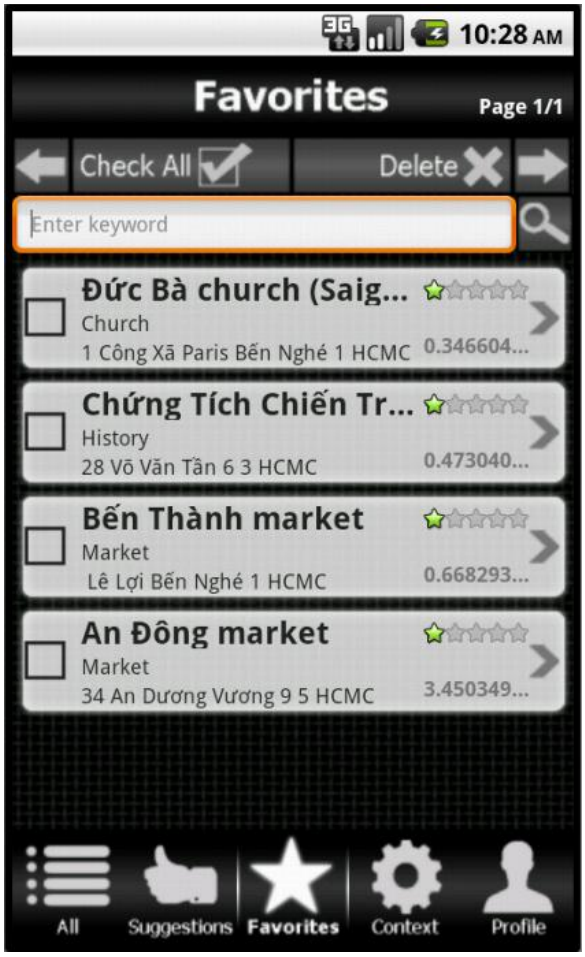


Hình 6.10: Màn hình Rate. (2)

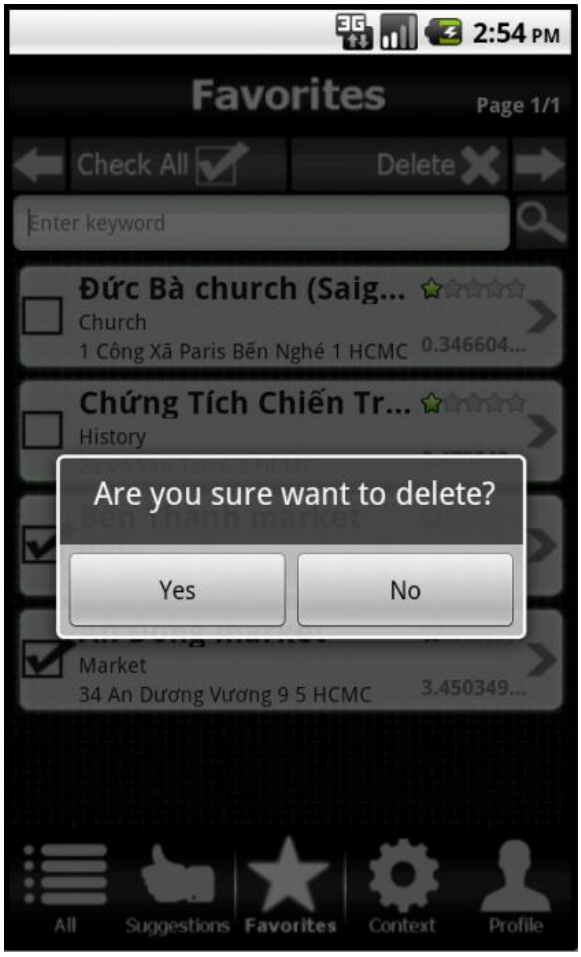
Mô tả hình 6.9 và hình 6.10:

Màn hình Rate với nút *Rate* cho phép người dùng gửi thông tin đánh giá (từ 1 đến 5, có thể chọn số lẻ 1.5, 2.5, 3.5, 4.5) cho địa điểm du lịch đang xét về máy chủ khuyến nghị. Thông tin đánh giá này ứng với các điều kiện ngữ cảnh được lưu bên tab Context.

Nếu người dùng chưa có tài khoản Google trong máy, chương trình sẽ yêu người dùng thiết lập tài khoản. Khi đó, lúc đánh giá, thông tin tài khoản Google sẽ được tự động lưu vào cơ sở dữ liệu trên máy chủ giúp phân biệt giữa các người dùng khác nhau. Cách làm này giúp tiện lợi hơn cho người dùng trong việc đăng ký tài khoản trong hệ thống khuyến nghị, hạn chế các thao tác tạo tài khoản, tạo tên truy cập, mật khẩu … có thể gây phiền phức đến cho người dùng.



Hình 6.11: Màn hình Favorites.



Hình 6.12: Màn hình Favorites. (2)

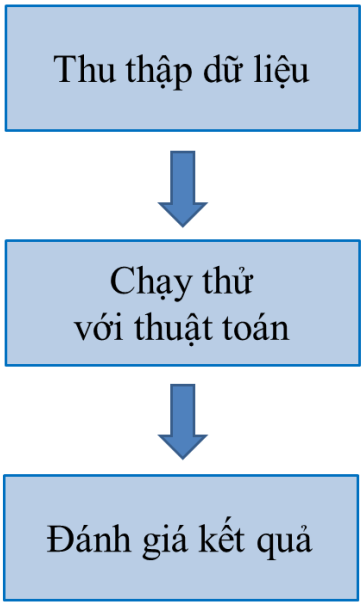
Mô tả hình 6.11 và hình 6.12:

Màn hình Favorites hiển thị những địa điểm du lịch ưa thích được người dùng lưu lại trong cơ sở dữ liệu cục bộ trên điện thoại.

Nút duyệt trang (previous, next) duyệt mỗi trang hiển thị 5 địa điểm du lịch. Nút *Check All* (đánh chọn tất cả địa điểm) để xóa. Nút *Delete* để xóa địa điểm, có thể xóa một hoặc nhiều địa điểm được chọn cùng lúc. Trước khi xóa, chương trình hiển thị yêu cầu xác nhận xóa, nếu đồng ý mới tiến hành xóa. Sau cùng là nút tìm kiếm hình chiếc kính lúp góc trên bên phải giúp tìm kiếm một địa điểm theo tên.

6.2. Thực nghiệm và đánh giá:

Quá trình thực nghiệm và đánh giá được thực hiện nhằm mục đích kiểm tra khả năng khuyến nghị của hệ thống trong thực tế với phương pháp đã được trình bày ở chương 3. Toàn bộ quá trình được minh họa khái quát qua hình vẽ sau đây:



Hình 6.13: Mô hình thực nghiệm.

Quá trình gồm ba bước chính là thu thập dữ liệu, chạy thử với thuật toán và cuối cùng là đánh giá kết quả. Cụ thể từng bước được thực hiện chi tiết như sau:

6.2.1. Thu thập dữ liệu:

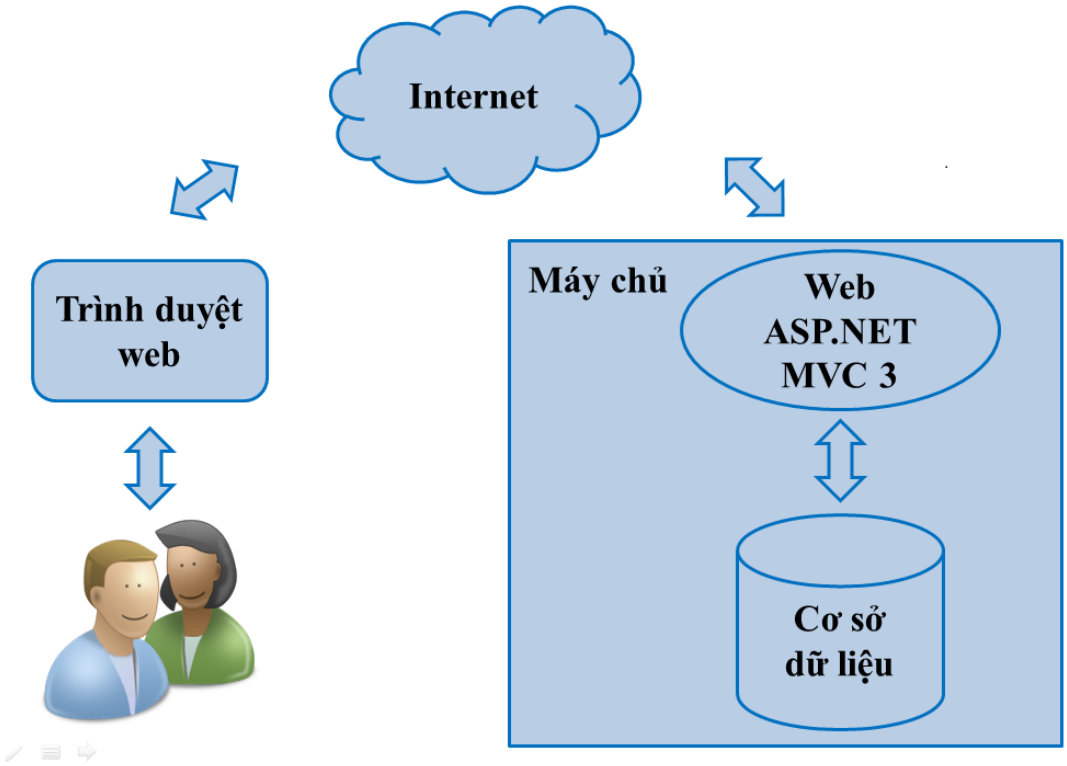
Như đã trình bày ở các chương trước, hệ thống khuyến nghị du lịch được xây dựng nhằm mục đích gợi ý những địa điểm du lịch đến người dùng tùy theo điều kiện ngữ cảnh của họ. Để gợi ý được, hệ thống cần có cơ sở dữ liệu các chỉ số đánh giá về các địa điểm du lịch của những người đã đi trước đó. Nhóm tác giả đã tìm kiếm trên Internet nhưng không tìm thấy tập dữ liệu mẫu nào có chứa các điều kiện ngữ cảnh thích hợp có thể sử dụng. Vì vậy, nhóm tác giả đã quyết định tự thu thập dữ liệu từ những người dùng thực tế.

Dữ liệu được thu thập cần chứa những thông tin sau đây, trong đó có các thông tin ngữ cảnh đã được mô tả ở chương 1, mục 1.3:

- ✓ Email: email người dùng được dùng để phân biệt những người dùng khác nhau trong hệ thống.
- ✓ Địa điểm du lịch: thông tin địa điểm du lịch người dùng đi.
- ✓ Thời tiết: thời tiết khi đi du lịch đến địa điểm được chọn.
- ✓ Kinh phí du lịch: kinh phí cho chuyến đi.
- ✓ Bạn đồng hành: bạn đồng hành trong chuyến đi.
- ✓ Thời gian du lịch: thời gian thực hiện chuyến đi.

