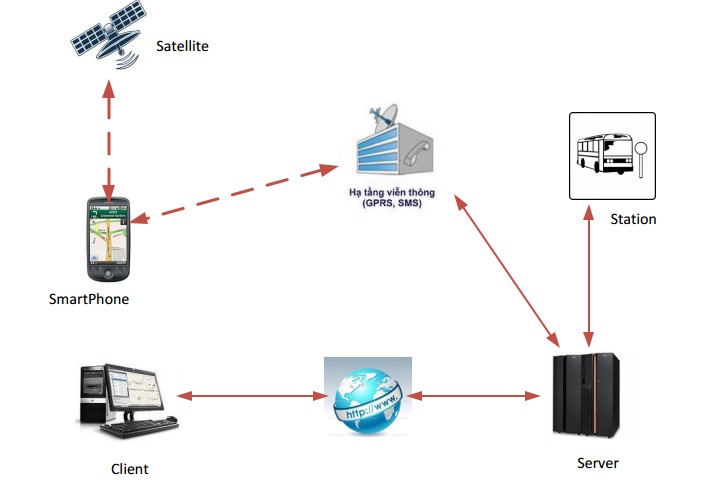
**PHẦN 3 : PHÂN TÍCH ỨNG DỤNG**

**HỆ THỐNG CHÍNH:**

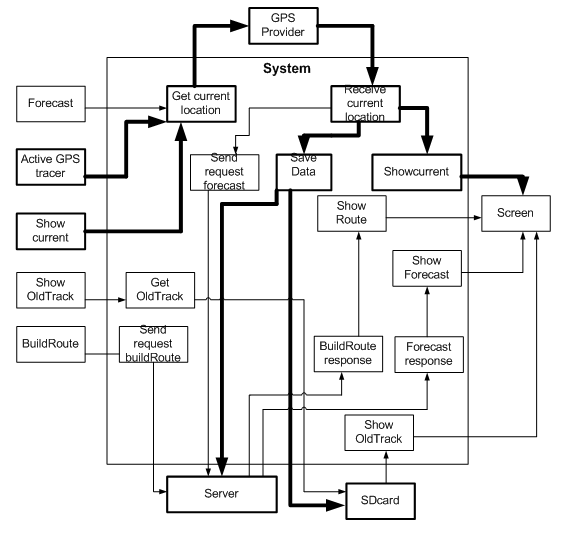
Các thành phần tham gia hệ thống :



Hệ thống bao gồm :

* Smart Phone : là các smartphone chạy hệ điều hành Android có hỗ trợ GPS. Các smart phone với nhiệm vụ chính là thu thập dữ liệu thu được từ GPS gửi về cho Server thông qua các hạ tầng viễn thông như Viettel , MobilePhone.. dữ liệu được gửi về thông qua 3G hoặc GPRS.Ngoài ra các smart phone cũng có nhiệm vụ hiển thị bản đồ dưới dạng GUI cho người dùng, nhận yêu cầu từ người dùng gửi về Server. Và nhận dữ liệu trả về từ Server và hiển thị lại cho người dùng.
* Satellite : Cung cấp định vị cho thiết bị di động , và các phương tiện giao thông có trang bị GPS.
* Hạ tầng viễn thông : Các nhà cung cấp dịch vụ viễn thông với nhiệm vụ cung cấp kết nối giữa các thiết bị smartphone với Server.
* Station : Ở đây là các trung tâm xe bus, xe taxi ... hoặc các hãng vận tải. Sẽ làm nhiệm vụ cung cấp dữ liệu cho Server.
* Server: Đây là thành phần tính toán chính của hệ thống .Nhiệm vụ xử lý dự liệu nhập vào , vì dữ liệu từ nhiều nguồn khác nhau nên phải xử lý để thống nhất, chính xác.Chạy các giải thuật để dự đoán tình hình giao thông trong khoảng thời gian xác định.Tìm ra đường đi nhanh nhất dữ vào các dữ liệu đã nhận được...Đáp ứng các yêu cầu của người dùng.
* Client : Là các máy tính cá nhân thông thường.Với chức năng có thể thị tình tình giao thông , dự báo kẹt xe , tìm đường đi với thời gian ngắn nhất.

