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 luận văn 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 luận văn.

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 luận văn, 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 và xin kính chúc quý thầy cô dồi dào sức khỏe và luôn luôn thành công hơn nữa trong sự nghiệp trồng người cao quý!

MỤC LỤC

[CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI 3](#_Toc311553483)

[1.1 ĐẶT VẤN ĐỀ: 3](#_Toc311553484)

[1.2 PHẠM VI NGHIÊN CỨU: 3](#_Toc311553485)

[1.3 ĐỊNH HƯỚNG CÁCH GIẢI QUYẾT: 3](#_Toc311553486)

[CHƯƠNG 2: KHẢO SÁT MỘT VÀI THUẬT TOÁN KHUYẾN NGHỊ ĐÃ CÓ 5](#_Toc311553487)

[2.1. THUẬT TOÁN KHUYẾN NGHỊ HAI CHIỀU TRUYỀN THỐNG: 5](#_Toc311553488)

[2.2. THUẬT TOÁN KHUYẾN NGHỊ ĐA NHIỀU: 10](#_Toc311553489)

[CHƯƠNG 3: HƯỚNG TIẾP CẬN VÀ GIẢI PHÁP 11](#_Toc311553490)

[CHƯƠNG 4: HIỆN THỰC VÀ CÀI ĐẶT 12](#_Toc311553491)

[4.1. KIẾN TRÚC TOÀN HỆ THỐNG: 12](#_Toc311553492)

[4.2. MỘT SỐ MÔ TẢ VỀ CÁC KỸ THUẬT: 12](#_Toc311553493)

[4.2.1. Cơ sở dữ liệu và kho dữ liệu hỗ trợ cài đặt thuật toán: 12](#_Toc311553494)

[4.2.2. Web services và windows communication foundation: 12](#_Toc311553495)

[4.2.3. Hệ điều hành Android cho điện thoại thông minh: 16](#_Toc311553496)

[4.3. XÂY DỰNG ỨNG DỤNG MINH HỌA TRÊN ANDROID: 21](#_Toc311553497)

[4.3.1. Kiến trúc ứng dụng: 21](#_Toc311553498)

[4.3.2. Sơ đồ các màn hình: 24](#_Toc311553499)

[4.3.3. Danh mục các use cases: 24](#_Toc311553500)

[CHƯƠNG 5: KẾT QUẢ ĐẠT ĐƯỢC 35](#_Toc311553501)

[CHƯƠNG 6: KẾT LUẬN 36](#_Toc311553502)

[TÀI LIỆU THAM KHẢO 37](#_Toc311553503)

# CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

## **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.

## PHẠM VI NGHIÊN CỨU:

**Phạm vi công nghệ:**

* Nền tảng di động Android.
* Cơ sở dữ liệu Microsoft SQL Server.
* Kho dữ liệu (data warehouse).
* Các dịch vụ web (Web services, WCF – Windows Communication Foundation)

**Phạm vi địa lý:** Thành phố Hồ Chí Minh.

**Đối tượng nghiên cứu:** Các thuật toán khuyến nghị.

## ĐỊNH HƯỚNG CÁCH GIẢI QUYẾT:

Do yêu cầu bài toán đặt ra liên quan đến nhiều điều kiện ngữ cảnh khác nhau trong quá trình khuyến nghị nên chúng tôi chọn hướng tiếp cận đa chiều (Multidimensional Approach). Để xây dựng một hệ thống khuyến nghị phụ thuộc những điều kiện ngữ cảnh theo hướng tiếp cận đa chiều, cần có một thuật toán thích hợp. Sau khi tìm hiểu và nghiên cứu các thuật toán đã được ứng dụng trong quá khứ, chúng tôi quyết định chọn cách kết hợp 2 thuật toán Memmory-based (một nhánh thuộc thuật toán Collaborative Filtering) và Reduction-based để giải quyết bài toán.

Về vấn để hiện thực thành ứng dụng đến tay người dùng, chúng tôi chọn nền tảng di động thông minh Android. Bên cạnh đó là những kỹ thuật khác về xây dựng cơ sở dữ liệu, kho dữ liệu, các dịch vụ web … để hoàn thiện hệ thống.

Tại sao lại chọn những thuật toán và kỹ thuật trên, chúng tôi sẽ đi vào chi tiết hơn ở những phần sau.