Do thời gian giới hạn nên nhóm tác giả quyết định chỉ tập trung vào 20 địa điểm nổi bật nhất trong thành phố Hồ Chí Minh được đa số người dùng biết đến. Để đạt được lượng dữ liệu đánh giá cần thiết, nhóm tác giả thực hiện cùng lúc hai hình thức thu thập dữ liệu là thu thập qua web và thu thập qua ứng dụng Android. Trước khi tiến hành triển khai ứng dụng thực tế và thu thập, cần phải thực hiện việc cài đặt một máy chủ tại nhà riêng, mở port cho modem, thiết lập các thông số cấu hình cho Internet Information Services (IIS) phiên bản 7.5, cơ sở dữ liệu SQL Server phiên bản 2008 kèm các công cụ khác như OLAP Cube, Data Warehouse …

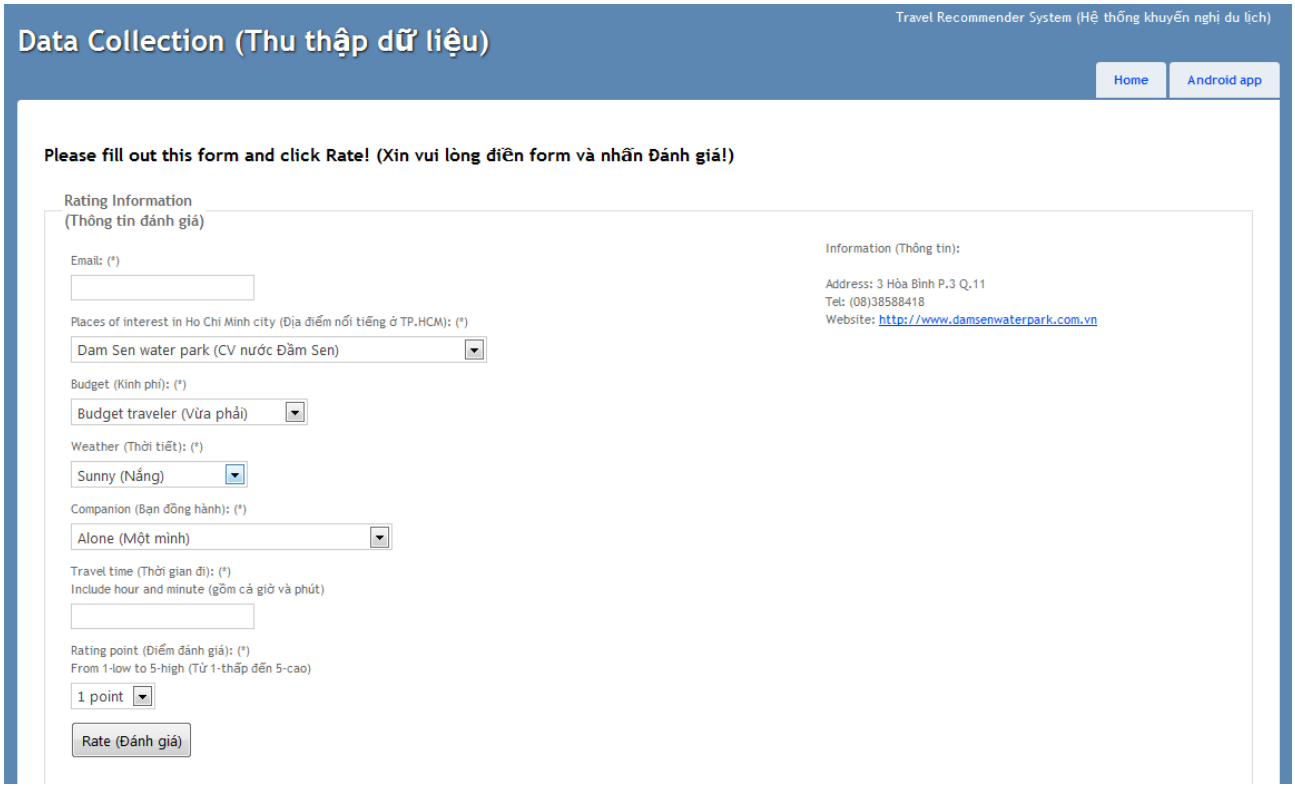
Quá trình thu thập qua web được thực hiện bằng cách xây dựng nhanh một trang web dùng công nghệ ASP.NET MVC 3 của Microsoft. Mô hình thu thập dữ liệu trên web và đưa dữ liệu vào cơ sở dữ liệu trên máy chủ như sau:



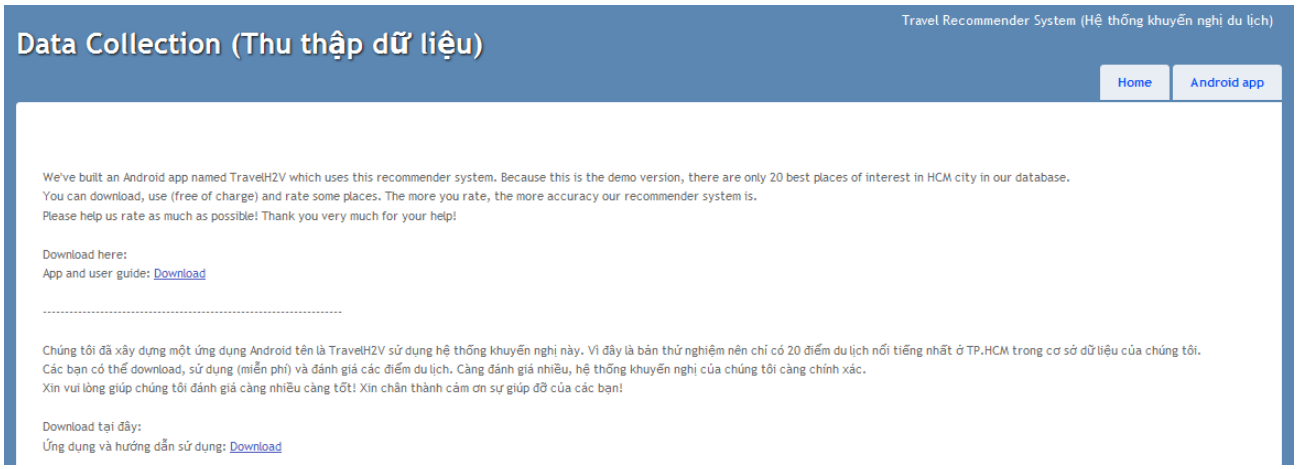
Hình 6.14: Thu thập dữ liệu qua web.

Người dùng truy cập vào trang web thu thập dữ liệu của nhóm, cung cấp các thông tin được yêu cầu. Thông tin được gửi từ trình duyệt web thông qua Internet đến máy chủ và được cập nhật xuống cơ sở dữ liệu. Trang web cũng bố trí một tab cho phép người dùng tải về ứng dụng Android nhóm tác giả xây dựng.

Sau đây là một số hình ảnh của trang web thu thập dữ liệu:

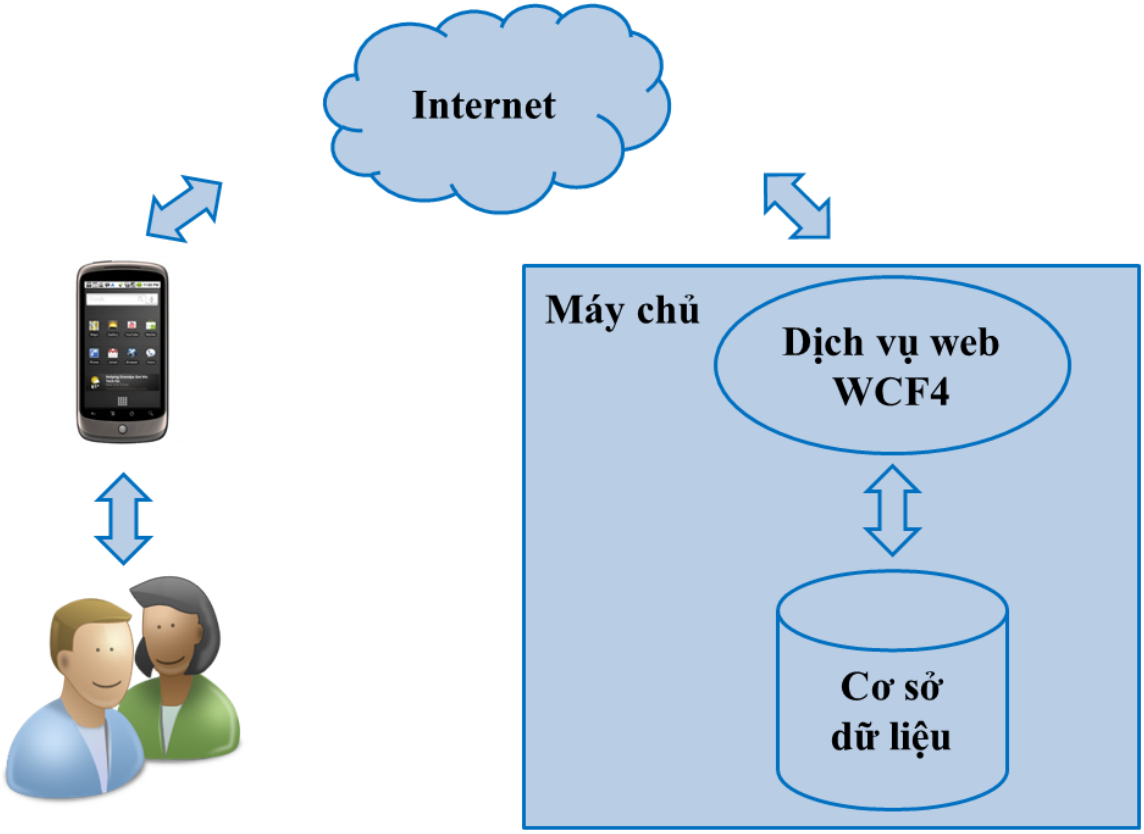


Hình 6.15: Trang thu thập dữ liệu.



Hình 6.16: Trang tải về ứng dụng Android.

Quá trình thu thập qua ứng dụng Android cũng gần tương tự. Trong ứng dụng Android, tab Context cho phép người dùng chọn điều kiện ngữ cảnh, email người dùng sẽ là tài khoản Google được thiết lập trên điện thoại. Khi người dùng vào xem thông tin chi tiết địa điểm và thực hiện đánh giá ở màn hình Rate, những thông đó sẽ được gửi lên Internet đến máy chủ, thông qua dịch vụ web WCF4, được cập nhật xuống cơ sở dữ liệu.



Hình 6.17: Thu thập dữ liệu qua ứng dụng Android.

6.2.2. Chạy thử với thuật toán:

Ở phần này, nhóm tác giả sẽ tiến hành chạy thử thuật toán theo hai hướng: một là thực nghiệm với tập dữ liệu không chứa các điều kiện ngữ cảnh (dùng tập dữ liệu Movielens có sẵn trên Internet), hai là thực nghiệm với tập dữ liệu chứa các điều kiện ngữ cảnh (dùng tập dữ liệu du lịch được thu thập thực tế ở trên). Qua đó, nhóm tác giả muốn đánh giá độ chính xác của thuật toán trong mỗi trường hợp cũng như đánh giá xem điều kiện ngữ cảnh ảnh hưởng thế nào đến việc khuyến nghị. Kết quả khuyến nghị sẽ ra sao khi có và khi không có điều kiện ngữ cảnh kèm theo?

a) Thực nghiệm với tập dữ liệu Movielens:

Movielens là một tập dữ liệu được nhóm nghiên cứu GroupLen (thuộc ngành Khoa Học Máy Tính trường Đại Học Minnesota tại Mỹ) thu thập được từ trang web http://movielens.umn.edu. Tập dữ liệu này là những đánh giá của rất nhiều người dùng đối với các bộ phim khác nhau. Chất lượng của tập dữ liệu được đánh giá là tốt do đã được các chuyên gia xử lý, chọn lọc và được sử dụng làm tập dữ liệu mẫu trong rất nhiều những nghiên cứu khác.