**Sơ đồ hoạt động của hệ thống :**



Phía bên trái của lược đồ là các chức năng chỉnh của hệ thống.

Phía bên phải của lược đồ là màn hình hiển thị lên các kết quả sau khi tính toán được của hệ thống.

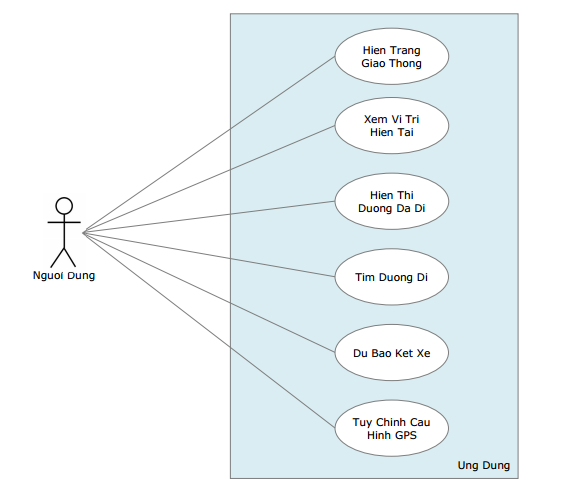
Ở giữa của lược đồ là toàn bộ quá trình xử lý của hệ thống khi nhận được các yêu cầu từ phần chức năng của hệ thống.

Các bước hoạt động các chức năng của hệ thống :

* Chức năng forecast chức năng dự đoán. Khi chức năng này được chọn. Bộ phận định vị sẽ lấy vị trí hiện tại của người dùng thông qua định vị GPS và nhà cung cấp GPS. Hệ thống sẽ nhận yêu cầu và gửi một yêu cầu dự đoán hiện trạng giao thông đến Server.Sau khi xử lý xong dữ liệu phía server, server sẽ gửi trả dữ liệu lại cho ứng dụng phía người dùng , cũng chính là module hiển thị . Sau khi Module này xử lý xong dữ liệu nhận về sẽ hiển thị lên trên màn hình cho người dùng.
* Chức năng lưu trữ đường đi của người dùng (Active GPS tracer).Đầu tiên hệ thống sẽ thực hiện việc lấy vị trí hiện tại của người dùng thông qua các hệ thống GPS được các nhà cung cấp hỗ trợ.Hệ thống sẽ nhận yêu cầu của người dùng. Sau đó thực hiện việc lưu trữ dữ liệu. Dữ liệu sẽ được lưu trữ ở hai phía cả server và client. Ở phía client dữ liệu sẽ được lưu vào các SD card.
* Chức năng hiển thị các đường đã đi qua. Về phía ứng dụng người dùng di động.Ứng dụng sẽ thực hiện yêu cầu bằng cách lấy danh sách các tracklist đã được lưu trữ trong thẻ nhớ hiển thị lên mà hình cho người dùng chọn. Sau khi người dùng chọn được đường đã đi trong danh sách. Ứng dụng sẽ hiển thị lên màn hình cho người dùng.Về phía server , sau khi người dùng chọn chức năng này, ứng dụng phía server ( ứng dụng viết trên nền web ) cũng sẽ hiển thị ra một danh sách các đường mà người dùng đã đi khi sử dụng ứng dụng trên thiết bị di động. Sau khi người dùng chọn được đường đi đã lưu. Ứng dụng sẽ hiển thị lên màn hình cho người dùng.
* Chức năng hiển thị vị trí hiện tại. Khi người dùng chọn yêu cầu hiển thị vị trí hiện tại. Hệ thống sẽ lấy vị trí người dùng thông hệ thống định vị GPS được các nhà cung cấp hỗ trợ. Sau khi nhận được vị trí hiện tại của người dùng.Ứng dụng trên web sẽ hiển thị vị trí của người dùng lên giao diện web. Ứng trên thiết bị di động sẽ hiển thị vị trí của người dùng lên thiết bị di động với lớp nền là Google Map.
* Chức năng tìm đường đi (build route) .Hệ thống có thể nhận được yêu cầu từ phía người dùng từ ứng dụng trên nền web hoặc trên thiết bị di động. Sau khi nhận được yêu cầu với chức năng tìm đường đi của người dùng. Hệ thống sẽ nhận yêu cầu tìm đường gửi lên phía server . Server sẽ xử lý yêu cầu và trả về dữ liệu hiển thị lên ứng dụng web hoặc ứng dùng trên di động tương ứng.

