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 :*

**HỆ THỐNG KHUYẾN NGHỊ HỖ TRỢ KHÁCH DU LỊCH DÙNG ĐIỆN THOẠI 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 15 tháng 12 năm 2011

**LỜI CẢM ƠN**

Để có thể hoàn thành khóa luận này, trước hết chúng tôi xin chân thành cảm ơn tập thể quý thầy cô trường Đại Học Đại Học Công Nghệ Thông Tin - ĐHQG TP.HCM, quý thầy cô khoa Hệ Thống Thông Thông Tin đã giảng dạy và trang bị cho chúng tôi 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, chúng tôi đặc biệt gửi lời cảm ơn chân thành sâu sắc đến thầy 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 chúng tôi từng bước hoàn thành khóa luận.

Qua thời gian hơn 4 tháng, chúng tôi 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 chúng tôi có thể hoàn thiện mình tốt hơn nữa phục vụ cho công việc trong tương lai.

Một lần nữa chúng tôi xin chân thành cảm ơn! Xin kính chúc quý thầy cô dồi dào sức khỏe và luôn luôn thành công hơn nữa trong sự nghiệp trồng người cao quý!

Nhóm tác giả

MỤC LỤC

[CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI 4](#_Toc312099119)

[1.1 Đặt vấn đề: 4](#_Toc312099120)

[1.2 Hướng giải quyết: 5](#_Toc312099121)

[CHƯƠNG 2. KHẢO SÁT MỘT VÀI PHƯƠNG PHÁP KHUYẾN NGHỊ ĐÃ CÓ 6](#_Toc312099122)

[2.1. Mô hình khuyến nghị truyền thống (hai chiều): 6](#_Toc312099123)

[2.1.1. Hệ thống khuyến nghị dựa trên nội dung: 7](#_Toc312099124)

[2.1.2. Hệ thống khuyến nghị Collaborative: 9](#_Toc312099125)

[2.1.3. Hệ thống khuyến nghị lai (hybrid): 12](#_Toc312099126)

[2.1.4. Đánh giá chung về mô hình khuyến nghị hai chiều: 13](#_Toc312099127)

[2.2. Mô hình khuyến nghị đa chiều: 14](#_Toc312099128)

[CHƯƠNG 3. PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ 20](#_Toc312099129)

[3.1. Hướng tiếp cận cắt giảm số chiều: 20](#_Toc312099130)

[3.2. Kết hợp phương pháp khuyến nghị cắt giảm số chiều và phương pháp khuyến nghị truyền thống: 23](#_Toc312099131)

[CHƯƠNG 4. KHẢO SÁT CÁC KỸ THUẬT XÂY DỰNG ỨNG DỤNG 26](#_Toc312099132)

[4.1. Cơ sở dữ liệu và kho dữ liệu hỗ trợ cài đặt thuật toán: 26](#_Toc312099133)

[4.2. Dịch vụ web và Windows Communication Foundation: 26](#_Toc312099134)

[4.2.1 Dịch vụ web là gì? 26](#_Toc312099135)

[4.2.2 Windows Communication Foundation (WCF) 27](#_Toc312099136)

[4.3. Hệ điều hành cho điện thoại thông minh: 29](#_Toc312099137)

[4.3.1. Tại sao triển khai hệ thống khuyến nghị trên điện thoại? 29](#_Toc312099138)

[4.3.2. Chọn lựa giữa ứng dụng thuần và ứng dụng web trên điện thoại? 30](#_Toc312099139)

[4.3.3. Tại sao chọn Android? 31](#_Toc312099140)

[CHƯƠNG 5 HIỆN THỰC VÀ CÀI ĐẶT 33](#_Toc312099141)

[5.1. Kiến trúc toàn hệ thống: 33](#_Toc312099142)

[5.2. Xây dựng ứng dụng trên Android: 34](#_Toc312099143)

[5.2.1. Kiến trúc ứng dụng Android: 34](#_Toc312099144)

[5.2.2. Sơ đồ các màn hình: 37](#_Toc312099145)

[5.2.3. Các use cases trong hệ thống: 38](#_Toc312099146)

[5.3. Xây dựng cơ sở dữ liệu trên máy chủ: 39](#_Toc312099147)

[5.4. Xây dựng dịch vụ web: 40](#_Toc312099148)

[CHƯƠNG 6. KẾT QUẢ ĐẠT ĐƯỢC 41](#_Toc312099149)

[6.1. Ứng dụng Android: 41](#_Toc312099150)

[6.2. Cơ sở dữ liệu dùng hiện thực thuật toán: 41](#_Toc312099151)

[6.3. Dịch vụ web WCF: 41](#_Toc312099152)

[CHƯƠNG 7 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 43](#_Toc312099153)

[7.1. Kết luận: 43](#_Toc312099154)

[7.2. Hướng phát triển: 44](#_Toc312099155)

[TÀI LIỆU THAM KHẢO 45](#_Toc312099156)

[PHỤ LỤC 49](#_Toc312099157)

[Phụ lục A: Mô tả chi tiết các use cases: 49](#_Toc312099158)

[Phụ lục B: Bảng so sánh giữa các hệ điều hành điện thoại thông minh 58](#_Toc312099159)

# 

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

---oOo---

Mở đầu, chúng tôi 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 chúng tôi sẽ triển khai.

## **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.

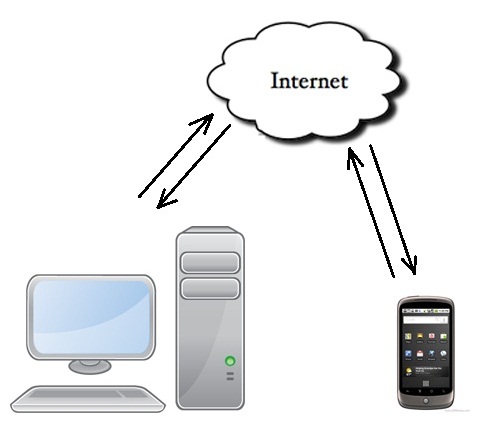
Ngày nay với nền công nghệ thông tin phát triển mạnh, nhiều ứng dụng và dịch vụ ra đời nhằm quảng bá hình ảnh du lịch của đất nước đến bạn bè thế giới. Những năm gần đây, điện thoại thông minh (smartphone) ngày càng trở nên phổ biến. Với lợi thế nhỏ gọn, tiện dụng và thông minh, điện thoại thông minh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

Thực tế khi đi du lịch, những điều kiện ngữ cảnh xung quanh người dùng (thời tiết, nhiệt độ, tâm trạng ...) luôn có ảnh hưởng đến quyết định lựa chọn điểm đến. Vì vậy nhóm chúng tôi 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, chúng sẽ đến nhanh hơn. Từ đó, chúng tôi hi vọng sẽ góp phần thúc đẩy ngành du lịch nước nhà phát triển tốt đẹp hơn.

## Hướng giải quyết:

Qua yêu cầu bài toán, để phát triển được ứng dụng trên điện thoại nói trên. Về cơ bản, chúng tôi nhận định 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, mọi người có thể hình dung kiến trúc cơ bản trên thông qua hình vẽ sau đây:



*Hình 1.1: Giao tiếp giữa máy chủ và điện thoại qua mạng Internet.*

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

# CHƯƠNG 2. KHẢO SÁT MỘT VÀI PHƯƠNG PHÁP KHUYẾN NGHỊ ĐÃ CÓ

---oOo---

Chương 1 trên đây là những hình dung ban đầu. Như đã nói, ta cần một cách thức cụ thể nào đó giúp giải quyết bài toán. Theo chúng tôi, 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, chúng tôi sẽ đi sâu vào tìm hiểu những phương pháp khuyến nghị đã có, từ đó lựa chọn ra phương pháp thích hợp nhất với đề tài của nhóm để triển khai.