# CHƯƠNG 2: KHẢO SÁT MỘT VÀI THUẬT TOÁN KHUYẾN NGHỊ ĐÃ CÓ

## 

## THUẬT TOÁN KHUYẾN NGHỊ HAI CHIỀU TRUYỀN THỐNG:

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 phim “Người nhện” với chỉ số là 4 (trên tổng mức 5).
* Người A đánh giá phim “Siêu nhân” với chỉ số là 5 (trên tổng mức 5).
* Người A đánh giá phim “Dị nhân” với chỉ số là 3 (trên tổng mức 5).
* Người B đánh giá phim “Người nhện” với chỉ số là 4 (trên tổng mức 5).
* Người B đánh giá phim “Siêu nhân” 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 phim “Dị nhân” 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: collaborative, content-based và hybrid. Chúng tôi sẽ lần lượt nói qua một cách vắn tắt về chúng ngay sau đây.

**Hệ thống khuyến nghị dựa trên nội dung (Content-based).**

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 bảng (*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 bảng, 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)* do Salton đưa ra (1989).

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*. 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.

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 bảng) hoặc được cung cấp một cách thủ công (thông tin của các bộ phim được mô tả cũng ở dạng văn bảng). Còn đối với những thông tin dạng đa phương tiện (hình ảnh, âm thanh …) thì thật sự rất khó khăn. 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 dựa trên ký ức (memory-based hay còn gọi là heuristic-based) và phương pháp dựa trên mô hình (model-based). Phương pháp 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 kết tụ (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 .

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.

Ngược lại với phương pháp khuyến nghị dựa trên ký ức, phương pháp khuyến nghị dựa trên mô hình (model-based) dùng tập hợp những chỉ số đánh giá của người dùng để học một mô hình (model), sau đó dùng mô hình đó để đưa ra những chỉ số dự đoán. Lấy ví dụ một kỹ thuật khuyến nghị dựa trên mô hình dùng cách tiếp cận xác suất thống kê, chỉ số đánh giá dự đoán được tính theo công thức:

Trong đó *x* là những giá trị đánh giá dạng số nguyên từ 0 đến n, Pr là xác suất có điều kiện người dùng *u* sẽ đánh giá đối tượng *i* với chỉ số là *x* sau khi người đó đã đánh giá các đối tượng *i’*.

Ngoài ra, còn có những kỹ thuật khác như là mô hình hồi quy tuyến tính, các giải thuật gom cụm … kết hợp với các kỹ thuật rút trích, cắt giảm số chiều của ma trận cũng được nghiên cứu. Người ta cũng đã tiến hành so sánh và ghi nhận phương pháp khuyến nghị dựa trên mô hình trong một vài trường hợp cho hiệu năng cao hơn, độ chính xác cao hơn so với phương pháp khuyến nghị dựa trên ký ức.

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 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 (hybrid). 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.

## THUẬT TOÁN KHUYẾN NGHỊ ĐA NHIỀU:

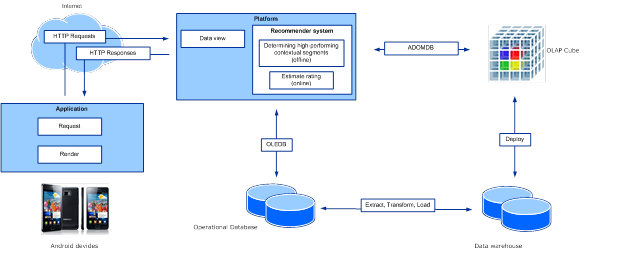
Trên đây là cái nhìn tổng quan về các hệ thống khuyến nghị truyền thống. Ta có thể nhận thấy điểm chung ở các hệ thống trên là chúng chỉ quan tâm đến người dùng và đối tượng (chỉ hai chiều) chứ không quan tâm đến những thông tin ngữ cảnh bên ngoài như thời gian, nơi chốn, thời tiết … Những điều kiện ngữ cảnh đó cũng có thể ảnh hưởng đến các chỉ số đánh giá, từ đó kéo theo sự ảnh hưởng của những lời gợi ý. Ví dụ với điều kiện trời mưa, người dùng sẽ không thích đi dã ngoại nhưng lại thích đi xem phim trong rạp chẳng hạn. 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).

# Xem 7., HandbookCHƯƠNG 3: HƯỚNG TIẾP CẬN VÀ GIẢI PHÁP

Thuật toán khuyến nghị đa chiều kết hợp 2 phương pháp Reduction-based và Collaborative Filtering 2 chiều

# CHƯƠNG 4: HIỆN THỰC VÀ CÀI ĐẶT

## KIẾN TRÚC TOÀN HỆ THỐNG:



Hệ thống gồm 3 phần chính:

* Phần 1 (Front-end): 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 (Back-end): 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 (trung gian): web services phụ trách việc truyền nhận thông tin giữa điện thoại với hệ thống khuyến nghị.