Có ba tập với kích cỡ khác nhau có thể được tải về từ trang web http://www.grouplens.org/:

- MovieLens 100k – gồm 100.000 đánh giá từ 1.000 người dùng đối với 1.700 phim.
- MovieLens 1M – gồm 1 triệu đánh giá từ 6.000 người dùng đối với 4.000 phim.
- MovieLens 10M – gồm 10 triệu đánh giá từ 72.000 người dùng đối với 10.000 phim.

Nhóm tác giả chọn tập dữ liệu MovieLens 100k để thực nghiệm. Do đây là tập dữ liệu chỉ có hai chiều (người dùng và phim), không có các chiều ngữ cảnh nên nhóm tác giả chỉ sử dụng mô hình hồi quy trong phương pháp khuyến nghị hai chiều được đề cập ở chương 3, mục 3.3, không dùng đến kỹ thuật thu giảm số chiều.

Tiêu chí đánh giá được lựa chọn là Mean Absolute Error (MAE). Với MAE, ta có thể dễ dàng hiểu và nhận thấy được độ sai lệch trong kết quả dự đoán của thuật toán.

Ví dụ: MAE = 1 nghĩa là thuật toán có khả năng dự đoán các chỉ số với độ chính xác (hay còn gọi là sai số) là ± 1.

Để thực nghiệm, nhóm tác giả dùng kỹ thuật Cross Validation với n = 10 đã được trình bày ở chương 3. Đầu tiên, tập dữ liệu MovieLens được chia ngẫu nhiên ra thành 10 phần bằng nhau (mỗi phần có 10.000 đánh giá). Sau đó, nhóm tác giả tiến hành chạy thử thuật toán 10 lần:

– Lần 1: tập dữ liệu đánh giá là phần 1, tập dữ liệu huấn luyện là các phần còn lại 2, 3, 4, 5, 6, 7, 8, 9, 10.

– Lần 2: tập dữ liệu đánh giá là phần 2, tập dữ liệu huấn luyện là các phần còn lại 1, 3, 4, 5, 6, 7, 8, 9, 10.

– Lần 3: tập dữ liệu đánh giá là phần 3, tập dữ liệu huấn luyện là các phần còn lại 1, 2, 4, 5, 6, 7, 8, 9, 10.

– …

– Lần 10: tập dữ liệu đánh giá là phần 10, tập dữ liệu huấn luyện là các phần còn lại 2, 3, 4, 5, 6, 7, 8, 9, 10.

Ở mỗi lần chạy, chỉ số MAE được tính toán và ghi nhận. Sau 10 lần, nhóm tác giả sẽ lấy trung bình cộng và được chỉ số MAE trung bình. Cụ thể, bảng kết quả như sau:

Lần	1	2	3	4	5	6	7	8	9	10
MAE	0.7456	0.7709	0.7522	0.7628	0.7438	0.7552	0.7472	0.7509	0.7775	0.7722

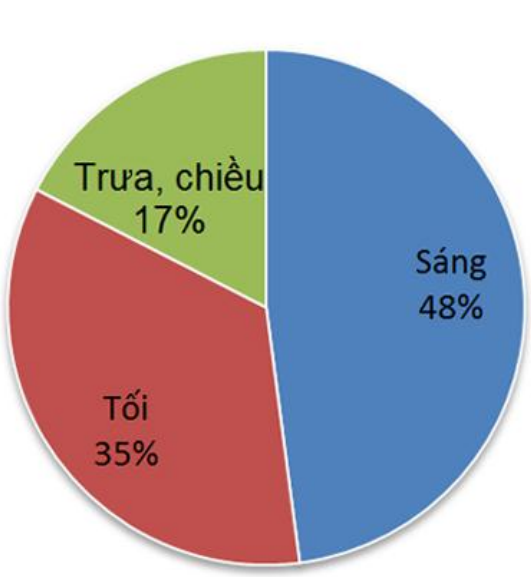
Từ đó tính được chỉ số MAE trung bình của 10 lần chạy trên là 0.7578.

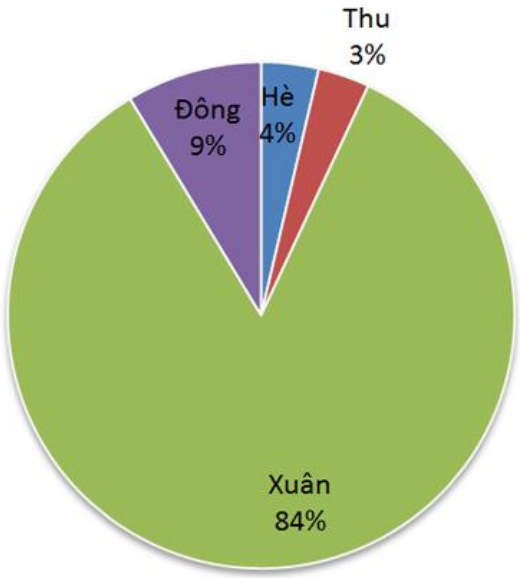
b) Thực nghiệm với tập dữ liệu du lịch thu thập thực tế:

Sau hơn ba tuần thu thập dữ liệu qua trang web, nhóm tác giả ghi nhận có *820* đánh giá trên *20* địa điểm được phản hồi từ *178* người dùng khác nhau. Với các số liệu thống kê các chỉ số đánh giá cụ thể như sau kèm một vài nhận xét. Những nhận xét này dựa trên lượng dữ liệu tương đối nhỏ chỉ từ *178* người dùng, có thể đúng trong phạm vi số người tham gia đánh giá này, nhưng với một lượng người dùng lớn hơn, có thể sẽ khác:

❖ Thời gian theo ngày, theo mùa:

Như đã trình bày ở phần trên, nhóm tác giả xây dựng cấu trúc phân cấp của thời gian, từ thời điểm đánh giá (ngày, tháng, năm, giờ, phút) nâng nên thành các cấp cao hơn như ngày, tuần, tháng, quý, mùa … Ở đây, nhóm tác giả thống kê và nhận xét ở hai cấp là ngày (buổi sáng, buổi trưa, buổi tối) và mùa (xuân, hạ, thu, đông) do chúng có sự chênh lệch tỉ lệ giữa các miền giá trị tương đối lớn.

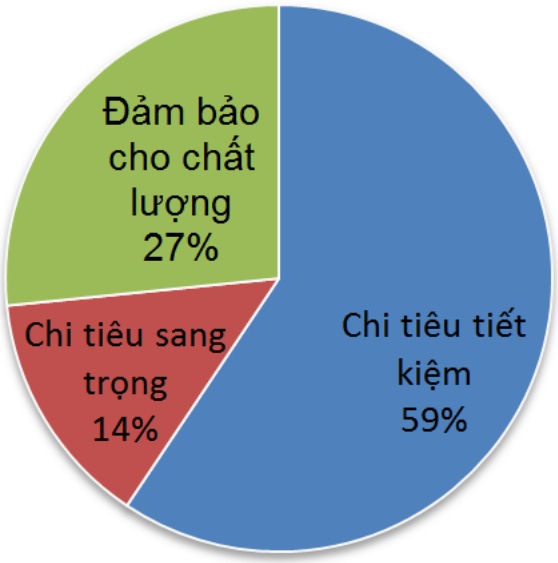




✓ Nhận xét: số người thích đi du lịch vào buổi sáng chiếm tỉ lệ đông nhất *48%*, kế đến là buổi tối. Vào buổi trưa, số người đi không nhiều bằng. Mùa xuân là mùa

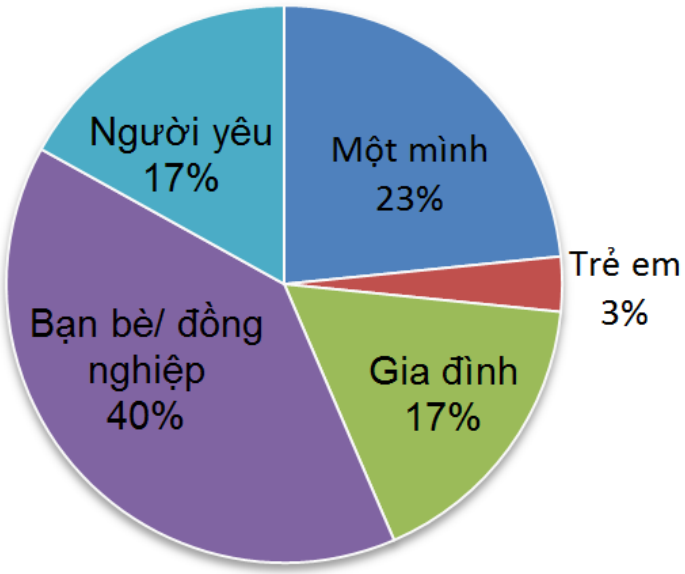
được chọn để du lịch nhiều nhất (chiếm tỉ lệ đến *84%*). Các mùa còn lại chiếm tỉ lệ ít hơn rất nhiều. Điều này cũng phù hợp với tâm lý thường thấy của người du lịch. Tuy nhiên, các đánh giá tập trung vào mùa xuân ở đây có lẽ vì khoảng thời gian nhóm tác giả tiến hành thu thập đánh giá là vào tháng một, người dùng chỉ nhớ đến những chuyến đi gần nhất trong thời gian đó.

❖ Theo kinh phí du lịch:



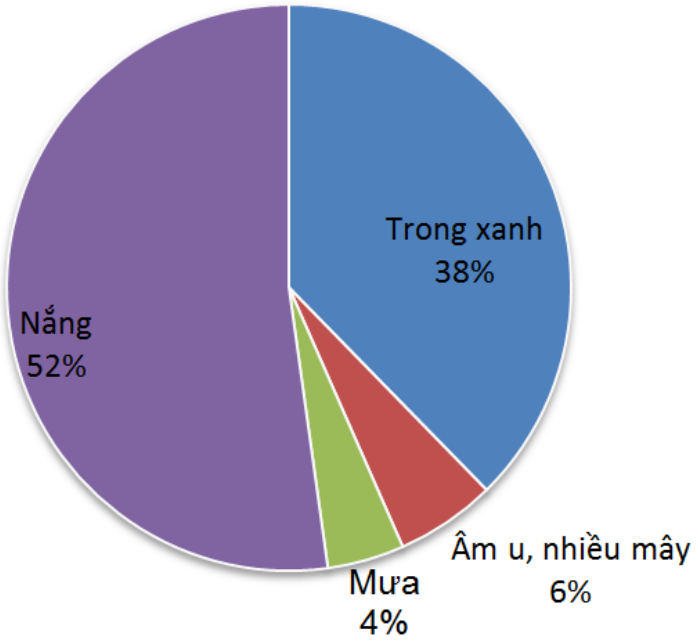
✓ Nhận xét: người đi du lịch cũng rất quan tâm đến vấn đề chi tiêu và tiết kiệm cho các chuyến đi. Số lượng người chọn cách chi tiêu tiết kiệm chiếm phần đông với tỉ lệ *59%*. Số lượng người có thu nhập cao, chịu chi tiêu sang trọng chiếm tỉ lệ thấp nhất *14%*.