## Mô hình khuyến nghị truyền thống (hai chiều):

Trong lĩnh vực xây dựng các hệ thống khuyến nghị trong quá khứ, người ta đã làm việc và nghiên cứu khá nhiều. Hầu hết công việc chủ yếu tập trung phát triền những phương pháp gợi ý những 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 2 thực thể là người dùng và đối tượng cần được được khuyến nghị đến cho người dùng (ở đây chúng tôi xin phép gọi tắt là đối tượng). 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 ta có một danh sách các chỉ số đánh giá như sau:

* Người A đánh giá điểm du lịch “DLA” với chỉ số là 4 (trên tổng mức 5).
* Người A đánh giá điểm du lịch “DLB” với chỉ số là 5 (trên tổng mức 5).
* Người A đánh giá điểm du lịch “DLC” với chỉ số là 3 (trên tổng mức 5).
* Người B đánh giá điểm du lịch “DLA” với chỉ số là 4 (trên tổng mức 5).
* Người B đánh giá điểm du lịch “DLB” với chỉ số là 4 (trên tổng mức 5).

Từ đó, người ta xây dựng một hàm R dùng để dự đoán các chỉ số đánh giá còn chưa biết. Ví dụ ta cần dự đoán người C sẽ đánh giá điểm du lịch “DLC” với chỉ số là bao nhiêu.

*R: Users* x *Items 🡪 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 các chỉ số đánh giá trong toàn bộ không gian (*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*.

Trong thực tế, người ta 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 3 loại: khuyến nghị dựa trên nội dung (content-based), khuyến nghị collaborative (collaborative) và khuyến nghị lai (hybrid) [6]. Chúng tôi sẽ lần lượt nói qua một cách vắn tắt về chúng ngay sau đây.

### 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 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, ta 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, người ta cần xác định một độ đo. Độ đo đó là những trọng số được tính toán dựa vào những thông tin thu thập được. Ví dụ như độ đo *term-frequency/inverse document frequency (TF-IDF)* [15].

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 , mô tả những đặc tính của người dùng *u* và đối tượng *i* [17]. 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 dụng 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 không phải dựa trên heuristic formula, mà 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 [18].

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”. Người ta 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 (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 [19]. 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 này, 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 bất cập nữa, là người dùng phải có đánh giá cho những 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.

### Hệ thống khuyến nghị Collaborative:

Theo truyền thống, những hệ thống khuyến nghị Collaborative 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 nghi kiểu Collaborative có thể được phân thành hai nhóm là phương pháp khuyến nghị sử dụng bộ nhớ (memory-based hay còn gọi là heuristic-based) và phương pháp khuyến nghị dựa trên mô hình (model-based) [20].

1. **Phương pháp khuyến nghị sử dụng bộ nhớ:**

Phương pháp khuyến nghị dựa trên bộ nhớ 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 người ta 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):

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:

Trong đó, *k* là tác nhân chuẩn hóa, thường được tính bằng công thức  
 , là chỉ số đánh giá trung bình của người dùng *u* đối với các đối tượng được định nghĩa bằng công thức với , *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:*

*Hệ số tương quan Cosine:*

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*, , là các chỉ số đánh giá trung bình của người dùng *x* và *y* vừa được nhắc đến ở trên, là tập hợp các đối tượng cùng được đánh giá bởi người dùng *x* và *y*, là tích vô hướng của 2 vector và . [21]

Người ta 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 ký ức, 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). Cũng có những 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. [21]

1. **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 models), 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 sử dụng bộ nhớ [12, 13]. 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, 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) chúng tôi tiến hành khảo sát các mô hình hồi qui, có thể tóm gọn như sau:

Quay trở lại phương pháp khuyến nghị sử dụng bộ nhớ 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 = (X1, X2, …, Xn) là tập biến số đại diện cho sự ưa thích của một người dùng đối với những item khác nhau. Mô hình hồi qui tuyến tính có thể được phát biểu như sau:

Trong đó α là một ma trận n × k, β = (β1, β2, …, βn) đại diện cho độ nhiễu trong lựa chọn của người dùng, Y là một ma trận n × m với Yij là đánh giá của người dùng i cho đối tượng j, X là một ma trận k × 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 [14] 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 collaborative 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 collaborative 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 nhưng 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.

### Hệ thống khuyến nghị lai (hybrid):

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 collaborative với nhau, dẫn đến sự ra đời của hệ thống khuyến nghị lai. Ví dụ như người ta lập trình cho hệ thống học và lưu giữ những thông tin cá nhân người dùng (user profiles) bằng những phương pháp rút trích, phân tích nội dung (content-based), sau đó tiến hành so sánh những thông tin cá nhân đó để xác định được những người dùng có tính chất, sở thích tương tự nhau để đưa ra gợi ý bằng phương pháp collaborative. Ví dụ như tùy vào độ tuổi, giới tính, sở thích phim ảnh khác nhau, người ta 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 người ta triển khai hai hệ thống riêng biệt, một dùng phương pháp dựa trên nội dung, một dùng phương pháp collaborative, chỉ số dự đoán sau cùng được đưa ra sau khi tổ hợp tuyến tính chỉ số của hai hệ thống riêng biệt. Hoặc tùy từng thời điểm mà người ta chọn dùng hệ thống này hay hệ thống kia tùy thuộc vào chất lượng thông tin gợi ý của hệ thống nào tốt hơn.

Người ta cũng đã nghiên cứu và nhận xét rằng hệ thống khuyến nghị lai sẽ cho ra 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 collaborative.[6,21]

### 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:

* *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, người ta nghiên cứu những kỹ thuật cắt giảm số chiều như Singular Value Decomposition (SVD) [23], Latent Semantic Indexing (LSI) [24,25], content-boosted CF [16], Tree Augmented Naïve Bayes Optimized By Extended Logistic Regression (TAN-ELR) [26], Maximum margin Matrix Factorizations (MMMF) [27,28]…
* *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 cắt giảm số chiều như SVD cũng giúp hạn chế được vấn đề, như thuật toán Incremental SVD [29] được cải tiến từ SVD. Hay thuật toán Item-based Pearson [30], gom cụm người dùng [31, 32] …
* *Sự tương tự, trùng lặp:* có những đối 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) [25] 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 collaborative [33].
* *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. Người ta nghiên cứu và 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) [34]. Cũng có những nghiên cứu đưa ra một mô hình giúp phát hiện những gian lận đánh giá này được nhắc đến ở tài liệu [35, 36].
* 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 … đã, đang và vẫn được tiếp tục nghiên cứu trên thế giới.

## 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* x *Items 🡪 Ratings*

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

*R: Users* x *Items* x *Contexts🡪 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 X là hấp dẫn nhất.

Để dễ hình dung khái niệm nhiều chiều, người ta thường dùng mô hình dữ liệu nhiều chiều dựa trên OLAP. [1] Giả sử ta có các chiều là *D1, D2, D3, …, Dn*. 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 hơp tích Descartes gồm nhiều thuộc tính. *Di Ai1* x *Ai2* x *…* x *Aik* , trong đó mỗi thuộc tính *Aik* 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 *Aik*).

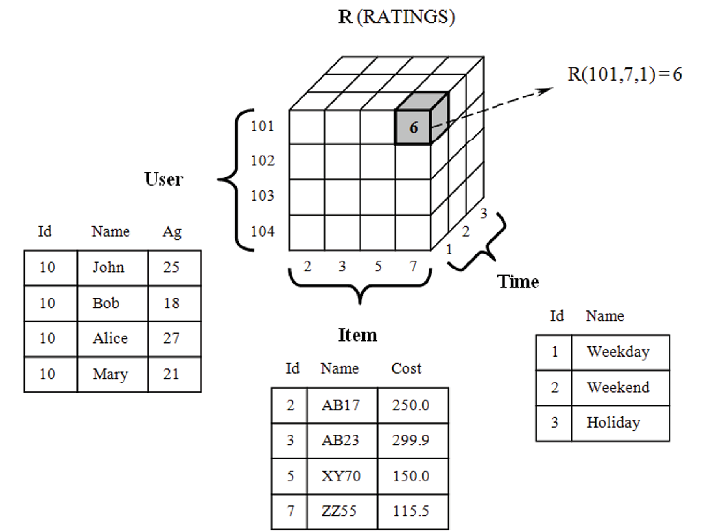
Ví dụ ta có không gian khuyến nghị ba chiều là Người dùng x Đối tượng x Thời gian. Trong đó:

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

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

*Thơi 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.

