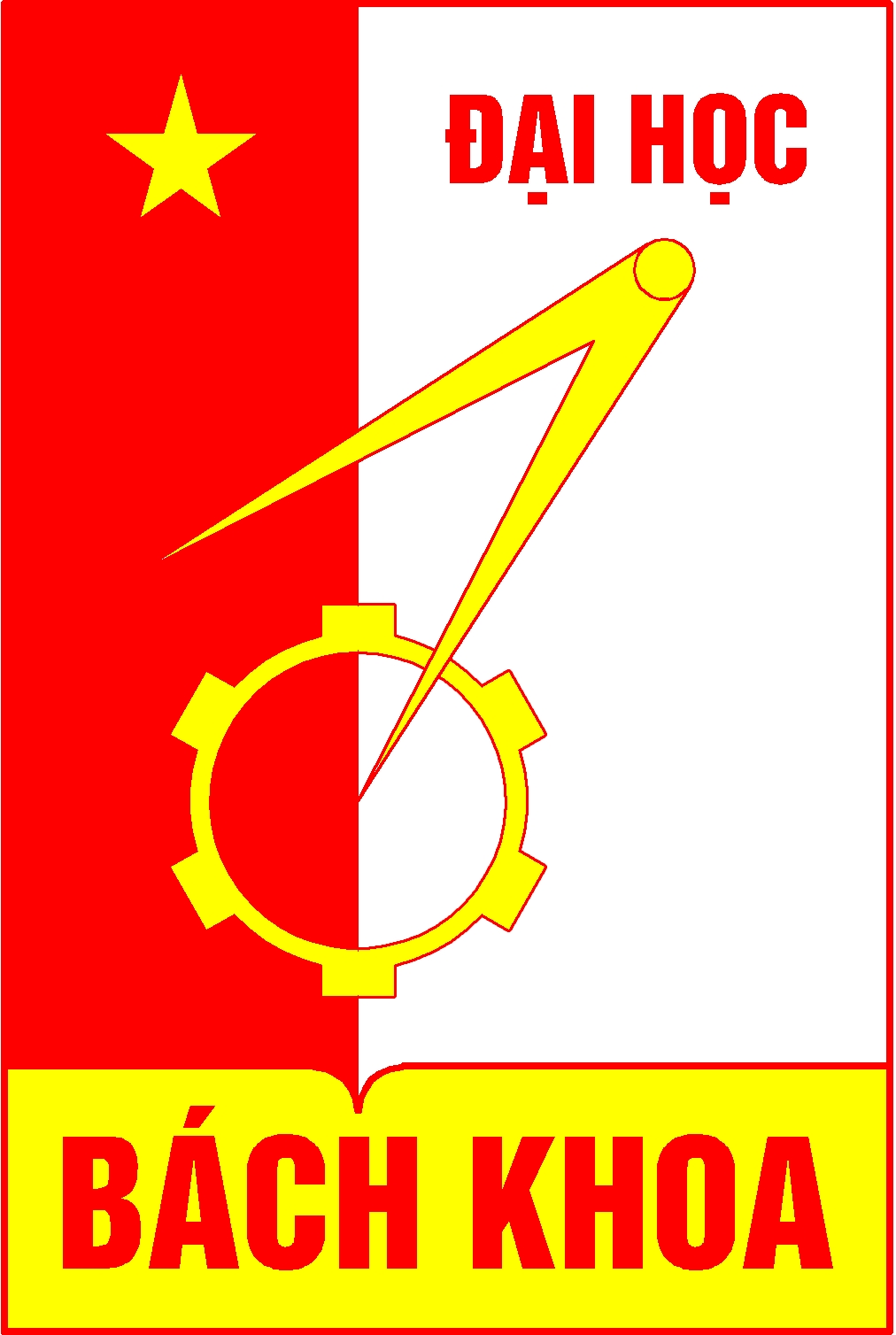
Hà Nội, 18 Tháng Mười Một 2014

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**BÁO CÁO BÀI TẬP LỚN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

Đề tài: phần mềm tra cứu tìm kiếm khách sạn trên nền tản di động

|  |  |
| --- | --- |
| Sinh viên thực hiện | MSSV |
| Hoàn Doãn Quân | 20093579 |
| Nguyễn Hồng Quân | 20122279 |
| Đàm Văn Tài | 20122371 |
| Nguyễn Gia Tuyến | 20122726 |
| Trần Hữu Đạt | 20121505 |

Mục lục

[**MỞ ĐẦU** 2](#_Toc404237512)

[**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG** 3](#_Toc404237513)

[**1.** **Cơ sở lý thuyết** 3](#_Toc404237514)

[**1.1.** **Lập trình Android** 3](#_Toc404237515)

[**1.2.** **WebServices** 5](#_Toc404237516)

[**1.3.** **JSON** 10](#_Toc404237517)

[**2.** **Các công nghệ sử dụng** 11](#_Toc404237518)

[**2.1.** **Công nghệ định vị GPS** 11](#_Toc404237519)

[**2.2.** **Google Maps API** 12](#_Toc404237520)

[**CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG** 15](#_Toc404237521)

[1. Biểu đồ triển khai hệ thống 15](#_Toc404237522)

[2. Biểu đồ ca sử dụng tổng quan 15](#_Toc404237523)

[3. Chức năng tìm kiếm khách sạn theo bán kính 16](#_Toc404237524)

[3.1. Biểu đồ ca sử dụng 16](#_Toc404237525)

[3.2. Biểu đồ tuần tự 17](#_Toc404237526)

[4. Chức năng thông tin chi tiết về khách sạn 18](#_Toc404237527)

[4.1. Biểu đồ ca sử dụng 18](#_Toc404237528)

[4.2. Biểu đồ tuần tự 19](#_Toc404237529)

[**CHƯƠNG 3: THIẾT KẾ CHI TIẾT VÀ GIẢI PHÁP KỸ THUẬT** 20](#_Toc404237530)

[1. Màn hình main (MainActivity) 20](#_Toc404237531)

[2. Màn hình hiện thị khách sạn (ActivityShowHotel) 20](#_Toc404237532)

[3. Màn hình hiển thị thông tin chi tiết khách sạn. 22](#_Toc404237533)

[**CHƯƠNG 4: KẾT LUẬN VÀ ĐÁNH GIÁ** 23](#_Toc404237534)

[**Các hình ảnh trong báo cáo** 24](#_Toc404237535)

[**Các bảng trong báo cáo** 25](#_Toc404237536)

[**Tài liệu tham khảo** 26](#_Toc404237537)

# **MỞ ĐẦU**

Sự phát triển mạnh mẽ của xu thế di động trong hiện tại và cả tương lai đã dẫn đến nhu cầu cấp thiết để phát triển các ứng dụng cho các nền tảng mobile. Trong số đó không thể không kể đến các ứng dụng sử dụng WebServices. WebServices là một công nghệ tuyệt vời của kĩ thuật lập ‎trình phân. ‎Giá trị cơ bản của WebServices cho phép các đối tượng khác nhau ở rất nhiều hệ thống khác nhau ‎truy cập đến.  
 Sinh viên lại là đối tượng giới trẻ nắm bắt công nghệ nhanh, có nhu cầu tìm kiếm địa điểm vui chơi, hẹn hò, ăn uống, quà tặng,.. cao ngoài những giờ học hành căng thẳng.  
 Vì vậy, nắm bắt được các vấn đề trên, em đã lựa chọn Đề tài: “*Sử dụng WebServices trong xây dựng ứng dụng Android*”. Mục tiêu của đồ án là xây dựng được ứng dụng giúp cho sinh viên tìm kiếm địa điểm thành phố Hà Nội một cách thuận tiện nhất và chọn lựa được địa điểm, đường đi hợp lý, tiết kiệm thời gian và chi phí di chuyển. Đồng thời ứng dụng cũng gợi ý những địa điểm nổi bật, địa điểm mới, địa điểm được đề cử cho người dùng.  
Để đạt được mục tiêu trên, các nhiệm vụ đặt ra là:

* Tìm hiểu và nghiên cứu các lý thuyết cơ sở liên quan như: Tìm hiểu về Hệ điều hành Android.
* Tìm hiểu về công nghệ WebServices
* Khảo sát và lựa chọn các công nghệ sử dụng: Công nghệ GPS, Google Map.
* Xây dựng cơ sở dữ liệu về các khác sạn ở Hà Nội.