❖ Bạn đồng hành:



✓ Nhận xét: đối tượng bạn bè hoặc đồng nghiệp được chọn lựa để đi du lịch cùng nhau chiếm phần lớn nhất *40%*. Trẻ em là đối tượng ít được lựa chọn làm người đồng hành trên các chuyến đi, chiếm tỉ lệ rất thấp chỉ *3%*. Ở đây, tỉ lệ chọn bạn bè cao nhất là bình thường. Tuy nhiên, tỉ lệ chọn trẻ em rất thấp có lẽ vì độ tuổi của những người tham gia đánh giá từ 20 đến 25 tuổi, đa số chưa lập gia đình, chưa có con nhỏ để trải nghiệm trong các chuyến du lịch. Và tâm lý ngại phải giữ trẻ nhỏ trong các chuyến đi chơi cũng là tâm lý thường thấy.

❖ Thời tiết:



✓ Nhận xét: phần đông mọi người thích đi du lịch vào những lúc thời tiết tốt. Trên đây trời nắng chiếm tỉ lệ cao nhất *52%*, kế đến là trời trong xanh chiếm *38%*. Thời tiết xấu không là thời điểm lý tưởng cho các chuyến du lịch, và số người chọn đi trong thời tiết đó là rất ít, trời mưa hay trời âm u chỉ chiếm tỉ lệ *10%*.

Các con số thống kê trên cho thấy các chỉ số đánh giá không trải đều cho từng điều kiện ngữ cảnh khác nhau. Có những điều kiện ngữ cảnh được nhiều người dùng quan tâm chọn lựa, cũng có những điều kiện ngữ cảnh người dùng ít chọn đến khi muốn đi du lịch. Qua đó cho thấy rõ ràng những điều kiện ngữ cảnh có ảnh hưởng đến việc người dùng chọn lựa địa điểm du lịch, và đương nhiên cũng ảnh hưởng đến mức độ thích thú, chỉ số đánh giá dành cho địa điểm du lịch đó.

Sau những nhận xét ban đầu trên, nhóm tác giả tiến hành thực nghiệm. Phương pháp thực nghiệm tương tự với tập dữ liệu MovieLens là dùng kỹ thuật Cross Validation với n = 10 và chỉ số MAE. Nhưng ở phần này nhóm tác giả sử

dụng tập dữ liệu du lịch thu thập thực tế (có chứa các điều kiện ngữ cảnh là thời gian, thời tiết, bạn đồng hành, kinh phí). Không gian khuyến nghị ở đây không còn là hai chiều nữa mà trở thành nhiều nhiều. Do đó, nhóm tác giả dùng pương pháp thu giảm số chiều kết hợp mô hình hồi qui đã được trình bày ở chương 3, mục 3.2 và 3.3 để tiến hành thực nghiệm.

Pha thứ nhất của phương pháp là xác định các phân khúc dữ liệu mạnh. Đầu tiên, xác định tổ hợp các phân khúc dữ liệu có thể có. Tiếp đến, với mỗi phân khúc dữ liệu, chia thành 10 phần, thực hiện 10 lần chạy của kỹ thuật Cross Validation. Ở mỗi lần chạy, tiến hành áp dụng thuật toán hồi qui trên phân khúc dữ liệu và trên toàn bộ dữ liệu (dùng cùng một bộ dữ liệu kiểm thử) để so sánh đánh giá. Sau 10 lần, sẽ tính được chỉ số MAE trung bình trên phân khúc dữ liệu đó cũng như chỉ số MAE trung bình trên dữ liệu toàn cục tương ứng với bộ dữ liệu kiểm thử được dùng.

Sau đó, chọn lọc ra những phân khúc dữ liệu mạnh dựa vào chỉ số MAE tương ứng (chỉ số MAE của phân khúc dữ liệu nhỏ hơn chỉ số MAE trên dữ liệu toàn cục).

Cuối cùng, trong tập các phân khúc dữ liệu mạnh vừa chọn lọc trên, nhóm tác giả tiến hành loại bỏ tất cả những phân khúc dữ liệu S nếu tồn tại một phân khúc dữ liệu Q sao cho $S \subset Q$ và $MAE(Q) < MAE(S)$ theo đúng các bước trong thuật toán. Qua các bước trên, bảng phân khúc dữ liệu được giữ lại sau cùng (theo thứ tự MAE tăng dần) là:

Phân khúc dữ liệu						**MAE (Dữ liệu cục bộ)**	**MAE (Dữ liệu toàn cục)**
Thời gian			**Kinh phí**	**Bạn đồng hành**	**Thời tiết**		
Thời điểm trong ngày	**Thời điểm trong tuần**	**Mùa**					
Sáng	Cuối tuần	Xuân	Cao	*	*	0.43372	0.44670
*	*	Xuân	Vừa đủ	*	Trong xanh	0.48457	0.49133
*	*	*	*	Người yêu	*	0.56927	0.57273
*	*	*	Vừa đủ	*	Trong xanh	0.57842	0.63945
Sáng	*	Xuân	Cao	*	*	0.60074	0.61275
Sáng	Trong tuần	*	*	*	Nắng	0.60995	0.67334
Sáng	*	*	Cao	*	*	0.61212	0.61542
Tối	Cuối tuần	Xuân	*	*	*	0.61438	0.61942
*	*	*	*	Bạn bè	Trong xanh	0.61633	0.61754
*	Trong tuần	*	Vừa đủ	*	*	0.63945	0.64539

*	Cuối tuần	*	*	*	Trong xanh	0.67543	0.81829
*	Cuối tuần	Xuân	*	Bạn bè	*	0.68348	0.76044
Sáng	*	Xuân	Vừa đủ	*	*	0.69206	0.90043
Tối	*	Xuân	*	*	Trong xanh	0.69549	0.95583
*	Cuối tuần	Xuân	*	*	Nắng	0.71159	0.76932
Sáng	*	*	Vừa đủ	*	*	0.71413	0.91596
Tối	Cuối tuần	*	Vừa đủ	*	*	0.71544	0.71544
Sáng	*	Xuân	*	*	*	0.73028	0.76854
*	Cuối tuần	*	Vừa đủ	*	*	0.74461	0.88036
Sáng	*	*	*	Bạn bè	*	0.74949	0.76367
Sáng	*	*	*	*	Nắng	0.78363	0.96781
*	*	*	Vừa đủ	*	Nắng	0.78476	0.89335
*	*	Xuân	*	Bạn bè	*	0.81526	0.81996
Sáng	*	*	*	*	Trong xanh	0.82395	0.88522
*	*	Xuân	Cao	*	*	0.83838	0.91555
*	Trong tuần	*	*	*	Trong xanh	0.85681	0.92658
*	*	*	Vừa đủ	Một mình	*	0.86454	1.04721
*	Trong tuần	*	*	*	*	0.86742	0.89478
Tối	*	*	*	*	*	0.89246	1.05671
*	*	*	Cao	Bạn bè	*	0.98586	1.00841
*	*	Xuân	*	*	Trong xanh	1.11469	1.17238
*	*	Xuân	*	Một mình	*	1.13553	1.16709
*	*	*	Rất cao	*	*	1.19250	1.19337

Sau khi có được phân khúc dữ liệu mạnh, pha thứ hai là dự đoán các chỉ số đánh giá theo điều kiện ngữ cảnh người dùng cung cấp và trả về kết quả cho người dùng.

6.2.3. Đánh giá kết quả:

- Kết quả thực nghiệm độ sai lệch tuyệt đối của mô hình hồi qui trên bộ dữ liệu Movielens (100.000 dòng dữ liệu) là khá đều nhau và có độ lệch trung bình là *0.7578*. Kết quả này cũng tiệm cận với các nghiên cứu trước đây của các tác giả khác [9].
- Kết quả thực nghiệm trên dữ liệu thu thập cũng có kết quả trung bình tiệm cận với các con số nói ở trên: *0.75233*.
- Khi tiến hành đánh giá thuật toán trên từng phân khúc dữ liệu theo ngữ cảnh ta có thể thấy sự khác nhau giữa độ lệch khi áp dụng trên dữ liệu cục bộ và toàn cục là vào khoảng từ *0* đến *0.26*. Rõ ràng phương pháp thu giảm số chiều đã giúp cải thiện độ chính xác cao hơn. Nhưng cũng trong nhiều trường hợp khác (bị loại bỏ khỏi danh sách trên) thì phương pháp này lại

không có hiệu quả, vì vậy việc áp dụng kết hợp phương pháp thu giảm số chiều và phương pháp truyền thống trong khóa luận này là lựa chọn tối ưu.

- Ngoài ra ta có thể thấy độ chênh lệch giữa các phân khúc dữ liệu một cách rõ ràng (*0.43372 ~ 1.19250*) bởi xu hướng đánh giá của người dùng ở điều kiện ngữ cảnh này có thể có độ tương đồng cao nhưng với điều kiện ngữ cảnh khác thì lại thấp, ta cũng không loại trừ trường hợp các người dùng "cá biệt" (đã đề cập ở mục 2.1.4) làm ảnh hưởng xấu tới kết quả. Để chứng minh một cách tương đối cho điều này, nhóm tác giả tiến hành tính toán thống kê các chỉ số sau (ứng với từng phân khúc dữ liệu):
  - Hệ số biến thiên CV: được tính bằng cách lấy độ lệch chuẩn chia cho giá trị trung bình. Hệ số này sẽ cho ta ý nghĩa về sự tương quan giữa các đánh giá của người dùng, nếu CV càng thấp nghĩa là người dùng càng có xu hướng đánh giá giống nhau, điều này sẽ có lợi cho việc dự đoán, ngược lại CV càng cao nghĩa là người dùng đánh giá càng khác nhau làm ảnh hưởng xấu đến việc dự đoán.

    Ví dụ có tập dữ liệu của một phân khúc như sau:

Địa điểm	**Đánh giá**
A	3
A	4
B	5
B	5
B	4
C	2
C	3
C	4
C	1

    Ta sẽ tính lần lượt hệ số biến thiên CV cho 3 địa điểm A, B, C theo công thức độ lệch chuẩn chia cho giá trị đánh giá trung bình. Mặt khác, vì số lượng đánh giá cho một địa điểm rõ ràng làm ảnh hưởng tới việc dự đoán nên 3 hệ số CV này phải được đánh trọng số khác nhau. Để đơn giản, nhóm tác giả tính trọng số bằng tỉ lệ số đánh giá cho điểm đó chia cho tổng số đánh giá của phân khúc dữ liệu.

Cuối cùng, hệ số biến thiên CV cho phân khúc dữ liệu S được tính toán theo các công thức sau:

$$CV_S = \sum_{P_i \in P} CV_{P_i} \times \frac{|P_i|}{|S|}$$

$$CV_{P_i} = \frac{\sqrt{E[(X - E(X))^2]}}{E(X)}$$

Với P là tập các địa điểm trong phân khúc dữ liệu S.