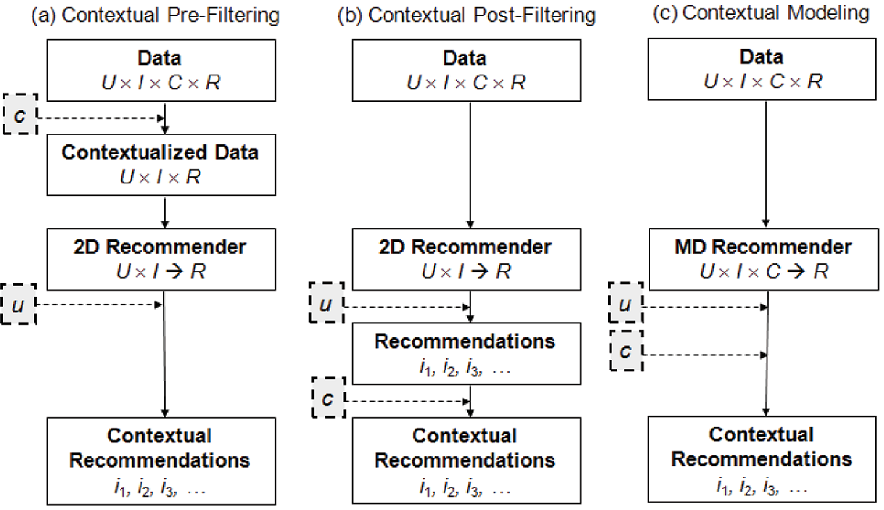


*Hình 2.1: Ví dụ không gian khuyến nghị ba chiều.*

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. Chúng ta cũng cần xem xét rằng không phải tất cả các điểu 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 có ảnh hưởng đển quyết định mua sách 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 người ta 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 điểu 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 cẩn 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 chúng ta sẽ sử dụng những thông tin ngữ cảnh trong quá trình đưa ra gợi ý vào lúc nào? Hình dưới đây mô tả ba trường hợp [21]:



*Hình 2.2: Thông tin ngữ cảnh trong hệ thống khuyến nghị.*

**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 điểu 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 cắt giảm số chiều (ở đây là chiều ngữ cảnh) *(reduction-based)*. 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 cắt giảm. Để dễ hình dung, ta tiếp tục ví dụ gợi ý phim với không gian gợi ý gồm người dùng, đối tượng và thời gian. Quá trình cắt giảm được minh họa như sau:

Ở đây, ta 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 tiền 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 cắt 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 (người dùng và đối tượng) 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 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* rối 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 ký ức, 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 nhiều 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. [21]

Do tính chất phức tạp cũng như những giới hạn trong quá trình thực hiện luận văn này, chúng tôi xin phép không đi vào chi tiết những kỹ thuật này. Phần khảo sát sơ bộ những hệ thống, những kỹ thuật khuyến nghị đã được những người đi trước thực hiện xin được phép dừng lại tại đây. Chương tiếp theo sẽ là phần trình bày chi tiết thuật toán chúng tôi nghiên cứu, tìm hiểu và sử dụng trong hệ thống khuyến nghị du lịch mà chúng tôi đang xây dựng.

# CHƯƠNG 3. PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ

---oOo---

Sau khi tiến hành khảo sát các phương pháp khuyến nghị ở những phần trên, ở phần này chúng tôi sẽ trình bày chi tiết về phương pháp đã được hiện thực trong khóa luận này. Chương này gồm có 2 phần: phần 1 là hướng tiếp cận cắt giảm số chiều (Reduction-based), phần 2 sẽ trình bày về mô hình hồi qui được sử dụng kết hợp để dự đoán đánh giá người dùng.

## Hướng tiếp cận cắt giảm số chiều:

Ý tưởng chính của phương pháp được chúng tôi lựa chọn này là đưa bài toán đa chiều về bài toán hai chiều (User × Item) 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 cùng bắt đầu quay lại với bài toán khuyến nghị hai chiều với ví dụ sau:

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, chúng ta có thể dự đoán bất kỳ một đánh giá nào. Tương tự ta thêm một chiều mới là thời gian thì sẽ có công thức dự đoán sẽ là:

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 3 chiều trên có thể biểu diễn thông qua hàm dự đoán 2 chiều như sau:

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 cắt 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 (u, i, t) mà thay vào đó là tập ngữ cảnh St (contextual segment) chứa các ngữ cảnh cao hơn hoặc bằng với t. Ví dụ, nếu muốn dự đoán 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 nhất” 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:

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

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

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, … Chúng ta cũng sử dụng thuật ngữ chuẩn trong lĩnh vực tiếp thị (marketing) để chỉ các tập dữ liệu đánh giá trích xuất theo mỗi giá trị của ngữ cảnh là “segment”, ví dụ ta có segment “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:

Tương tự segment “Cuối tuần – Bạn bè” sẽ chứa các đánh giá cho những địa điểm được đi vào cuối tuần với người yêu

Ta minh họa cách thức hoạt động của phương pháp cắt 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 , 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 với tập dữ liệu là các đánh giá vào buổi sáng. Sau đó chúng ta có thể áp dụng bất kỳ thuật toán dự đoán hai chiều nào như đã trình bày ở chương 2 (phương pháp chúng tôi lựa chọn là sử dụng mô hình hồi qui sẽ được trình bày chi tiết ở những phần sau). Ý tưởng của phương pháp này khá đơn giản: nếu ta muốn môt đánh giá vào buổi sáng của một người dùng cho một đối tượng cụ thể, chúng ta chỉ 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 mặc nhiên bỏ qua). Chúng ta cùng xem xét ví dụ sau:

**Ví dụ:** Ta có không gian khuyến nghị ba chiều , với X là chiều ngữ cảnh có hai giá trị là h và t. Gọi các giá trị đánh giá khi là *nt* và khi là *nh*. 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 nt hoặc nh tùy ngữ cảnh) và .

Với giả định trên phương pháp cắt 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 đánh giá sử dụng bộ nhớ:

Do đó, nếu ta sử dụng các thông tin ngữ cảnh về đối tượng đã được đánh giá, trong trường hợp segment , tất cả các đánh giá đều bằng nt, và do đó bất chấp độ đo tương quan giữa các người dùng. Tương tự với segment chúng ta luôn có . Do đó tất cả dự đoán đều chính xác với thực tế khi áp dụng phương pháp cắt 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 *nt* và *nh* 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 (khi chỉ những dữ liệu đánh giá sai được lựa chọ để tính toán).

Lý do mà phương pháp khuyến nghị cắt 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 *nt* và nhóm các đánh giá cho *X = h* là *nh* với ). Tuy nhiên nếu thì chiều X sẽ không còn giá trị trong mục đích khuyến nghị (việc đánh giá trong hai trường hợp và là như nhau, nghĩa là không ảnh hưởng tới quyết định của người dùng). Trong trường hợp này, phương pháp khuyến nghị cắt giảm số chiều có độ chính xác không hơn phương pháp khuyến nghị truyền thống vì việc cắt giảm số chiều ở đây là không có ý nghĩa. Trong trường hợp tổng quát, thì khả năng X1 = X2 = … = Xn là rất thấp, do đó phương pháp khuyến nghị cắt 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 segment bằng phương pháp cắt 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 segment sau khi cắt giảm là quá ít (Dữ liệu thưa thớt). Một trong những giải pháp cho vấn đề này là kết hợp phương pháp khuyến nghị cắt giảm số chiều với phương pháp khuyến nghị truyền thống sẽ được trình bày ở phần tiếp theo.

