TRƯỜNG ĐẠI HỌC MỎ ĐỊA CHẤT

KHOA CÔNG NGHỆ THÔNG TIN

-----\*\*\*-----



PHÂN TÍCH THIẾT KẾ HỆ THỐNG

ĐỀ TÀI: PHÂN TÍCH THIẾT KẾ HỆ THỐNG PHẦN MỀM MOBILE GIS

|  |  |
| --- | --- |
| ****Giảng viên hướng dẫn****  Sinh viên thực hiện  Nhóm 09 | Trần Trung Chuyên  Lớp 4080103-07  Nguyễn Thanh Tuấn (1321050776)  Vũ Bảo Dương (1321050051)  Tạ Quốc Đạt (1321050483)  Nguyễn Đức Hiển (1321050073) |

**Hà Nội – 2016**

Mục Lục

[Lời nói đầu 1](#_Toc469436642)

[Chương I: Tổng quan về GIS 2](#_Toc469436643)

[1.1. Giới thiệu 2](#_Toc469436644)

[1.2. Các thành phần của GIS 2](#_Toc469436645)

[1.3. Chức năng của GIS 3](#_Toc469436646)

[Chương II: Thiết kế bản đồ trên thiết bị di động 4](#_Toc469436647)

[2.1. Giới thiệu 4](#_Toc469436648)

[2.1.1 Thế nào là Mobile GIS? 4](#_Toc469436649)

[2.1.2 Tại sao sử dụng Mobile GIS? 4](#_Toc469436650)

[2.1.3 Tại sao Mobile GIS chưa được sử dụng nhiều? 5](#_Toc469436651)

[2.1.4 Khuynh hướng thiết bị di động 5](#_Toc469436652)

[2.2. Các vấn đề nghiên cứu khi phát triển một ứng dụng Mobile GIS 5](#_Toc469436653)

[2.2.1 Lựa chọn kiểu kiến trúc phù hợp cho Mobile GIS 6](#_Toc469436654)

[2.2.2 Lựa chọn mô hình dữ liệu không gian cho Mobile GIS 6](#_Toc469436655)

[2.2.3 Cấu trúc dữ liệu cho hiển thị bản đồ trên thiết bị di động 6](#_Toc469436656)

[2.2.4 Cách thức truy vấn thông tin đối tượng di động 6](#_Toc469436657)

[2.3. Kiến trúc xây dựng một ứng dụng Mobile GIS 7](#_Toc469436658)

[2.3.1 Kiến trúc 1: Stand-Alone Client 7](#_Toc469436659)

[2.3.2 Kiến trúc 2: Client – Server 7](#_Toc469436660)

[2.3.3 Kiến trúc 3: Distributed Client-Server 9](#_Toc469436661)

[2.3.4 Kiến trúc 4: Services 10](#_Toc469436662)

[2.3.5 Kiến trúc 5: Peer-to-Peer 10](#_Toc469436663)

[2.4. Công cụ phát triển các ứng dụng Mobile GIS 11](#_Toc469436664)

[2.4.1 Platform: J2ME 11](#_Toc469436665)

[2.4.2 Platform: .Net Compact Framework 12](#_Toc469436666)

[2.4.3 Mobile GIS Software: ESRI ArcPad 13](#_Toc469436667)

[2.4.4 Công nghệ WAP 14](#_Toc469436668)

[CHƯƠNG III. Công nghệ XMLvà webservice 15](#_Toc469436669)

[3.1. Công nghệ XML 15](#_Toc469436670)

[3.1.1 XML là gì 15](#_Toc469436671)

[3.1.2 Cấu trúc của file XML và tài liệu XML 15](#_Toc469436672)

[3.1.3 Đặc điểm của XML 15](#_Toc469436673)

[3.1.4 Ngôn ngữ đặc tả cấu trúc 16](#_Toc469436674)

[3.1.5 Ưu điểm của XML 16](#_Toc469436675)

[3.2 Webservice 16](#_Toc469436676)

[3.2.1 Webservice là gì? 16](#_Toc469436677)

[3.2.2 Một số đặc điểm cơ bản của webservice 17](#_Toc469436678)

[3.2.3 Phân loại webservice 18](#_Toc469436679)

[CHƯƠNG IV: GOOGLE MAP VÀ GOOLE MAPS API WEB SERVICES 19](#_Toc469436680)

[4.1. Google Maps 19](#_Toc469436681)

[4.2. Google Maps API Web Service 19](#_Toc469436682)

[4.2.1 Google Places API 20](#_Toc469436683)

[4.2.2 Directions API 20](#_Toc469436684)

[4.2.3 Geocoding API 20](#_Toc469436685)

[4.2.4 Google Distance Matrix API 21](#_Toc469436686)

[V. Phân Tích thiết kế hệ thống 22](#_Toc469436687)

[5.1. Đặc tả hệ thống 22](#_Toc469436688)

[5.2. Mô tả về hệ thống 22](#_Toc469436689)

[5.2.1 Yêu cầu chức năng 22](#_Toc469436690)

[5.2.2 Yêu cầu phi chức năng 23](#_Toc469436691)

[5.2.3 Yêu cầu hệ thống 23](#_Toc469436692)

[5.3. Phân tích thiết kế 23](#_Toc469436693)

[5.3.1 Mô hình hóa yêu cầu 23](#_Toc469436694)

[5.3.2 Xây dựng Use case 24](#_Toc469436695)

[5.3.3 Biểu đồ use case 31](#_Toc469436696)

[5.3.5 Biểu đồ hoạt động 33](#_Toc469436697)

[5.4. Các vấn đề trong quá trình xây dựng ứng dụng 38](#_Toc469436698)

[5.4.1 Giao diện UI trong lập trình ứng dụng Android 38](#_Toc469436699)

[5.4.2 Đăng ký bản đồ Google Map trong ứng dụng 40](#_Toc469436700)

[5.4.3 Đánh dấu địa điểm và hiện thông tin trên bản đồ 41](#_Toc469436701)

[5.4.4 Sử dụng SQLite trong hệ điều hành Android 41](#_Toc469436702)

[5.4.5 Phân tích cú pháp XML 42](#_Toc469436703)

[5.4.6 Bắt vị trí hiện tại bằng các Location Provider 43](#_Toc469436704)

[5.4.7 Dùng Observer Template để thông báo cho các Activity mỗi khi địa điểm hiện tại thay đổi 43](#_Toc469436705)

[VI. Kết Luận và hướng phát triển 45](#_Toc469436706)

[6.1. Ưu điểm, khuyết điểm của ứng dụng 45](#_Toc469436707)

[6.2. Hướng phát triển 45](#_Toc469436708)

[BẢNG PHÂN CÔNG CÔNG VIỆC 46](#_Toc469436709)

# Lời nÓI đầu

Ngày nay nhu cầu đi chơi, giải trí của mọi người rất cao và để đến được những địa điểm đó thì chúng ta cần phải có bản đồ. Tuy nhiên với bản đồ giấy thì khá là bất tiện khi luôn phải cầm một tờ giấy to và loay hoay xem hướng, xem địa điểm và cũng thể vác theo một cái máy tính để bàn hay laptop theo để tìm kiếm bản đồ.

Với sự phát triển của thị trường mobile như hiện nay thì việc ứng dụng GIS lên mobile để cung cấp bản đồ, địa điểm cho phép người sử dụng tìm kiếm, xem thông tin sẽ giải quyết vấn đề nêu trên.

# Chương I: Tổng quan về GIS

## 1.1. Giới thiệu

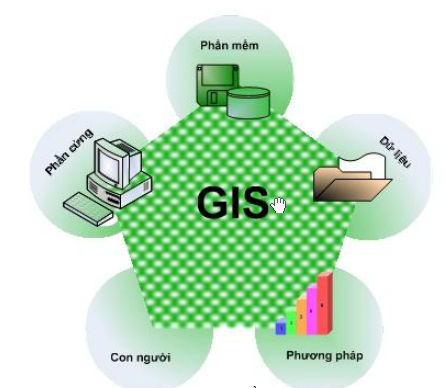
Hệ thông tin địa lý (HTTĐL) – Geographic Information System (GIS) là một nhánh của công nghệ thông tin, đã hình thành từ những năm 60 của thế kỷ trước và phát triển rất mạnh trong những năm gần đây. GIS được sử dụng nhằm xử lý đồng bộ các lớp thông tin không gian (bản đồ) gắn với các thông tin thuộc tính, phục vụ nghiên cứu, quy hoạch và quản lý các hoạt động theo lãnh thổ.

Hiện nay, có rất nhiều định nghĩa về HTTĐL nhưng đều thống nhất quan niệm chung: “Hệ thông tin địa lý *là một hệ thống thông tin có khả năng thu thập, cập nhật, quản trị và phân tích, biểu diễn dữ liệu địa lý phục vụ giải quyết các bài toán ứng dụng có liên quan tới vị trí địa lý trên, trong và ngoài bề mặt trái đất* hoặc được định nghĩa như *là một hệ thống thông tin với khả năng tru nhập, tìm kiếm, xử lý, phân tích và truy xuất dữ liệu địa lý nhằm hỗ trợ cho công tác quản lý, quy hoạch và quản lý tài nguyên thiên nhiên và môi trường”*.

Công nghệ HTTĐL kết hợp các thao tác cơ sở dữ liệu thông thường (như cấu trúc hỏi đáp) và cho phép phân tích thống kê, phân tích địa lý, phân tích các sự kiện, dự đoán tác động và hoạch định chiến lược. Những khả năng này phân biệt HTTĐL với các hệ thống thông tin khác và khiến cho HTTĐL có phạm vi ứng dụng rộng trong nhiều lĩnh vực khác nhau.

## 1.2. Các thành phần của GIS

Một hệ thống GIS hoàn chỉnh gồm có hệ thống máy tính (phần cứng, phần mềm…); Cơ sở dữ liệu; Con người và phương pháp.



Thành phần của GIS

**Phần cứng:** Hệ thống máy tính, có thể là máy chủ trung tâm hay các máy trạm hoạt động độc lập hoặc liên kết mạng

**Phần mềm:** Hiện nay có rất nhiều phần mềm phổ biến đã được thương mại hóa như: ArcGIS, Mapinfo, EVNI, Microstation… Các thành phần chính trong phần mềm:

* Công cụ nhập vào thao tác trên các thông tin địa lý
* Hệ quản trị cơ sở dữ liệu (DBMS)
* Công cụ hỗ trợ hỏi đáp, phân tích và hiển thị địa lý
* Giao diện đồ họa Người – Máy để truy cập công cụ dễ dàng

**Cơ sở dữ lệu:** Được coi là thành phần quan trọng của GIS. Các dữ liệu địa lý và dữ liệu thuộc tính liên quan có thể được người sử dụng tự tập hợp hoạc được mua từ nhà cung cấp dữ liệu thương mại. Các nguồn dữ liệu phải cung cấp được các thông tinmaf hệ thống yêu cầu như: Tọa độ địa lý, quy mô, đặc điểm thuộc tính, các mối quan hệ.

**Con người và phương pháp:** Là thành phần quan trọng của GIS. Những người làm công tác quản lý hệ thống thong tin địa lý cần có khả năng nhận định về tính chính xác, phạm vi suy diễn thông tin, kết nối các mảng thông tin trong hệ thống.

