**PHẦN 4 : HIỆN THỰC ỨNG DỤNG**

**HIỂN THỊ HIỆN TRẠNG GIAO THÔNG :**

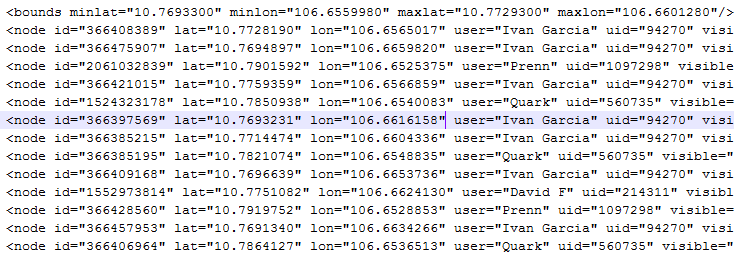
**Dữ liệu về đường đi :**

GoogleMap API chỉ cung cấp cho người dùng một số chức năng như :

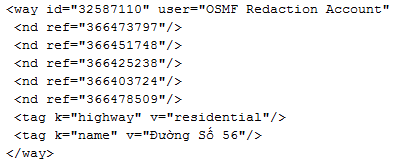
* Chiếu một điểm có tọa độ xuống bản đồ.
* Cho người dùng lấy dữ liệu về độ zoom của bản đồ.

Để hiện thực việc vẽ đường đi chúng tôi lựa chọn giải pháp lấy dữ liệu từ OpenStreetMap để có được tọa độ các điểm dùng để vẽ đường đi.

Dữ liệu được lấy xuống trên OpenStreetMap sẽ là file XML có chứa dữ liệu thông tin về đường đi của toàn bản đồ.



Node\_Id sẽ chưa thông tin về kinh tuyến và vĩ tuyến.



Way\_Id sẽ chứa tên đường đi, và loại đường đi.Và tham khảo đến các node của nó

Ngoài ra còn nhiều thông tin khác. Nhưng để vẽ hiện trạng giao thông chúng ta chỉ cần chú ý đến hai loại dữ liệu nêu trên.

Sau khi lấy được file XML chứa dữ liệu đường đi của toàn thành phố một cách thủ công.Dữ liệu sẽ được đọc bằng SAX(Simple API for XML) , chỉ lấy các giá trị liên quan đến kinh độ vĩ độ, loại đường, chỉ số đường , chỉ số node. Mục đích là để cho dữ liệu thu được là ít nhất. Vừa đủ để phục vụ yêu cầu ứng dụng. Dữ liệu sau khi được sử lý sẽ được lưu xuống cơ sở dữ liệu được trình bày ở bên dưới. Dữ liệu này là một trong những dữ liệu quan trong cho ứng dụng.

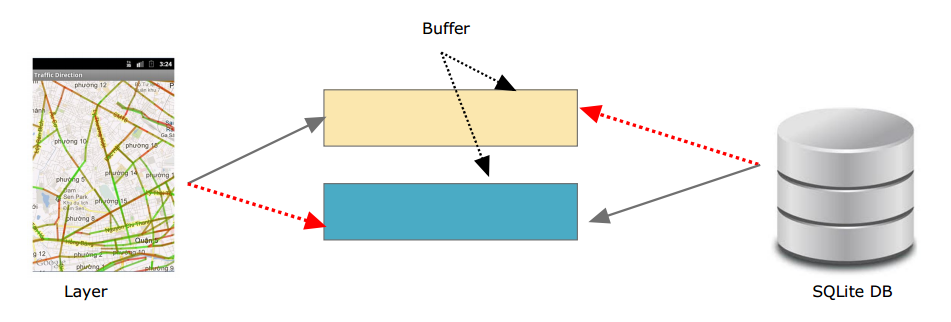
**Lấy dữ liệu theo đúng màn hình.**

Đầu tiên sẽ xác định điểm trung tâm của màn hình . Sau đó sẽ tính ra vị trí của 4 góc màn hình. Và tình ra được tất cả các điểm chứa trên màn hình. Từ đó chúng ta sẽ lấy lượng dữ liệu cần thiết dùng đủ cho màn hình. Làm tăng tốc độ và thời gian đáp ứng cho ứng dụng.

Bên cạnh đó sẽ load dữ liệu trước tám màn hình để giảm thời gian load cho ứng dụng . Mỗi khi người dùng di chuyển.

**Cách truy xuất dữ liệu nhanh :**

Dùng hai buffer để truy xuất dữ liệu.Trong quá trình đọc dữ liệu từ dưới cơ sở dữ liệu lên , sẽ sử dụng hai thread. Một thread dùng để đọc dữ liệu từ cơ sở dữ liệu và lưu vào buffer. Một thread dùng để đọc dữ liệu từ buffer còn lại và vẽ lên bản Overlay. Hai thread này sẽ hoạt động liên tục. Và hoán đổi buffer với nhau để giảm thời gian chờ khi đọc dữ liệu.