Xây dựng các mô đun của ứng dụng: Mô đun tìm kiếm khách sạn, Mô đun dẫn đường.Với các mục tiêu và nhiệm vụ nêu trên, nội dung Đồ án gồm 4 chương:  
**Chương 1**: Cở sở lý thuyết và các công nghệ sử dụng  
Chương này trình bày các cơ sở lý thuyết và các công nghệ được áp dụng trong đồ án.  
**Chương 2**: Phân tích và thiết kế hệ thống  
Nội dung chương này là phân tích hệ thống để xác định các use case và xây dựng biểu đồ tuần tự.  
**Chương 3:** Thiết kế chi tiết và giải pháp kỹ thuật  
Chương này sẽ thiết kế chi tiết cho các lớp trong biểu đổ lớp đã thiết kế trong chương 2.  
**Chương 4**: Kết luận và đánh giá  
Kết luận và đánh giá các kết quả đạt được.

# **CHƯƠNG 1: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG**

Chương này bao gồm các cơ sở lý thuyết cần tìm hiểu và các công nghệ sẽ sử dụng trong quá trình thực hiện đồ án. Chương bao gồm 2 phần:

* Cơ sở lý thuyết: gồm có lý thuyết về lập trình Android, JSON, WebServices.
* Các công nghệ sử dụng: Công nghệ định vị GPS, định vị bằng 3G/Wifi; Google Map API, Facebook API.

1. **Cơ sở lý thuyết**
   1. **Lập trình Android**
      1. **Giới thiệu chung về HĐH Android**

Android là một hệ điều hành di động dựa trên nền tảng linux có mã nguồn mở, miễn phí, mạnh mẽ và được ưa chuộng ‎cao trên thế giới.‎

Hệ điều hành Android một hệ điều hành rất mạnh , có tính bảo mật cao, hỗ trợ ‎nhiều công nghệ tiên tiến như 3G, GPS, EDGE, Wifi.. tương thích với nhiều ‎phần cứng, hỗ trợ nhiều loại bộ nhập dữ liệu như keyboard, touch và trackball. ‎Android là hệ điều hành di động nên có khả năng kết nối cao với các mạng không ‎dây. Android hỗ trợ công nghệ OpenGL nên có khả năng trình diễn các đồ họa cực tốt.‎

* + 1. **Bức tranh tổng thể Android**

Bức tranh về Android được Google vẽ lên như trong hình 2.1.

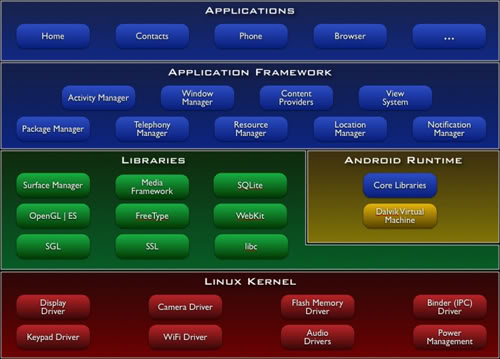
* *Linux Kernel - Trái tim của Android:*

Android được viết trên linux kernel có chỉnh sửa lại và tối ưu cho nền tảng di động. Kernel đảm nhiệm các chức năng cơ bản của hệ điều hành (Quản lý bộ nhớ, lập lịch, quản lý process, file System…). Đặc biệt trong kernel có các driver để cung cấp giao tiếp phần cứng cho từng mẫu di động. Muốn nhúng Android cho một mẫu di động mới, hãng phát triển phải viết lại toàn bộ hoặc các phần driver trong kernel.

* *Library-thư viện mở cho người dùng:*

Android gồm có 9 nhóm thư viện chính:

* Surface Manager: Thư viện quản lý các đối tượng đồ họa.
* Media Framework:Thư viện hỗ trợ giải mã và ghi âm các chuẩn âm thanh, hình ảnh, video phổ biến.
* SQlite:thư viện tạo, lưu trữ và phân tích cơ sở dữ liệu.
* OpenGL | ES:thư viện đồ họa 3D
* FreeType: thư viện giải mã và hiển thị các font chữ
* Webkit: thư viện phân tích và giả mã các trang web
* SGL: thư viện đồ họa 2D
* SSL: thư viện hỗ trợ kết nối tới máy chủ thông qua giao thức SSL
* LibC: thư viện biên dịch mã nguồn các chương trình viết bằng C



Hình 1-Bức tranh tổng thể Android

* *Android runtime:*

Android bao gồm một tập hợp các thư viện cơ bản mà cung cấp hầu hết các chức năng có sẵn trong các thư viện lõi của ngôn ngữ lập trình Java. Tất cả các ứng dụng Android đều chạy trong tiến trình riêng. Máy ảo Dalvik đã được viết để cho một thiết bị có thể chạy nhiều chương trình hiệu quả. Các máy ảo Dalvik thực thi các tập tin thực thi Dalvik (dex), định dạng được tối ưu hóa cho bộ nhớ tối thiểu. Máy ảo dựa trên register-based, và chạy các lớp đã được biên dịch bởi một trình biên dịch Java để chuyển đổi thành các định dạng dex. Các máy ảo Dalvik dựa vào nhân Linux cho các chức năng cơ bản như luồng và quản lý bộ nhớ thấp.

* *Application Framework - Biển lớn giành cho người phát triển phần mềm:*

Android application framework cung cấp sẵn các class để cho người lập trình kế thừa một cách dễ dàng. Kèm theo đó là môi trường phát triển phần mềm tích hợp (Android SDK) khiến cho việc tạo ra các phần mềm ứng dụng cho nền tảng này dễ dàng hơn, thuận lợi hơn.

* + 1. **Các thành phần cơ bản của ứng dụng Android**

Ứng dụng Android được viết bằng Java.Một ứng dụng Android được cấu thành từ 4 thành phần cơ bản:

* Activity
* Service
* Broadcast Reciever
* Content Provider

Các thành phần này không nhất thiết phải có mặt đầy đủ trong ứng dụng. Chúng ta có thể xem các thành phần nào được sử dụng trong ứng dụng bằng việc xem khai báo trong file AndroidManifest.xml.