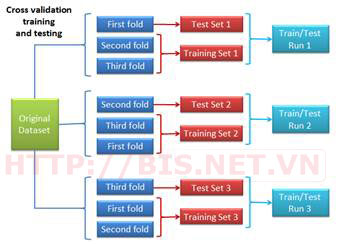
## Kết hợp phương pháp khuyến nghị cắt giảm số chiều và phương pháp khuyến nghị truyền thống:

Trước khi đi vào phương pháp kết hợp này chúng tôi xin trình bày một vài khái niệm tiên quyết.

Để kết hợp hai phương pháp đầu tiên chúng ta cần những thước đo độ hiệu quả để so sánh phương pháp nào tốt hơn phương pháp nào trên những segment 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, chúng tôi sử dụng một hàm đo độ hiệu quả trừu tượng là 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 . Với mỗi , gọi *d.R* là giá trị đánh giá thực cho điểm d, *d.RA,X* là đánh giá được dự đoán bằng thuật toán A trên tập dữ liệu huấn luyện X cho điểm d. Hàm được định nghĩa như một hàm thống kê trên hai tập dữ liệu đánh giá và . Ví dụ độ sai lệch trung bình tuyệt đối được định nghĩa như sau:

Như đã đề cập ở trên, trong thì , nói cách khác tập dữ liệu huấn luyện và tập dữ liệu đánh giá phải riêng biệt với nhau. Trong các nghiên cứu người ta thường sử dụng kỹ thuật “N-fold Cross Validation“, dữ liệu gốc sẽ được chia thành n phần bằng nhau, và quá trình huấn luyện/đánh giá cho model thực hiện lặp lại n lần. Tại mỗi lần huấn luyện/đánh giá, 1 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 3 phần, và có 3 lần thực hiện quá trình huấn luyện/đánh giá. Lần thứ nhất, phần 1 làm tập đánh giá, 2 phần còn lại làm tập huấn luyện. Lần thứ hai, phần 2 làm tập đánh giá. Lần thứ ba, tới phần 3 làm tập đánh giá …



*Hình 3.1: 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 Test Set 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.

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

---oOo---

Ở chương 3, chúng tôi đã trình bày phần phương pháp được lựa chọn để giải quyết bài toán. Sang chương 4, chúng tôi sẽ đi sâu vào tìm hiểu những kỹ thuật quan trọng cần thiết cho việc hiện thực hóa hệ thống.

## Cơ sở dữ liệu và kho dữ liệu hỗ trợ cài đặt thuật toán:

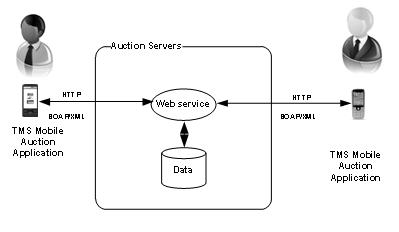
## Dịch vụ web và Windows Communication Foundation:

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, người ta sử dụng những dịch vụ web (web services).

### 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 ta xây dựng một ứng dụng với số lượng người dùng lên đến hàng trăm, hàng ngàn người và phân tán ở những địa điểm khác nhau.

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 gừi một yêu cầu lên máy chủ, máy chủ xử lý và trả về kết quả là chuỗi JSON chứa các địa điểm cần tìm. Người dùng có thể mở đường link sau trên trình duyệt web sẽ thấy kết quả.

[https://ajax.googleapis.com/ajax/services/search/local?v=1.0&rsz=2&sll=10,106  
&q=hotel&start=0](https://ajax.googleapis.com/ajax/services/search/local?v=1.0&rsz=2&sll=10,106&q=hotel&start=0)

*Ý nghĩa:* tìm kiếm các khách sạn xung quanh điểm có tọa độ (10,106), rsz = 2: tập kết quả trả về gồm 2 kết quả, sll = 10,106: tọa độ (hoành độ, tung độ) điểm hiện tại, q = hotel: tìm khách sạn, start = 0: chỉ số thứ tự của kết quả tìm kiếm trả về đầu tiên.

### Windows Communication Foundation (WCF)

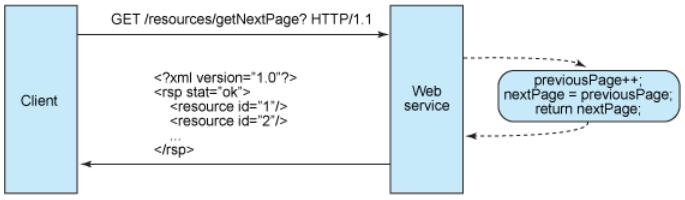
Do phần hệ thống back-end gồm cơ sở dữ liệu cũng như kho dữ liệu chúng tôi chọn những công nghệ của Microsoft, nên chúng tôi cũng chọn WCF để xây dựng dịch vụ web. WCF là một phần của .NET Framework cung cấp một mò hình lập trình thống nhất hỗ trợ xây dựng một cách nhanh chóng và dễ dàng các ứng dụng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

Chúng 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:

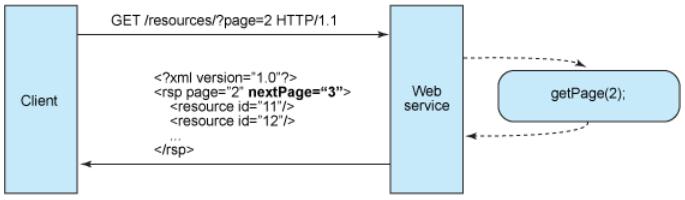
* Hỗ trợ truyền nhận thông tin không chỉ qua giao thức HTTP, mà còn có thể thông qua TCP hoặc những giao thức mạng khác.
* Khả năng chuyển đổi giữa các giao thức truyền nhận thông tin một cách dễ dàng.
* Hỗ trợ việc host services không những trên một web máy chủ mà còn trên nhiều dạng khác như: WinForms 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, chúng tôi xây dựng một WCF 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 4.2: Thiết kế có lưu giữ trạng thái (stateful)*



*Hình 4.3: Thiết kế phi trạng thái (stateless)*

* Hiển thị cấu trúc thư mục như URls: ví dụ để tập hợp và duyệt các bài viết theo những đề tài khác nhau, chúng ta có thể xây dựng đường link cho web service có cấu trúc như sau:[http://www.myservice.org/discussion/topics/{topic}](http://www.myservice.org/discussion/topics/%7btopic%7d)

Phần gốc, /discussion, có một nút /topics bên dưới. Dưới topics là tên các tên chủ đề khác nhau, như văn hóa, xã hội, kỹ thuật, ... Trong cấu trúc này, ta dễ dàng truy xuất đến chủ đề con nào đó bằng cách điền thêm một vài từ sau nút /topics/.

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

Những mô tả kỹ thuật chi tiết hơn về WCF, REST ... chúng tôi xin phép không đề cập ở đây. Người đọc có thể tìm hiểu thêm thông tin WCF từ MSDN - Microsoft, REST từ IBM:

<http://msdn.microsoft.com/en-us/library/dd560536.aspx>

<http://www.ibm.com/developerworks/webservices/library/ws-restful/>

hoặc từ nhiều nguồn khác.

## Hệ điều hành cho điện thoại thông minh:

### 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, chúng tôi 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 họ ngay lập tức có thể sử dụng ở mọi lúc mọi nơi. Thay vì họ 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 khác. Ở thời điểm khác, điều kiện khác, họ thay đổi quyết định. Vì vậy, luôn mang bên mình chiếc điện thoại, họ 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 nghe 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.