## MỘT SỐ MÔ TẢ VỀ CÁC KỸ THUẬT:

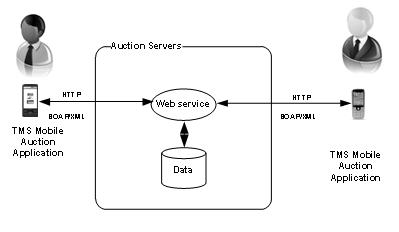
### Cơ sở dữ liệu và kho dữ liệu hỗ trợ cài đặt thuật toán:

### Web services và windows communication foundation:

1. **Web services là gì?**

Có nhiều cách định nghĩa khác nhau về web services. Theo tổ chức W3C, web services 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 server đến nhiều thiết bị điện thoại khác nhau cho nhiều người ở nhiều nơi thông qua web services. Không những điện thoại, bất kỳ thiết bị nào khác có hỗ trợ Internet đều có thể truyền nhận dữ liệu.



Điều này giúp thông tin được truyền tải theo cách độc lập nền tảng thiết bị và ngôn ngữ lập trình vì mọi thứ đều được chuẩn hóa về dạng web. Web services 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.

1. **Ví dụ về Google Local Search**

Google Local Search là một dạng web services được Google xây dựng nhằm mục đích hỗ trợ người dùng tìm kiếm các địa điểm xung quanh mình. Trên điện thoại gừi một HTTP request lên server, server xử lý và trả về kết quả là chuỗi JSON chứa các địa điểm cần tìm.

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

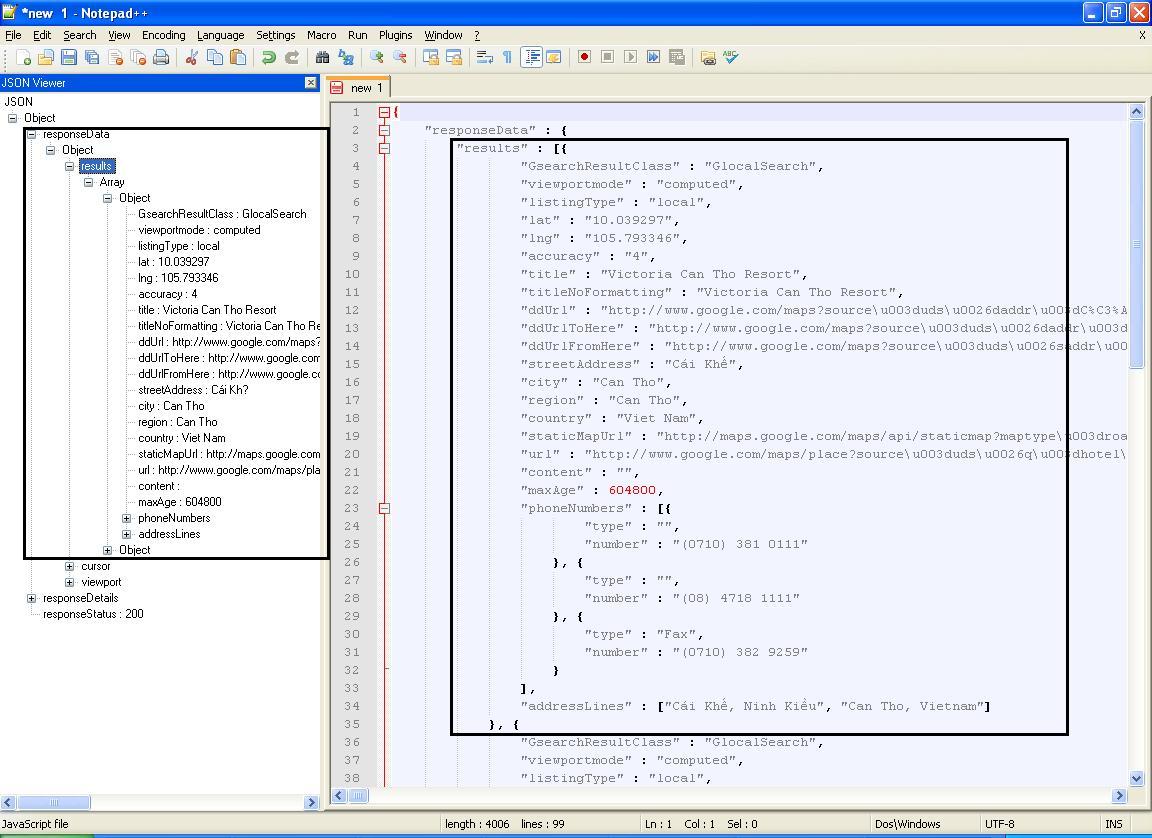
***Ý nghĩa:*** tìm kiếm các khách sạn xung quanh điểm có tọa độ (10,106)

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.