## 1.3. Chức năng của GIS

Một hệ thống phần mềm xử lý HTTĐL yêu cầu tối thiểu phải có ba chức năng sau:

* Tự động hóa bản đồ (Mapping office)
* Quản lý cơ sở dữ liệu (Data base management system - DBMS)
* Xử lý dữ liệu: Đây là chức năng quan trọng để phân biệt một phần mềm HTTĐL với một phần mềm bản đồ khác

Các chức năng của GIS:

* Tổ chức dữ liệu
* Lưu trữ dữ liệu
* Tìm kiếm dữ liệu
* Phân tích dữ liệu
* Hiển thị dữ liệu
* Xuất dữ liệu

# Chương II: Thiết kế bản đồ trên thiết bị di động

## 2.1. Giới thiệu

### 2.1.1 Thế nào là Mobile GIS?

Mobile GIS là: “GIS trên các thiết bị di động” Tuy nhiên, thuật ngữ Mobile GIS cũng có thể bao gồm các thiết bị tại các trạm cố định giao tiếp với các thiết bị di động thông qua một mạng.

Mobile GIS được xem như một phần nhỏ nằm trong lĩnh vực GIS. Tuy nhiên, Mobile GIS có những đặc điểm riêng biệt cần được chú ý đến khi phát triển ứng dụng, đó là sự hạn chế về tài nguyên phần cứng, sự chuyển động của phía client, và cơ chế truyền nhận dữ liệu qua mạng không dây. Để có thể thiết kế bản đồ cho việc hiển thị trên các thiết bị di động, một trong những yêu cầu quan trọng đòi hỏi người thiết kế phải hiểu được cách hiển thị bản đồ trên màn hình có kích thước nhỏ và hiểu được cách tương tác với bản đồ kỹ thuật số trên các thiết bị di động như PDAs (Personal Digital Assistants) và cell phone. Mobile GIS cho phép người sử dụng phóng to, thu nhỏ, thêm và bỏ các tầng dữ liệu được hiển thị, để thay đổi sự xuất hiện của bản đồ trên màn hình hiển thị, và thậm chí có thể sửa đổi được bộ dữ liệu hiển thị. Những khả năng mới trong việc hiển thị bản đồ trên thiết bị di động, từ khả năng tương tác với dữ liệu đến việc cập nhật các bộ dữ liệu phức tạp vào các vùng trên bản đồ, làm cho Mobile GIS trở lên linh hoạt hơn.

### 2.1.2 Tại sao sử dụng Mobile GIS?

Chúng ta đã tìm hiểu thế nào là Mobile GIS, câu hỏi tiếp theo là tại sao lại sử dụng Mobile GIS, có rất nhiều lý do song một vài lý do chủ yếu nhất được đưa ra như sau:

* Có thể truy cập được dữ liệu địa lý trên những vùng rộng lớn
* Có khả năng lấy được dữ liệu trên những vùng rộng lớn và theo thời gian thực
* Có thể bổ sung thông tin vị trí vào cơ sở dữ liệu
* Thực hiện được các chức năng của GIS trên một phạm vi rộng lớn

Từ quan điểm tích hợp, Mobile GIS là một hệ thống ứng dụng tích hợp của hệ thông tin địa lý với truyền thông di động bao gồm các thiết bị GPS, Wireless LBS, truyền thông di động (mạng GSM, GPRS, và CDMA ...), mạng Internet không dây, Mobile GIS không chỉ được ứng dụng trong các lĩnh vực chuyên môn (dự báo thời tiết, quản lý thống kê tài nguyên, ...) mà còn được ứng dụng rộng rãi hơn trong các dịch vụ công cộng như các dịch vụ định vị, dịch vụ giám sát các đối tượng chuyển động...

### 2.1.3 Tại sao Mobile GIS chưa được sử dụng nhiều?

Nếu Mobile GIS thực sự hấp dẫn như vậy, tại sao nó không được triển khai một cách rộng rãi? Cũng có rất nhiều lý do:

* Tài nguyên phần cứng trên các thiết bị di động thường không thích hợp để chạy các ứng dụng GIS chuyên biệt
* Phần mềm: đặc biệt là các phần mềm GIS thường không được hỗ trợ sẵn trên các thiết bị di động
* Thiết lập kết nối giữa các vùng thường rất thưa, tạo ra khó khăn trong kết nối thiết bị di động đến thiết bị di động khác hay mạng
* Thông thường chi phí cho các thiết bị di động tương đối lớn trong khi nó chỉ phát huy được sức mạnh khi đối tượng sử dụng bao gồm cả những người thu nhập thấp

### 2.1.4 Khuynh hướng thiết bị di động

Công nghệ di động gần đây đang được phát huy một cách hiệu quả. Các thiết bị di động ngày nay đang có sức mạnh và bộ nhớ của nó tương ứng với bộ nhớ của máy Desktop. Các nhà sản xuất phần cứng đang tìm cách nén hơn nữa kích thước màn hình vào các thiết bị di động nhỏ hơn và các nhà cung cấp phần mềm đã tạo ra giao diện người sử dụng làm tăng tối đa lợi ích trên các màn hình nhỏ ấy. Các kết nối không dây đã trở lên phổ biến hơn trước đây.

## 2.2. Các vấn đề nghiên cứu khi phát triển một ứng dụng Mobile GIS

Hệ thống ứng dụng Mobile GIS là một hệ thống dịch vụ với đầy đủ các chức năng cung cấp các thông tin về sự chuyển động trong không gian, nó được tích hợp nhiều các công nghệ của mạng không dây và quản lý dữ liệu không gian. Sự phát triển của Mobile GIS có được từ sự phát triển của các công nghệ này. Tuy nhiên, sẽ không có cách nào để xây dựng một hệ thống Mobile GIS nếu chúng ta chỉ đơn giản mang chúng đặt cạnh nhau. Ví dụ, sự tích hợp của GIS, thiết bị di động đầu cuối, mạng di động và công nghệ mạng máy tính sẽ phải đối mặt với các vấn đề như làm thế nào để hiện thực hoá một hệ thống GIS nhỏ trên thiết bị di động có tài nguyên hạn chế trong khi không thể đưa công nghệ phát triển GIS trên máy tính cá nhân vào nó. Hơn nữa, để có thể xử lý được với tài nguyên phần cứng hạn chế trên thiết bị đầu cuối di động, mạng Internet không dây và công nghệ mạng máy chủ phải được sử dụng. Do đó, việc tìm hiểu các vấn đề trên Mobile GIS cần phải được dựa trên các quan điểm hệ thống.

### 2.2.1 Lựa chọn kiểu kiến trúc phù hợp cho Mobile GIS

Do khả năng giới hạn phần cứng của các thiết bị di động như dung lượng lưu trữ dữ liệu, tốc độ xử lý chậm..., hệ điều hành trên các thiết bị di động cũng không thể linh hoạt như trên các máy tính cá nhân, nên để giải quyết được các vấn đề đặt ra, thiết bị di động đầu cuối cần kết nối với máy chủ qua mạng không dây để gửi các yêu cầu về phía máy chủ xử lý. Với những đặc điểm hạn chế của mạng không dây, tuỳ thuộc vào từng ứng dụng cụ thể mà người phát triển cần xây dựng một kiến trúc phù hợp cho hệ thống.

### 2.2.2 Lựa chọn mô hình dữ liệu không gian cho Mobile GIS

Cách đây 30 năm, công nghệ GIS đã được phát triển mạnh mẽ, tuy nhiên đối tượng nghiên của của GIS là các thực thể hình học tĩnh. Làm thế nào để miêu tả một đối tượng chuyển động, và mối quan hệ giữa các đối tượng chuyển động với các thực thể hình học tĩnh là vấn đề cốt lõi trong Mobile GIS. Các vấn đề nghiên cứu bao gồm mô hình khái niệm Mobile GIS, mô hình logic cũng như mô hình vật lý. Mặt khác trong thực tế, mô hình dữ liệu được sử dụng chỉ cho các vấn đề đặc biêt; không có mô hình dữ liệu cho tất cả các vấn đề trong Mobile GIS. Trong Mobile GIS, chúng ta phải xây dựng mô hình dữ liệu cho các vấn đề chuyên biệt. Như chúng ta đã biết, Mobile GIS tích hợp công nghệ của các công nghệ như định vị trên mạng không dây, vị trí của một thiết bị đầu cuối di động sẽ được chỉ ra như một điểm di động, làm thế nào để mô tả sự di chuyển của đối tượng, lưu giữ, vấn tin đối tượng chuyển động là một vấn đề cốt lõi của hệ thống.

### 2.2.3 Cấu trúc dữ liệu cho hiển thị bản đồ trên thiết bị di động

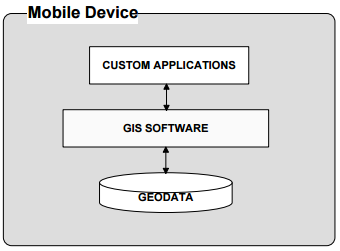
Có rất nhiều sự khác biệt giữa thiết bị di động với môt máy tính cá nhân, chẳng hạn như RAM, CPU, hệ điều hành ...có ảnh hưởng đến việc hiển thị dữ liệu, đặc biệt là dữ liệu không gian. Để hiển thị dữ liệu không gian trên các thiết bị di động, người phát triển ứng dụng cần nghiên cứu xây dựng một cấu trúc dữ liệu phù hợp.

### 2.2.4 Cách thức truy vấn thông tin đối tượng di động

Như đã được đề cập, một trong những vấn đề cốt lõi trong Mobile GIS là mối liên hệ giữa các đối tượng di động và các thực thể không gian, đặc biệt trong các ứng dụng định vị hay giám sát. Truy vấn thông tin đối tượng di động là vấn đề cần được đặt ra cho người phát triển hệ thống.

## 2.3. Kiến trúc xây dựng một ứng dụng Mobile GIS

### 2.3.1 Kiến trúc 1: Stand-Alone Client



Kiến trúc Stand-Alone Client

Một kiến trúc Mobile GIS đơn giản nhất là kiến trúc Stand-Alone Client. Với kiểu kiến trúc này, toàn bộ ứng dụng GIS được cài đặt trên thiết bị di động. Thiết bị di động này lưu giữ dữ liệu không gian, cài đặt phần mềm GIS để biên dịch và hiển thị dữ liệu, và các tuỳ chọn ứng dụng khác, được xây dựng ở mức đỉnh của phần mềm GIS.

Một tiếp cận khác là phát triển ứng dụng tuỳ chọn từ sự hỗn tạp và thực hiện biên dịch, hiển thị thông tin trực tiếp, loại bỏ những thứ không cần thiết cho một ứng dụng GIS. Trong khi một vài ứng dụng đem lại lợi ích từ cách tiếp cận này thì hầu hết các ứng dụng phát triển theo cách tiếp cận này mất thời gian nhiều nhất cho việc phát triển và kiểm thử.