**ỨNG DỤNG TRÊN ANDROID:**

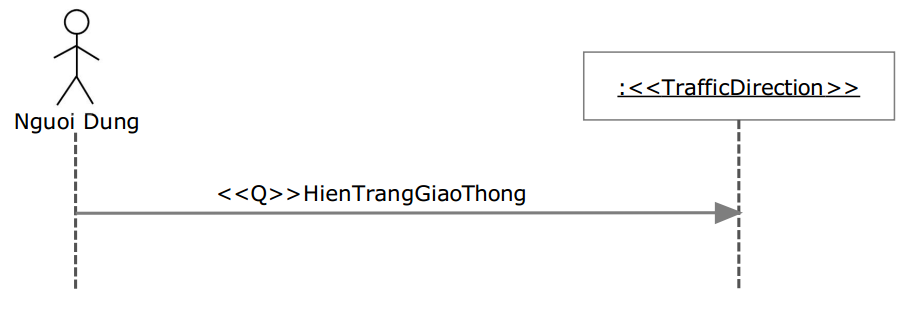
Lược đồ User-Case cho module hiển thị dữ liệu :



Các chức năng chính của ứng dụng :

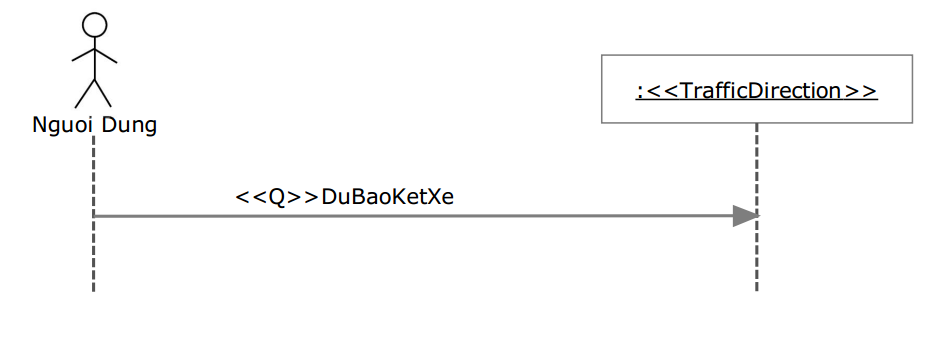
* Hiện trạng giao thông : Chức năng này để xem hiện trạng giao thông của thành phố tại một thời điểm được ghi trên bản đồ. Mục đích của tính năng này giúp người dùng có thể xác định được đâu là các điểm có lưu lượng xe tham gia giao thông đông, để tránh các điểm đó khi di chuyển.
* Dự báo tình trạng kẹt xe : Chức năng này giúp người dùng có thể biết trước được hiện trạng giao thông với một thời điểm gần trong tương lại.
* Tìm đường đi : Dựa trên hiện trạng giao thông hiện tại ,và chức năng dự báo tính trạng kẹt xe. Ứng dụng trả về cho người dùng con đường đi với thời gian ngắn nhất để đi đến được đích mong muốn.
* Xem vị trí hiện tại : Chức năng này giúp người dùng có thể biết được vị trí hiện tại của mình trên bản đồ.
* Hiển thị đường đã đi qua( tracking) : Chức năng này giúp người dùng có thể xem lại đoạn đường mình đã đi qua, với vận tốc của từng quãng đường. Từ chức năng này người dùng có thể biết được đoạn đường nào đi với tốc độ cao, đoạn đường nào với tốc độ thấp, và rút kinh nghiệm cho lần sau khi đi qua đoạn đường đó.
* Tùy chỉnh cấu hình GPS. Hệ điều hành Android chạy trên rất nhiều loại thiết bị khác nhau. Nên chất lượng của các thiết bị thu sóng từ vệ tinh cũng có độ chính xác khác nhau. Vì vậy chức năng này giúp cho người dùng tùy chỉnh lại cấu hình GPS để có thể bắt được sóng GPS chính xác nhất.Việc thu sóng chính xác là rất quan trọng , nó sẽ làm giảm thời gian xử lý dữ liệu phía Server làm cho hệ thống hoạt động hiệu quả hơn.