Kết quả trả về với mảng gồm 2 đối tượng địa điểm và thông tin chi tiết của từng điểm (tọa độ, tên, địa chỉ, điện thoại, bản đồ ...).

1. **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 web services – một thành phần trung gian giúp truyền nhận thông tin giữa cơ sở dữ liệu trên server và ứng dụng trên điện thoại Android.

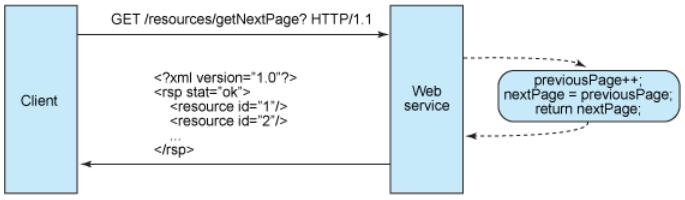
WCF là một phần của .NET Framework cung cấp một mò hình lập trình thống nhất hỗ trợ xây dựng một cách nhanh chóng và dễ dàng các ứng dụng theo hướng dịch vụ (service-oriented) mà giữa chúng truyền thông với nhau thông qua web. Được tích hợp lần đầu tiên trong .NET Framework 3.0 năm 2006, phiên bản mới nhất hiện tại là .NET Framework 4.0.

Chúng ta hoàn toàn có thể xây dựng một web service dùng ASP.NET Web Services (ASMX). Tại sao cần đến WCF? WCF cung cấp nhiều lợi ích hơn ASMX, có thể kể đến như:

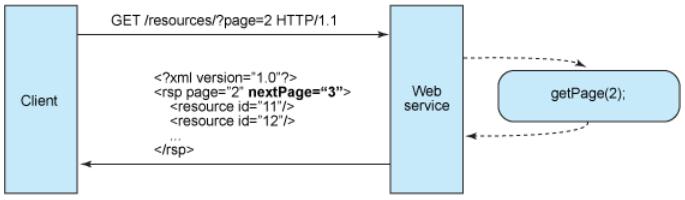
* Hỗ trợ truyền nhận thông tin không chỉ qua giao thức HTTP, mà còn có thể thông qua TCP hoặc những giao thức mạng khác.
* Khả năng chuyển đổi giữa các giao thức truyền nhận thông tin một cách dễ dàng.
* Hỗ trợ việc host services không những trên một web server mà còn trên nhiều dạng khác như: WinForms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



Hình: 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 trên Internet.

### Hệ điều hành Android cho điện thoại thông minh:

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, 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.

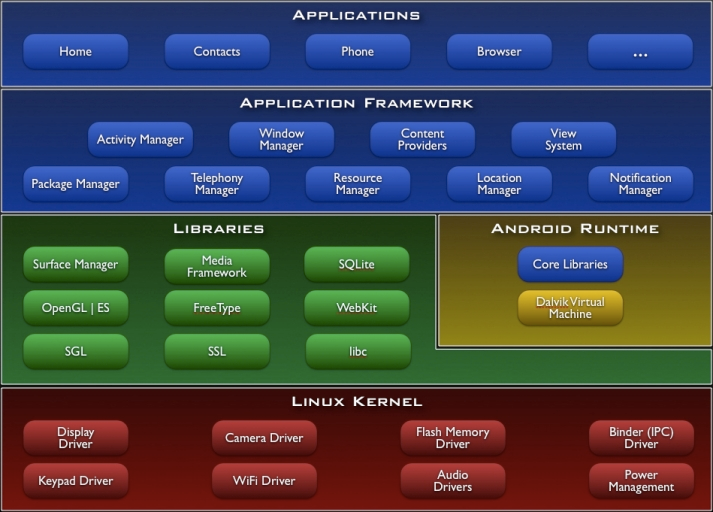
Bên cạnh những lợi thế trên, điện thoại cũng có những nhược điểm. Một trong số đó là giao diện người dùng. Điện thoại đa số có cỡ màn hình từ 3-4 inches, không phải màn hình rộng 19-21 inches như máy tính. Việc bố trí các chức năng, cách hiển thị thông trên một khung nhìn nhỏ cũng phải rất cẩn thận sao 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.

1. **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), Windows Phone (của Microsoft) và RIM OS (của BlackBerry).

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 ngày 5 tháng 11 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.