- Số lượng đánh giá trung bình cho một địa điểm và tổng số lượng đánh giá: như đã nói ở trên, số lượng đánh giá cho một địa điểm trong một phân khúc dữ liệu sẽ làm ảnh hưởng tới việc dự đoán nên chúng tôi tiến hành thống kê giá trị trung bình này cũng như tổng số lượng đánh giá cho tất cả địa điểm có trên phân khúc dữ liệu đó.

Theo đó, bảng kết quả thống kê dữ liệu của các phân khúc dữ liệu được trình bày như sau:

	Phân khúc dữ liệu						MAE (Dữ liệu cục bộ)	MAE (Dữ liệu toàn cục)	Hệ số biến thiên CV	Trung bình số đánh giá cho một địa điểm	Tổng số đánh giá
	Thời gian			Kinh phí	Bạn đồng hành	Thời tiết					
	Thời điểm trong ngày	Thời điểm trong tuần	Mùa								
1	Sáng	Cuối tuần	Xuân	Cao	*	*	0.43372	0.44670	0.158203177	4.785714286	67
2	*	*	Xuân	Vừa đủ	*	Trong xanh	0.48457	0.49133	0.181744161	7.421052632	141
3	*	*	*	*	Người yêu	*	0.56927	0.57273	0.18964369	7.722222222	139
4	*	*	*	Vừa đủ	*	Trong xanh	0.57842	0.63945	0.17714198	8.315789474	158
5	Sáng	*	Xuân	Cao	*	*	0.60074	0.61275	0.191853438	6.4	96
6	Sáng	Trong tuần	*	*	*	Nắng	0.60995	0.67334	0.189985244	6.7	134
7	Sáng	*	*	Cao	*	*	0.61212	0.61542	0.179638903	6.764705882	115
8	Tối	Cuối tuần	Xuân	*	*	*	0.61438	0.61942	0.115879779	2.3	23
9	*	*	*	*	Bạn bè	Trong xanh	0.61633	0.61754	0.189630478	6.777777778	122
10	*	Trong tuần	*	Vừa đủ	*	*	0.63945	0.64539	0.20807787	15.1	302
11	*	Cuối tuần	*	*	*	Trong xanh	0.67543	0.81829	0.17070911	8.45	169
12	*	Cuối tuần	Xuân	*	Bạn bè	*	0.68348	0.76044	0.19095328	7.444444444	134
13	Sáng	*	Xuân	Vừa đủ	*	*	0.69206	0.90043	0.211142609	10.26315789	195
14	Tối	*	Xuân	*	*	Trong xanh	0.69549	0.95583	0.153413557	5.611111111	101
15	*	Cuối tuần	Xuân	*	*	Nắng	0.71159	0.76932	0.206751108	7.842105263	149
16	Sáng	*	*	Vừa đủ	*	*	0.71413	0.91596	0.20625079	12.36842105	235
17	Tối	Cuối tuần	*	Vừa đủ	*	*	0.71544	0.71544	0.141677536	3.888888889	70
18	Sáng	*	Xuân	*	*	*	0.73028	0.76854	0.216655024	16.5	330
19	*	Cuối tuần	*	Vừa đủ	*	*	0.74461	0.88036	0.18784688	9.25	185

20	Sáng	*	*	*	Bạn bè	*	0.74949	0.76367	0.192736424	8.611111111	155
21	Sáng	*	*	*	*	Nắng	0.78363	0.96781	0.203756563	11.5	230
22	*	*	*	Vừa đủ	*	Nắng	0.78476	0.89335	0.21478557	14.3	286
23	*	*	Xuân	*	Bạn bè	*	0.81526	0.81996	0.202723593	13.57894737	258
24	Sáng	*	*	*	*	Trong xanh	0.82395	0.88522	0.185737381	7.157894737	136
25	*	*	Xuân	Cao	*	*	0.83838	0.91555	0.207247432	10.11111111	182
26	*	Trong tuần	*	*	*	Trong xanh	0.85681	0.92658	0.189441417	7.368421053	140
27	*	*	*	Vừa đủ	Một mình	*	0.86454	1.04721	0.176259894	7.7	154
28	*	Trong tuần	*	*	*	*	0.86742	0.89478	0.222321878	22.5	450
29	Tối	*	*	*	*	*	0.89246	1.05671	0.212568977	14.94736842	284
30	*	*	*	Cao	Bạn bè	*	0.98586	1.00841	0.186808772	6.235294118	106
31	*	*	Xuân	*	*	Trong xanh	1.11469	1.17238	0.197316145	13.25	265
32	*	*	Xuân	*	Một mình	*	1.13553	1.16709	0.199223364	8.75	175
33	*	*	*	Rất cao	*	*	1.19250	1.19337	0.226836744	9.583333333	115

Từ bảng số liệu trên, nhóm tác giả có một số nhận xét:

- Các chỉ số về hệ số biến thiên CV, số lượng đánh giá trung bình trên một địa điểm, tổng số lượng đánh giá trên tất cả địa điểm ở mỗi phân khúc đã phần nào phản ánh chính xác kết quả của việc thực nghiệm thuật toán. Theo bảng trên, phân khúc dữ liệu tốt nhất có hệ số biến thiên CV tương đối nhỏ: *0.158203177* (phân khúc 1), ngược lại hệ số biến thiên CV cao nhất là *0.226836744* (phân khúc 33) và nó cũng ứng với chỉ số MAE cao nhất (*1.19250*).
- Ở phân khúc 8, hệ số biến thiên CV thấp nhất là *0.115879779* tuy nhiên vì phân khúc này có quá ít đánh giá (*23* đánh giá) và trung bình số đánh giá cho một địa điểm thấp (*2.3*) nên MAE không đột biến cao (gần về *0* hơn) nhưng vẫn có giá trị tốt (*0.61438*).
- Khi so sánh hai phân khúc dữ liệu thì không hoàn toàn đúng khi kết luận một phân khúc dữ liệu có hệ số biến thiên thấp hơn, trung bình số đánh giá trên một địa điểm cao hơn và số lượng đánh giá nhiều hơn thì sẽ có MAE nhỏ hơn, chẳng hạn như cặp phân khúc 2 và 11. Các chỉ số của phân khúc số 11 đều rất tốt nhưng MAE lại không tốt (*0.81829*). Điều này có thể được lý giải bởi thật sự số lượng các đánh giá thu thập vẫn còn hạn chế (tổng số lượng đánh giá ở phân khúc này là *169*). Thuật toán thực hiện chia ngẫu nhiên tập này thành 10 phần bằng nhau (mỗi phần chỉ có khoảng *17* đánh giá, con số này là rất thấp). Sau đó tiến hành đánh giá độ sai lệch trung bình của 10 phần theo phương pháp Cross Validation. Như vậy, dù đã cố gắng thực hiện đánh giá khách quan nhưng quá trình ngẫu nhiên của việc chia dữ liệu vẫn có ảnh hưởng nhất định tới kết quả. Nhược điểm này sẽ được khắc phục nếu bộ dữ liệu thu thập lớn hơn và điều này cũng đã được chứng minh ở bộ thực nghiệm dữ liệu Movielens với 100.000 dòng dữ liệu, kết quả chạy thuật toán là tốt.

Trên đây là toàn bộ quá trình thực nghiệm, thống kê, đánh giá, nhận xét kết quả thu được.

CHƯƠNG 7
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

---oOo---

Chương này sẽ là phần kết luận chung về những gì nhóm tác giả đã thực hiện ở khóa luận này. Đồng thời cũng bàn luận về những hướng phát triển cho đề tài trong tương lai.

7.1. Kết luận:

Sau hơn bốn tháng làm việc và nghiên cứu, nhóm tác giả đã thành công bước đầu trong việc xây dựng một hệ thống khuyến nghị trong ngành du lịch mà trước hết, đối tượng phục vụ là những khách hàng sử dụng điện thoại thông minh nền tảng Android. Hệ thống có thể vận hành tương đối tốt các chức năng cơ bản đã được trình bày ở các chương trước. Bên cạnh đó, nhóm tác giả đã xây dựng thành công một hệ khuyến nghị nhiều chiều giúp hệ thống có thể dựa vào đó đưa ra những lời gợi ý về các địa điểm du lịch cho du khách. Tuy nhiên, hệ thống hiện tại cần được phát triển nhiều hơn nữa mới có thể triển khai rộng rãi vào thực tế. Cụ thể, nguồn cơ sở dữ liệu cần được làm giàu thêm, sau đó tìm cách quảng bá hệ thống đến nhiều người dùng hơn nữa để họ tiến hành sử dụng và đánh giá những điểm du lịch. Vì có càng nhiều chỉ số đánh giá của người dùng, hệ thống có thể đưa ra những lời gời ý càng chính xác.

Qua quá trình xây dựng hệ thống trên, nhóm tác giả đã có cơ hội củng cố những gì đã biết và học hỏi thêm nhiều kiến thức mới như kiến thức về những phương pháp khuyến nghị, xây dựng các ứng dụng Android, xây dựng các dịch vụ web, xây dựng cơ sở dữ liệu, ... Trong đó, quý giá nhất có thể nói đến mảng kiến thức về những phương pháp khuyến nghị. Những hệ thống khuyến nghị giúp ích được rất nhiều cho người sử dụng trong thực tế, giúp những thông tin phù hợp có thể đến với họ nhanh hơn những cách thức bình thường khác. Do thời gian có hạn nên nhóm tác giả vẫn chưa thể hiểu thấu đáo toàn bộ những phương pháp khuyến nghị, vẫn còn đó rất nhiều phương pháp cần được tìm hiểu và nghiên cứu thêm. Và hy vọng với những kiến thức tích lũy được trong khóa luận này sẽ là nền tảng giúp nhóm tác giả có thể phát triển tốt hơn những hệ thống khuyến nghị khác trong tương lai.

Nhìn chung, nhóm tác giả nhận xét rằng những gì thực hiện trong khóa luận này đã đáp ứng được yêu cầu và mục tiêu được đề ra ban đầu.

7.2. Hướng phát triển:

Như đã nói trên, còn rất nhiều việc cần thực hiện trong tương lai để có một hệ thống khuyến nghị hoàn hảo hơn nữa. Cụ thể, hệ thống có thể được mở rộng theo những hướng như sau:

- ✓ Nghiên cứu cài đặt thêm những thuật toán khuyến nghị khác vào hệ thống.
- ✓ Nghiên cứu mở rộng hệ thống khuyến nghị qua những lĩnh vực khác: phim ảnh, sách vở, mua sắm, ... chứ không chỉ là lĩnh vực du lịch như hiện tại.
- ✓ Mở rộng ứng dụng sang các nền tảng hệ điều hành điện thoại khác như Windows Phone, iOS ...
- ✓ Mở rộng ứng dụng sang nền tảng web cho phép những người không có điện thoại thông minh vẫn có thể sử dụng được.
- ✓ Phát triển thêm những chức năng khác cho ứng dụng trên điện thoại.