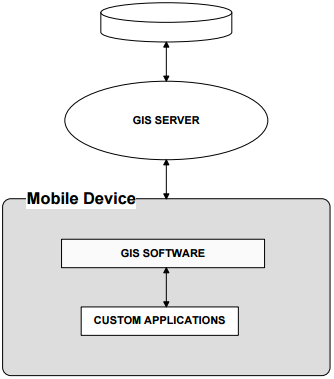
Kiến trúc Mobile GIS này cũng có một vài hạn chế chính: Thứ nhất, tài nguyên phần cứng của thiết bị di động giới hạn dung lượng dữ liệu không gian mà ứng dụng GIS có thể hỗ trợ. Thứ hai, ứng dụng này không cho phép giao tiếp với bất kỳ các ứng dụng khác hay với người cộng tác sử dụng cùng một ứng dụng.

### 2.3.2 Kiến trúc 2: Client – Server

Để khắc phục những hạn chế từ mô hình Stand-Alone Client, mô hình Client –Server được xây dựng, trong đó phần dữ liệu không gian được đặt trên một máy tính riêng biệt và phục vụ client bởi phần mềm GIS phía server. Các ứng dụng tuỳ chọn và phần mềm GIS phía client vẫn được cài đặt trên các thiết bị di động.

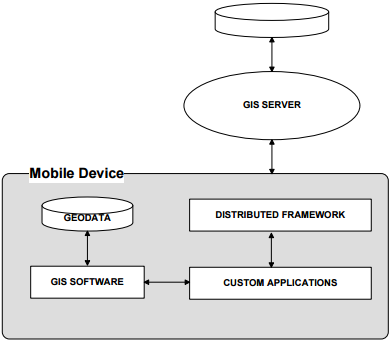
Kiến trúc này có một vài ưu điểm so với kiến trúc Stand-Alone Client: Thứ nhất, dữ liệu không gian bây giờ được lưu không giới hạn về tài nguyên trên một máy chủ riêng biệt. Thứ hai, nhiều thiết bị di động chạy cùng một ứng dụng có thể truy cập đến máy chủ đồng thời tạo ra kiến trúc nhiều người sử dụng (multi-user).

Không may, kiến trúc này cũng đưa ra một biến đổi mới: Truyền thông. Cái gì sẽ xảy ra khi thiết bị di động không thể giao tiếp với GIS Server do giới hạn giao thoa (nhiễu) hoặc các nhân tố ảnh hưởng khác? Thì khi đó ứng dụng không thể truy cập đến cơ sở dữ liệu không gian và nó trở lên vô ích. Bởi vì sự xung đột trong kết nối là một thuộc tính cố hữu, vấn đề chung trong các ứng dụng di động, vấn đề này cần phải được xem xét.



Kiến trúc Client-Server

### 2.3.3 Kiến trúc 3: Distributed Client-Server



Kiến trúc Distributed Client -Server

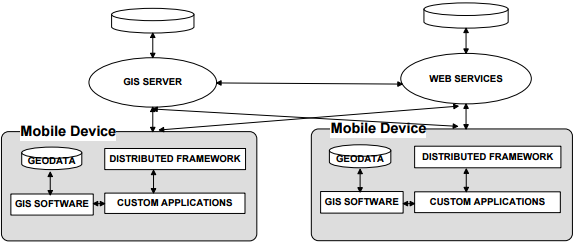
Để giải quyết vấn đề mâu thuẫn trong kết nối, cần triển khai hệ thống phân tán đảm bảo được hai vấn đề sau: liên tục (persistence) và quản lý tài nguyên.

* Liên tục: Khi thiết bị di động không thể kết nối đến máy chủ, nó liên tục thử kết nối lại
* Quản lý tài nguyên: Khi một thiết bị di động không thể kết nối đến máy chủ, nó sử dụng một bộ nhớ nhỏ cache chứa dữ liệu nằm cục bộ trên thiết bị di động

Khi thiết bị di động có thể kết nối trở lại, nó đồng bộ dữ liệu của nó (bộ nhớ cache) với máy chủ. Thành phần có thể thực hiện logic hai vấn đề liên tục và quản lý tài nguyên được gọi là khuôn dạng phân tán (distributed framework).

Kiến trúc này hỗ trợ được hầu hết các ứng dụng GIS di động một cách mạnh mẽ, tin cậy, nhưng nó không hỗ trợ được các chức năng mở rộng cho các chương trình phụ trợ (back-end) – các chương trình ứng dụng phía sau.

### 2.3.4 Kiến trúc 4: Services



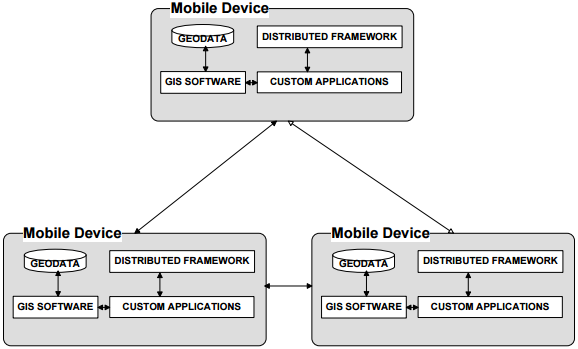
Kiến trúc Services

Để cung cấp khả năng mở rộng cho các chương trình phụ trợ, kiến trúc này xem GIS Server như một dịch vụ Web và cho phép các dịch vụ web khác cũng là một phần của ứng dụng. Do các dịch vụ web sử dụng cùng môt giao thức truyền thông, các thiết bị di động có thể giao tiếp với tất cả các dịch vụ này. Hơn thế nữa, các dịch vụ web cũng có thể giao tiếp với nhau. Một giao thức truyền thông chung phù hợp được đưa ra là SOAP XML, chuẩn công nghiệp cho đóng gói các thông điệp giữa các thành phần phần mềm.

Kiến trúc này hỗ trợ truyền thông một cách mạnh mẽ giữa một số lượng bất kỳ các thiết bị di động và các dịch vụ web nhưng nó có thể không phải là lý tưởng đối với một vài ứng dụng chẳng hạn như chúng được sử dụng thiết kế cho sự tác nghiệp trên các vùng xa nơi mà các kết nối đến máy chủ không thực hiện được.

### 2.3.5 Kiến trúc 5: Peer-to-Peer

Trong hoàn cảnh ấy, kiến trúc Peer-to-Peer sẽ cho phép giao tiếp giữa các thiết bị di động trong khi. Với kiến trúc này, không còn máy chủ sẵn sàng cho việc lưu trữ toàn bộ dữ liệu không gian nữa, dữ liệu phải được lưu giữ trên chính các thiết bị di động. Nhưng mỗi một thiết bị di động không thể lưu trữ 100% dữ liệu (nếu không nó sẽ trở thành mô hình Standard Alone). Để cho phép nhiều dữ liệu được lưu giữ thông qua ứng dụng, mỗi một thiết bị di động lưu giữ một bộ con dữ liệu. Khi thiết bị di động A cần dữ liệu, nó dựa vào khuôn dạng quản lý tài nguyên phân tán của chính nó. Nếu nó không có, nó sẽ thực hiện truy cập đến dữ liệu trên thiết bị di động B. Như ví dụ đã chỉ ra, kiến trúc này đòi hỏi phải sử dụng một khái niệm: naming - nhận dạng duy nhất cho mỗi thiết bị và mối quan hệ logic giữa chúng. Nhưng trong trường hợp B ở bên ngoài vùng kết nối của A? A không thể truy cập được cơ sở dữ liệu mà nó cần. Để giải quyết vấn đề này, mô hình kiến trúc peer-to-peer đưa ra khái niệm: redundancy – lưu trữ dữ liệu trên một hoặc nhiều thiết bị khác nhau. Vì vậy trong trường hợp này, bản khuôn dạng phân tán (distributed framework) trên thiết bị di động A cho biết rằng dữ liệu mà nó cần cũng có thể tìm được ở thiết bị di động C.



Kiến trúc Peer-to-Peer

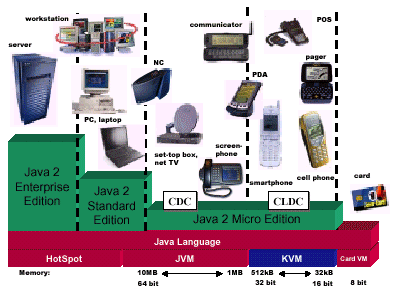
Như vậy chúng ta có nhiều kiểu kiến trúc khác nhau để lựa chọn phát triển một ứng dụng Mobile GIS. Người phát triển cần lựa chọn một kiến trúc tốt nhất, phù hợp nhất cho ứng dụng của mình.

## 2.4. Công cụ phát triển các ứng dụng Mobile GIS

### 2.4.1 Platform: J2ME

J2ME được phát triển từ kiến trúc Java Card, Embeded Java và Personal Java của phiên bản Java 1.1. Đến sự ra đời của Java 2 thì Sun quyết định thay thế Personal Java và đươc gọi với tên mới là Java 2 Micro Edition, hay viết tắt là J2ME. Đúng với tên gọi, J2ME là nền tảng cho các thiết bị có tính chất nhỏ, gọn (Micro có nghĩa là nhỏ trong tiếng Anh). Thị trường J2ME được mở rộng cho nhiều chủng loại thiết bị:

* Các loại thẻ cá nhân như Java Card
* Máy điện thoại di động
* Máy PDA
* Các hộp điều khiển dành cho tivi, thiết bị gia dụng

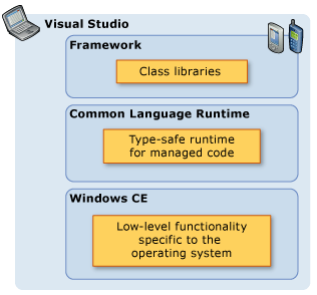


Để phát triển một ứng dụng J2ME, người phát triển phải lựa chọn một cấu hình; ứng dụng sau đó có thể phát triển sử dụng cấu hình đó. Có 4 tuỳ chọn cấu hình:

* Connected Limited Device Configuration (CLDC): Đây là cấu hình sử dụng KVM, một phiên bản cực kỳ nhẹ của Java Virtual Machine (JVM)
  + Mobile Information Device Profile (MIDP): MIDP chỉ là một profile sẵn có cho CLDC và được sử dụng cho các thiết bị di động có tài nguyên rất hạn chế, chẳng hạn như điện thoại di động
* Connected Device Configuration (CDC): cấu hình này sử dụng một JVM đầy đủ, nhưng profile của nó quyết định có bao nhiêu chức năng được hỗ trợ
  + Foundation Profile (FP): như tên của nó đã chỉ ra, profile này chỉ cung cấp các lớp cơ sở
  + Personal Foundation Profile (PFP): PFP cung cấp các lớp cơ sở và bổ sung thêm các chức năng về đồ hoạ
  + Personal Profile (PP): profile này là bộ chứa nhiều tính năng cấp cao so với PFP và cung cấp toàn bộ Advanced Windows Toolket (AWT) cho người phát triển

### 2.4.2 Platform: .Net Compact Framework

.Net Compack Framework là một sản phẩm của Microsoft được dùng cho phát triển các ứng dụng trên thiết bị di động. Những gì mà người phát triển có thể làm giống như việc phát triển một ứng dụng trên máy để bàn. Mỗi lần người phát triển cấu hình thuộc tính được xây dựng trên IDE của anh ta để triển khai đến một thiết bị di động thay cho một máy để bàn, IDE của anh ta quan tâm đến sự thay đổi. Do đó, việc phát triển ứng dụng di động với CF không đòi hỏi phải có nhiều kinh nghiệm của chuyên gia trong lĩnh vực phát triển ứng dụng di động.