Sơ đồ dưới đây thể hiện 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 …)

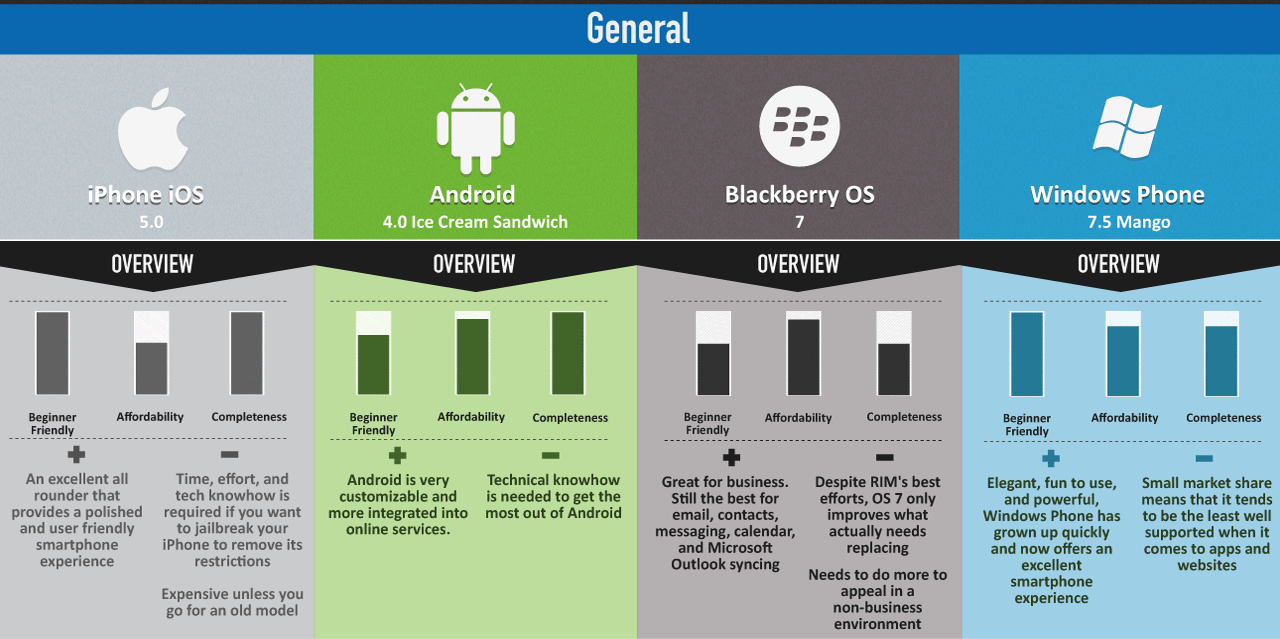
**Libraries:** gồm một tập các thư viện C/C++ được viết sẵn hỗ trợ xử lý âm thanh, hình ảnh, hiệu ứng đồ họa 2D, 3D, trình duyệt web, cơ sở dữ liệu SQLite …

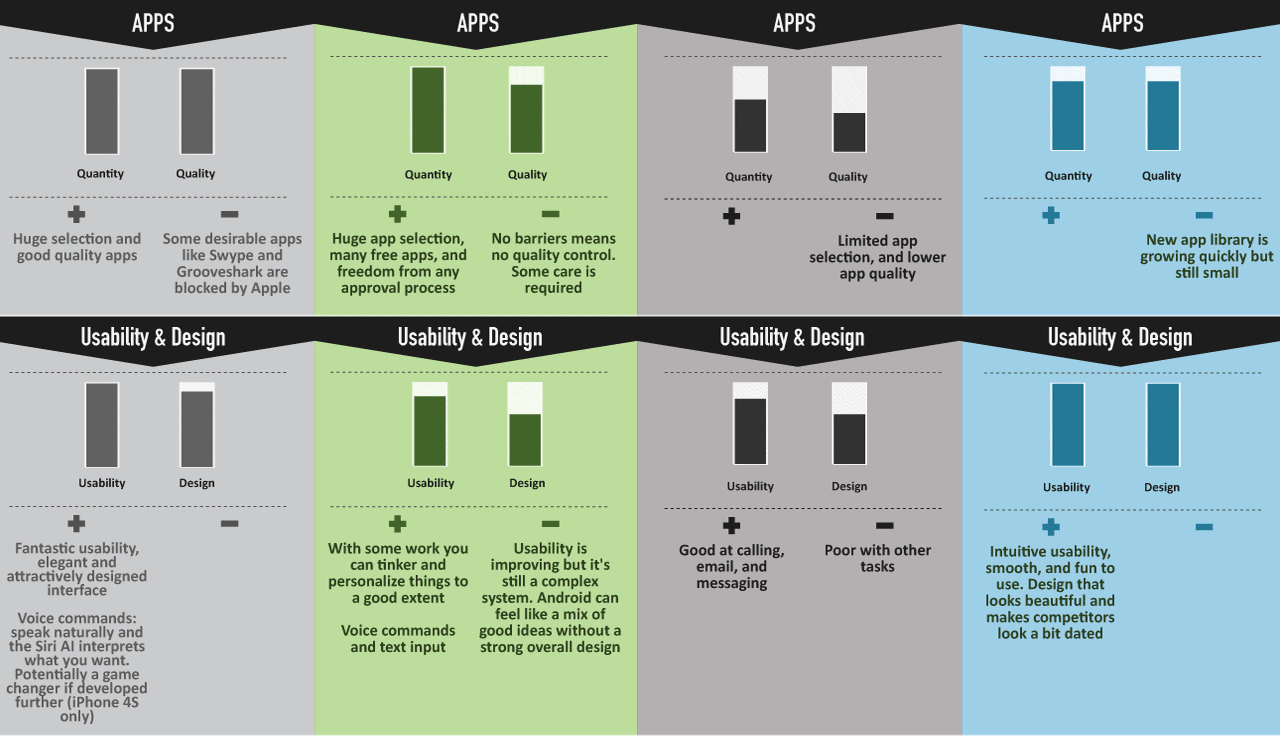
**Android Runtime:** mỗi ứng dụng Android chạy trong một thể hiện của máy ảo Dalvik. Trên Android hỗ trợ chạy đa nhiệm. Máy ảo Dalvik thực thi những file ứng dụng ở dạng .dex (Dalvik Executable) được tối ưu hóa cho bộ nhớ và phần cứng điện thoại.