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

**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 chúng tôi 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 offline khi không có Internet.
* Trong tương lai, có thể dễ dàng thương mại hóa trên các cửa hàng ứng dụng (App Store).

### 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.

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, TV ... 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:

**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 …

Chi tiết hơn về hệ điều hành Android, người đọc có thể tìm hiểu thêm trên trang <http://developer.android.com/>

Tóm lại, chúng tôi quyết định chọn nền tảng Android vì những lý do chính 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 đúc trên toàn cầu.
* Điện thoại sử dụng Android ngày càng chiếm thị phần lớn do giá thành rẻ hơn so với các nên 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.

# 

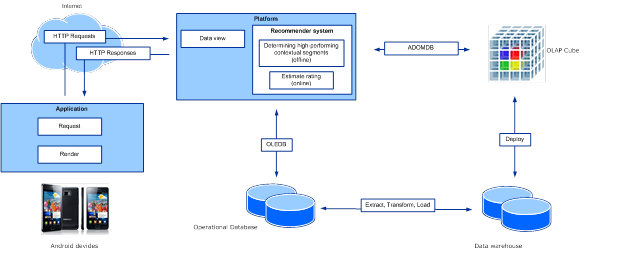
# CHƯƠNG 5 HIỆN THỰC VÀ CÀI ĐẶT

---oOo---

Qua các chương trên, chúng tôi 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, chúng tôi sẽ cụ thể hóa việc xây dựng hệ thống như thế nào.

## Kiến trúc toàn hệ thống:

Như đã phát họa sơ lược ở phần đầu của báo cáo, hệ thống chúng tôi cần xây dựng phải có ít nhất hai thành phần chính là máy chủ chứa cơ sở dữ liệu và máy khách là các điện thoại thông mình sẽ gởi các yêu cầu khuyến nghị và nhận dữ liệu trả về từ máy chủ. Sau khi tìm hiểu kỹ hơn về mặt kỹ thuật, bây giờ chúng tôi có thể đưa ra một kiến trúc hoàn chỉnh hơn cho hệ thống. Hình dưới đây minh họa cho kiến trúc đó.



*Hình 5.1: Kiến trúc toàn hệ thống.*

Hệ thống gồm 3 phần chính:

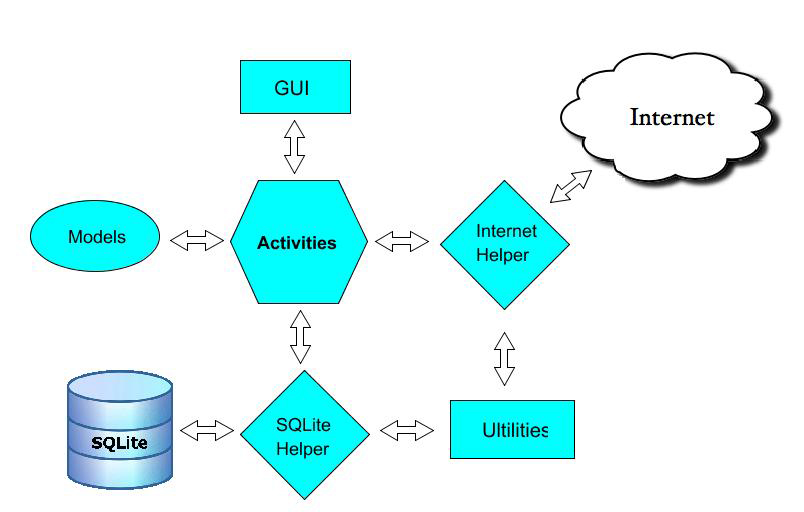
* Phần 1: ứng dụng được viết trên nền tảng hệ điều hành Android hỗ trợ khách du lịch các chức năng như: tìm kiếm thông tin các điểm du lịch, các gợi ý về các điểm du lịch phù hợp với người dùng, bản đồ …
* Phần 2: hệ thống khuyến nghị người dùng dựa trên ngữ cảnh với cơ sở dữ liệu, kho dữ liệu và OLAP Cube. Đây là nơi thực hiện những thuật toán khuyến nghị và trả kết quả về cho điện thoại.
* Phần 3: dịch vụ web phụ trách việc truyền nhận thông tin giữa điện thoại với hệ thống khuyến nghị.

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

## Xây dựng ứng dụng trên Android:

### Kiến trúc ứng dụng Android:

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



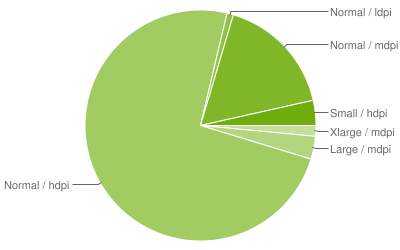
*Hình 5.2: Kiến trúc của ứng dụng Android.*

Những thành phần quan trọng nhất gồm:

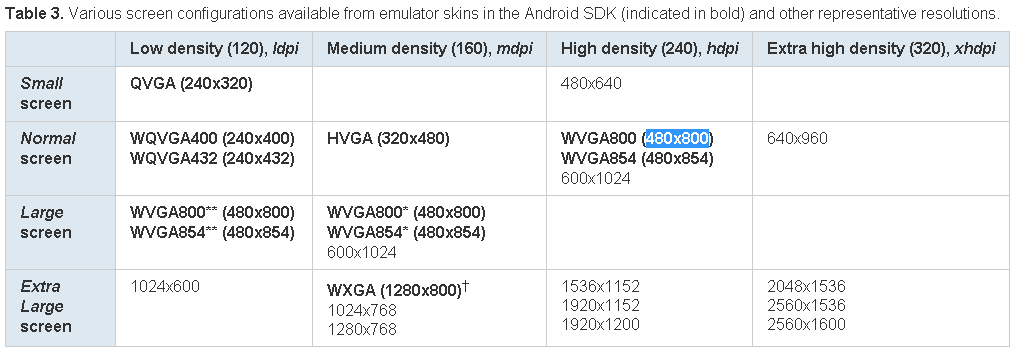
* **Activies:** giữ vai trò chủ đạo tương tự như Form trong lập trình Windows Form. Ứng với mỗi màn hình trên điện thoại là một activity phụ trách việc hiển thị, điều khiển các chức năng tương ứng.
* [**Graphical user interface**](http://en.wikipedia.org/wiki/Graphical_user_interface) **(GUI):** giao diện người dùng được thiết kế bởi những file XML và các file ảnh định dạng PNG, JPG, BMP.

Ở đây, chúng tôi xin nói thêm vài điều về việc thiết kế giao diện trong Android. Do đây là hệ điều hành mã nguồn mở nên hỗ trợ rất nhiều dòng điện thoại với kích cỡ màn hình rất đa dạng. Theo khảo sát của Google thì sự phân bố các loại màn hình trên thị trường hiện nay như sau:

|  |  |  |  |
| --- | --- | --- | --- |
|  | ldpi | mdpi | hdpi |
| small |  |  | 3.5% |
| normal | 0.9% | 16.9% | 74% |
| large |  | 3.2% |  |
| xlarge |  | 1.5% |  |