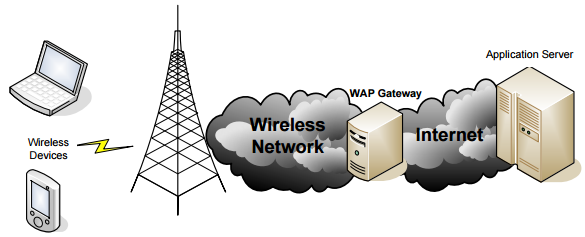


### 2.4.3 Mobile GIS Software: ESRI ArcPad

Có thể không chính xác khi tuyên bố rằng ESRI ArcPad chỉ là phần mềm GIS dành cho thiết bị di động, nhưng khá chính xác khi ESRI tuyên cho rằng ARCPad sẽ còn tiến xa và đi đầu trong lĩnh vực công nghiệp phần mềm. Hiện tại phiên bản 6.0.2 cho phép kết nối trực tiếp đến dữ liệu không gian cục bộ hoặc kết nối từ xa đến máy chủ bản đồ của ESRI. Thêm vào đó, giao diện tương tác của nó cung cấp nhiều công cụ GIS chuẩn như phóng to, thu nhỏ …

Người phát triển ứng dụng di động có thể tuỳ chọn hoá ArcPad sử dụng ArcPad Studio. Studio cho phép tạo các thanh công cụ tuỳ chọn và các form tuỳ chọn được viết bằng VBScript. Chúng có thể được biên dịch thành một “applet” (đoạn nhúng), chính là những ứng dụng rất nhỏ của ArcPad được lưu trên thiết bị di động như ArcPadXML. Việc mở rộng ArcPad có thể được thực hiện sử dụng ngôn ngữ C/C++ với các thiết bị cài đật Windows CE.

### 2.4.4 Công nghệ WAP



Các giao thức WAP được thiết kế trên nền của các giao thức web. Mục đích của WAP là sử dụng lại cấu trúc cơ sở của web, để từ đó nâng cao quá trình giao tiếp giữa các nhà cung cấp và các thiết bị di động, giúp quá trình này trở lên hiệu quả và tốn ít thời gian hơn là sử dụng chính các giao thức web.

Do kiến trúc của WAP được thiết kế gần giống với web, nên nó cũng kế thừa mô hình client-server được dùng trên Internet của Web. Điểm khác nhau chính đó là sự có mặt của WAP Gateway dùng cho việc chuyển đổi giữa HTTP và WAP.

Để truy cập vào một ứng dụng trên server, client khởi tạo một kết nối với WAP Gateway và gửi đi yêu cầu của mình. Gateway sẽ chuyển đổi những yêu cầu này sang định dạng được dùng trên Internet (HTTP), và sau đó chuyển đến server cung cấp dịch vụ. Nội dung trả về được gửi từ server đến gateway, tại đây nó sẽ được chuyển sang định dạng WAP để sau đó gửi về thiết bị di động. Như vậy, gateway đã giứp Internet có thể giao tiếp với môi trường mạng không dây.

Như đoạn trên đã minh hoạ, có nhiều tuỳ chọn sẵn có cho việc phát triển một ứng dụng GIS di động. Việc lựa chọn phụ thuộc vào yêu cầu của ứng dụng và kỹ năng của người phát triển.

# CHƯƠNG III. Công nghệ XMLvà webservice

## 3.1. Công nghệ XML

### 3.1.1 XML là gì

XML (viết tắt từ tiếng Anh eXtensible Markup Language, “Ngôn ngữ đánh dấu mở rộng”) là ngôn ngữ đánh dấu do W3C đề nghị. Mục đích chính của XML là đơn giản hóa việc chia sẻ dữ liệu giữa các hệ thống khác nhau, đặc biệt là các hệ thống được kết nối Internet.

XML là một ngôn ngữ đánh dấu tương đối vì nó là một subset (một phần nhỏ hơn) của (derived from) một ngôn ngữ đánh dấu ra đời trước tên là Standard Generalized Markup Language (SGML).

### 3.1.2 Cấu trúc của file XML và tài liệu XML

**Cấu trúc của file XML:**

*Document Prolog:* Lưu trữ metadata của XML gồm 2 phần đó là khai báo XML và khai báo kiểu dữ liệu trong XML. Phần khai báo XML (XML declaration) bao gồm các thông tin về version của XML, charset, encoding… Phần khai báo kiểu dữ liệu trong XML (DTD) dùng để khai báo cấu trúc của các thẻ dùng trong XML.

*Root element* hay còn gọi là Document Element: chứa tất cả các phần tử và nội dung của nó. 1 phần tử của XML phải có thẻ mờ và thẻ đóng.

**Cấu trúc tài liệu XML:**

Gồm có hai phần đó là cấu trúc logic và cấu trúc vật lý

* Cấu trúc logic: Định nghĩa các phần tử, các thuộc tính, kiểu dữ liệu
* Cấu trúc vật lý: Chứa dữ liệu và các phần tử chứa dữ liệu như text, hình ảnh, media…

### 3.1.3 Đặc điểm của XML

* XML cung cấp một phương tiện dùng văn bản (text) để mô tả thông tin và áp dụng một cấu trúc kiểu cây cho thông tin đó.
* Đơn vị cơ sở của XML là các ký tự theo định nghĩa của Universal Character Set (Bộ ký tự toàn cầu)
* Các tệp XML có thể dùng cho nhiều loại dữ liệu đa phương tiện. RFC3023 định nghĩa các loại “application/xml” và “text/xml”, với ý rằng dữ liệu được biểu diễn bằng XML mà không nói gì đến ngữ nghĩa của dữ liệu
* XML được dùng kết hợp với HTML, rất hữu ích cho việc trao đổi dữ liệu và tạo ra tùy biến cho các tags

### 3.1.4 Ngôn ngữ đặc tả cấu trúc

*DTD (Document Type Definition)*

Một DTD xác định ngữ pháp của một tài liệu XML bằng một tập những qui tắc của phần tử (elements) và thuộc tính (attributes)

*XML Schema*

XML Schema mô tả cấu trúc tài liệu XML, thay thế cho DTD.

XML Schema mô tả:

* Các phần tử và thuộc tính trong tài liệu XML
* Thứ tự và số lượng các phần tử con
* Các kiểu dữ liệu của phần tử và thuộc tính

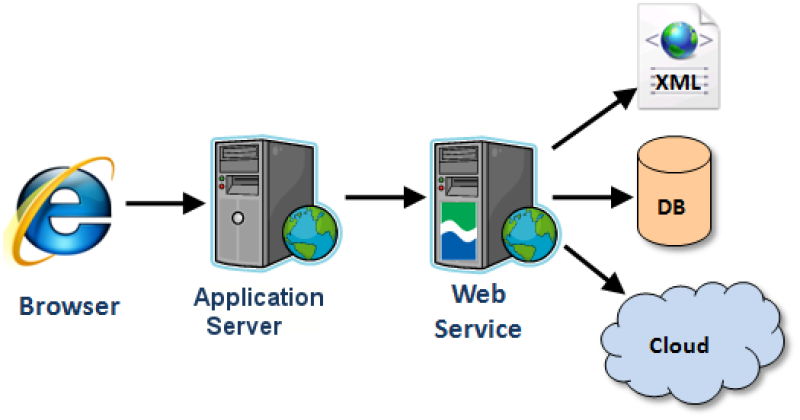
### 3.1.5 Ưu điểm của XML

* Dễ dàng xử lý, chuyển tải và trao đổi dữ liệu
* Mô tả dữ liệu và cách thể hiện dữ liệu tông qua các thẻ
* Tổ chức dữ liệu và cấu trúc phân cấp
* Dữ liệu độc lập là ưu điểm chính của XML. Do XML chỉ dùng để mô tả dữ liệu bằng dạng text nên tất cả các chương trình đều có thể đọc được XML
* Dễ dàng tạo 1 file XML
* Lưu trữ cấu hình cho website
* Sử dụng cho phương thức Remote Procedure Calls (RPC) phục vụ web service

## 3.2 Webservice

### 3.2.1 Webservice là gì?

Webservice (dịch vụ web) là sự kết hợp các máy tính cá nhân với các thiết bị khác, các cơ sở dữ liệu và các mạng máy tính để tạo thành một cơ cấu tính toán ảo mà người sử dụng có thể làm việc thông qua các trình duyệt hoặc ứng dụng có kết nối mạng



Một web service có thể được mô tả như là một chứ năng có thể duy trì trên web, và có thể được gọi bởi bất kỳ ứng dụng hay dịch vụ nào. Nó có thể là một business application hay một system function.

Bản thân các dịch vụ này sẽ chạy trên các máy chủ trên nền Internet chứ không phải là các máy tính cá nhân, do vậy có thể chuyển các chức năng từ máy tính cá nhân lên Internet. Người sử dụng có thể làm việc với các dịch vụ thông qua bất kỳ loại máy hay phần mềm nào có hỗ trợ web service và có truy cập internet, kể cả các thiết bị cầm tay. Do đó các web service sẽ làm internet biến đổi thành một nơi làm việc chứ không phải là một phương tiện để xem và tải nội dung.

Điều này cũng sẽ đưa các dữ liệu và các ứng dụng từ máy tính cá nhân tới các máy phục vụ của một nhà cung cấp dịch vụ web. Các máy phục vụ này cũng cần trở thành nguồn cung cấp cho người sử dụng cả về độ an toàn, độ riêng tư và khả năng truy nhập.

Các máy phục vụ ứng dụng sẽ là một phần quan trọng cảu các web service bởi vì thường thì các máy chủ này thực hiện các hoạt động ứng dụng phức tạp dựa trên sự chuyển giao giữa người sử dụng và các chương trình kinh doanh hay các cơ sở dữ liệu của một tổ chức nào đó.

Web service chủ yếu dựa trên một lời gọi thủ tục từ xa không chặt chẽ mà có thể thay thế các lời gọi thủ tục từ xa chặt chẽ, đòi hỏi các kết nối API phù hợp đang phổ biến hiện nay. Dịch vụ web sử dụng XML chứ không phải C hay C++, để gọi các quy trình, Một số chuyên gia lại cho rằng web service là một dạng API dựa trên phần mềm trung gian, có sử dụng XML để tạo giao diện trên nền Java 2 (J2EE) hay các server ứng dụng .NET. Giống như các phần mềm trung gian, web service sẽ kết nối server ứng dụng cới các chương trình khách hàng.

### 3.2.2 Một số đặc điểm cơ bản của webservice

Một webservice có thể được truy cập thông qua web.

Một webservice có giao diện dịch vụ. Giao diện này giúp cho webservice có thể được gọi bởi bất kỳ một ứng dụng nào hay bởi webservice nào khác. Giao diện dịch vụ là một tài liệu XML. Bởi vì XML có thể hiện rõ vai trò trong công nghệ trao đổi thông tin toàn cục (global exchange technology) được chấp nhận bởi phần lớn công nghệ hiện nay.

Các dịch cụ web dùng giao thức tiêu chuẩn web để giao tiếp, không như COM, RMI hay CORBA. XML được dùng để trao đổi thông tin giữa các chương trình ứng dụng và dịch vụ.