**Hiển thị dữ liệu theo độ zoom của bản đồ:**

Với mỗi cấp độ zoom khác nhau của bản đồ .Sẽ có những dữ liệu khác nhau được lựa chọn hiển thị.

Đường đi ở mức zoom cao : (những chấm đỏ là số node cần để hiển thị)

Đường đi ở mức zoom thấp hơn :

Hiển thị theo mức độ zoom và các loại đường. Ta có các loại đường như sau.

* Loại số 4 hẻm nhỏ , bao gồm các con hẻm.
* Loại số 3 là các đường nhỏ. Ví dụ : đường Thành Thái , Tô Hiến Thành …
* Loại số 2 là các đường lớn .Ví dụ : đường Lý Thường Kiệt , 3 tháng 2
* Loại số 1 là các đường quốc lộ.

Tùy theo mức độ zoom của bản đồ mà ta sẽ vẽ các thêm các loại đường hoặc loại bỏ đi một số đường mục đích là làm giảm lượng dữ liệu cần tải , cải thiện tốc độc cho ứng dụng.



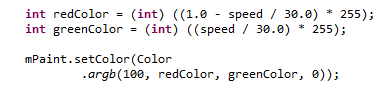
**Xác định màu sắc và vận tốc :**

C:\Users\DucNguyen\Dropbox\GoogleMapAndroid\open-gps-tracker\OpenGPSTracker\application\bin\res\drawable-mdpi\speedindexbar.png

0 (km/h) >40 (km/h)

Màu săc ở đây sẽ được chọn từ màu đỏ cho đến màu xanh non (green). Để hiển thị vận tốc của trên các đoạn đường .Ứng dụng sẽ hiển thị vận tốc từ 0(km/h) tới 40(km/h). Đây là dải vận tốc chính của các phương tiện lưu thông trong thành phố.

Đối với các dải vận tốc từ 0 (km/h) đến 40(km/h). Để biểu diễn các khoảng vận tốc trong đó chúng ta sẽ sử dụng công thức nội suy dưới đây (dạng code java).



**Làm mới dữ liệu**

Mặc định dữ liệu thời gian yêu cầu dữ liệu sẽ nằm trong khoảng 5-10 giây.

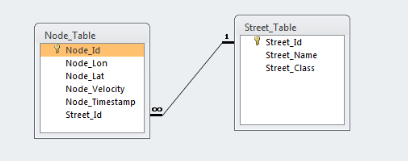
Và sau một khoảng thời gian là 15-20 phút thì ứng dụng sẽ gửi yêu cầu tới server để server cập nhật dữ liệu giao thông và gửi dữ liệu mới nhất về.

Việc hiển thị dữ liệu giao thông lấy được từ server lên trên màn hình ứng dụng cũng sử dụng buffer để có được tốc độ nhanh hơn.

**Thiết kế cơ sở dữ liệu cho ứng dụng :**

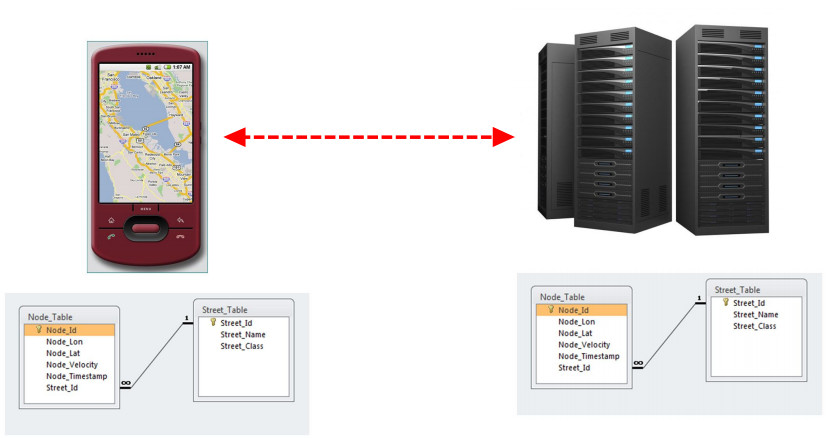
Cơ sở dữ liệu sẽ được thiết kế cả ở server và trong ứng dụng.

Cơ sở dữ liệu :



Thực hiện việc đánh index trên các cột NodeId , StreetId, StreetClass để có truy cập dữ liệu nhanh hơn.