*Hình 5.3: Độ phổ biến của các chuẩn màn hình Android*

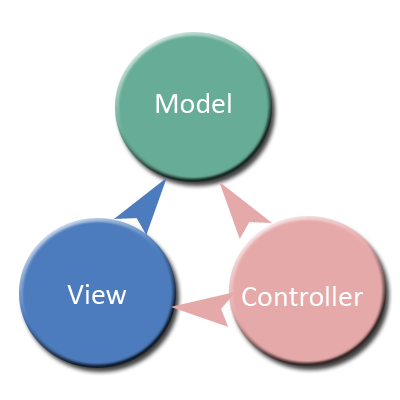


*Hình 5.4: Độ phân giải các loại màn hình Android*

Trong đó, small, normal, large là kích cỡ màn hình từ nhỏ đến lớn; ldpi, mdpi, hdpi là mật độ điểm ảnh từ thấp đến cao (DPI viết tắt của Dots Per Inches – số điểm ảnh trên một đơn vị diện tích tính bằng inch). Màn hình cỡ normal với mật độ hdpi chiếm phần lớn. Kế đến là loại normal/mdpi. Do thời gian có hạn, ứng dụng chúng tôi xây dựng sẽ chỉ được thiết kế giao diện sao cho tương thích tốt nhất với 2 loại màn hình trên. Ở các loại màn hình khác, giao diện có thể hiển thị không đúng vị trí, tỉ lệ hoặc bị vỡ ảnh. Do có nhiều loại mật độ điểm ảnh nên trong Android, người ta đưa ra loại đơn vị *Density-independent pixel* (DIP) được dùng khi thiết kế giao diện. Đây là đơn vị giúp ánh xạ giữa kích thước ảnh thật ứng với từng màn hình với mật độ điểm ảnh khác nhau. 1 px = 1 dip \* (dpi / 160). Với mỗi loại màn hình, sẽ có một bộ giao diện riêng với các ảnh có kích cỡ tương ứng. Điều này giúp ảnh sẽ hiện thị đúng tỉ lệ trên các màn hình dù cho chúng có mật độ điểm ảnh khác nhau.

* **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 chúng tôi 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 …

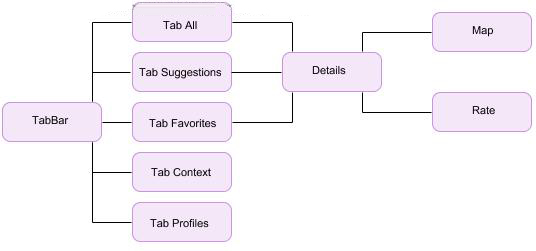
Về mô hình tổ chức code, chúng tôi sử dụng mô hình MVC (Model – View - Controller) quen thuộc.



*Hình 5.5: 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).

### Sơ đồ các màn hình:

****

*Hình 5.6: Sơ đồ các màn hình trong ứng dụng.*

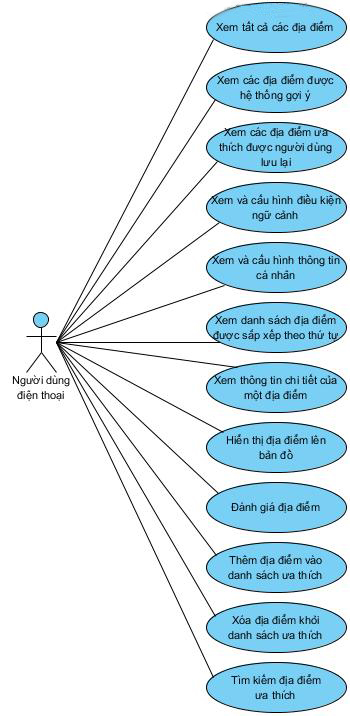
Ứng dụng gồm 5 tab:

* 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, chúng tôi sẽ chụp những hình minh họa ở chương tiếp theo.

### Các use cases trong hệ thống:

Để xây dựng được hệ thống, chúng tôi dùng phương pháp phân tích thiết kế hệ thống theo hướng UML. Tuy nhiên nếu trình bày chi tiết tất cả quá trình phân tích thiết kế thì sẽ rất dài dòng. Vậy nên, chúng tôi xin phép bỏ qua các sơ đồ khác trong phân thích 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.



*Hình 5.7: Sơ đồ use case*

Phần mô tả chi tiết từng use case, người đọc có thể xem tại phần phụ lục ở cuối quyển báo cáo này.

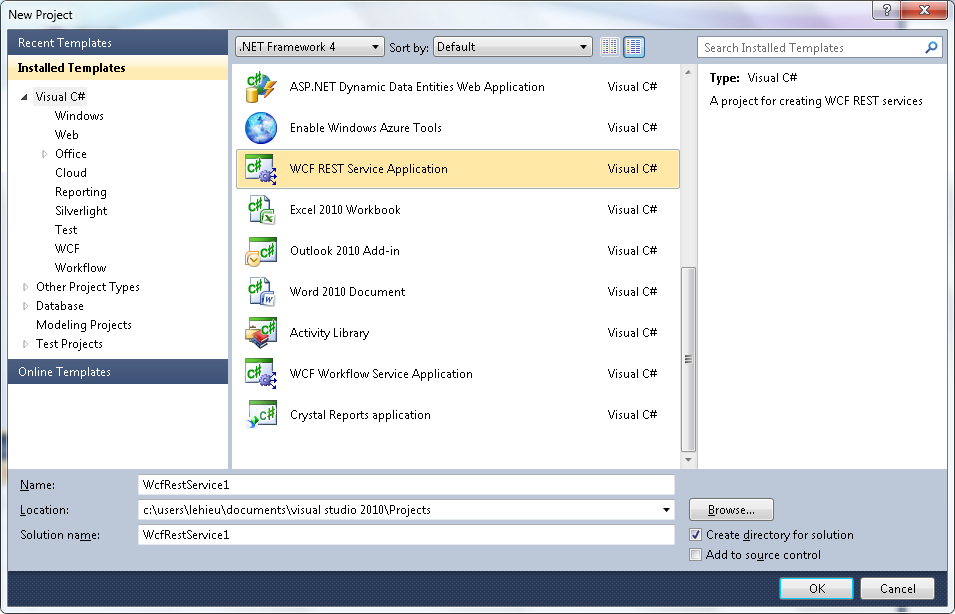
## Xây dựng cơ sở dữ liệu trên máy chủ:

## Xây dựng dịch vụ web:

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. Mô hình mẫu đó được download miễn phí tại đây:

<http://visualstudiogallery.msdn.microsoft.com/fbc7e5c1-a0d2-41bd-9d7b-e54c845394cd>

Sau đó, tiến hành tạo project với mô hình mẫu đó và tiếp tục xây dựng theo những hướng dẫn trên trang MSDN.



*Hình5.8: Tạo project dùng mô hình WCF REST Service 4.0*

Phần code để tạo một dịch vụ web WCF rất đơn giản. Chúng tôi xin phép không trình bày chi tiết hơn ở đây. Người đọc có thể tìm được mã nguồn được đính kèm trong đĩa CD chứa báo cáo này.

# CHƯƠNG 6. KẾT QUẢ ĐẠT ĐƯỢC

---oOo---

Chương này chúng tôi sẽ trình bày những hình ảnh minh họa cho các phần trình bày ở các chương trước cũng như một vài nhận xét về những kết quả đạt được sau khi tiến hành xây dựng toàn bộ hệ thống.