TÀI LIỆU THAM KHẢO

---oOo---

1. Francesco Ricci, L.R., Bracha Shapira, Paul B. Kantor, *Recommender Systems Handbook* 2011: Springer.
2. Xiaoyuan Su, T.M.K., *A Survey of Collaborative Filtering Techniques*, 2009.
3. Ricardo Baeza, Y.B.R., Neto, *Modern Information Retrieval* 1999: Addison - Wesley.
4. Daniel Billsus, M.J.P., *Learning collaborative information filters*, 1998.
5. Upendra Shardanand, P.M., *Social information filtering: Algorithms for automating 'word of mouth'*, 1995.
6. Paul Resnick, N.I., Mitesh Suchak, Peter Bergstrom, John Riedl, *An open architecture for collaborative filtering of netnews*, 1994.
7. John S. Breese, D.H., Carl Kadie, *Empirical analysis of predictive algorithms for collaborative filtering*, 1998.
8. Chumki Basu, H.H., William Cohen, *Recommendation as classification: using social and content-based information in recommendation*, 1998.
9. Slobodan Vucetic, Z.O., *Collaborative filtering using a regression-based approach*, 2005.
10. Canny, J., *Collaborative filtering with privacy via factor analysis*, 2002.
11. Gediminas Adomavicius, R.S., Shahana Sen, Alexander Tuzhilin *Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach*, 2005.
12. Gediminas Adomavicius, A.T., *Multidimensional Recommender Systems: A Data Warehousing Approach*, 2001.
13. Murphy, M., *Beginning Android 2* 2010: Apress.
14. Sayed Y. Hashimi, S.K., Dave MacLean *Pro Android 2* 2010: Apress.

PHỤ LỤC

---oOo---

Phụ lục A: Mô tả chi tiết các use cases:

Mã số	Tên use case
UC_01	Xem tất cả các địa điểm.
UC_02	Xem các địa điểm được hệ thống gợi ý.
UC_03	Xem các địa điểm ưa thích được người dùng lưu lại.
UC_04	Xem và cấu hình điều kiện ngữ cảnh.
UC_05	Xem và cấu hình thông tin cá nhân.
UC_06	Xem danh sách địa điểm được sắp xếp theo thứ tự.
UC_07	Xem thông tin chi tiết của một địa điểm.
UC_08	Đánh giá địa điểm.
UC_09	Hiển thị địa điểm lên bản đồ.
UC_10	Thêm địa điểm vào danh sách ưa thích.
UC_11	Xóa địa điểm khỏi danh sách ưa thích.
UC_12	Tìm kiếm địa điểm ưa thích.

a) Use case "Xem tất cả các địa điểm":

Mã số	UC_01	
Tên	Xem tất cả các địa điểm	
Mục đích	Người dùng muốn hiển thị danh sách tất cả các địa điểm lên màn hình điện thoại.	
Actor	Người dùng điện thoại.	
Tiền điều kiện	Điện thoại có kết nối Internet (có thể dùng Wi-fi hoặc 3G).	
Hậu điều kiện	Danh sách tất cả địa điểm được hiển thị.	
Dòng sự kiện chính	1. Người dùng chọn tab All.	2. Điện thoại gửi yêu cầu truy vấn tất cả địa điểm đến máy chủ. 3. Máy chủ xử lý yêu cầu và trả về

		chuỗi JSON chứa những thông tin địa điểm. 4. Màn hình điện thoại hiển thị danh sách tất cả địa điểm.
Các xử lý ngoại lệ	**UC_01_2E:** nếu điện thoại không kết nối được Internet thì hiển thị thông báo yêu cầu kiểm tra kết nối Internet.	

b) Use case "Xem các địa điểm được hệ thống gợi ý":

Mã số	UC_02	
Tên	Xem các địa điểm được hệ thống gợi ý.	
Mục đích	Người dùng muốn hiển thị danh sách các địa điểm được hệ thống gợi ý (tùy theo điều kiện ngữ cảnh của người dùng) lên màn hình điện thoại.	
Actor	Người dùng điện thoại.	
Tiền điều kiện	Điện thoại có kết nối Internet (có thể dùng Wi-fi hoặc 3G). Người dùng đã cấu hình thông tin các ngữ cảnh ở tab Context.	
Hậu điều kiện	Danh sách tất cả địa điểm gợi ý được hiển thị.	
Dòng sự kiện chính	1. Người dùng chọn tab Suggestions.	2. Điện thoại gửi yêu cầu truy vấn các địa điểm cần gợi ý đến máy chủ. 3. Máy chủ xử lý yêu cầu và trả về chuỗi JSON chứa những thông tin địa điểm được gợi ý. 4. Màn hình điện thoại hiển thị danh sách địa điểm được gợi ý.
Các xử lý ngoại lệ	**UC_02_1E:** nếu người dùng chưa cấu hình các thông tin ngữ cảnh ở tab Context thì hiện thông báo yêu cầu cấu hình và chuyển sang tab Context. **UC_02_2E:** nếu điện thoại không kết nối được Internet thì hiển thị thông báo yêu cầu kiểm tra kết nối Internet.	

c) Use case "Xem các địa điểm ưa thích được người dùng lưu lại":

Mã số	UC_03	
Tên	Xem các địa điểm ưa thích được người dùng lưu lại.	
Mục đích	Người dùng muốn hiển thị danh sách các địa điểm ưa thích đã được lưu lại trong cơ sở dữ liệu cục bộ của điện thoại lên màn hình.	
Actor	Người dùng điện thoại.	
Tiền điều kiện	Không có.	
Hậu điều kiện	Danh sách tất cả địa điểm ưu thích của người dùng được hiển thị.	
Dòng sự kiện chính	1. Người dùng chọn tab Favorites.	2. Điện thoại kết nối cơ sở dữ liệu cục bộ SQLite. 3. SQLite xử lý yêu cầu và trả về những thông tin địa điểm ưa thích. 4. Màn hình điện thoại hiển thị danh sách địa điểm ưa thích.
Các xử lý ngoại lệ	**UC_03_2E:** nếu điện thoại không kết nối được SQLite thì hiển thị thông báo lỗi.	

d) Use case "Xem thông tin và cấu hình điều kiện ngữ cảnh":

Mã số	UC_04
Tên	Xem và cấu hình điều kiện ngữ cảnh.
Mục đích	Người dùng muốn xem và cấu hình các điều kiện ngữ cảnh.
Actor	Người dùng điện thoại.
Tiền điều kiện	Không có.
Hậu điều kiện	Các thông tin điều kiện ngữ cảnh người dùng được cập nhật trong cơ sở dữ liệu cục bộ SQLite.

Dòng sự kiện chính	1. Người dùng chọn tab Context. 5. Người dùng tùy chỉnh thông tin ngữ cảnh và nhấn nút Save.	2. Điện thoại kết nối cơ sở dữ liệu cục bộ SQLite. 3. SQLite truy vấn và trả về những thông tin ngữ cảnh. 4. Màn hình điện thoại hiển thị các thông tin ngữ cảnh. 6. Thông tin được cập nhật xuống cơ sở dữ liệu cục bộ.
Các xử lý ngoại lệ	**UC_04_2E:** nếu điện thoại không kết nối được SQLite thì hiển thị thông báo lỗi.	

e) Use case "Xem và cấu hình thông tin cá nhân":

Mã số	UC_05	
Tên	Xem và cấu hình thông tin cá nhân.	
Mục đích	Người dùng muốn xem và cấu hình thông tin cá nhân.	
Actor	Người dùng điện thoại.	
Tiền điều kiện	Không có.	
Hậu điều kiện	Các thông tin cá nhân người dùng được cập nhật trong cơ sở dữ liệu cục bộ SQLite.	
Dòng sự kiện chính	1. Người dùng chọn tab Profile.	2. Điện thoại kết nối cơ sở dữ liệu cục bộ SQLite. 3. SQLite truy vấn và trả về những thông tin cá nhân.

	5. Người dùng tùy chỉnh thông tin cá nhân và nhấn nút Save.	4. Màn hình điện thoại hiển thị các thông tin cá nhân. 6. Thông tin được cập nhật xuống cơ sở dữ liệu cục bộ.
Các xử lý ngoại lệ	**UC_04_2E:** nếu điện thoại không kết nối được SQLite thì hiển thị thông báo lỗi.	

f) Use case "Hiển thị danh sách địa điểm được sắp xếp theo thứ tự":

Mã số	UC_06	
Tên	Hiển thị danh sách địa điểm được sắp xếp theo thứ tự.	
Mục đích	Người dùng danh sách các địa điểm được hiển thị lên màn hình theo một thứ tự nào đó (sắp xếp theo tên, theo chỉ số đánh giá, theo khoảng cách, theo phân loại)	
Actor	Người dùng điện thoại.	
Tiền điều kiện	Điện thoại đã lấy về được danh sách các địa điểm.	
Hậu điều kiện	Danh sách các địa điểm được sắp xếp.	
Dòng sự kiện chính	1. Người dùng chọn nút A-Z (hoặc Rating, Distance, Type) tùy theo ý muốn sắp xếp theo kiểu nào.	2. Điện thoại xử lý danh sách địa điểm và sắp xếp theo yêu cầu. 3. Điện thoại hiện lại danh sách mới đã được sắp xếp.
Các xử lý ngoại lệ	Không có.	

g) Use case "Xem thông tin chi tiết của một địa điểm":

Mã số	UC_07	
Tên	Xem thông tin chi tiết của một địa điểm.	
Mục đích	Người dùng muốn xem thông tin chi tiết của một địa điểm	
Actor	Người dùng điện thoại.	
Tiền điều kiện	Điện thoại đã có danh sách các địa điểm hiển thị lên ở một trong các màn hình: All places, Suggestions, Favorites.	
Hậu điều kiện	Màn hình hiển thị thông tin chi tiết của địa điểm được hiển thị.	
Dòng sự kiện chính	1. Người dùng chọn một địa điểm trong danh sách các địa điểm đang hiển thị.	2. Điện thoại chuyển qua màn hình Details và hiển thị thông tin chi tiết của địa điểm được chọn.
Các xử lý ngoại lệ	Không có.	

h) Use case "Đánh giá địa điểm":

Mã số	UC_08
Tên	Đánh giá địa điểm.
Mục đích	Người dùng muốn đánh giá mức độ ưa thích đối với một địa điểm ứng với điều kiện ngữ cảnh họ đã thiết lập ở tab Context.
Actor	Người dùng điện thoại.
Tiền điều kiện	Đã đăng nhập vào hệ thống. Đang ở màn hình Details.
Hậu điều kiện	Thông tin đánh giá được gửi lên máy chủ.