**Lược đồ tuần tự cho use-case hiển thị hiện trạng giao thông :**

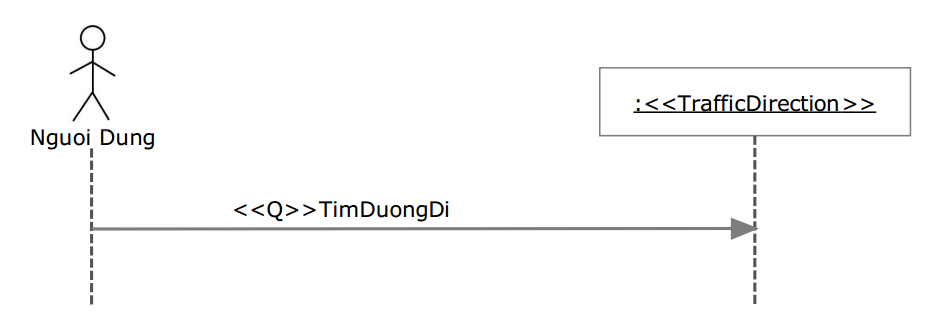


Khi người dùng chọn chức năng xem hiện trạng giao thông từ menu của ứng dụng.Ứng dụng gửi yêu cầu lên Server. Sau đó Server sẽ nhận yêu cầu xử lý và gửi dữ liệu lại cho ứng dụng.Ứng dụng sẽ hiển thị thông tin về hiện trang giao thông cho người dùng xem.

**Lược đồ tuần tự use-case dự báo kẹt xe:**

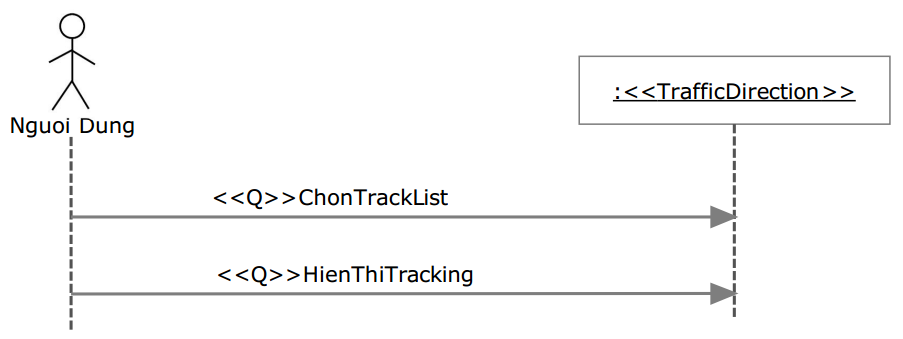


Khi người dùng chọn chức năng dự báo kẹt xe thông qua menu của ứng dụng. Ứng dụng sẽ gửi yêu cầu lên server, sau khi xử lý xong server sẽ gửi thông tin xuống cho ứng dụng.Ứng dụng sẽ làm nhiệm vụ hiển thị thông tin về hiện trạng giao thông trong một thời gian gần trong tương lai lên cho người dùng xem.Từ đó người dùng có những lựa chọn thích hợp cho lộ trình của mình.