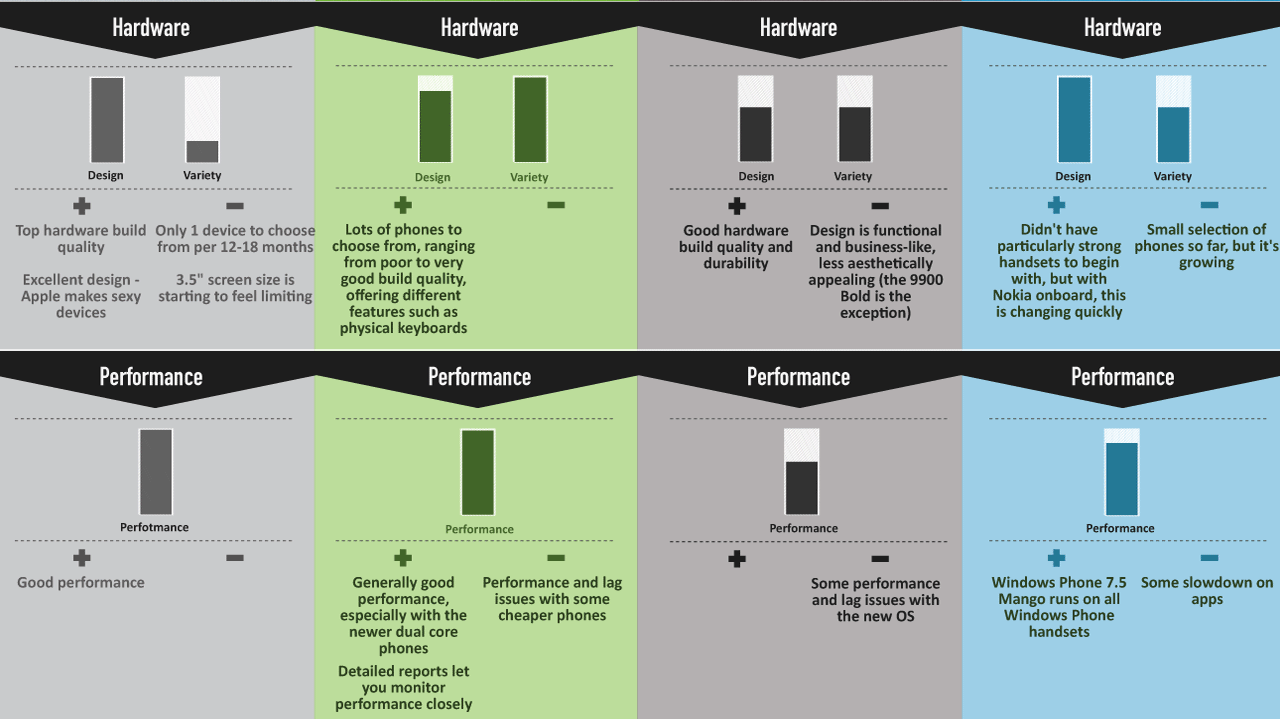
**Linux Kernel:** Android dựa trên nhân Linux phiên bản 2.6 cung cấp khả năng bảo mật, quản lý bộ nhớ, quản lý tiến trình …