Dòng sự kiện chính	1. Người dùng nhất nút Rate. 3. Ở màn hình Rate, người dùng chọn số sao muốn đánh giá và nhấn nút Rate.	2. Điện thoại chuyển qua màn hình Rate. 4. Điện thoại gửi thông tin đánh giá lên máy chủ và máy chủ cập nhật vào cơ sở dữ liệu.
Các xử lý ngoại lệ	**UC_08_4E:** nếu người dùng chưa đăng nhập thì hiện thông báo yêu cầu đăng nhập.	

i) Use case "Hiển thị địa điểm lên bản đồ":

Mã số	UC_09	
Tên	Hiển thị địa điểm lên bản đồ.	
Mục đích	Người dùng muốn xem vị trí của một địa điểm trên bản đồ.	
Actor	Người dùng điện thoại.	
Tiền điều kiện	Đang ở màn hình Details.	
Hậu điều kiện	Địa điểm được hiển thị lên bản đồ (Google Maps).	
Dòng sự kiện chính	1. Người dùng nhất nút Show on map.	2. Điện thoại chuyển qua màn hình Map và hiển thị vị trí địa điểm cùng với đường đi từ vị trí người dùng đến địa điểm.
Các xử lý ngoại lệ	Không có.	

j) Use case "Thêm địa điểm vào danh sách ưa thích":

Mã số	UC_10	
Tên	Thêm địa điểm vào danh sách ưa thích	
Mục đích	Người dùng muốn thêm một địa điểm vào danh sách ưa thích.	
Actor	Người dùng điện thoại.	
Tiền điều kiện	Đang ở màn hình Details.	
Hậu điều kiện	Địa điểm được thêm vào cơ sở dữ liệu cục bộ trên điện thoại.	
Dòng sự kiện chính	1. Người dùng nhất nút Add favorite.	2. Thông tin địa điểm được thêm vào cơ sở dữ liệu SQLite.
Các xử lý ngoại lệ	**UC_10_2E:** nếu địa điểm đã có trong SQLite do người dùng đã thêm trước đó thì hiển thị thông báo cho người dùng và không thêm.	

k) Use case "Xóa địa điểm khỏi danh sách ưa thích":

Mã số	UC_11	
Tên	Xóa địa điểm khỏi danh sách ưa thích.	
Mục đích	Người dùng muốn xóa một địa điểm khỏi danh sách ưa thích.	
Actor	Người dùng điện thoại.	
Tiền điều kiện	Đang ở màn hình Favorites.	
Hậu điều kiện	Địa điểm bị xóa khỏi cơ sở dữ liệu cục bộ trên điện thoại.	
Dòng sự kiện chính	1. Người dùng chọn một hoặc nhiều địa điểm cần xóa và nhấn nút Delete. 3. Người dùng xác nhận.	2. Điện thoại hiện thông báo yêu cầu xác nhận xóa.

		4. Địa điểm được xóa khỏi cơ sở dữ liệu SQLite.
Các xử lý ngoại lệ	**UC_11_2E:** nếu người dùng từ chối xóa thì không xóa.	

l) Use case "Tìm kiếm địa điểm ưa thích":

Mã số	UC_12	
Tên	Tìm kiếm địa điểm ưa thích.	
Mục đích	Người dùng muốn tìm một địa điểm trong danh sách ưa thích.	
Actor	Người dùng điện thoại.	
Tiền điều kiện	Đang ở màn hình Favorites.	
Hậu điều kiện	Địa điểm cần tìm được hiển thị lên màn hình.	
Dòng sự kiện chính	1. Người dùng nhập tên địa điểm và nhấn nút Tìm kiếm (hình chiếc kính lúp).	2. Điện thoại tiến hành truy vấn cơ sở dữ liệu cục bộ SQLlite. 3. Thông tin địa điểm được hiển thị trên màn hình.
Các xử lý ngoại lệ	**UC_12_2E:** nếu không tìm thấy địa điểm trong cơ sở dữ liệu thì hiện thông báo không tìm thấy.	

Phụ lục B: Bảng so sánh giữa các hệ điều hành điện thoại thông minh

(nguồn http://www.pcworld.com/)

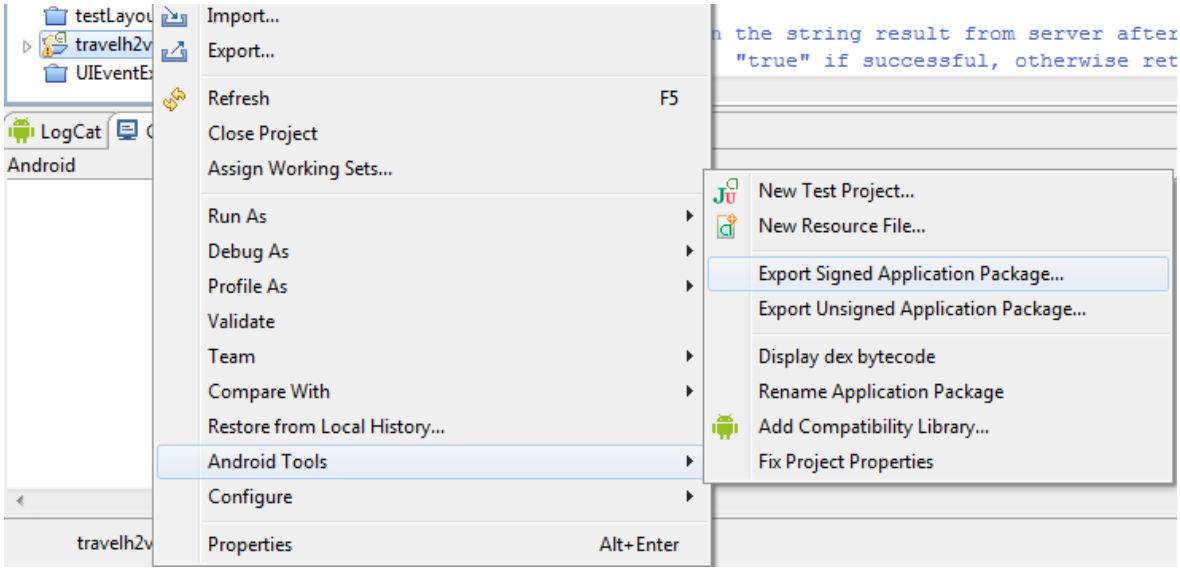
Thành phần	iOS 5	Android 4.0	Windows Phone 7.5
Đa nhiệm	Có	Có	Có
Chép/Cắt/Dán	Có	Có	Có
Hỗ trợ phần cứng	Chỉ những sản	Đa dạng	Rất ít

	phẩm của Apple		
Bảo mật	Tốt	Khá	Tốt
Mạng xã hội tích hợp	Twitter	Facebook, Twitter	Facebook, Twitter, Linkedin, Windows Live
Cửa hàng phim	iTunes	Android Market	Zune
Cửa hàng âm nhạc	iTunes	Google Musics	Zune
Cửa hàng sách	iBooks	Google Books	Không
Máy ảnh số	Tốt	Khá	Khá
Trò chơi	Rất đa dạng	Đa dạng	Đa dạng
Hỗ trợ Flash	Không	Có	Không
Hỗ trợ HTML5	Có	Có	Có
Bộ máy tìm kiếm	Google	Google	Bing
Màn hình chủ	Icons	Icons và Widgets	Icons và Widgets
Ứng dụng văn phòng	iWorks	Google Docs	Office Mobile
Nhận dạng giọng nói	Rất tốt	Khá	Khá
Kết nối WI-FI, WI-FI direct, GPS	Có	Có	Có
Hỗ trợ máy tính bảng	Có	Có	Không
Cập nhật	Có	Có	Có
Dịch vụ lưu trữ đám mây	iCloud	Google Sync	Skydrive
Khả năng tùy biến	Cực kỳ giới hạn	Dễ dàng	Không thể
Kho ứng dụng	Trên 500.000	Trên 250.000	Trên 30.000
Công nghệ NFC	Không	Có	Không

Tin nhắn trực tiếp giữa 2 điện thoại	Có	Không	Không
Màn hình cảm ứng	Rất mượt	Khá	Mượt

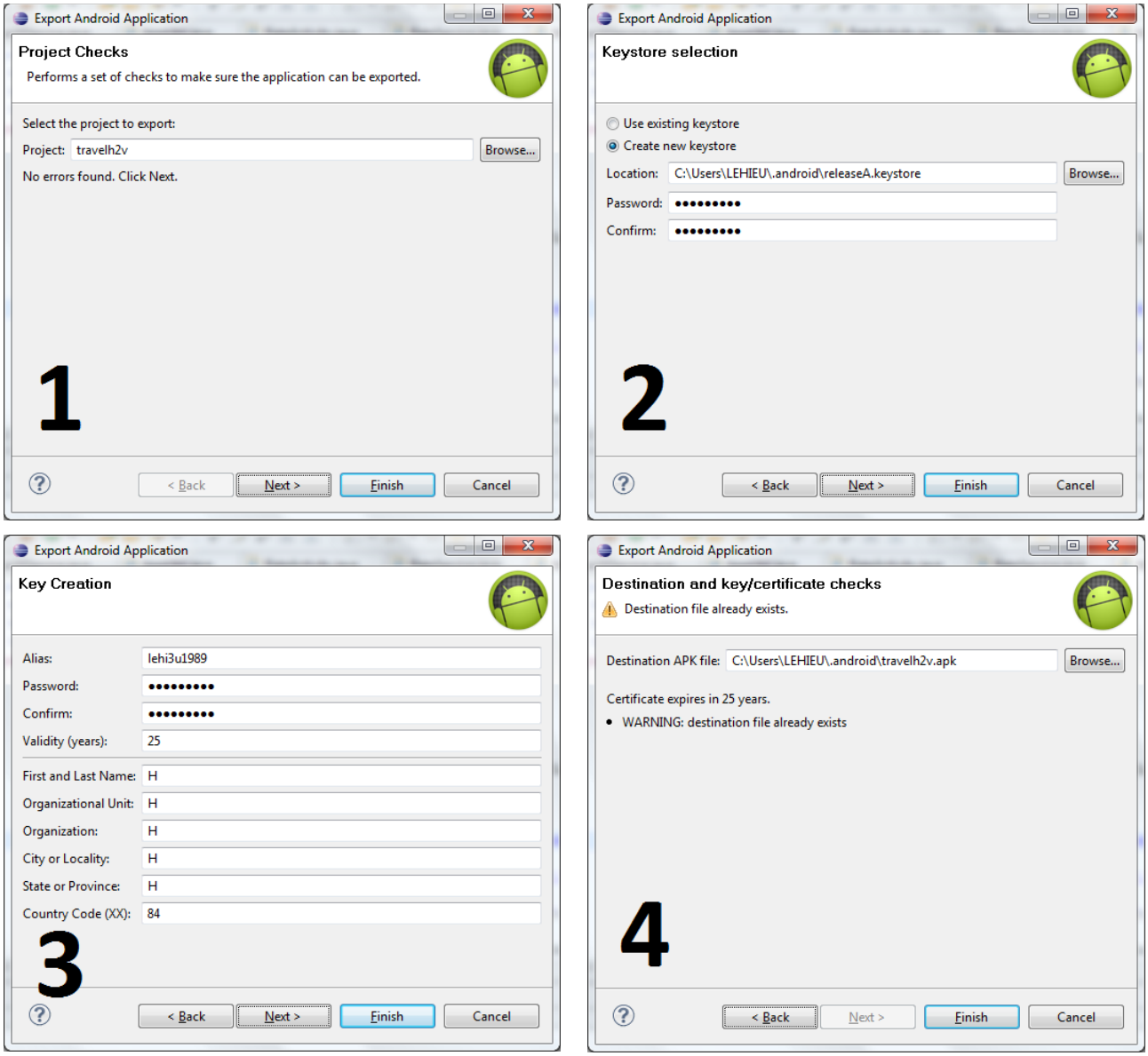
Phụ lục C: Cách đăng ký Google Maps API key cho ứng dụng Android.