Cơ sở dữ liệu trên ứng dụng và cơ sở dữ liệu ở Server



**Gửi nhận dữ liệu trên Server :**

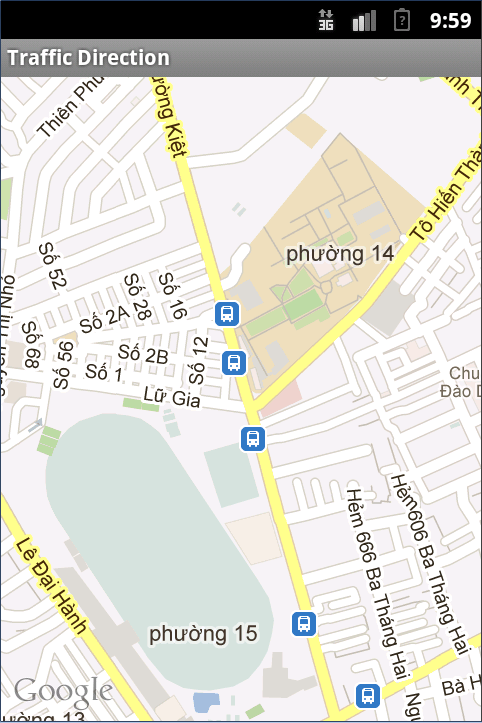
Gửi nhận dữ liệu server và ứng dụng dùng TCP/IP . Mặc dù phương pháp này tốn rất nhiều chi phí để duy trì kết nối. Nhưng nó cũng cho phép việc liên lạc giữa server được với ứng dụng.

Để giảm chi phí overhead khi thiết lập kết nối TCP/IP. Sẽ sử dụng các biện pháp như sau.

* Sử dụng dữ liệu dạng String để gửi nhận dữ liệu trên với server. Mục đích là để đơn giản quá trình xử lý dữ liệu vì dữ liệu gửi nhận đơn giản sẽ giảm được thời gian truy xuất dữ liệu.
* Vì hiện trạng giao thông không thể thay đổi trong thời gian tức thì, nên ứng dụng sẽ gửi nhận dữ liệu một cách tự động sau khoảng 15-20 phút một lần.
* Thời gian để gửi và nhận dữ liệu từ server đến ứng dụng chỉ dưới hạn dưới 1 phút.

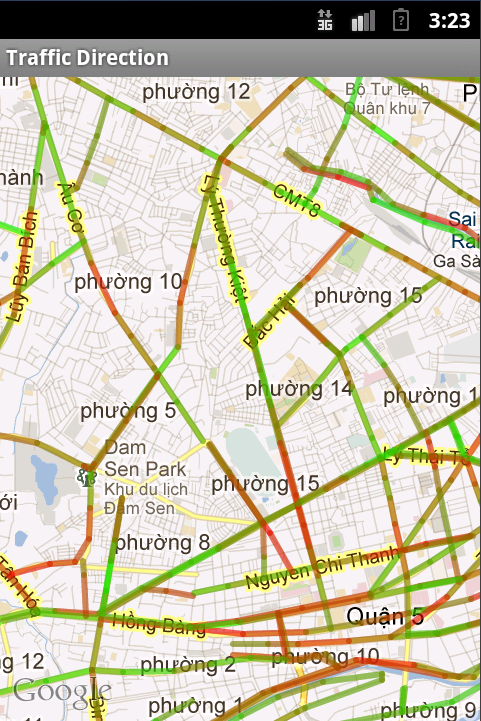
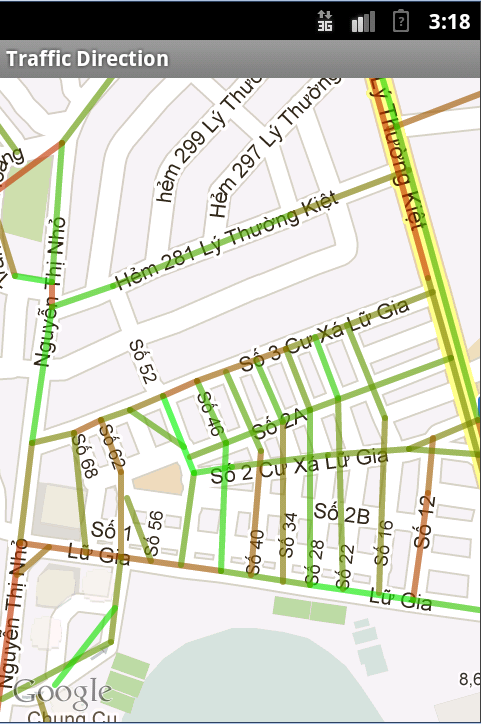
**HOÀN THÀNH ỨNG DỤNG:**