### 3.2.3 Phân loại webservice

Webservice có hai loại:

Dịch vụ ứng dụng và dịch vụ hệ thống (Application and System Service). Một Application Service thể hiện một hành động của user như duyệt mail, hay kiểm tra tỷ số hối đoái… Một System Service thể hiện yêu cầu của kiến trúc hệ thống và sự quản lý như bảo mật, lưu trữ và chịu lỗi, quản lý transaction hay messagin.

Một dịch vụ có hai đặc điểm chính: Interface và Registration. Sử dụng Interface, một dịch vụ có thể được gọi từ một chương trình khác. Một service được đăng ký trong một registry.

Một vấn đề tổng quát với service hệ thống là các client cần thiết chỉ rõ giao thức để yêu cầu các dịch vụ từ cá hệ thống. Web service là một hệ thống phát triển từ hệ thống hướng dịch vụ, dùng giao tiếp tổng quát để chuyển tải các thông điệp giữa các hệ thống.

# CHƯƠNG IV: GOOGLE MAP VÀ GOOLE MAPS API WEB SERVICES

Để làm việc tốt với hệ thống bản đồ của google, thì cần sử dụng một số các dịch vụ cần thiết để lập trình. Điểm chung của các dịch vụ này là đều chạy trên môi trường web và đều trả kết quả qua định dạng JSON hoặc XML.

## 4.1. Google Maps

Google Maps là một dịch vụ ứng dụng và công nghệ bản đồ trực tuyến trên web miễn phí được cung cấp bởi Google và hỗ trợ nhiều dịch vụ dựa vào bản đồ như Google Ride Finder và một sốt có thể dùng để nhúng vào các trang web của bên thứ ba thông qua Google Maps API. Nó cho phép thấy bản đồ đường sá, đường đi cho xe đạp, cho người đi bộ (những đường đi ngắn), xe hơi, và những địa điểm khinh doanh trong khu vực cũng như khắp nơi trên thế giới.

## 4.2. Google Maps API Web Service

Google Maps API Web Service là một tập các giao diện HTTP cung cấp thông tin địa lý cho ứng dụng.

Google Maps API cung cấp dịch vụ như là các giao diện phục vụ cho việc yêu cầu dữ liệu địa lý và sử dụng dữ liệu đó trong ứng dụng của nhà phát triển. Các dịch vụ này được thiết kế để sử dụng với một ứng dụng bản đồ.

Các dịch vụ này sử dụng HTTP request, thiết lập các chuỗi URL request để gửi yêu cầu cho web service. Thông thường, web service sẽ trả kết quả là tập tin JSON hoặc XML. Phân tích cú pháp kết quả trả về để sử dụng.

URL request có dạng:

*http://maps.googleapis.com/maps/api/service/output?parameters*

* service: loại dịch vụ.
* output: kết quả ở dạng tập tin JSON hay XML.
* parameters: các tham số phù hợp.

**SSL Access**

Có thể truy cập Google Maps API Web Services thông qua HTTPS. Truy cập Maps API Web Service thông qua HTTPS được khuyến khích sử dụng nếu ứng dụng có chứa những dữ liệu nhạy cảm như tọa độ người dùng.

[*https://maps.googleapis.com/maps/api/service/output?parameters*](https://maps.googleapis.com/maps/api/service/output?parameters)

**Theo dõi sử dụng với tham số sensor**

Sử dụng Google Maps API yêu cầu nhà phát triển phải chỉ ra rằng ứng dụng có sử dụng một cảm biến (ví dụ cảm biến GPS) để xác định vị trí người dùng trong bất cứ yêu cầu nào cho service. Nó khá quan trọng cho điện thoại. Nếu ứng dụng Google Map API sử dụng bất cứ dạng cảm biến để xác định vị trí của thiết bị phải thiết lập giá trị của tham số sensor=true.

Trong trường hợp ứng dụng không sử dụng cảm biến thì vẫn phải gán giá trị sensor=false.

**Processing Responses**

Kết quả trả về của Google Direction API khá dễ hiểu nhưng lại ít thân thiện với người sử dụng. Khi giởi yêu cầu và nhận được kết quả, cần phải phân tích cú pháp kết quả nhận được và chỉ trích xuất một số thông tin có ích.

Phân tích cú pháp kết quả nhận được phụ thuộc vào loại tập tin trả về là XML hay JSON và có thể sử dụng nhiều kỹ thuật và ngôn ngữ lập trình khác nhau.

Google Maps API Web Service bao gồm:

* Directions API
* Distance Matrix API
* Elevation API
* Geocoding API
* Places API

### 4.2.1 Google Places API

Google Places API là một dịch vụ của Google INC mà khi sử dụng chúng ta sẽ nhận được các thông tin về các địa điểm đã được đánh dấu trên bản đồ Google Maps bởi các tổ chức, công ty và cá nhân ở khắp nơi trên thế giới. Để sử dụng được dịch vụ này chúng ta phải đăng ký API key dịch vụ Google Places với google (sử dụng miễn phí có giới hạn số lần requesst/day).

### 4.2.2 Directions API

Google Direction API là dịch vụ của Google INC tính toán đường đi giữa hai điểm sử dụng HTTP request. Thông tin đầu vào là điểm đầu và điểm cuối, sau đó sẽ ta sẽ nhận được thông tin về khaorng cách giữa 2 điểm, tọa độ GPS, thời gian đi lại dự kiến với các phương như xe máy, oto hay là đi bộ.

### 4.2.3 Geocoding API

Mã hóa địa lý là tiến trình chuyển địa chỉ sang tọa độ địa lý để đánh dấu trên bảng đồ. Google Geocoding API cung cấp một phương thức trực tiếp để truy cập bộ mã hóa địa lý thông qua giao thức HTTP. Ngoài ra dịch vụ Google Geocoding API còn cho phép giải mã từ tọa độ sang địa chỉ.

### 4.2.4 Google Distance Matrix API

Google Distance Matrix API là một dịch vụ cung cấp khoảng cách và thời gian di chuyển cho một ma trận các điểm gốc và điểm đích. Thông tin trả về phụ thuộc vào đường đi được đề nghị giữa điểm gốc và điểm đích theo tính toán của Google Distance Matrix API, bao gồm các giá trị khoảng cách và thời gian cho mỗi cặp.

Dịch vụ Google Distance Matrix API không cung cấp chi tiết đường đi. Muốn tìm thông tin chi tiết đường đi có thể sử dụng dịch dụ Google Direction API với chỉ một cặp điểm gốc và điểm đích.

# V. Phân Tích thiết kế hệ thống

## 5.1. Đặc tả hệ thống

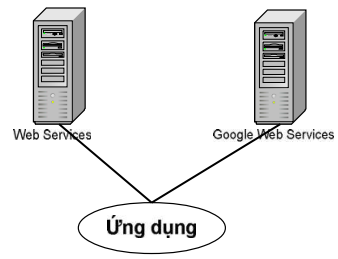
Xây dựng chương trình quản lý địa điểm như ATM, ngân hàng, điểm du lịch, trạm xe bus, khách sạn… theo danh sách và hiển thị trên bản đồ.

Các điểm sẽ được chia làm nhiều thể loại. Loại địa điểm đặc trưng cho djch vụ hay hoạt động mà địa điểm đó có, chia làm các loại: ẩm thực, khách sạn, điểm tham quan, ATM, bệnh viện…

Mỗi địa điểm sẽ cung cấp thông tin: Địa chỉ (gồm số nhà và đường), mô tả tóm tắt địa điểm, số điện thoại, tọa độ địa lý dùng để hiển thị và chỉ đường trên bản đồ.

## 5.2. Mô tả về hệ thống

Hệ thống được xây dựng theo mô hình trao đổi dữ liệu client-server. Trong đó ứng dụng phía thiết bị Android là máy client tạo các truy vấn lên 2 máy chủ là máy chủ web của ứng dụng và máy chủ các dịch vụ cần thiết của google để lấy dữ liệu phù hợp.



### 5.2.1 Yêu cầu chức năng

***Các chức năng của hệ thống bao gồm:***

* Mở hệ thống
* Đóng hệ thống
* Tìm kiếm
* Hiển thị kết quả tìm kiếm, thông tin đối tượng của lớp bản đồ hiện tại
* Lịch sử tìm kiếm
* Quản lý lớp
* Zoom in
* Zoom out
* Chứ năng Pan
* Gửi góp ý
* Xem góp ý
* Cập nhật danh mục tỉnh thành
* Cập nhật danh mục quận huyện
* Cập nhật danh sách địa điểm

### 5.2.2 Yêu cầu phi chức năng

* Chương trình có thể đảm bảo làm việc 24/7
* Chương trình có thể vận hành tốt khi mà cơ sở dữ liệu về địa điểm có thể tăng trong tương lai
* Chương trình có thể thực hiện tìm kiếm và trả về kết quả với sai số nhỏ với cơ sở dữ liệu lớn
* Chương trình phải dễ nhìn và đơn giản để hướng tới mọi đối tượng sử dụng

### 5.2.3 Yêu cầu hệ thống

* Sử dụng hệ điều hành Android phiên bản 4.4 trở lên
* Có kết nối Internet và hỗ trợ GPS
* Máy chủ web dịch vụ của ứng dụng chạy được PHP và cài đặt CSDL MySQL

## 5.3. Phân tích thiết kế

### 5.3.1 Mô hình hóa yêu cầu

***Xác định tác nhân:***

* **Người sử dụng:** Là người sử dụng phần mềm để tìm kiếm địa chỉ đường, các địa điểm như: cây xăng, ATM, khác sạn…
* **Quản trị viên:** Là người sử dụng phần mềm

***Các trường hợp sử dụng:***

* Khởi động hệ thống
* Đóng hệ thống
* Tìm kiếm
* Hiển thị kết quả tìm kiếm, thông tin đối tượng của lớp bản đồ hiện tại
* Lịch sử tìm kiếm
* Quản lý lớp
* Zoom in
* Zoom out
* Chức năng Pan
* Gửi góp ý
* Xem góp ý
* Cập nhật danh mục tỉnh thành
* Cập nhật danh mục quận huyện
* Cập nhật danh sách địa điểm

### 5.3.2 Xây dựng Use case

#### 5.3.2.1 Use case 1: Khởi động hệ thống

|  |  |  |
| --- | --- | --- |
| **Tên Use Case:** Khởi động hệ thống | **ID:** UC-1 | **Ưu tiên:** Cao |
| **Tác nhân:** Người sử dụng, Quản trị viên | | |
| **Miêu tả:** Bắt đầu phiên làm việc Người sử dụng sẽ mở ứng dụng để tìm kiếm, Quản trị viên mở services để cập nhật danh mục, xem các góp ý | | |
| **Trigger:** Người sử dụng, Quản trị viên khởi động hệ thống  **Kiểu:** 🗹 **Kích hoạt ngoài** 🞏 **Kích hoạt trong** | | |
| **Điều kiện tiên quyết:**   1. Đã tải ứng dụng về 2. Có kết nối internet | | |
| **Các bước thông thường:**  1.0 Khởi động hệ thống   1. Bật kết nối mạng 2. Chạm mở ứng dụng | | |
| **Các bước thay thế:**  1.1 Ứng dụng bật lên kiểm tra không thấy kết nối mạng (xảy ra ở bước 2)   1. Thông báo bật mạng 3G hoặc wifi 2. Ứng dụng hỏi thử kết nối lại hay thoát 3. Người dùng chọn kết nối lại 4. Ứng dụng thiết lập kết nối với máy chủ   5b. Người dùng chọn thoát  6b. Chấm dứt use case | | |