Bước 1: Nhấp phải chuột lên project, chọn Android Tools, Export Signed Application Package.



Bước 2: Điền các thông tin theo hướng dẫn.

Ở màn hình Keystore Selection, nếu muốn tạo key mới thì chọn "Create new keystore" rồi chọn vị trí lưu key, điền các thông tin khác như tên, mật khẩu, thời hạn sử dụng … Nếu đã tạo key trước đó rồi và muốn dùng key đó để đăng ký cho ứng dụng thì chọn "Use existing keystore". Sau cùng chọn nơi lưu file .apk của ứng dụng. File này sẽ được đăng ký với key vừa tạo.



Bước 3: Lấy Google Maps API key tương ứng với file key vừa tạo để đăng ký vào file XML, ở vị trí MapView dùng hiển thị bản đồ.

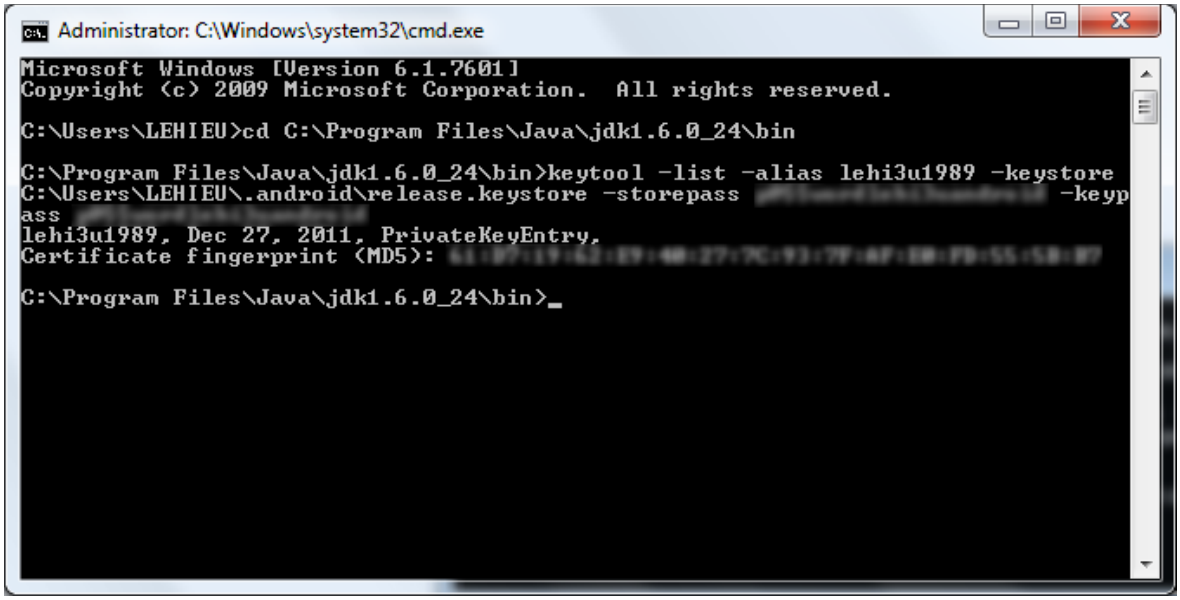
Mở Command Line của Windows, dẫn đến vị trí:

C:\Program Files\Java\jdk1.6.0_24\bin

Gõ dòng lệnh:

*keytool –list –alias ***** –keystore C:\Users\LEHIEU\.android\release.keystore –storepass ***** –keypass *****

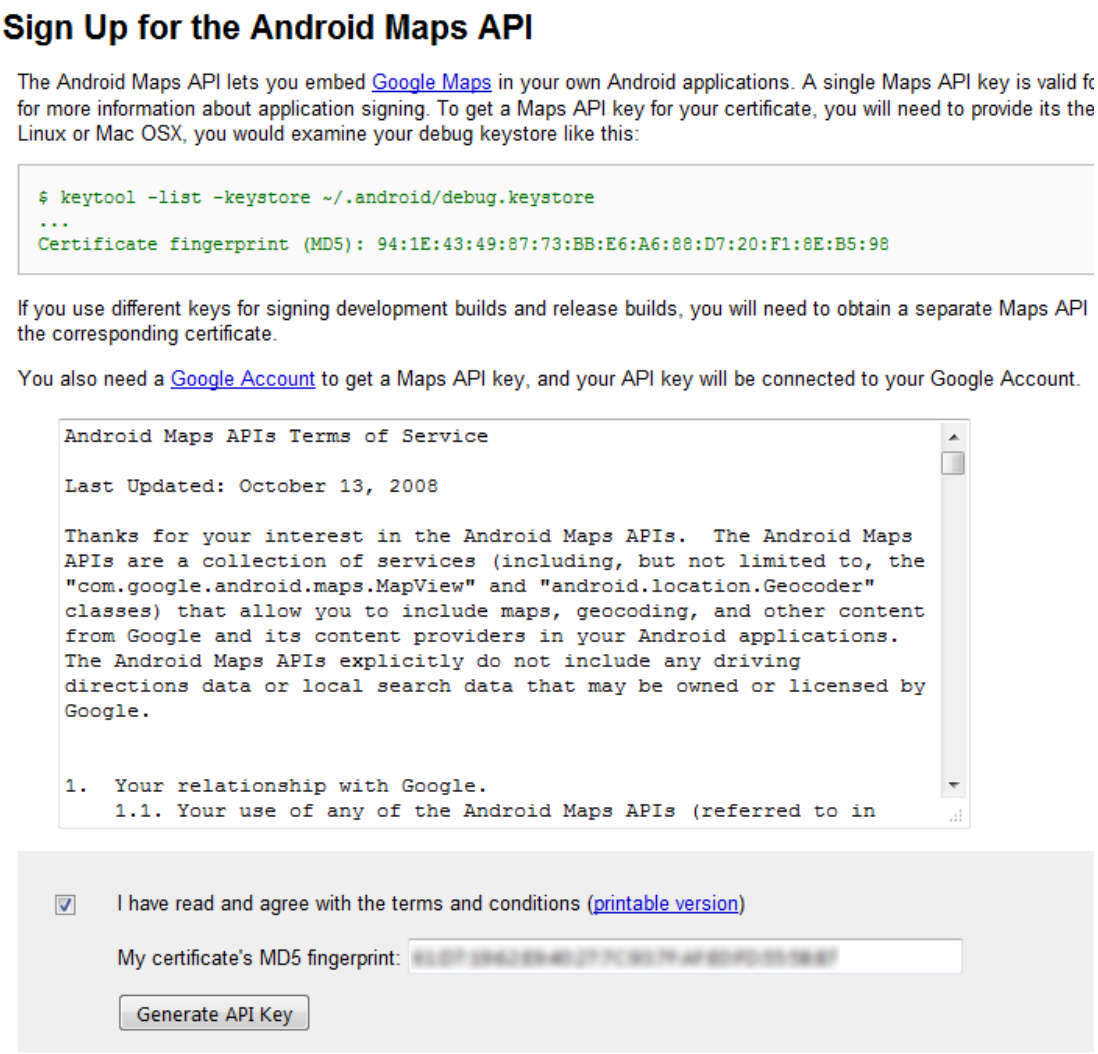
Với vị trí thư mục lưu key, key alias, storepass và keypass là những thông tin đã điền ở màn hình tạo key trong bước 2 ở trên.



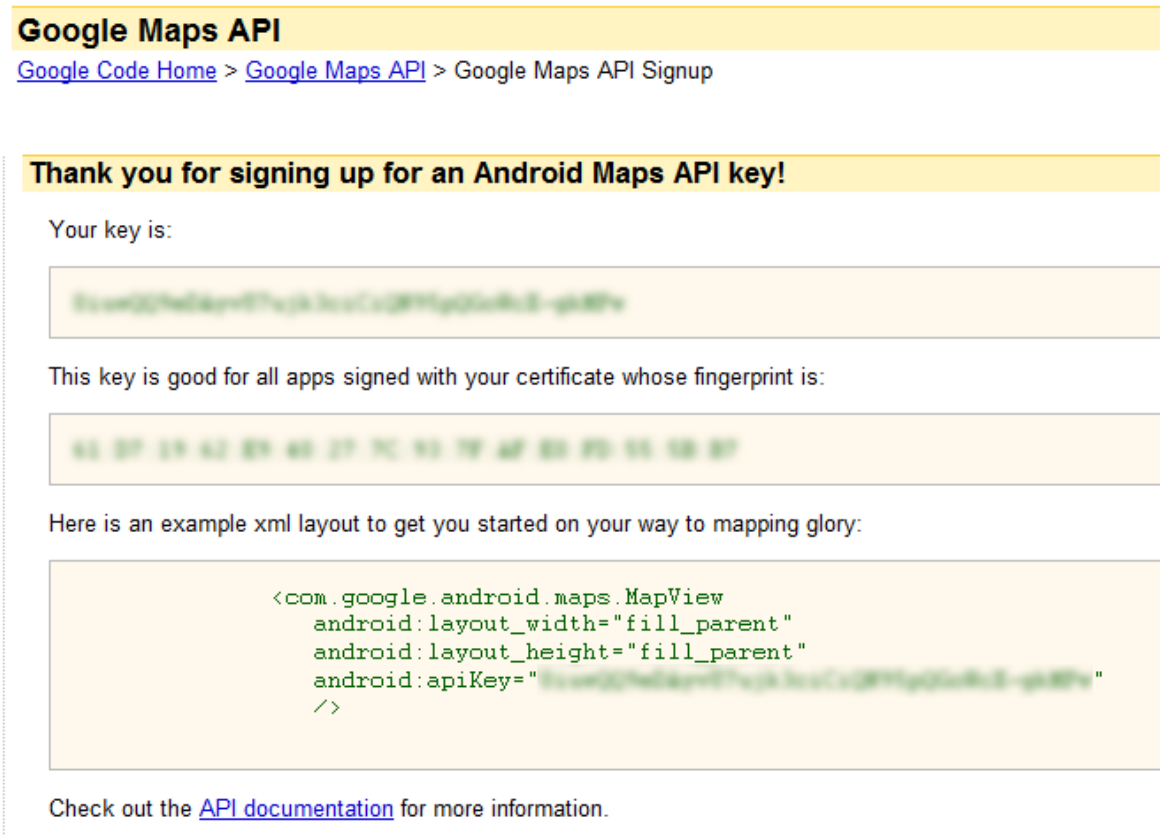
Ta sẽ được một chuỗi MD5. Sau đó vào trang web

http://code.google.com/android/maps-api-signup.html

Chép chuỗi MD5 vào và nhấn nút "Generate API key".



Ta sẽ được đoạn XML chứa key, chép đoạn này vào file XML, vị trí cần hiển thị bản đồ (MapView).



Google Maps API

Google Code Home > Google Maps API > Google Maps API Signup

Thank you for signing up for an Android Maps API key!

Your key is:

[redacted]

This key is good for all apps signed with your certificate whose fingerprint is:

[redacted]

Here is an example xml layout to get you started on your way to mapping glory:

```
<com.google.android.maps.MapView
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:apiKey="[redacted]"
   />
```

Check out the API documentation for more information.

Trên đây là toàn bộ quá trình đăng ký Google Maps API key cho một ứng dụng Android.