**Màn hình chính của ứng dụng :**



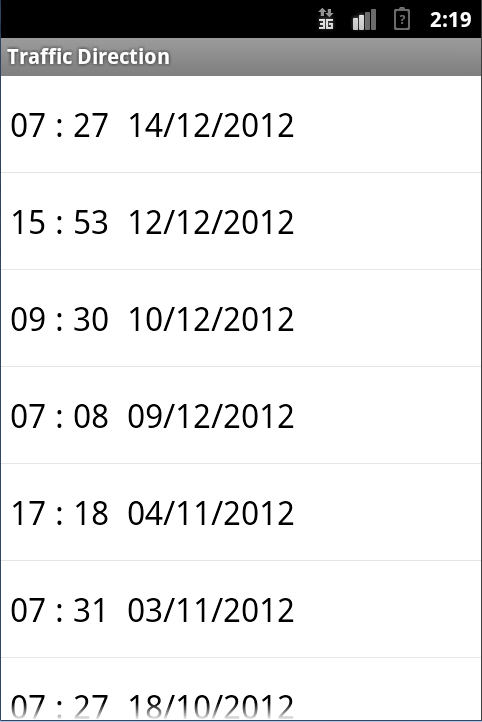
Về giao diện của ứng dụng.Ở đây vì màn hình của các thiết bị di động nhỏ, nên chúng tôi chọn phương pháp. Ban đầu ứng dụng chỉ hiện thị GoogleMap . Mọi chức năng sẽ do người dùng tự quyết định bằng cách nhấn vào phím Menu. Khi đó Menu của chương trình sẽ tự hiển thị lên cho người dùng lựa chọn.

**Hiện thị trạng thái giao thông của một vùng:**

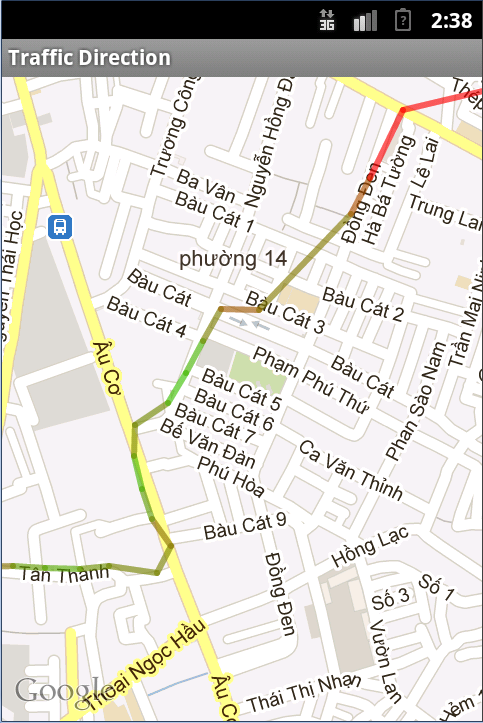


Để hiện thị hiện trạng giao thông. Sẽ có một lớp OverLay lên trên GoogleMap. Lớp Overlay này sẽ hiển thị đường đường đi với các màu khác nhau tương ứng với vận tốc khác nhau.

**Hiển thị đường đã đi qua : (Tracking)**

****

Hiển thị danh sách các đường đã đi qua. Sau khi người dùng chọn được đường đi đã track. Dữ liệu sẽ được hiển thị như sau.



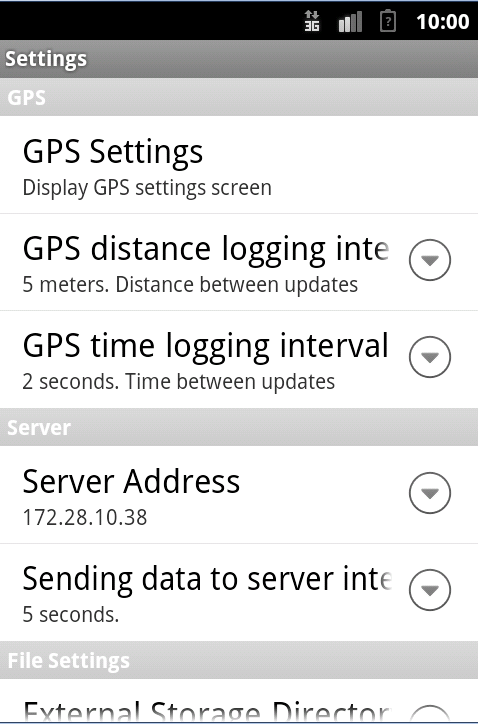
Hiện thị đường đã đi qua. Với màu sắc của từng đoạn đường tương ứng với vận tốc khác nhau.

**Hiển thị đường đi cho chức năng tìm đường đi.**



Sau khi nhập dữ liệu với điểm bắt đầu và điểm kết thúc. Ứng dụng sẽ xuất ra đường đi với thời gian ngắn nhất.Đường đi đã được server tính toán trước vận tốc đựa trên dữ liệu dự doán từ server.

**Màn hình hiển tùy chỉnh cấu hình GPS:**



Màn hình tùy chọn với nhiều tính năng có tăng độ chính xác của thiết bị.

**CÀI ĐẶT ỨNG DỤNG:**