#### 5.3.2.2 Use case 2: Tìm kiếm

|  |  |  |
| --- | --- | --- |
| **Tên Use Case:** Tìm kiếm | **ID:** UC-2 | **Ưu tiên:** Cao |
| **Tác nhân:** Người sử dụng | | |
| **Miêu tả:** Người sử dụng sẽ sử dụng chức năng này để tìm kiếm đường đi, địa chỉ cần đến, ATM, khách sạn… theo các tiêu chí tìm kiếm, có thể tìm kiếm trong cơ sở dữ liệu của web services của chương trình hoặc từ dịch vụ google | | |
| **Trigger:** Người sử dụng tìm kiếm địa điểm  **Kiểu:** 🗹 **Kích hoạt ngoài** 🞏 **Kích hoạt trong** | | |
| **Điều kiện tiên quyết:**   1. Ứng dụng đang mở 2. Có kết nối mạng | | |
| **Các bước thông thường:**  2.0 Tìm kiếm địa điểm   1. Người sử dụng cần tìm địa điểm 2. Nhập tên địa điểm vào thanh tìm kiếm 3. Ấn tìm kiếm 4. Hệ thống thực hiện tìm kiếm 5. Địa điểm trên bản đồ được gửi về hiển thị trên máy | | |
| **Các bước thay thế:**  2.1 Ứng dụng không tìm thấy địa điểm cần tìm (xảy ra ở bước 4)   1. Thông báo không tìm thấy địa điểm cần tìm 2. Yêu cầu nhập lại tên địa điểm   2.2 Địa điểm đã được tìm trước đấy (xảy ra ở bước 2)   1. Hiển thị lịch sử địa điểm đã tìm | | |

#### 5.3.2.3 Use case 3: Lịch sử tìm kiếm

|  |  |  |
| --- | --- | --- |
| **Tên Use Case:** Lịch sử tìm kiếm | **ID:** UC-3 | **Ưu tiên:** Bình thường |
| **Tác nhân:** Người sử dụng | | |
| **Miêu tả:** Người sử dụng có thể xem những địa điểm đã tìm để chọn tìm kiếm nhanh hơn, gợi ý địa điểm đã tìm khi người dùng nhập địa điểm trong thanh tìm kiếm | | |
| **Trigger:** Người sử dụng tìm kiếm địa điểm  **Kiểu:** 🗹 **Kích hoạt ngoài** 🞏 **Kích hoạt trong** | | |
| **Điều kiện tiên quyết:**   1. Đã tìm kiếm địa điểm | | |
| **Các bước thông thường:**   1. Tìm kiếm địa điểm 2. Người sử dụng cần tìm địa điểm 3. Nhập tên địa điểm vào thanh tìm kiếm 4. Gợi ý lịch sử tìm kiếm | | |
| **Các bước thay thế:**  3.1 Xem địa điểm đã tìm (xảy ra ở bước 1)   1. Chọn lịch sử tìm kiếm 2. Hiển thị những địa điểm đã tìm | | |

#### 5.3.2.4 Use case 4: Quản lý lớp

|  |  |  |
| --- | --- | --- |
| **Tên Use Case:** Quản lý lớp | **ID:** UC-4 | **Ưu tiên:** Cao |
| **Tác nhân:** Người sử dụng, Quản trị viên | | |
| **Miêu tả:** Sau khi mở một file bản đồ, tất cả các lớp ứng dụng được hiển thị mặc định. Nếu ta bỏ chọn một lớp nào đó thì lớp đó sẽ không hiện lên màn hình. Với chức năng này thì ta cũng có thể thêm một lớp bản đồ khác vào bản đồ hiện hành, hoặc xem các thông tin của một lớp nào đó | | |
| **Trigger:** Cần xem thông tin một lớp bản đồ  **Kiểu:** 🗹 **Kích hoạt ngoài** 🞏 **Kích hoạt trong** | | |
| **Điều kiện tiên quyết:**   1. Bản đồ được hiển thị trên máy | | |
| **Các bước thông thường:**   1. Bật tắt lớp bản đồ 2. Bỏ tích lớp bản đồ không cần thông tin 3. Hệ thống hiển thị lớp còn lại 4. Tích chọn lớp bản đồ cần xem thông tin 5. Hệ thống hiển thị lớp bản đồ vừa tích chọn | | |

#### 5.3.2.5 Use case 5: Zoom bản đồ

|  |  |  |
| --- | --- | --- |
| **Tên Use Case:** Zoom bản đồ | **ID:** UC-5 | **Ưu tiên:** Bình thường |
| **Tác nhân:** Người sử dụng | | |
| **Miêu tả:** Dùng để phóng to, thu nhỏ một cùng bản đồ để người sử dụng có thể nhìn rõ hơn các đối tượng cần quan tâm | | |
| **Trigger:** Xem rõ đối tượng trên bản đồ  **Kiểu:** 🗹 **Kích hoạt ngoài** 🞏 **Kích hoạt trong** | | |
| **Điều kiện tiên quyết:**   1. Bản đồ đã hiển thị trên máy | | |
| **Các bước thông thường:**   1. Zoom đối tượng 2. Chọn đối tượng 3. Phóng to đối tượng để nhìn rõ hơn 4. Thu nhỏ đối tượng để nhìn bao quát hơn | | |

#### 5.3.2.6 Use case 6: Pan

|  |  |  |
| --- | --- | --- |
| **Tên Use Case:** Pan | **ID:** UC-6 | **Ưu tiên:** Bình thường |
| **Tác nhân:** Người sử dụng | | |
| **Miêu tả:** Dùng để dịch chuyển bản đồ đến vị trí mong muốn mà không có thay đổi tỷ lệ trên bản đồ | | |
| **Trigger:** Người sử dụng di chuyển bản đồ  **Kiểu:** 🗹 **Kích hoạt ngoài** 🞏 **Kích hoạt trong** | | |
| **Điều kiện tiên quyết:**   1. Bản đồ đã hiển thị trên máy | | |
| **Các bước thông thường:**   1. Di chuyển bản đồ 2. Giữ bản đồ 3. Di chuyển đến vị trí muốn xem | | |

#### 5.3.2.7 Use case 7: Gửi góp ý

|  |  |  |
| --- | --- | --- |
| **Tên Use Case:** Gửi góp ý | **ID:** UC-7 | **Ưu tiên:** Bình thường |
| **Tác nhân:** Người sử dụng | | |
| **Miêu tả:** Người sử dụng sẽ sử dụng chức năng này để gửi góp ý về cho người phát triển về các vấn đề khi sử dụng chương trình | | |
| **Trigger:** Người sử dụng muốn góp ý  **Kiểu:** 🗹 **Kích hoạt ngoài** 🞏 **Kích hoạt trong** | | |
| **Điều kiện tiên quyết:**   1. Ứng dụng đang mở 2. Có kết nối mạng | | |
| **Các bước thông thường:**   1. Gửi góp ý 2. Nhập nội dung 3. Gửi đi | | |
| **Các bước thay thế:**  7.1 Mất kết nối (xảy ra ở bước 2)   1. Thông báo mất kết nối, không gửi được 2. Hiển thị gửi lại hay tắt chức năng 3. Người sử dụng chọn gửi lại 4. Hệ thống thực hiện lại kết nối   5b. Người sử dụng chọn tắt chức năng  6b. Kết thúc use case | | |

#### 5.3.2.8 Use case 8: Xem góp ý

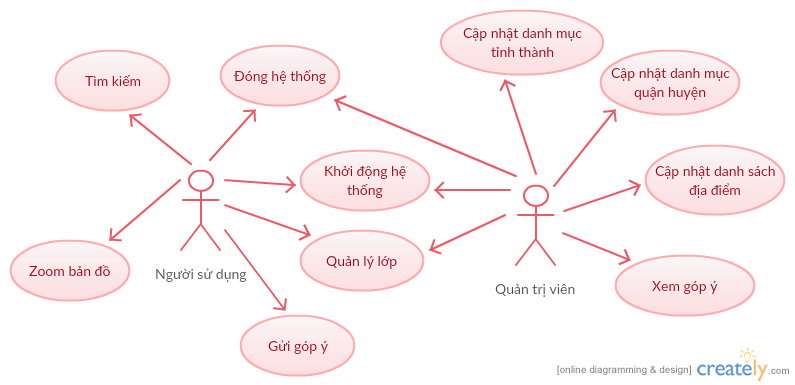
|  |  |  |
| --- | --- | --- |
| **Tên Use Case:** Xem góp ý | **ID:** UC-8 | **Ưu tiên:** Bình thường |
| **Tác nhân:** Quản trị viên | | |
| **Miêu tả:** Xem các góp ý được gửi đến từ người sử dụng chương trình | | |
| **Trigger:** Quản trị viên xem góp ý của người sử dụng  **Kiểu:** 🗹 **Kích hoạt ngoài** 🞏 **Kích hoạt trong** | | |
| **Điều kiện tiên quyết:**   1. Có thư góp ý của người sử dụng | | |
| **Các bước thông thường:**   1. Xem thư 2. Chọn thư được gửi đến 3. Xem thư | | |

#### 5.3.2.9 Use case 9: Cập nhật danh mục, địa điểm

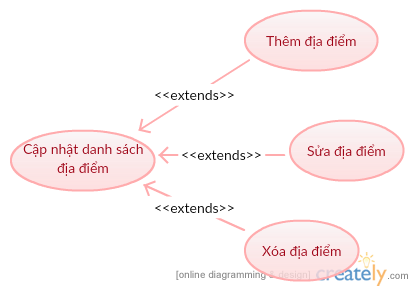
|  |  |  |
| --- | --- | --- |
| **Tên Use Case:** Cập nhật | **ID:** UC-9 | **Ưu tiên:** Cao |
| **Tác nhân:** Quản trị viên | | |
| **Miêu tả:** Cập nhật tỉnh thành, quận huyện, các địa điểm: ATM, khách sạn, ngân hàng… | | |
| **Trigger:** Cập nhật bản đồ  **Kiểu:** 🗹 **Kích hoạt ngoài** 🞏 **Kích hoạt trong** | | |
| **Điều kiện tiên quyết:**   1. Dữ liệu mới, chính xác | | |
| **Các bước thông thường:**   1. Cập nhật dữ liệu 2. Sử dữ liệu 3. Thêm mới dữ liệu 4. Xóa dữ liệu cũ không còn giá trị | | |