**Lược đồ tuần tự use-case tìm đường đi:**

Khi người dùng chọn chức năng tìm đường đi.Tại menu của ứng dụng. Ứng dụng sẽ hiển thị lên menu chọn đường đi bằng cách chọn trên map. Sau đó ứng dụng sẽ gửi yêu cầu lên server. Server sẽ nhận và xử lý dữ liệu sau đó gửi lại ứng dụng.Ứng dụng hiển thị lên đường đi cho người dùng trên map.

**Lược đồ tuần tự hiển thị Tracking:**

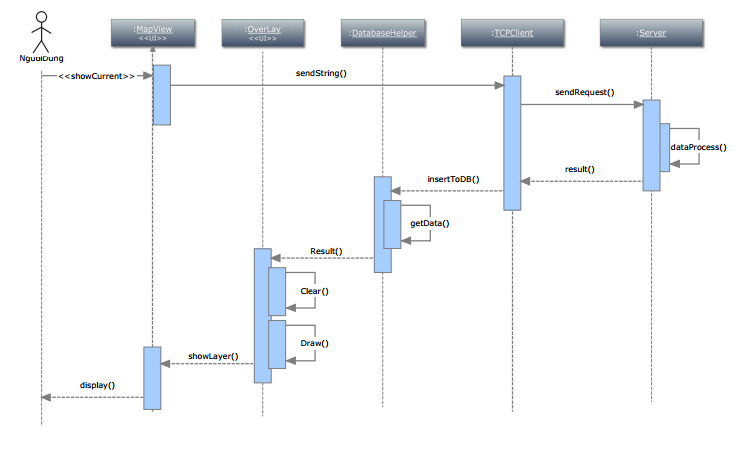


Khi người dùng chọn chức năng hiển thị tracking từ menu của ứng dụng.Ứng dụng sẽ hiển thị ra một danh sách các file tracking mà người dùng đã track lại.Người dùng sẽ chọn file muốn hiện thị.Kết quả là quãng đường đã đi qua của người dùng với với vận tốc được tính trong từng khoảng. Từ đó người dùng có thể xem lại các vận tốc các đoạn đường đã đi qua để có thể có lựa chọn tốt hơn trong lần sau đi trên đoạn đường đó.