* 1. **WebServices**
     1. **Giới thiệu về WebServices**

Theo định nghĩa của W3C (World Wide Web Consortium), WebServices là một giao thức tương tác giữa, được thiết kế để hỗ trợ khả năng tương tác giữa các ứng dụng trên các máy tính khác nhau thông qua mạng Internet. Các phần mềm được viết bởi những ngôn ngữ lập trình khác nhau và chạy trên những nền tảng khác nhau có thể sử dụng WebServices để chuyển đổi dữ liệu thông qua mạng Internet theo cách giao tiếp tương tự bên trong một máy tính. WebServices là tài nguyên phần mềm có thể xác định bằng địa chỉ URL, thực hiện các chức năng và trả lại các thông tin người dùng yêu cầu. Một WebServices được tạo nên bằng cách lấy các chức năng và đóng gói chúng sao cho các ứng dụng khác dễ dàng nhìn thấy và có thể truy cập đến những dịch vụ mà nó thực hiện, đồng thời có thể yêu cầu thông tin từ WebServices khác. Nó bao gồm các mô đun độc lập cho hoạt động của khách hàng và doanh nghiệp và bản thân nó được thực thi trên server.



Hình 2-Mô hình tương tác giữa máy khách với WebServices và CSDL

* + 1. **Đặc điểm WebServices**
* WebServices cho phép ứng dụng Client/Service tương tác với nhau ngay cả trong môi trường khác nhau.
* Phần lớn WebServices được xây dựng trên mã nguồn mở và được phát triển từ các chuẩn được công nhận như XML.
* Một WebServices bao gồm nhiều mô đun và được triển khai lên internet.
  + 1. **Kiến trúc WebServices**

WebServices gồm 3 chuẩn chính: SOAP (Simple Object Access Protocol), WSDL (WebServices Description Language) và UDDI (Universal Description, Discovery and Integration).



Hình 3-Các thành phần WebServices và sự tương tác của chúng

Hình trên mô tả thành phần của WebServices, trong đó UDDI được sử dụng để đăng ký và khám phá WebServices đã được miêu tả cụ thể trong WSDL. Giao tác UDDI sử dụng SOAP để nói chuyện với UDDI Server, sau đó các ứng dụng SOAP yêu cầu một WebServices. Các thông điệp được gửi đi chính xác bởi HTTP và TCP/IP.

Chồng giao thức WebServices là tập hợp các giao thức mạng máy tính được sử dụng để định nghĩa, xác định vị trí, thi hành và tạo nên WebServices tương tác với những ứng dụng hay dịch vụ khác. Chồng giao thức này có 4 thành phần chính:

* Dịch vụ vận chuyển (Service Transport)
* Thông điệp XML
* Mô tả dịch vụ
* Khám phá dịch vụ
  + 1. **Các thành phần WebServices**

1. *XML - eXtensible Markup Language*

Là một chuẩn mở do W3C đưa ra cho cách mô tả dữ liệu, nó được sử dụng để định nghĩa các thành phần dữ liệu trang Web và những tài liệu về dịch vụ trực tuyến. Về hình thức XML có cấu trúc hoàn toàn giống như ngôn ngữ HTML. Với XML, các thẻ được lập trình viên tự tạo ra trên mỗi trang Web và được lựa chọn bởi định dạng thông điệp chuẩn và tính phổ biến và hiệu quả của mã nguồn mở.

1. *WSDL – WebServices Description Language*

WSDL định nghĩa cách mô tả WebServices theo cú pháp tổng quát của XML, bao gồm các thông tin:

* Tên dịch vụ
* Giao thức và kiểu mã hóa sẽ được sử dụng khi gọi các hàm WebServices
* Loại thông tin: thao tác, tham số, những kiểu dữ liệu

1. *UDDI – Universal Description , Discovery, and Integration*

Để có thể sử dụng dịch vụ, trước tiên client phải tìm kiếm dịch vụ, ghi nhận thông tin về cách sử dụng và biết được đối tượng nào cung cấp dịch vụ. UDDI định nghĩa một số thành phần cho biết các thông tin này, cho phép các Client truy tìm và nhận những thông tin được yêu cầu khi sử dụng WebServices.

1. *SOAP – Simple Object Access Protocol*

Để truy xuất vào các WebServices ta sử dụng giao thức gọi là Simple Object Access Protocol - SOAP. Nói cách khác chúng có thể truy xuất đến UDDI registry bằng các lệnh gọi hoàn toàn theo định nghĩa của SOAP.

SOAP là giao thức giao tiếp có cấu trúc như XML, nó được xem là cấu trúc xương sống của các ứng dụng phân tán được xây dựng từ nhiều ngôn ngữ và các hệ điều hành khác nhau. SOAP là giao thức thay đổi các thông điệp dựa trên XML qua mạng máy tính, thông thường sử dụng giao thức HTTP.

Thông điệp theo định dạng SOAP là một văn bản XML bình thường bao gồm các phần tử sau:

* Phần tử gốc – envelop: phần tử bao trùm nội dung thông điệp, khai báo văn bản XML như là một thông điệp SOAP
* Phần tử đầu trang – header: chứa các thông tin tiêu đề cho trang, phần tử này không bắt buộc khai báo trong văn bản. Header còn có thể mang những dữ liệu chứng thực, những chứ ký số, thông tin mã hóa hay cài đặt cho các giao dịch khác.
* Phần tử khai báo nội dung chính trong thông điệp – body, chứa các thông tin yêu cầu và thông tin được phản hồi.
* Phần tử đưa ra các thông tin về lỗi - fault, cung cấp thông tin lỗi xảy ra trong quá trình xử lý thông điệp.

Một SOAP đơn giản trong body sẽ lưu các thông tin về tên thông điệp, tham chiếu tới một thể hiện của dịch vụ, một hoặc nhiều tham số. Có 3 kiểu thông báo sẽ được đưa ra khi truyền thông tin: request message (tham số gọi thực thi một thông điệp), respond message (các tham số trả về, được sử dụng khi yêu cầu được đáp ứng) và cuối cùng là fault message (thông báo tình trạng lỗi).

* + 1. **Tương tác với WebServices** .

Hiện nay dùng 2 giao thức phổ biến là REST (Represenemtion Semte Transfer) và SOAP (Simple Object Accesss Protocol) để tương tác giữa ứng dụng với WebServices. Vì phần lớn chúng dùng giao thức HTTP nên hầu như thiết bị nào hỗ trợ web cũng có thể tương tác với WebServices được. Android không hỗ trợ giao thức SOAP, muốn sử dụng ta phải dùng thêm thư viện của bên thứ ba như KSoap. Trong đồ án này em chỉ sử dụng giao thức REST nên sẽ không trình bày về giao thức SOAP.

1. **Sử dụng giao thức REST**

* **Tổng quan về REST**

REST định nghĩa các quy tắc kiến trúc để bạn thiết kế WebServicess chú trọng vào tài nguyên hệ thống, bao gồm các trạng thái tài nguyên được định dạng như thế nào, được chuyển tải qua HTTP thông qua số lượng lớn người dùng và được viết bởi những ngôn ngữ khác nhau. REST được dùng một cách khá lỏng lẻo để mô tả một khung làm việc để chuyển tải dữ liệu qua một giao thức chẳng hạn như HTTP mà không phải bổ sung các lớp ngữ nghĩa hoặc quản lý phiên làm việc.

Một WebServices sử dụng REST sẽ tuân thủ theo bốn nguyên tắc thiết kế cơ bản sau:

* Sử dụng các phương thức HTTP một cách rõ ràng
* Phi trạng thái
* Hiển thị cấu trúc thư mục như URIs (uniform resource identifiers)
* Chuyển đổi dữ liệu thành JSON (JavaScript Object Noemtion) và XML hoặc cả hai.