### 5.3.3 Biểu đồ use case

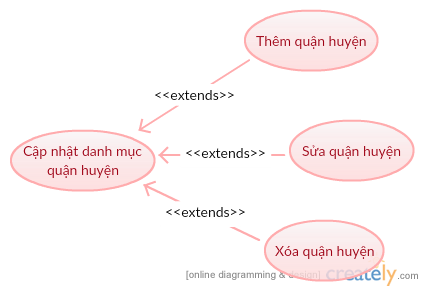
* Use case tổng quát



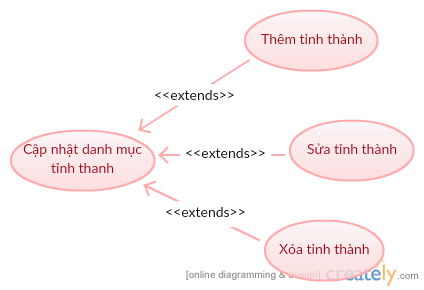
* Use case cập nhật địa điểm



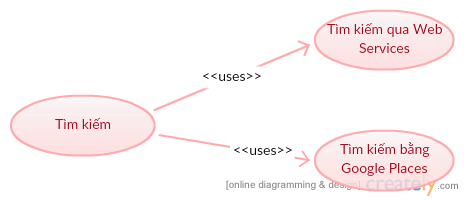
* Use case cập nhật Quận huyện



* Use case cập nhật Tỉnh thành



* Use case Tìm kiếm



### 5.3.5 Biểu đồ hoạt động

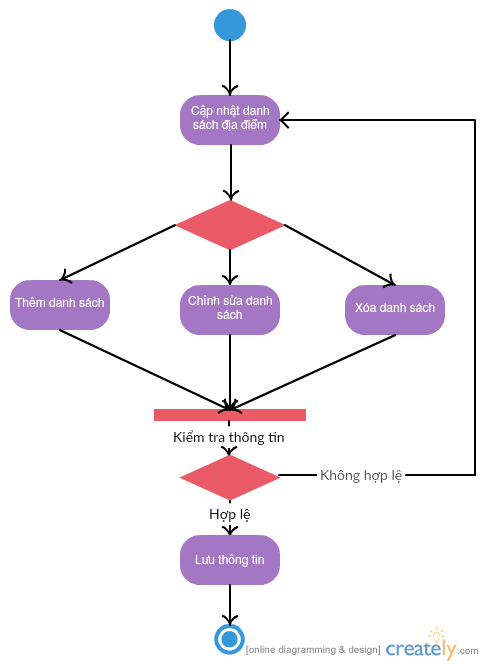
* Biểu đồ hoạt động mở hệ thống



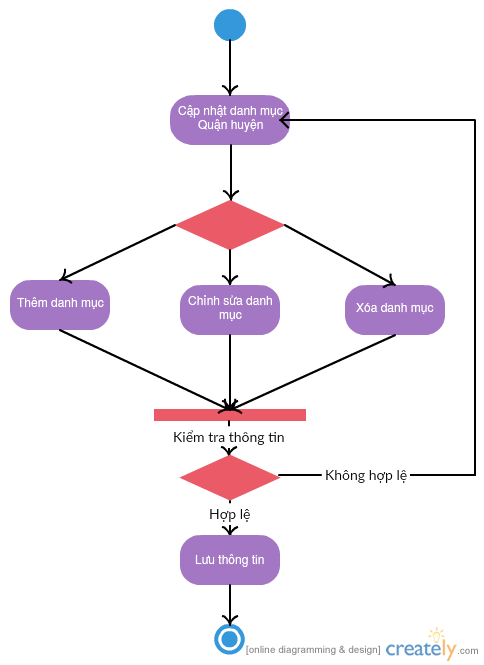
* Biểu đồ Đóng hệ thống



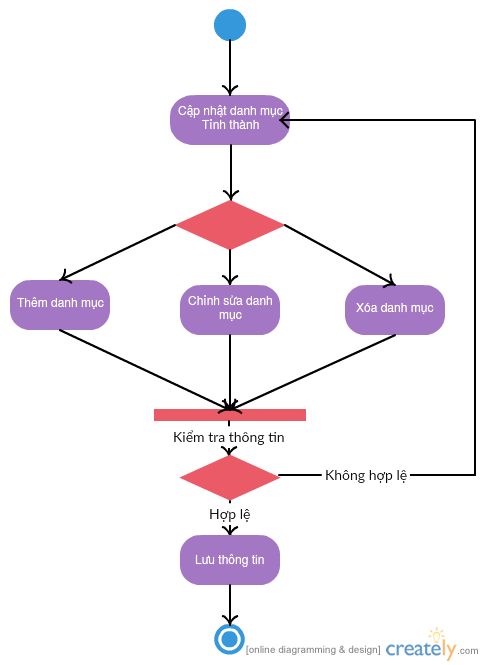
* Biểu đồ Cập nhật danh sách địa điểm



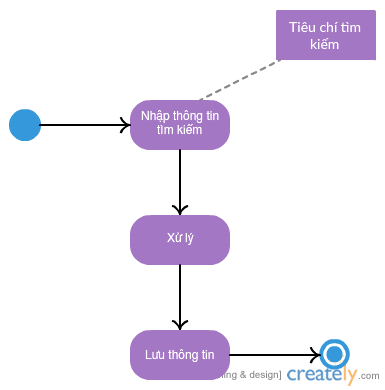
* Biểu đồ Cập nhật danh mục quận huyện



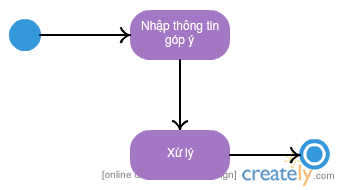
* Biểu đồ Cập nhật Danh mục tỉnh thành



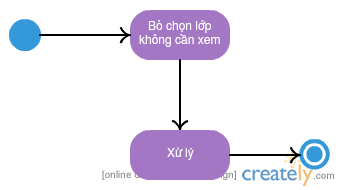
* Biểu đồ Tìm kiếm



* Biểu đồ Góp ý



* Biểu đồ Quản lý lớp



## 5.4. Các vấn đề trong quá trình xây dựng ứng dụng

### 5.4.1 Giao diện UI trong lập trình ứng dụng Android

Giao diện người dùng của ứng dụng trên hệ điều hành Android là các Activity. Mỗi Activity tương ứng với một giao diện gồm nhiều đối tượng View. Các đối tượng View được chia làm hai loại:

* View: là một UI cụ thể có khả năng hiển thị và tương tác với người dùng. Ví dụ: TextView, Button…
* View Group: chứa các View và View Group khác, mục đích của View Group là xắp xếp các thành phần View theo một trật tự nào nhất định

Các loại View thường sử dụng:

* TextView: hiển thị text, một dòng hay nhiều dòng
* EditText: chỉnh sửa text, có thể chọn nhiều kiểu EditText ví dụ: chỉ nhập số, vùng nhập địa chỉ mail, vùng nhập tự đoán nội dung…
* ImageView: hiển thị ảnh
* Button: nút nhấn phân biệt bằng text
* ImageButton: nút nhất phân biệt bằng hình ảnh
* RadioButton: chọn một trong danh sách
* CheckBox: chọn nhiều trong danh sách
* ListView: hiển thị các item con theo kiểu danh sách dọc.
* ExpandableListView: hiển thị các item con theo từng nhóm theo kiểu cây thư mục
* ToggleButton: nút nhấn hai trạng thái.
* ProgressBar: thanh tiến trình của tác vụ.

Các loại View Group thường xử dụng:

* LinearLayout: dùng thường xuyên nhất, xắp xếp các View con theo chiều ngang hay chiều dọc, mỗi View con tương ứng một cột hay một hàng
* RelativeLayout: xắp xếp các View con theo vị trí tương đối với các View con khác và với View cha
* ScrollView: Tự động chuyển sang thành cuộn khi nội dung dài quá chiều dài màng hình theo chiều dọc
* TableLayout: Xắp xếp các view theo kiểu bảng, gồm nhiều TableRow

Cách đơn giản và phổ biết nhất để thiết kế UI trên ứng dụng Android là viết tập tin Layout bằng XML. Mỗi Activity sẽ tương ứng với một XML Layout. Lưu ý: luôn luôn có duy nhất một View Group gốc chứa toàn bộ các View còn lại.

Các thuộc tính của View được thiếp lập bằng giá trị các thuộc tính của thể XML. Các thuộc tính quan trọng thường sử dụng trên mọi kiểu View:

* **Id:** tên vủa View hay View Group, để xác định và lấy ra sử dụng trong code
* **layout\_width:** độ rộng của View
* **layout\_height:** độ cao của View

Giá trị của layout\_width và layout\_height có thể là giá trị số kèm theo đơn vị (5px, 5dp …):

* **px:** pixel, đơn vị điểm ảnh chính xác, không nên dùng khi thiết kế cho nhiều kích thước màn hình
* **dp** hoặc **dip** (Density-independent Pixels): đơn vị phụ thuộc kích thước màn hình, nên dùng, thích hợp cho thiết kết UI trên nhiều độ phân giải
* **sp** (Scale-independent Pixels): giống dp nhưng thay đổi khi người dùng tùy chọn kích thước font chữ cho hệ thống, tích hợp cho font chữ
* **pt:** point, 1/72 của một inches
* **in:** inches

Hoặc có thể là giá trị cố định sau:

* **fill\_parent:** lấy hết kích thước View cha.
* **match\_parent:** như fill\_parent kể từ API lever 8.
* **wrap-content:** kích thước vừa đủ nội dung.
* **layout\_weight:** thuộc tích khác đặt biệt nhưng sử dụng khá nhiều, giá trị số, cho biết View sẽ tràn hết khoảng trống còn lại của View theo giá trị được đặt
* **gravity:** thiết lập vị trí tương đối của View con: trái, phải, giữa, trên, dưới…
* **layout\_gravity:** thiết lập vị trí tương đối so với View cha: trái, phải, giữa, trên, dưới…
* **padding:** khoảng cách của nội dung hay View con cách mép trong trái, phải trên, dưới, không làm thay đổi kích thước View
* **layout\_margin:** thêm phần khoảng trống vào View ở 4 mép ngoài trái, phải trên, dưới…làm thay đổi kích thước View

Ngoài ra với mỗi kiểu View sẽ có rất nhiều thuộc tính khác nhau có thể thiết lập.

Thao tác trên View

Mỗi loại View hay View Group sẽ có các hành động khác nhau.

Thêm sự kiện thao tác trên View hoặc View Group

Để bắt sự kiện người dùng thao tác trên View phải tạo các Listener tương ứng với hành động của người dùng và tương ứng với mỗi loại View. Các loại Listener thường dùng:

* **OnClickListener:** sự kiện click vào View
* **OnLongClickListener:** sự kiện nhấn và giữ View hơn 1s
* **OnItemClickListener:** sự kiện chọn item (ListView …)
* **OnCheckedChangeListener:** sự kiện thay đổi trạng thái check (ToogleButton, CheckBox)
* **OnDoubleTapListener:** sự kiện click đôi
* **OnTouchListener:** sự kiện trược trên View

### 5.4.2 Đăng ký bản đồ Google Map trong ứng dụng

Để sử dụng bản đồ Google Map trong ứng dụng cần phải đăng ký một chuỗi APIkey. Mỗi phiên bản của Android SDK được cài đặt trên máy tính sẽ có thể lấy một APIkey duy nhất và không thể dùng được trên máy khác.

Để hiển thị bản đồ Google Map lên UI cần phải khai báo một View đặt biệt là: com.google.android.maps.MapView.