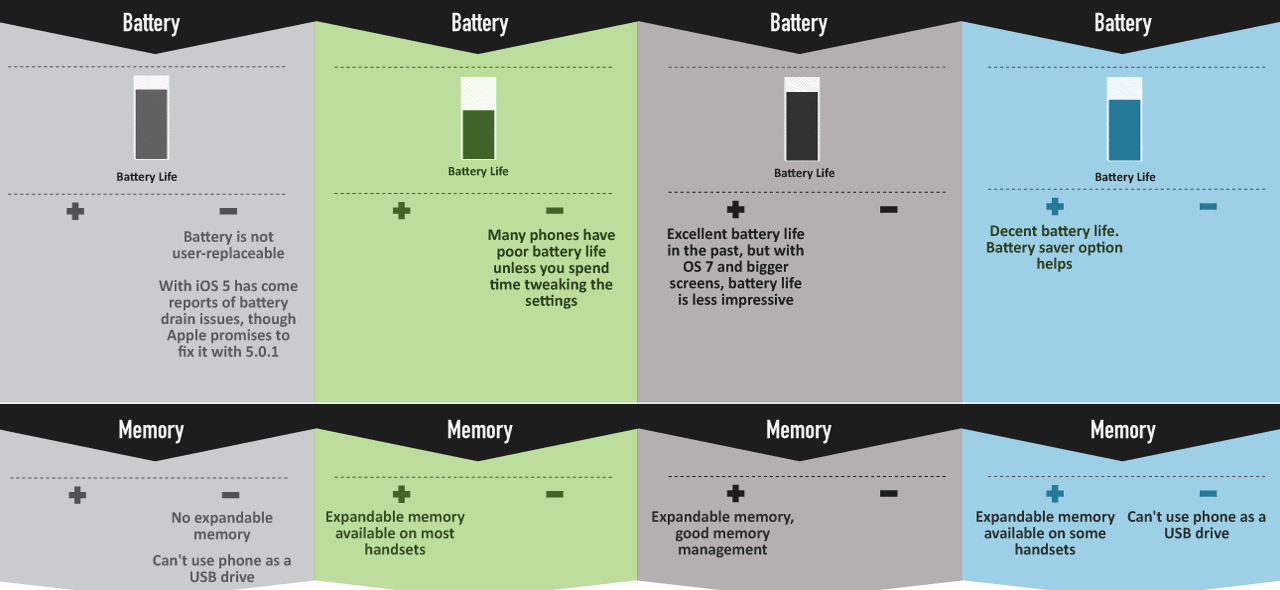
Chi tiết hơn về hệ điều hành Android, người đọc có thể tìm hiểu thêm trên trang <http://developer.android.com/>

Dưới đây là một phần của bảng so sánh hoàn chỉnh các hệ điều hành được cập nhật vào tháng 11/2011 (nguồn: <http://myphonedeals.co.uk> )









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.

1. **Chọn lựa giữa Native-app và Web-app trên điện thoại?**

**Native-app** (tạm dịch là ứng dụng thuần, ứng dụng gốc): 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.

**Web-app** (ứng dụng nền tảng web): 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.

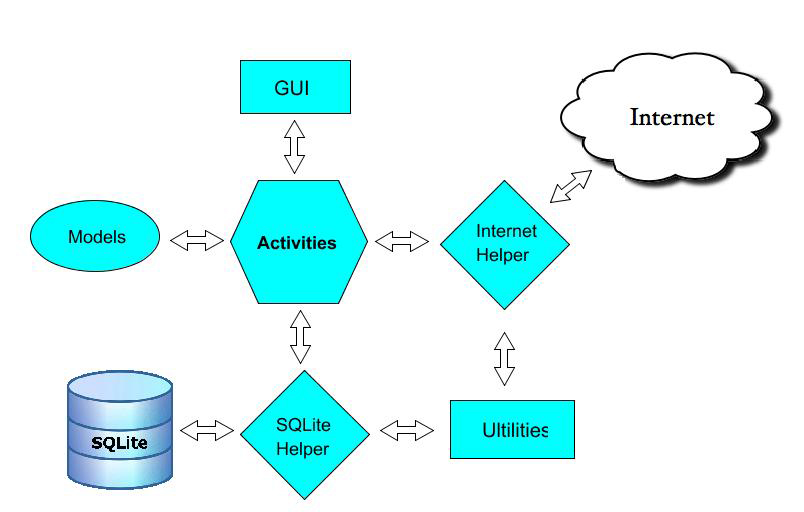
Mỗi loại đều có ưu nhược điểm riêng, nhưng chúng tôi đã quyết định chọn native-app vì những lý do sau:

* 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).