REST sử dụng các bộ định danh tài nguyên đồng bộ URIs (uniform resource identifiers) để định vị và truy cập một đại diện đã cho của tài nguyên.  Đại diện Nguồn, gọi là Trạng thái đại diện có thể được tạo, lấy ra, sửa đổi và xoá bỏ. Ví dụ, bạn có thể áp dụng REST cho tài liệu xuất bản để làm cho các tài liệu này sẵn có cho các độc giả. Tại bất kỳ thời điểm nào cho trước, nhà xuất bản có thể trình bày các URL Web để các độc giả có thể truy cập thông tin (trạng thái đại diện) về các tài liệu của nhà xuất bản. Các độc giả của tài liệu chỉ cần biết các URL để đọc thông tin tài liệu, và nếu được ủy quyền thì được phép sửa đổi thông tin.

REST định nghĩa các thực thể chủ yếu sau đây:

* Các phần tử dữ liệu: dữ liệu, các bộ định danh (các URL và URI), và các

đại diện của dữ liệu chẳng hạn như tài liệu HTML, tài liệu XML, và hình ảnh.

* Các thành phần: các máy chủ nguồn như Apache httpd và các dịch vụ thông tin như Internet Microsoft® (IIS), các cổng vào chẳng hạn như Squid và CGI, các proxy chẳng hạn như Gauntlet và Netscape và các đại lý người sử dụng chẳng hạn như các trình duyệt Web hoặc các thiết bị lưu động.
* Các đầu nối: các đầu nối khách hàng chẳng hạn như libwww, các đầu nối máy chủ chẳng hạn như NSAPI, các cache, chẳng hạn như một cache trình duyệt, và một số khác.
* REST và HTTP

Theo luận án của Fielding, trong một hệ thống REST dựa trên HTTP, phải sử

dụng các phương thức HTTP chuẩn —có nghĩa là GET, PUT, POST, và DELETE

để truy cập trạng thái đại diện cho tài nguyên:

* GET : Sử dụng phương thức này để chuyển giao trạng thái đại diện hiện tại của một tài nguyên từ nhà xuất bản đến người tiêu dùng.
* PUT : Sử dụng phương thức nàyđể chuyển giao trạng thái đại diện được sửa đổi của một tài nguyên từ người tiêu dùng đến nhà xuất bản.
* POST : Sử dụng phương thức này để chuyển giao trạng thái đại diện mới của một tài nguyên từ người tiêu dùng đến nhà xuất bản.
* DELETE : Sử dụng phương thức này để chuyển giao thông tin cần có để thay đổi trạng thái đại diện của một tài nguyên sang xóa .

Để lấy về tài liệu, một máy khách gửi một truy vấn GET đến máy chủ qua một URL. Sau đó máy chủ phản hồi lại cho máy khách dữ liệu máy khách yêu cầu.

Mỗi phần tử sưu tập trong một tài liệu nội quan thể hiện vùng chứa mà trong đó thông tin về mỗi tài nguyên bộ phận thuộc bộ sưu tập được lưu lại. Mỗi sưu tập trong một tài liệu nội quan có URI duy nhất. Việc chạy một truy vấn HTTP GET cho URI đó sẽ trả về một tài liệu có chứa thông tin về tài nguyên bộ phận có trong sưu tập.

Để tạo các mục tài nguyên bộ phận mới trong sưu tập, máy khách gửi các truy vấn HTTP POST đến sưu tập URI cùng với một trạng thái đại diện của các tài nguyên bộ phận đã được bổ sung vào sưu tập. Các lối vào tài nguyên bộ phận mới có URI riêng duy nhất mà bạn có thể truy vấn để lấy ra trạng thái đại diện của mỗi lỗi vào tài nguyên bộ phận. Để sửa đổi một mục tài nguyên bộ phận, một máy khách lấy ra tài nguyên bằng cách chạy một truy vấn HTTP GET đến URI của tài nguyên bộ phận, thực hiện các sửa đổi mong muốn cho trạng thái đại diện của tài nguyên mà lấy truy vấn GET trả về, sau đó gửi trạng thái đại diện đã được sửa đổi trở lại sưu tập bằng cách sử dụng một truy vấn HTTP PUT. Việc xoá bỏ lối vào tài nguyên bộ phận khỏi sưu tập là dễ làm khi máy khách chạy một truy vấn HTTP DELETE đến URI tài nguyên bộ phận thích hợp. Thông thường Dữ liệu WebServices trả về là JSON hoặc xml.

* **Sử dụng REST trong Android**

Các bước sử dụng REST trong Android dựa trên giao thức HTTP, em làm như sau:

* Định nghĩa 1 URL là đường dẫn đến WebServices
* Tạo 1 đối tượng HTTPClient
* Dùng HTTPGet để tạo request tới WebServices các tham số truyền đi được đặt sau đường dẫn URL theo cú pháp định sẵn (hoặc HTTPPost nếu muốn thay đổi dữ liệu trên server)
* Sử dụng đối tượng HTTPClient ở trên gọi phương thức execute() tham số là đối tượng HTTPGet (hay HTTPPost).
* Dùng HTTPResponse để lấy response từ WebServices
* Dùng HTTPEntity để lấy đối tượng dữ liêu trả về trong response
* Tùy vào dạng dữ liệu trả về là kiều gì thì ta dùng kiểu phân tích tương ứng để phân tích dữ liệu. Ví dụ: JSON thì dùng GSON, XML thì dùng DOM, SAX,…
  1. **JSON**

Trong đồ án này dữ liệu WebServices trả về là JSON. JSON (JavaScript Object Noattion) là một chuẩn để định dạng dữ liệu. JSON rất dễ dàng để đọc, viết và phân tích. Khi sử dụng JSON không cần phải có các bước phân tích phức tạp như đối với XML. Mà có thể truy vấn trực tiếp giá trị theo tên (khóa) được định nghĩa trong JSON, đứng về mặt tốc độ của các ứng dụng di động đây là một ưu điểm rất lớn.

***JSON được xây dựng trên 2 cấu trúc:***

* Là tập hợp của các cặp tên và giá trị name-value.
* Là 1 tập hợp các giá trị đã được sắp xếp.

***Các đặc điểm của JSON:***

* JSON là văn bản trơn (không có định dạng(màu sắc, cỡ chữ,…))
* JSON là "tự mô tả" (người dùng có thể hiểu được)
* JSON là phân cấp (có cấu trúc cây)
* JSON có thể được phân tích cú pháp (parse) bởi JavaScript
* Dữ liệu JSON có thể được truyền đi bằng AJAX
* Không có thẻ kết thúc
* Ngắn hơn
* Đọc và ghi nhanh hơn
* Có thể được phân tích cú pháp bằng hàm dựng sẵn trong JavaScript
* Sử dụng mảng (Array)

1. **Các công nghệ sử dụng**
   1. **Công nghệ định vị GPS**
      1. **Giới thiệu hệ thống GPS**

GPS – Global Positioning System - Hệ thống định vị toàn cầu, nó là hệ thống định vị dựa vào vệ tinh. Đây là một hệ thống rất đồ sộ, đắt tiền và có liên quan đến rất nhiều công nghệ vệ tinh, hiện đại khác. Tuy nhiên cơ chế hoạt động của nó lại khá đơn giản và trực quan. Cơ chế hoạt động được mô tả cụ thể phía dưới.