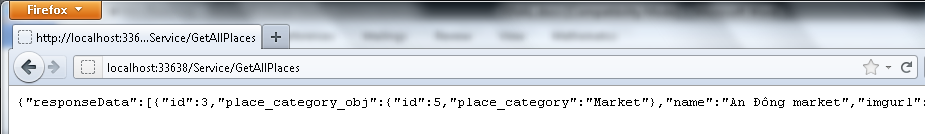
## Ứng dụng Android:

## Cơ sở dữ liệu dùng hiện thực thuật toán:

## Dịch vụ web WCF:

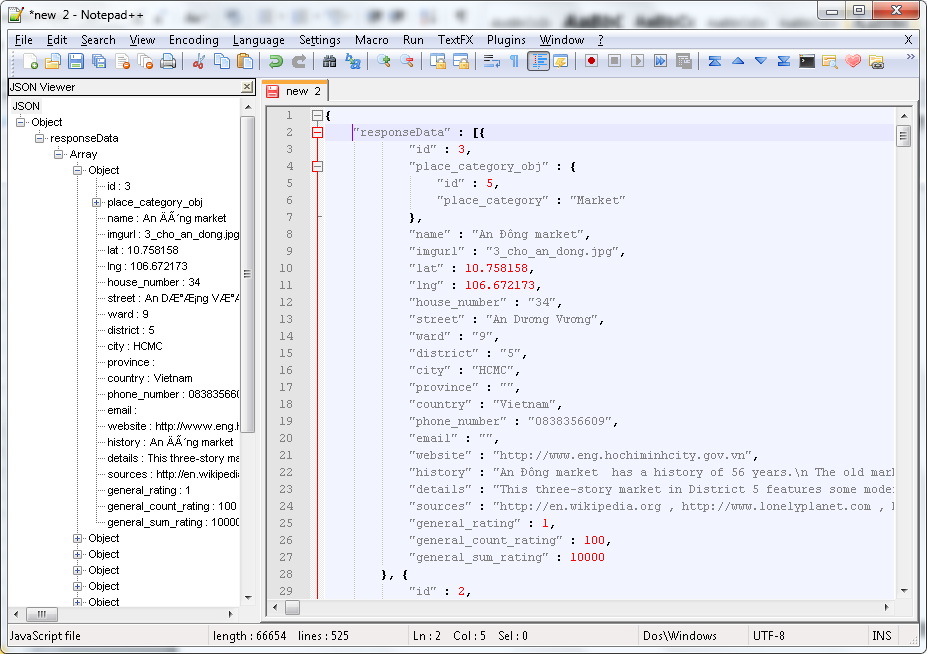
Sau khi triển khai dịch vụ web WCF trên localhost, để thử nghiệm dịch vụ, chúng tôi dùng một trình duyệt bất kì. Trên trình duyệt nhập vào đường link của dịch vụ web muốn sử dụng. Quá trình nhập này là để thử nghiệm. Thực tế trên điện thoại người dùng không phải nhập tay thế này. Ứng với các chức năng cần dùng dịch vụ web, điện thoại sẽ kích hoạt đường dẫn đến dịch vụ web cần thiết và nhận dữ liệu trả về.

Ví dụ sau gọi dịch vụ web để lấy danh sách tất cả các địa điểm du lịch từ cở sở dữ liệu và trả về với dạng chuỗi JSON. Tương tự, chúng tôi cũng dùng dịch vụ web này để trả về danh sách các địa điểm được hệ thống gợi ý cho người dùng tùy theo ngữ cảnh họ cung cấp, để truyền thông tin đánh giá một địa điểm của người dùng từ điện thoại lên máy chủ …



*Hình 6.: Thử nghiệm dịch vụ web WCF trên trình duyệt FireFox 8.*

Để thấy rõ hơn, chúng tôi xem chuỗi JSON trên bằng cách dùng phần mềm Notepad++ với hai plugin là JSMin (dùng để định dạng, xuống dòng cho chuỗi JSON giúp dễ nhìn) và JSON Viewer (dùng để xem cấu trúc chuổi JSON ở dạng cây). Kết quả hiển thị như sau:



*Hình 6.: Chuỗi JSON do dịch vụ web WCF trả về.*

Hình trên cho thấy một mảng các đối tượng được trả về, mỗi đối tượng là một địa điểm du lịch trong cơ sở dữ liệu gồm các thông tin chi tiết của địa điểm đó. Phía điện thoại sau khi nhận chuỗi JSON sẽ tiến hành phân tích và hiển thị thông tin chứa trong chuỗi lên màn hình cho người dùng.

# **CHƯƠNG 7 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

---oOo---

Chương này sẽ là phần tự nhận xét và đánh giá chung về những gì chúng tôi đã cố gắng thực hiện cùng nhau trong khóa luận suốt vài tháng qua.

# Kết luận:

Khoảng thời gian gần 4 tháng từ lúc bắt đầu chọn được đề tài đến khi hoàn thành khóa luận có thể nói là khoảng thời gian khiến chúng tôi căng thẳng nhất ở những năm đại học. Sau khi vượt qua được khoảng thời gian đó, chúng tôi 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 chúng tôi đã trình bày ở các chương trước. Bên cạnh đó, chúng tôi đã xây dựng thành công một trí thông minh cơ bản 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 là chúng tôi phải làm giàu thêm nguồn cơ sở dữ liệu cũng nhưng tìm cách quảng bá hệ thống đến cho nhiều người dùng để họ tiến hành sử dụng và đánh giá những điểm du lịch. Những chỉ số đánh giá của người dùng là cơ sở để hệ thống có thể đưa ra những lời gời ý thích hợp.

Qua quá trình xây dựng hệ thống trên, chúng tôi đã có cơ hội củng cố và học hỏi thêm nhiều nguồn kiến thức mới như kiến thức về 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 mà chúng tôi đã tìm hiểu và triển khai. 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 chúng tôi 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 chúng tôi có thể phát triển tốt hơn những hệ thống khuyến nghị trong tương lai.

Nhìn chung, không gì có thể hoàn hảo tuyệt đối ngay lúc ban đầu, cần phải có rất nhiều thời gian để hoàn thiện từng bước. Và chúng tôi nhận xét rằng những gì thực hiện trong khóa luận này cũng đã đáp ứng được yêu cầu và mục tiêu chúng tôi đã đề ra ban đầu.

# Hướng phát triển:

Như đã nói trên, còn rất nhiều việc chúng tôi 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 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] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, Alexander Tuzhilin (2005), “Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach”.

[2] Kaijian Xu, Manli Zhu, Daqing Zhang, Tao Gu (2008), “Context-Aware Content Filtering & Presentation for Pervasive & Mobile Information Systems”.

[3] Gediminas Adomavicius, Alexander Tuzhilin (2001),” Multidimensional Recommender Systems: A Data Warehousing Approach”.

[4] Linas Baltrunas, Bernd Ludwig, Stefan Peer, and Francesco Ricci (2011), “Context-Aware Places of Interest Recommendations for Mobile Users”.

[5] Stefan Peer (author), Prof. Dr. Francesco Ricci, Linas Baltrunas (supervisor) (2010), “Real-Time Context-Aware Recommendations for Mobile Users” (thesis).

[6] Xiaoyuan Su, Taghi M.Khoshgoftaar (2009), “A Survey of Collaborative Filtering Techniques”.

[7] Jun Wang, Arjen P.de Vries, Marcel J.T. Reinders (2006), “Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion”.

[8] Mark L.Murphy (2010), Beginning Android 2, Apress publisher.

[9] Sayed Y.Hashimi, Satya Komatineni, Dave MacLean (2010), Pro Android 2, Apress publisher.

[10] <http://developer.android.com/>

[11] <http://msdn.microsoft.com/>

[12] J. Breese, D. Heckerman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative ﬁltering,” in Proceedings of the 14th Conference on Uncertainty in Artiﬁcial Intelligence (UAI ’98), 1998

[13] C. Basu, H. Hirsh, and W. Cohen, “Recommendation as classiﬁcation: using social and content-based information in recommendation,” in Proceedings of the 15th National Conference on Artiﬁcial Intelligence (AAAI ’98), pp. 714–720, Madison,Wis, USA, July 1998

[14] J. Canny, “Collaborative ﬁltering with privacy via factor analysis,” in Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 238–245, Tampere, Finland, August 2002.

[15] Salton, G., 1989. Automatic Text Processing. Addison-Wesley.

[16] P. Melville, R. J. Mooney, and R. Nagarajan, “Content-boosted collaborative filtering for improved recommendations,” in Proceeding s of the 18th National Conference on Artificial Intelligence (AAAI ’02) , pp. 187–192, Edmonton,Canada, 2002.

[17] Baeza-Yates, R. and B. Ribeiro-Neto, 1999. Modern Information Retrieval. Addison-Wesley.

[18] Billsus, D. and M. Pazzani, 1998. Learning collaborative information filters. In International Conference on Machine Learning, Morgan Kaufmann Publishers.