**THIẾT KẾ ỨNG DỤNG :**

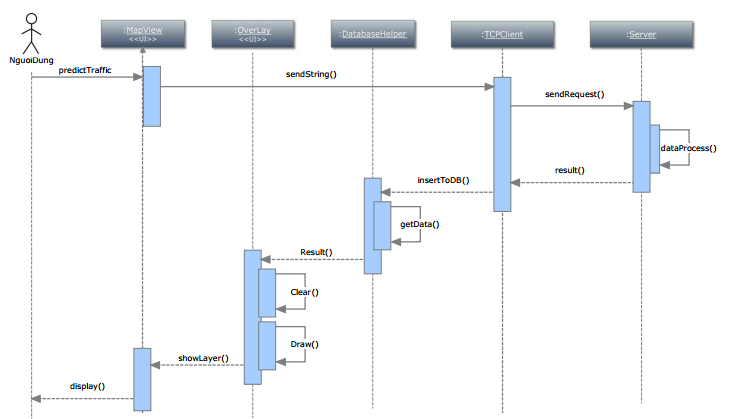
**LƯỢC ĐỒ TUẦN TỰ :**

**Chức năng hiển thị hiện trạng giao thông.**



Sau khi người dùng chọn chức năng hiện thị trạng thái hiện tại của tình trạng giao thông trong menu của lớp MapView. Yêu cầu sẽ được ghi nhận và gửi về cho lớp TCPClient . Dữ liệu sẽ được lớp TCPClient gửi về cho Server dưới dạng String. thông qua giao thức TCP/IP. Server nhận được yêu cầu sẽ xử lý và gửi kết quả lại cho ứng dụng thông qua phương thức TCP/IP. Lớp TCPClient sẽ nhận yêu cầu và xử lý dữ liệu sau đó gửi cho lớp DatabaseHelper() Lớp này có nhiệm vụ so sánh dữ liệu hiện có trong cơ sở dữ liệu với dữ liệu mới nhận được. Nếu dữ liệu mới nhận được có thay đổi so với dữ liệu trong CSDL thì dữ liệu trong CSDL sẽ được cập nhật lại. Sau khi cập nhật lại CSDL.Dữ liệu sẽ được sử lớp OverLayer truy xuất. Lớp OverLay sẽ gọi hàm Clear() để xóa toàn bộ các Layer có trước và vẽ Layer hiển thị hiện trạng giao thông bằng dữ liệu mới nhận được lên.Sau khi vẽ xong lớp OverLayer sẽ được hiển thị bên trên lớp MapView.

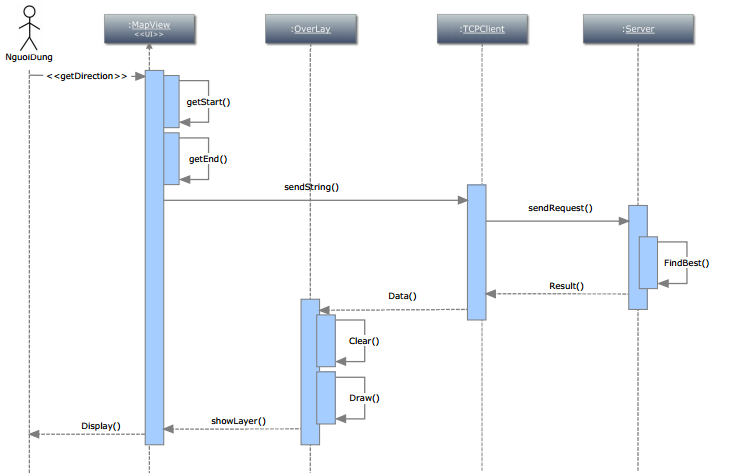
**Chức năng dự báo hiện trạng giao thông :**



Sau khi người dùng chọn chức năng dự báo hiện trạng giao thông trong một khoảng thời gian gần trong tương lai trong menu của lớp MapView. Yêu cầu sẽ được ghi nhận và gửi về cho lớp TCPClient . Dữ liệu sẽ được lớp TCPClient gửi về cho Server dưới dạng String thông qua giao thức TCP/IP. Server nhận được yêu cầu sẽ xử lý và gửi kết quả lại cho ứng dụng thông qua phương thức TCP/IP. Lớp TCPClient sẽ nhận yêu cầu và xử lý dữ liệu sau đó gửi cho lớp DatabaseHelper() Lớp này có nhiệm vụ so sánh dữ liệu hiện có trong cơ sở dữ liệu với dữ liệu mới nhận được. Nếu dữ liệu mới nhận được có thay đổi so với dữ liệu trong CSDL thì dữ liệu trong CSDL sẽ được cập nhật lại. Sau khi cập nhật lại CSDL.Dữ liệu sẽ được lớp OverLayer truy xuất. Lớp OverLay sẽ gọi hàm Clear() để xóa toàn bộ các Layer có trước và vẽ Layer hiển thị hiện trạng giao thông bằng dữ liệu mới nhận được lên.Sau khi vẽ xong lớp OverLayer sẽ được hiển thị bên trên lớp MapView.