* + 1. **Các thành phần của GPS**

Hệ thống GPS gồm 3 thành phần

* **Trạm không gian**(Space Segment) : bao gồm các vệ tinh, chúng truyền những tin hiệu cần thiết cho hệ thống hoạt động.
* **Trung tâm điều khiển**(Cotrol Segment): bao gồm các hệ thống được đặt trên mặt đất, thực hiện nhiệm vụ theo dõi vệ tinh, tính toán quỹ đạo cần thiết cho sự quản lý mảng không gian.
* **Mảng người sử dụng**(User Segment): bao gồm toàn thể các thiết bị thu và kỹ thuật tính toán để cung cấp cho người sử dụng thông tin về vị trí.
  + 1. **Các cách định vị vị trí**

1. **Định vi bằng GPS**

Định vi bằng GPS là định vị dựa vào hệ thống định vị toàn cầu. Việc định vị này không cần dùng mạng, thậm chí cũng ko cần lắp sim, chỉ cần có bản đồ nơi mình đang sống và máy có hỗ trợ chức năng GPS là có thể xác định được vị trí. Định vị GPS có thể làm việc vượt ra ngoài phạm vi mạng lưới và không có phí sử dụng dữ liệu. Tuy nhiên nếu dùng cách này thì độ chính xác không cao. Độ chính xác cao chỉ được dùng trong quân sự Mỹ và nếu muốn dùng thì phải trả phí rất đắt. Và một nhược điểm lớn của việc định vị bằng GPS là nó phụ thuộc vào thời tiết, các công trình kiến trúc và các trướng ngại vật tự nhiên. Nếu người dùng đang ở trong nhà hoặc ở trong tầng hầm thì điện thoại sẽ phải rất “ vất vả” để kết nối với vệ tinh, và hầu như là không thể định vị được.

1. **Định vị bằng 3G/Wifi**

Định vị thông qua dịch vụ nhà mạng: để khắc phục nhược điểm của định vị bằng GPS, các nhà mạng cung cấp dịch vụ 3G/GPRS, người dùng có thể định vị nhanh và chính xác, tại mọi điều kiện, miễn là có sóng 3G/GPRS và phải trả phí. Đây là phương pháp phổ biến được dùng để định vị trong các smartphone hiện nay.

* 1. **Google Maps API**

Google Maps là dịch vụ bản đồ mà người dùng có thể xem trong các trình duyệt Web của mình.Tùy thuộc vào vị trí của mình, người dùng có thể xem bản đồ và thông tin về doanh nghiệp địa phương ở dạng cơ bản hoặc tùy chỉnh,bao gồm vị trí,thông tin liên hệ và chỉ đường lái xe của doanh nghiệp. Người dùng có thể nhấp và kéo, thu phóng và quay bản đồ để xem ngay các khu vực lân cận hay xem hình ảnh vệ tinh của vị trí mong muốn.

Hiện tại Google Maps cung cấp các API trong ứng dụng Map riêng của mình:

* Direction API: dùng để tính toán tìm đường đi chi tiết giữa hai địa điểm.
* Place API: dùng để truy vấn và tìm các thông tin địa điểm trên bản đồ.
* Geocoding API: dùng để chuyển đổi giữa các địa chỉ và các mô tả khác sang vĩ độ/kinh độ của toạ độ và ngược lại.
* Elevation API: cung cấp dữ liệu độ cao cho mọi vị trí trên trái đất.
* Distance Matrix API: tính khoảng cách và thời gian đi lại giữa điểm bắt đầu và điểm kết thúc nhưng không có thông tin đường đi chi tiết.
  + 1. **Đăng ký Key từ Google Maps API**

Để bản đồ hoạt động hiển thị được và debug được trên máy thật thì cần có Google API key.

1. **Lấy mã SHA1**

Trước tiên ta cần phải xác định vị trí của file debug.keystore, đường dẫn thường được sử dụng là:

C:\Users\<user>\.Android\debug.keystore

Nếu sử dụng Eclipse thì ta có thể vào Windows->Prefs->Android->Build để lấy đường dẫn. Sau khi lấy được đường dẫn của debug.keystore. Ta bật cmd và chạy lệnh sau để lấy mã SHA1 fingerprint:

keytool -list -alias Androiddebugkey -keystore <path\_to\_debug\_keystore>.keystore -storepass Android -keypass Android

Kết quả của dòng lệnh trên như sau:

75:27:32:DD:15:F1:F3:D4:44:49:25:77:7C:D9:13:C5:14:2C:54:5E

1. **Đăng ký Key Google Maps API**

Chúng ta truy cập vào đường dẫn sau: [http://code.google.com/Android/maps-api-signup.html](http://code.google.com/android/maps-api-signup.html) rồi làm theo các hướng dẫn trên trang web. Sau khi đã đăng ký với dịch vụ Google Maps và đã có được một Maps API Key, ta phải bổ sung thêm cho các đối tượng MapView trong ứng dụng.

* + 1. **Google Direction API**

1. **Giới thiệu**

Google Direction API là một dịch vụ tính toán đường đi giữa các địa điểm sử dụng giao thức HTTP Request. Bạn có thể tìm đường đi cho nhiều loại phương tiện giao thông như: ô tô, xe máy, đi bộ hoặc đi xe đạp. Direction API có thể tìm ra nhiều đường đi có thể qua các điểm trung gian khác nhau. Thông tin đi kèm các tuyến đường bao gồm thời gian di chuyển, độ dài quãng đường, các chỉ dẫn rẽ trái phải và các điểm trung gian sẽ đi qua. Dịch vụ này thường được thiết kế để tính định hướng cho địa chỉ tĩnh, không được thiết kế để đáp ứng trong thời gian thực cho người sử dụng.

Direction API được thiết kế dành cho các trang web và các nhà phát triển ứng dụng trên điện thoại di động. Có thể sử dụng các dữ liệu được cung cấp bởi Direction API để xây dựng các ứng dụng trên bản đồ…

1. **Direction request và Direction response**

* Direction request:

Một câu lệnh direction request có dạng:

<http://maps.googleapis.com/maps/api/directions/output?parameters>

* Đầu ra: Quy định định dạng dữ liệu phản hồi của Server Goolge, nó nhận 1 trong 2 dạng:
* JSON: File phản hồi là 1 file JSON
* XML: File phản hồi là 1 file XML
* Các tham số của câu lệnh lênh request:
* Tham số bắt buộc:
* Orgin: Địa chỉ hoặc kinh độ/ vĩ độ điểm xuất phát
* Destination: Địa chỉ hoặc kinh độ/ vĩ độ đích đến
* Sensor (true/false):
* Tham số khác:
* Mode: (transit/ driving/ walking/ bicycling): mặc định nhận giá trị là driving. Tham số quy định loại phương tiện
* Languge: Quy định ngôn ngữ của dữ liệu phản hồi
* Alternative: (true/false)
* …

Ví dụ một câu lệnh request:

http://maps.googleapis.com/maps/api/directions/xml?origin=Chicago,IL&destination=Los+Angeles,CA&waypoints=Joplin,MO|Oklahoma+City,OK&sensor=false

* Direction response: Có 2 dạng dữ liệu trả về là
* JSON: Dữ liệu dạng file JSON
* XML: Dữ liệu dạng File XML

File dữ liệu trả về chứa các thông tin liên quan đến tuyến đường đi, bao gồm các thông tin như: địa chỉ, tọa độ điểm xuất phát; địa chỉ, tọa độ đích đến; thời gian di chuyển; độ dài quãng đường; một mảng dữ liệu các đoạn đường trong lộ trình; mỗi một đoạn đường lại bao gồm các chỉ dẫn, điểm đầu, điểm cuối, thời gian, độ dài và các điểm trong đoạn đường đã được mã hóa ( phục vụ việc vẽ đường trên bản đồ).