[19] Shardanand, U. and P. Maes, 1995. Social information filtering: Algorithms for automating ‘word of mouth’. In Proceedings of the Conference on Human Factors in Computing Systems.

[20] Resnick, P., N. Iakovou, M. Sushak, P. Bergstrom, and J. Riedl, 1994. GroupLens: An open architecture for collaborative filtering of netnews. In Proceedings of the 1994 Computer Supported Cooperative Work Conference.

[21] Francesco Ricci · Lior Rokach · Bracha Shapira ·Paul B. Kantor, “Recommender Systems Handbook”, Springer 2011.

[23] D. Billsus and M. Pazzani, “Learning collaborative information filters,” in Proceedings of the 15th International Conference on Machine Learning (ICML ’98), 1998.

[24] T. Landauer, M. Littman, and Bell Communications Research (Bellcore), “Computerized cross-language document retrieval using latent semantic indexing,” US patent no.5301109, April 1994.

[25] S.Deerwester,S.T.Dumais,G.W.Furnas,T.K.Landauer, and R. Harshman, “Indexing by latent semantic analysis,” Journal of the American Society for Information Science, vol.41, no. 6, pp. 391–407, 1990.

[26] R. Greinemr, X. Su, B. Shen, and W. Zhou, “Structural extension to logistic regression: discriminative parameter learning of belief net classifiers,” Machine Lear ning, vol. 59, no. 3, pp. 297–322, 2005.

[27] N. Srebro, J. D. M. Rennie, and T. Jaakkola, “Maximum margin matrix factorization,” in Advances in Neural Information Processing Systems , vol. 17, pp. 1329–1336, 2005.

[28] J. D. M. Rennie and N. Srebro, “Fast maximum margin matrix factorization for collaborative prediction,” in Proceedings of the 22nd International Conference on Machine Learning. (ICML ’05) , Bonn, Germany, August 2005.

[29] B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Incremental SVD-based algorithms for highly scaleable recommender systems,” in Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT ’02) , 2002.

[30] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in Proceedings of the 10th International Confere nce on World Wide We b ( WWW ’01) , pp. 285–295, May 2001.

[31] M. O’Connor and J. Herlocker, “Clustering items for collaborative filtering,” in Proceedings of the ACM SIGIR Workshop on Recommender Systems ( SIGIR ’99) , 1999.

[32] G.-R. Xue, C. Lin, Q. Yang, et al., “Scalable collaborative filtering using cluster-based smoothing,” in Proceedings of the ACM SIGIR Conference, pp. 114–121, Salvador, Brazil, 2005.

[33] M. Claypool, A. Gokhale, T. Miranda, et al., “Combining content-based and collaborative filters in an online newspaper,” in Proceeding s of the SIGIR Workshop on Recommender Systems: Algorithms and Evaluation, Berkeley, Calif, USA, 1999.

[34] S. K. Lam and J. Riedl, “Shilling recommender systems for fun and profit,” in Proceedings of the 13th International World Wide Web Conference ( WWW ’04), pp. 393–402, New York, NY, USA, May 2004.

[35] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams, “Effective attack models for shilling item-based collaborative filtering systems,” in Proceedings of the WebKDD Workshop, August 2005.

[36] M. O’Mahony, N. Hurley, N. Kushmerick, and G. Silvestre, “Collaborative recommendation: arobustness analysis,” ACM Transactions on Internet Technology,vol.4,no.4,pp.344–377, 2004.

[37] G. Karypis, “Evaluation of item-based top-N recommendation algorithms,” in Proceedings of the International Conference on Information and KnowledgeManagement (CIKM’01), pp. 247–254, Atlanta, Ga, USA, November 2001

# PHỤ LỤC

---oOo---

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

|  |  |
| --- | --- |
| **Mã use case** | **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. |

1. **Use case “Xem tất cả các địa điểm”:**

|  |  |  |
| --- | --- | --- |
| **Mã**  **use case** | 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. | 1. Điện thoại gửi yêu cầu truy vấn tất cả địa điểm đến máy chủ. 2. Máy chủ xử lý yêu cầu và trả về chuỗi JSON chứa những thông tin địa điểm. 3. 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. | |

1. **Use case “Xem các địa điểm được hệ thống gợi ý”:**

|  |  |  |
| --- | --- | --- |
| **Mã**  **use case** | 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. | 1. Điện thoại gửi yêu cầu truy vấn các địa điểm cần gợi ý đến máy chủ. 2. Máy chủ xử lý yêu cầu và trả về chuỗi JSON chứa những thông tin địa điểm được gợi ý. 3. 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. | |

1. **Use case “Xem các địa điểm ưa thích được người dùng lưu lại”:**

|  |  |  |
| --- | --- | --- |
| **Mã**  **use case** | 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. | 1. Điện thoại kết nối cơ sở dữ liệu cục bộ SQLite. 2. SQLite xử lý yêu cầu và trả về những thông tin địa điểm ưa thích. 3. 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. | |

1. **Use case “Xem thông tin và cấu hình điều kiện ngữ cảnh”:**

|  |  |  |
| --- | --- | --- |
| **Mã**  **use case** | 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. 2. Người dùng tùy chỉnh thông tin ngữ cảnh và nhấn nút Save. | 1. Điện thoại kết nối cơ sở dữ liệu cục bộ SQLite. 2. SQLite truy vấn và trả về những thông tin ngữ cảnh. 3. Màn hình điện thoại hiển thị các thông tin ngữ cảnh. 4. 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. | |

1. **Use case “Xem và cấu hình thông tin cá nhân”:**

|  |  |  |
| --- | --- | --- |
| **Mã**  **use case** | 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. Người dùng tùy chỉnh thông tin cá nhân và nhấn nút Save. | 1. Điện thoại kết nối cơ sở dữ liệu cục bộ SQLite. 2. SQLite truy vấn và trả về những thông tin cá nhân. 3. Màn hình điện thoại hiển thị các thông tin cá nhân. 4. 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. | |

1. **Use case “Hiển thị danh sách địa điểm được sắp xếp theo thứ tự”:**

|  |  |  |
| --- | --- | --- |
| **Mã**  **use case** | 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. | 1. Điện thoại xử lý danh sách địa điểm và sắp xếp theo yêu cầu. 2. Điện thoại hiện lại danh sách mới đã được sắp xếp. |
| **Các xử lý ngoại lệ** | Không có. | |

1. **Use case “Xem thông tin chi tiết của một địa điểm”:**

|  |  |  |
| --- | --- | --- |
| **Mã**  **use case** | 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. | 1. 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. | |

1. **Use case “Đánh giá địa điểm”:**

|  |  |  |
| --- | --- | --- |
| **Mã**  **use case** | 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. 2. Ở màn hình Rate, người dùng chọn số sao muốn đánh giá và nhấn nút Rate. | 1. Điện thoại chuyển qua màn hình Rate. 2. 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. | |

1. **Use case “Hiển thị địa điểm lên bản đồ”:**

|  |  |  |
| --- | --- | --- |
| **Mã**  **use case** | 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. | 1. 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. | |

1. **Use case “Thêm địa điểm vào danh sách ưa thích”:**

|  |  |  |
| --- | --- | --- |
| **Mã**  **use case** | 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. | 1. 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. | |

1. **Use case “Xóa địa điểm khỏi danh sách ưa thích”:**

|  |  |  |
| --- | --- | --- |
| **Mã**  **use case** | 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. 2. Người dùng xác nhận. | 1. Điện thoại hiện thông báo yêu cầu xác nhận xóa. 2. 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. | |

1. **Use case “Tìm kiếm địa điểm ưa thích”:**

|  |  |  |
| --- | --- | --- |
| **Mã**  **use case** | 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). | 1. Điện thoại tiến hành truy vấn cơ sở dữ liệu cục bộ SQLlite. 2. 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://myphonedeals.co.uk> )