Activity của hiển thị bản đồ cần phải kế thừa lớp MapActivity. Đưa thư viện com.google.android.maps vào ứng dụng bằng cách khai báo trong tập tin AndroidManifest.xml

Để thao tác trên bản đồ Google Map có hai đối tượng quan trọng:

* **MapView:** là đối tượng bản đồ Google Map. Các thao tác thêm thông tin, thêm đánh dấu, thêm đường đi, cập nhật bản đồ, lấy tọa độ… thực hiện trên đối tượng này
* **MapController:** thao tác trên bản đồ: phóng to, thu nhỏ, di chuyển đến một tạo độ nhất định…

Một tọa độ trên bản đồ Google Map là một thể hiện của lớp GeoPoint. Thuộc tính của thể hiện của lớp GeoPoint gồm có: latitudeE6(int) và longitudeE6(int).

### 5.4.3 Đánh dấu địa điểm và hiện thông tin trên bản đồ

Để đánh dấu địa điểm, vẽ thông tin, lấy tạo độ trên bản đồ cần sử dụng thể hiện của lớp Overlay. Lớp ItemizedOverlay là một lớp kế thừa từ lớp Overlay dùng để vẽ một danh sách điểm có kiểu hoặc kế thừa từ lớp OverLayItem.

Trong mỗi thể hiện của lớp MapView có chứa một danh sách ArrayList các Overlay.

Để hiển thị Overlay lên bản đồ phải thêm nó vào danh sách Overlay của MapView. Nội dung của Overlay được thêm vào sau sẽ đè lên các Overlay đã thêm trước đó.

Để vẽ thông tin lên bản đồ, cần phải cài đặt lệnh vẽ cho mỗi Overlay trong phương thức draw(). Lớp ItemizedOverlay là lớp đã kế thừa từ lớp Overlay và cài đặt sẵn phương thức vẽ các điểm đánh dấu bới đối số là ảnh Bitmap hay Drawable để đánh dấu mỗi OverLay, nên nếu chỉ cần vẽ điểm đánh dấu thì không cần override phương thức draw(). Tuy nhiên nếu muốn vẽ thêm nội dung khác ví dụ: hình ảnh, đoạn thằng, hình tròn … thì phải phải override phương thức onDraw.

Khi vẽ lên bản đồ cần phải chuyển từ tạo độ GeoPoint sang tọa độ điểm trên màn hình. Thực hiệu điều đó bằng cách sử dụng lớp Projection.

Mỗi OverlayItem bao gồm các thuộc tính: point (GeoPoint), title (String), snippet(String).

Lớp ItemizedOverlay là lớp kế thừa từ lớp Overlay mà Android SDK cung cấp để vẽ đánh dấu điểm nhanh chóng của một danh sách các OverlayItem. Ngoài ra có thể kế thừa tiếp tục lớp ItemizedOverlay và OverlayItem để tạo ra các lớp có khả năng mở rộng hơn. Lớp Balloon ItemizedOverlay là lớp kế thừa từ lớp ItemizedOverlay để vẽ một bảng thông tin lên điểm đánh dấu trên bản đồ khi người dùng click vào.

### 5.4.4 Sử dụng SQLite trong hệ điều hành Android

Mặc định SQLite không có công cụ quản lý riêng, vì vậy cơ sở dữ liệu phải thao tác trên code. Android SDK cung cấp một lớp để thao tác trên cơ sở dữ liệu SQLite là SQLiteOpenHelper.

Thao tác trên cơ sở dữ liêu bằng cách override hai phương thức:

* onCreate(SQLiteDatabase db): gọi để tạo cơ sở dữ liệu, là nơi cài đặt các phương thức tạo cơ sở dữ liệu, tạo bảng, tạo trigger… Được gọi khi SQLiteOpenHelper khởi tạo và chưa có cơ sở dữ liệu
* onUpgrade(SQLiteDatabse db, int oldVersion, int newVersion): gọi khi cập nhật phiên bản cho cơ sở dữ liệu. Là nơi cài đặt các thao tác cập nhật cơ sở dữ liệu

**Sử lý đối tượng Cursor để lấy dữ liệu truy vấn**

Toàn bộ kết quả truy vấn sẽ nằm trong đối tượng Cursor. Các câu lệnh để sử dụng Cursor:

* boolean moveToNext(): chuyển đến dòng tiếp theo
* boolean moveToFirst(): chuyển đến dòng đầu tiên
* boolean moveToPosition(int position): chuyển đến dòng ở vị trí position
* boolean moveToPrevious(): đến dòng vừa trước
* boolean moveToLast(): chuyển đến dòng cuối cùng

Các phương thức trên trả về false nếu không có nội dung ở dòng chuyển đến.

**Kiểu dữ liệu trong SQLite**

* INTEGER: số nguyên, chứa trong 1, 2, 3, 4, 6, hay 8 byte phụ thuộc vào độ lớn
* REAL: số thực, kiểu 8-byte của IEEE
* TEXT. kiểu chuỗi hỗ trợ UTF-8, UTF-16BE or UTF-16LE
* BLOB: kiểu dữ liệu để lưu trữ thông tin bất kỳ, truyền vào thế nào thì nhận lại như vậy

Để đơn giản hóa, trong SQLite không có kiểu boolean, kiểu datetime… nên phải dùng kiểu int để thay thế.

### 5.4.5 Phân tích cú pháp XML

Một nhiệm vụ của chương trình là phân tích tập tin xml kết quả trả về của các API webservice để trích xuất nội dung: Có nhiều cơ chế để phân tích một tài liệu xml bằng ngôn ngữ Java trong Android. Một là phân tích hướng sự kiện mà tiêu biểu là sử dụng API SAX do Sun đưa ra. Hai là phân tích hướng đối tượng bằng cách sử dụng API DOM do W3C đưa ra.

Mỗi một mô hình phân tích đều có ưu nhược điểm riêng. Mô hình phân tích hướng đối tượng mang lại khả năng tìm kiếm. Trong khi mô hình phân tích heo hướng sự kiện duyệt tài liệu từ trên xuống dưới một chiều, chủ yếu dùng để đọc xml là chính. Ưu điểm của phân tích hướng sự kiện là tốc độ đọc nhanh, ít tốn bộ nhớ. Vì vậy trong trường hợp này ta sẽ chọn phân tích file Rss theo phương pháp hướng sự kiện sử dụng API SAX.

Sử dụng SAX bằng cách tạo một lớp kế thừa từ lớp DefaultHandler và override các phương thức:

* characters(char[] ch, int start, int length): con trỏ đang duyệt đến nội dung của mỗi thẻ
* startElement(String uri, String localName, String qName, Attributes attributes): con trỏ duyệt đến đầu thẻ
* endElement(String uri, String localName, String qName): con thỏ duyệt đến kết thúc thẻ

Con trỏ của XML SAX sẽ duyện lần lược từ đầu tài liệu, đến vị trí tương ứng nó sẽ gọi phương thức tương ứng, nhiệm vụ phải cài đặt các câu lệnh nhận biết con trỏ đang ở vị trí nào để lấy dữ liệu thích hợp.

### 5.4.6 Bắt vị trí hiện tại bằng các Location Provider

Trong hệ thống Android có cung cấp sẵn các dịch vụ để lấy vị trí của người dùng. Để tính toán và xác định vị trí người dùng, hệ thống Android sử dụng dữ liệu GPS (chính xác nhất), tín hiệu Wifi, tín hiệu sóng điện thoại.

Để truy cập và sử dụng các dịch vụ vị trí của hệ thống, phải sử dụng lớp LocationManager.

Implement interface LocationListener tạo ra một lớp lắng nghe mỗi khi vị trí thay đổi và đăng ký nó với đối tượng của lớp LocationManager. Có thể thiết lập điều kiện để LocationListener quyết định cập nhật vị trí mới, điều kiện bao gồm: sau một thời gian nhất định, cách vị trí đã xác định lần trước một khoảng nhất định, loại thành phần cung cấp vị trí.

Thông tin vị trí bắt được sẽ là một đối tượng của lớp Location. Mỗi thể đối tượng Location sẽ chứa vị trí địa lý (GeoPoint) và các thông tin kèm theo: loại provider, độ chính xác, tốc độ, thời gian.

### 5.4.7 Dùng Observer Template để thông báo cho các Activity mỗi khi địa điểm hiện tại thay đổi

Sử dụng LocationListener đăng ký với LocationManager, mỗi lần vị trí mới được cập nhật đối tượng LocationListener sẽ chạy phương thức của nó là onLocationChanged(Location location).

Mỗi khi vị trí hiện tại mới thay đổi, phải cập nhật sự thay đổi đó ngay trên UI. Nếu muốn cập nhật trên một Activity cụ thể có thể tùy biến lớp LocationListener để đưa một tham chiếu của Activity đó vào LocationListener và phương thức onLocationChanged(Location location) của LocationListener sẽ gọi một phương thức cập nhật UI ủa Activity đó.

Vấn đề đặt ra là sẽ có nhiều Activity khác nhau cần phải cập nhật khi vị trí hiện tại thay đổi và mỗi Activity sẽ thay đổi UI một cách khác nhau. Giải pháp là sử dụng mẫu Observer để xây dựng cơ chế thông báo mỗi khi vị trí thay đổi.

Trong Java SDK có cung cấp lớp cơ sở Observable và interface Observer.Xây dựng lớp MyLocationListener implement interface LocationListener và đồng thời kế thừa lớp Observable để nó trở thàng một Subject thông báo cho các Observer của nó.

Đối với các Activity muốn cập nhật UI mỗi khi vị trí hiện tại thay đổi sẽ implement interface Observer, override phương thức update() cài đặt các hành động phù hợp. Mỗi khi Activity khởi tạo sẽ đăng ký nó với MyLocationListener. Mỗi khi vị trí có thay đổi MyLocationListener sẽ gọi phương thức notifyObservers() trong onLocationChanged(Location location) và cập nhật các Activity.

# VI. Kết Luận và hướng phát triển

## 6.1. Ưu điểm, khuyết điểm của ứng dụng

**Ưu điểm:**

* Cung cấp khá rõ thông tin về các địa điểm, địa chỉ cho người sử dụng
* Phân nhóm loại địa điểm nên khá tiện lợi cho người sử dụng khi tìm kiếm địa điểm
* Vì ứng dụng trên mobile nên có tính linh hoạt cao

**Nhược điểm:**

* Còn thiếu một số chức năng như: lưu điểm yêu thích, chia sẻ vị trí hiện tại, người dùng thêm địa điểm mới, tìm kiếm bằng giọng nói…
* Không có đồng bộ hóa dữ liệu, khi xóa ứng dụng thì dữ liệu người dùng sẽ mất
* Phải có kết nối mạng 3G hoặc wifi khi sử dụng

## 6.2. Hướng phát triển

* Bổ sung thêm dữ liệu
* Thêm các chức năng mới
* Có thể sử dụng bản đồ offline
* Đồng bộ hóa cơ sở dữ liệu

# BẢNG PHÂN CÔNG CÔNG VIỆC

|  |  |  |
| --- | --- | --- |
| **STT** | **Họ và tên** | **Công việc** |
| 1 | Nguyễn Thanh Tuấn | Tìm hiểu google map, google map API web services, phân tích thiết kế hệ thống ứng dụng |
| 2 | Vũ Bảo Dương |
| 3 | Tạ Quốc Đạt | Tìm hiểu về XML, Webservice, đánh giá ứng dụng |
| 4 | Nguyễn Đức Hiển | Tìm hiểu về GIS, thiết kế bản đồ trên mobile, đánh giá ứng dụng |