# **CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

## Biểu đồ triển khai hệ thống

Hình 4-Biểu đồ triển khai hệ thống

Hệ thống gốm có 4 thành phần: máy chủ Google có các WebServices cung cấp dữ liệu hiển thị bản đồ cho người dùng tìm đường đi đến khách sạn; CSDL của ứng dụng; máy chủ có WebServices cung cấp các thông tin khách sạn.

## **Biểu đồ ca sử dụng tổng quan**

Hình 5-Biểu đồ ca sử dụng tổng quan

Tác nhân chính là người dùng. Các tác nhân phụ bên ngoài tương tác với hệ thống là hệ thống GPS.Các chức năng chính của ứng dụng là:

* Định vị vị trí người dùng.
* Tìm kiếm khách sạn theo bán kính (so với vị trí của người dùng).
* Xem thông tin về khách sạn.
* Tìm đường tới khách sạn.
* Liên hệ với khách sạn qua số điện thoại.

## Chức năng tìm kiếm khách sạn theo bán kính

### 3.1. Biểu đồ ca sử dụng

Hình 6-Biểu đồ ca sử dụng chức năng tìm kiếm theo bán kính

Bảng 1 dưới đây mô tả chi tiết các bước tìm kiếm khách sạn.

|  |  |  |
| --- | --- | --- |
| **Tên Use case: Tìm kiếm khách sạn theo bán kính** | | |
| 1 | Mô tả | Người dùng khởi động chương trình sau đó chọn một bán kính bất kỳ và sau đó sẽ lên web service lấy các khách sạn thảo mãn. |
| 2 | Actor | Người dùng |
| 3 | Điều kiện tiên quyết | Không có |
| 4 | Kịch bản chính | 1. Người dùng bật ứng dụng 2. Ứng dụng kiểm tra kết nối mạng, nếu có ứng dụng kết nối với WebServices gửi request yêu cầu lấy danh sách khách sạn. WebServices sẽ trả về một list khách sạn. Thực hiện tính toán để đưa ra các khách sạn thỏa mãn và hiện thị lên một 1 listview. |
| 5 | Kịch bản con | C1: Click vào nút Back trên màn hình kết quả để quay trở lại màn hình trước.  C2: Tìm kiếm khách sạn theo tên(nhập tên vào Edittext) |
| 6 | Ngoại lệ | E1: Không có kết nối mạng, một dialog thông báo cần kết nối mạng hiện lên, người dùng đồng ý sẽ chuyển đến trình cài đặt của Android.  E2: Kết nối mạng bị lỗi. Việc gửi request lên server và nhận response trả về bị lỗi. Một dialog thông báo lỗi hiện lên.  E3: Không bật định vị GPS trên điện thoại. Một dialog thông báo cần bật GPS, người dùng đồng ý sẽ chuyển đến trình cài đặt GPS. |

Bảng 1-Đặc tả Use case chức năng tìm kiếm khách sạn theo bán kính.

### 3.2. Biểu đồ tuần tự

Hình 7-Biểu đồ tuần tự chức năng tìm kiếm theo bán kính

Các bước tuần tự như sau:

* Bước 1: Người dùng chạy chương trình và nhập bán kính tìm kiếm
* Bước 2: Chương trình gửi request lấy danh sách khách sạn tới WebServices.
* Bước 3:Service sẽ lấy toàn bộ khách sạn trong database và sau đó gửi lại dữ liệu tìm được cho ứng dụng trên máy Client.
* Bước 4: Chương trình nhận dữ liệu gửi về từ WebServices, phân tích dữ liệu JSON thực hiện tính toán sau đó hiển thị danh sách khách sạn thỏa mãn trên listview.

## 4. Chức năng thông tin chi tiết về khách sạn

### 4.1. Biểu đồ ca sử dụng

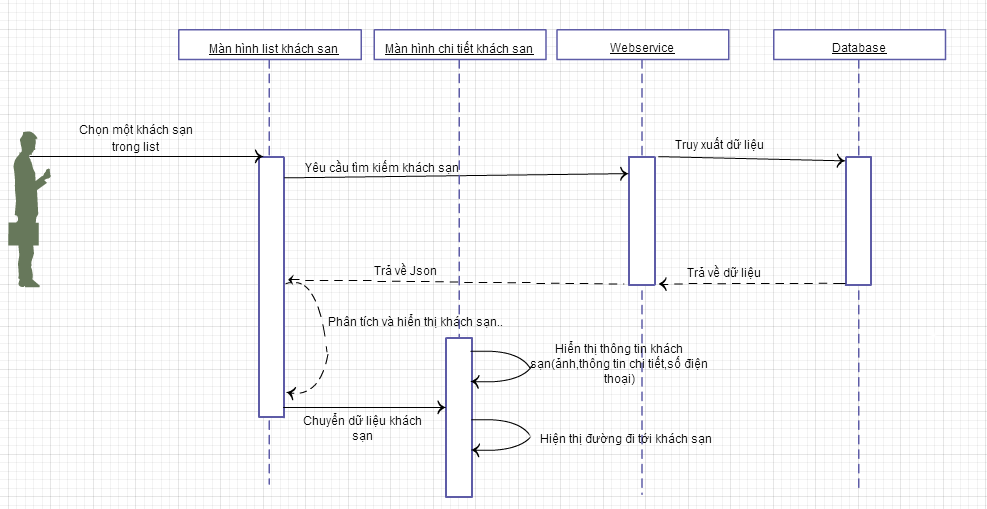
Hình 8-Biểu đồ ca sử dụng chức năng xem thông tin chi tiết về khách sạn

Bảng 2 mô tả chi tiết các bước trong Use case thông tin về khách sạn

|  |  |  |
| --- | --- | --- |
| **Tên Use case: Thông tin chi tiết về khách sạn** | | |
| 1 | Mô tả | Người dùng chọn một khách sạn trong listView |
| 2 | Actor | Người dùng |
| 3 | Điều kiện tiên quyết | Có danh sách khách sạn |
| 4 | Kịch bản chính | 1. Người dùng nhập chọn một khách sạn trong listView(bằng cách ấn vào khách sạn một lát) 2. Chương trình thực hiện lấy dữ liệu khách sạn đã lấy về từ màn hình trước.Màn hình hiện thị chi tiết khách sạn sẽ nhận dữ liệu để hiện thị ảnh,thông tin,số điện thoại và map dẫn đường. |
| 5 | Kịch bản con | C1: Click vào nút Back trên màn hình kết quả để quay trở lại màn hình trước.  C2: Có thể slide hình ảnh  C3: Click vào số điện thoại thì thực hiện gọi tới cho khách sạn. |
| 6 | Ngoại lệ | E1: Không tìm thấy địa điểm nào, xuất hiện dialog thông báo ko tìm thấy địa điểm, nhắc nhập bán kính lớn hơn  E2: Kết nối mạng bị lỗi. Việc gửi request lên server bị lỗi. Một dialog thông báo lỗi hiện lên. |

Bảng 2-Đặc tả chi tiết các bước trong Use case thông tin chi tiết về khách sạn.