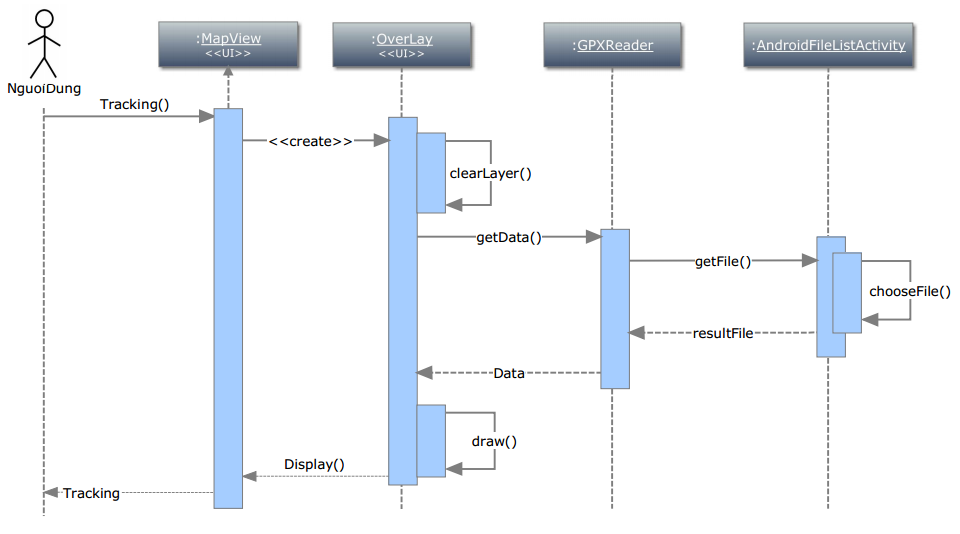
Về căn bản yêu cầu dự báo hiện trạng giao thông trong tương lai cũng giống như yêu cầu dự báo hiện trạng giao thông hiện tại nhưng có khác thời gian.

**Chức năng tìm đường.**



Khi người dùng chọn chức năng tìm đường khi nhấn nút menu. Menu hoạt động trên lớp MapView . Từ menu người dùng sẽ chọn điểm bắt đầu và điểm kế thúc . Sau đó sẽ gửi yêu cầu dạng String lại lại cho lớp TCP Client xử lý.Lớp TCPClient sẽ thực hiện gửi một yêu cầu lên trên server. Sau khi Server nhận yêu cầu . Server sẽ xử lý dữ liệu và tính toán ra đường đi với thời gian ngắn nhất , Dữ liệu tìm được sẽ được chuyển về dạng String sau đó gửi xuống lại cho Ứng dụng . Lớp TCPClient sẽ làm nhiệm vụ nhận thông tin trả về từ phí Server , sau khi xử lý dữ liệu. Dữ liệu sẽ được gửi xống lớp Overlay.Lớp Overlay sẽ thực hiện hàm Clear() với mục đích là xóa tất cả các Layer có trước đó. Sau đó sẽ vẽ đường đi dựa trên dữ liệu mới nhận được và hiển thị lên lớp MapView.

**Chức năng hiện thị tracking của người dùng.**



Người dùng chọn chức năng hiển thị tracking từ menu trong lớp MapView . Lớp MapView sẽ gửi yêu cầu tao Overlay() và xóa hết các Overlay cũ đi . Sau đó yêu cầu được lớp MapView gửi lên GPXReader() để lấy dữ liệu vẽ. Tại lớp GPXReader() sẽ gửi yêu cầu đến lớp AndroidListFilesActiviy tại lớp này người dùng sẽ chọn file Tracking trong quá khứ để hiện thị. Sau khi file được chọn xong sẽ được chuyển về cho lớp GPXReader() lớp này sẽ dùng SAX để parser dữ liệu GPX dạng XML ra dữ liệu thông thường (dữ liệu bao gồm các trường kinh độ , vĩ độ , thời gian, vận tốc) và gửi về cho lớp OverLay(). Lớp Overlay() sẽ làm nhiệm vụ nhận dữ liệu gửi về và vẽ lên OverLay(). Sau khi vẽ xong sẽ hiển thị lên MapView cho người dùng.