## XÂY DỰNG ỨNG DỤNG MINH HỌA TRÊN ANDROID:

### Kiến trúc ứng dụng:

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

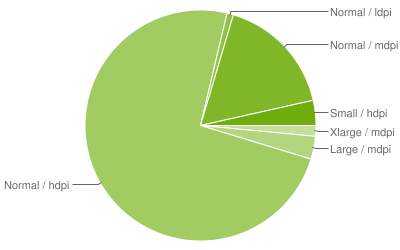


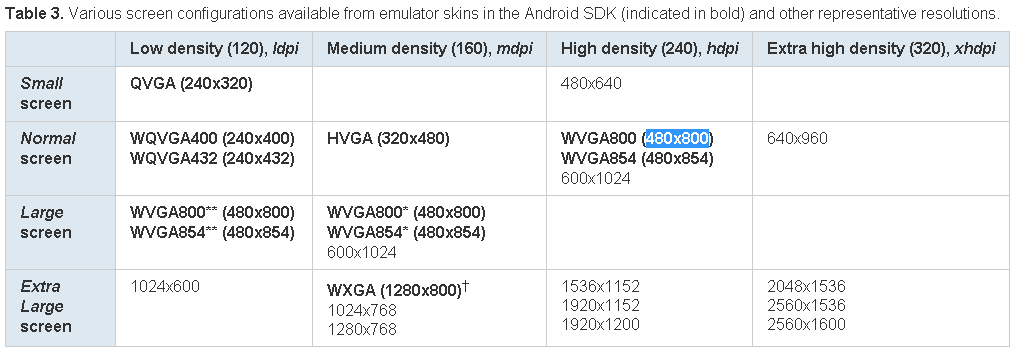
Những thành phần quan trọng nhất gồm:

* **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% |  |



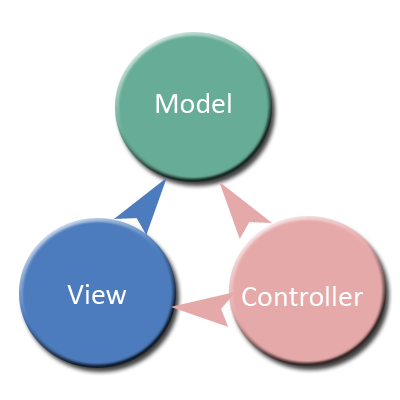


Trong đó, small, normal, large là kích cỡ màn hình từ nhỏ đến lớn; ldpi, mdpi, hdpi là mật độ điểm ảnh từ thấp đến cao (DPI viết tắt của Dots Per Inches – số điểm ảnh trên một đơn vị diện tích tính bằng inch). Màn hình cỡ normal với mật độ hdpi chiếm phần lớn. Kế đến là loại normal/mdpi. Do thời gian 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.

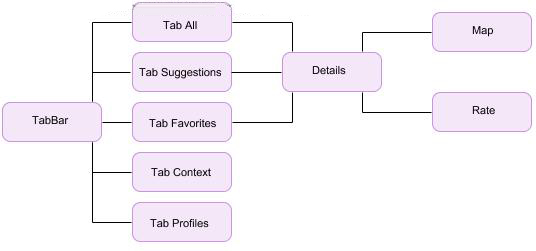
* **Cơ sở dữ liệu SQLite:** cơ sở dữ liệu cục bộ trên điện thoại được xây dựng nhằm lưu trữ những thông tin cấu hình ngữ cảnh, địa điểm du lịch ưa thích mà người dùng muốn lưu lại, cùng một số thông tin khác. Thiết kế cơ sở dữ liệu này tương tự với cơ sở dữ liệu ở back-end. Và SQLite Helper là thành 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.



* **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:

****

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.

### Danh mục các use cases:

Để tránh dài dòng, chúng tôi xin phép bỏ qua các sơ đồ khác trong phân thích thiết kế UML. Tuy nhiên vẫn mô tả chi tiết tất cả những use cases trong ứng dụng để người đọc có thể hiểu rõ các chức năng.

|  |  |
| --- | --- |
| **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 | Hiển thị 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 server. 2. Server 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 server. 2. Server 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 server. | |
| **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 server và server cập nhật vào cơ sở dữ liệu. |
| **Các xử lý ngoại lệ** | * **UC\_11\_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\_13\_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\_14\_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\_15\_2E:** nếu không tìm thấy địa điểm trong cơ sở dữ liệu thì hiện thông báo không tìm thấy. | |

# CHƯƠNG 5: KẾT QUẢ ĐẠT ĐƯỢC

Chụp hình minh họa demo …

Nhận xét đánh giá …

CHƯƠNG 6: KẾT LUẬN

# TÀI LIỆU THAM KHẢO

[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/>