### 4.2. Biểu đồ tuần tự



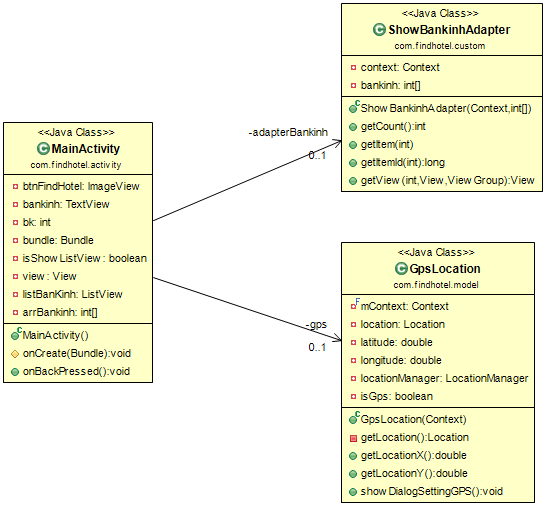
Hình 9-Biểu đồ tuần tự chức năng xem thông tin chi tiết về khách sạn

Các bước tuần tự như sau:

* Bước 1 : Người dùng chọn một khách sạn trong list
* Bước 2 : Dữ liệu của khách sạn được chọn sẽ được chuyển sang màn hình mới(màn hình thông tin chi tiết khách sạn)
* Bước 3 : Ở mà hình thông tin chi tiết khách sạn sẽ thực hiện nhận dữ liệu và xử lý để hiện thị thông tin nhận được(ảnh, thông tin,số điện thoại,map).

# **CHƯƠNG 3: THIẾT KẾ CHI TIẾT VÀ GIẢI PHÁP KỸ THUẬT**

## Màn hình main (MainActivity)

* Cho phép người dùng có thể chọn hoặc nhập bán kính tìm kiếm.
* Kiếm tra xem người dùng đã bật mạng và GPS chưa nếu chưa hiện thị Dialog cho người dùng vào bật mạng hoặc GPS.
* Click vào Imagview (View) để tiến hành tìm kiếm khách sạn.

Hình 10-Biểu đồ UML màn hình main

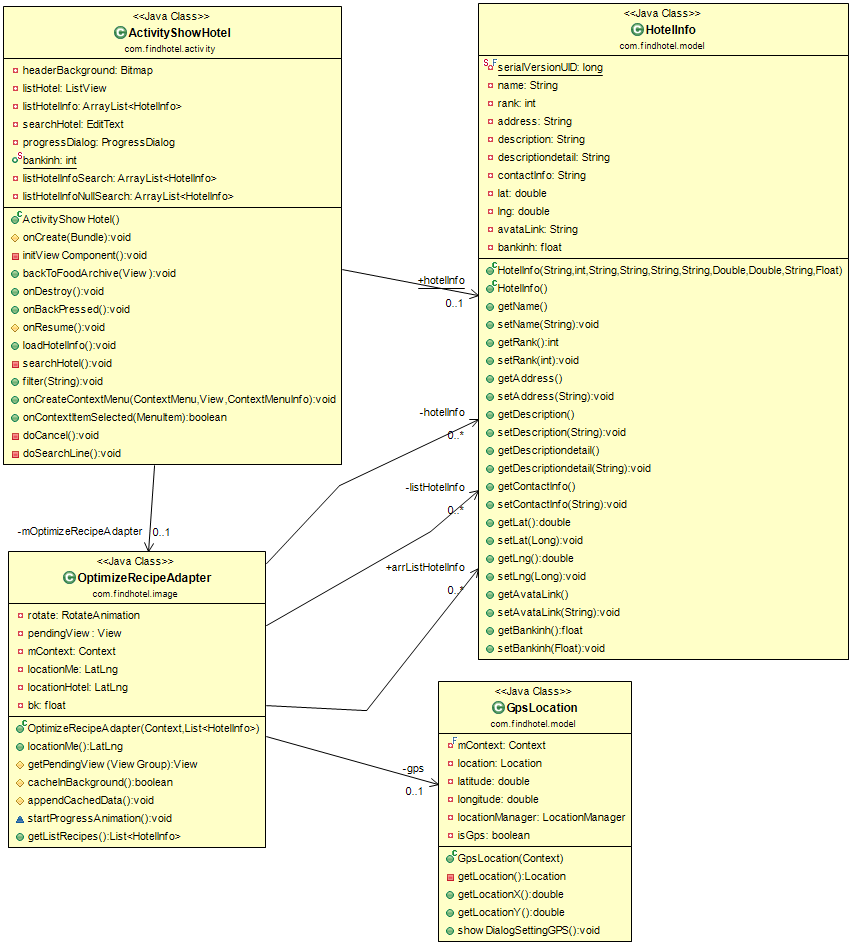
Check GPS and get location

## Màn hình hiện thị khách sạn (ActivityShowHotel)

* Hiển thị tất cả khách sạn nằm trong bán kính đã chọn trước đó lên listView (tên,khoảng cách,thông tin,hạng…)
* Cho phép người dùng tìm kiếm theo tên các khách sạn trong listView.
* Nếu ấn lâu vào một khách sạn trong list sẽ xuất hiện Menu cho phép hiển thị thông tin chi tiết khách sạn hoặc quay trở lại chọn một bán kính khác.

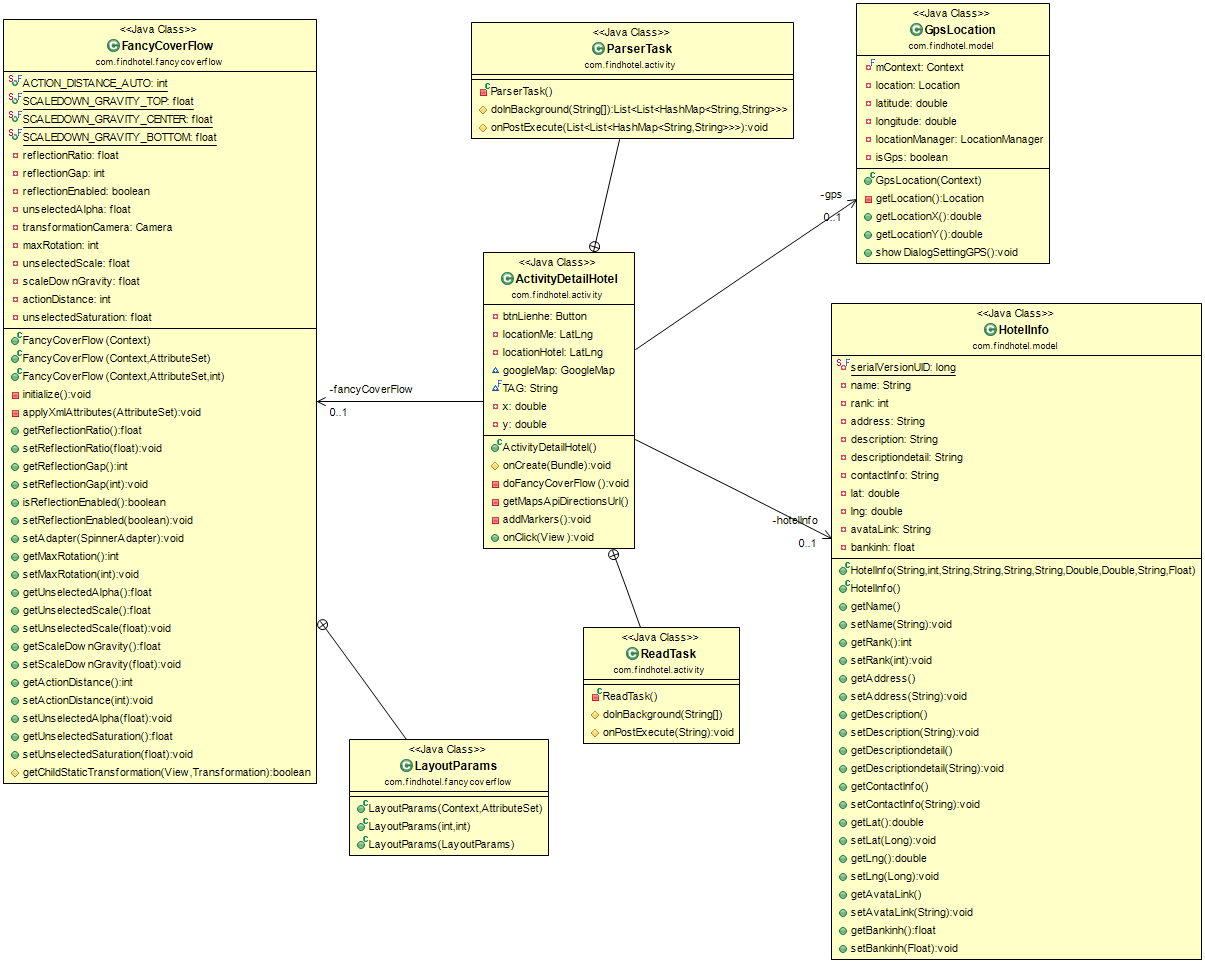
- Class HotelInfo chứa tất cả các thuộc tính của khách sạn.

- Class OptimizeRecipeAdapter có nhiệm vụ là Path Json mà Server trả về.Sau khi lấy được dữ liệu về sẽ thực hiện tính toán khoảng cách từ vị trí của người dùng đến vị trí của khách sạn.Nếu khoảng cách đó nhỏ hơn hoặc bằng bán kính tìm kiếm thì sẽ cho khách sạn đó vào arrayList nếu không thì bỏ qua.

-Class ActivityShowHotel có nhiệm vụ hiển thị các khách sạn thỏa mãn lên listView.

Hình 11-Biểu đồ UML màn hình hiển thị khách sạn

## Màn hình hiển thị thông tin chi tiết khách sạn.

* Cho phép người dùng biết được quy mô khách sạn cũng như những địa địa điểm xung quanh khách sạn.
* Cho phép người dùng biết được đường đi từ vị trí của họ đến khách sạn.
* Người dùng có thể liên lạc với khách sạn bằng cách click vào số điện thoại.
* Người dùng xem các hình ảnh về khách sạn với hiệu ứng chuyển Slide.

Hình 12-Biểu đồ UML màn hình hiển thị thông tin chi tiết khách

# **CHƯƠNG 4: KẾT LUẬN VÀ ĐÁNH GIÁ**

1. Những khó khăn chính

* Việc tạo Services mất nhiều thời gian và vẫn chưa được đầy đủ các chức năng.
* Số lượng khách sạn vẫn còn hạn chế.

1. Cách giải quyết

* Các thành viên luôn luôn liên lạc và trao đổi với nhau về vấn đề gặp phải.
* Các thành viên tích cực tìm hiểu tài liệu từ nhiều nguồn
* Lập kế hoạch tỉ mỉ, chi tiết, chặt chẽ.
* Chương trình được thiết kế theo các modul riêng biệt và tuân theo nguyên lý hướng đối tượng.

# **Các hình ảnh trong báo cáo**

[Hình 1-Bức tranh tổng thể Android 4](#_Toc404235788)

[Hình 2-Mô hình tương tác giữa máy khách với WebServices và CSDL 5](#_Toc404235789)

[Hình 3-Các thành phần WebServices và sự tương tác của chúng 6](#_Toc404235790)

[Hình 4-Biểu đồ triển khai hệ thống 15](file:///C:\Users\Tuyen%20Ng.Gia\Downloads\Báo%20cáo%20oop%20.docx#_Toc404235791)

[Hình 5-Biểu đồ ca sử dụng tổng quan 15](file:///C:\Users\Tuyen%20Ng.Gia\Downloads\Báo%20cáo%20oop%20.docx#_Toc404235792)

[Hình 6-Biểu đồ ca sử dụng chức năng tìm kiếm theo bán kính 16](file:///C:\Users\Tuyen%20Ng.Gia\Downloads\Báo%20cáo%20oop%20.docx#_Toc404235793)

[Hình 7-Biểu đồ tuần tự chức năng tìm kiếm theo bán kính 17](file:///C:\Users\Tuyen%20Ng.Gia\Downloads\Báo%20cáo%20oop%20.docx#_Toc404235794)

[Hình 8-Biểu đồ ca sử dụng chức năng xem thông tin chi tiết về khách sạn 18](file:///C:\Users\Tuyen%20Ng.Gia\Downloads\Báo%20cáo%20oop%20.docx#_Toc404235795)

[Hình 9-Biểu đồ tuần tự chức năng xem thông tin chi tiết về khách sạn 19](#_Toc404235796)

[Hình 10-Biểu đồ UML màn hình main 20](file:///C:\Users\Tuyen%20Ng.Gia\Downloads\Báo%20cáo%20oop%20.docx#_Toc404235797)

[Hình 11-Biểu đồ UML màn hình hiển thị khách sạn 21](file:///C:\Users\Tuyen%20Ng.Gia\Downloads\Báo%20cáo%20oop%20.docx#_Toc404235798)

[Hình 12-Biểu đồ UML màn hình hiển thị thông tin chi tiết khách 22](file:///C:\Users\Tuyen%20Ng.Gia\Downloads\Báo%20cáo%20oop%20.docx#_Toc404235799)

# **Các bảng trong báo cáo**

[Bảng 1-Đặc tả Use case chức năng tìm kiếm khách sạn theo bán kính. 17](#_Toc404234834)

[Bảng 2-Đặc tả chi tiết các bước trong Use case thông tin chi tiết về khách sạn. 19](#_Toc404234835)

# **Tài liệu tham khảo**

* Chris Haseman, Android Essentials, Appres, 2008.
* Johannes Knutsen, WebServices Clients on Mobile Android Devices, Norwegian University of Science and Technology, 2009.
* Reo Meier, Professional Android Development, John Wiley & Sons, 2010.
* Scott Davis, Google Maps API, 2nd, Appress, 2013.
* Johannes Knutsen, WebServices Clients on Mobile Android, O’Reilly, 2009.
* Mark L. Murphy, Beginning Android. Appress, 2009.
* Sayed Y. Hashimi and Satya Komatineni, Pro Android, Appress, 2009.
* Johannes Knutsen, WebServices Clients on Mobile Android Devices, Wrox, 2010.
* <https://developers.google.com/maps/documenemtion/directions/>
* <https://developers.google.com/>
* <https://stackoverflow.com/>