**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**

****

**ĐỒ ÁN TỐT NGHIỆP**

**ĐỊNH VỊ VÀ GIÁM SÁT ĐỐI TƯỢNG**

**DI ĐỘNG VỚI GPS/GPRS**

Giảng viên hướng dẫn:

**ThS. Lê Hoài Nghĩa**

**TS. Chung Tấn Lâm**

Sinh viên thực hiện:

1. **Nguyễn Nguyệt Minh – *06520292***
2. **Đào Văn Thiện – *06520450***

Lớp: **KTMT01**

Khóa: **2006**

***Tp. Hồ Chí Minh, tháng 2 năm 2011***

**LỜI MỞ ĐẦU**

Hiện nay, đất nước ta đang phát triển một cách mạnh mẽ trên mọi lĩnh vực kèm theo đó là sự phát triển của cơ sở vật chất hạ tầng mà ở đây là mạng lưới giao thông. Ở những thành phố như Hà Nội, Đà Nẵng, Tp.Hồ Chí Minh có lượng phương tiện tham gia giao thông lớn đã làm cho mạng lưới giao thông bị quá tải, việc vận chuyển hàng hóa trở nên khó khăn hơn dẫn tới giá cả bị đẩy lên cao để bù lại chi phí vận chuyển.

Để sản phẩm của mình có thể cạnh tranh được khi đưa ra thị trường, các công ty phải giảm thiểu tối đa những chi phí phát sinh trong đó việc vận chuyển và phân phối sản phẩm là một vấn đề đặc biệt quan trọng. Nếu tối ưu hóa được công đoạn này, công ty sẽ dễ dàng quản lý được thị trường của mình cũng như giảm thiểu chi phí vận chuyển kèm theo đó là giá thành sản phẩm sẽ có sức cạnh tranh hơn.

Đó cũng là mục tiêu mà đề tài tốt nghiệp: **Định vị và giám sát đối tượng di động với GPS/GPRS** hướng đến.

**Nguyễn Nguyệt Minh – MSSV: 06520292**

**Đào Văn Thiện – MSSV: 06520450**

Chúng em xin chân thành cảm ơn!

**LỜI CẢM ƠN**

**NHẬN XÉT**

**(Giảng viên hướng dẫn)**

**MỤC LỤC**

**DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH**

# Giới thiệu đề tài

1. Tổng quan

# Cơ sở lý thuyết

# 

# Nội dung

## Xây dựng phần mềm.

* 1. **Phần mềm GPSControl**

***Chức năng:***

* Quản lý đội xe.
* Xem thông tin đội xe, xe, tài xế.
* Theo dõi vị trí, tốc độ, mức nhiên liệu của xe trên bản đồ.
* Theo dõi lịch sử hoạt động của xe trong một khoảng thời gian nhất định.
* Liên lạc với xe.
* Giao diện:
* Phần mềm sử dụng cấu trúc giao diện hình cây (TreeView).
* Sử dụng các CustomPanel để hiển thị thông tin về đối tượng (Đội xe, tài xế, xe).
* ***Hiển thị bản đồ:***
  + *Yêu cầu*: Hiển thị chính xác thông tin tọa độ để từ đó xác định dễ dàng vị trí của xe. Mặt khác phải đảm bảo tốc độ hiển thị bản đồ trong khi theo dõi xe.
  + *Giải quyết vấn đề:*

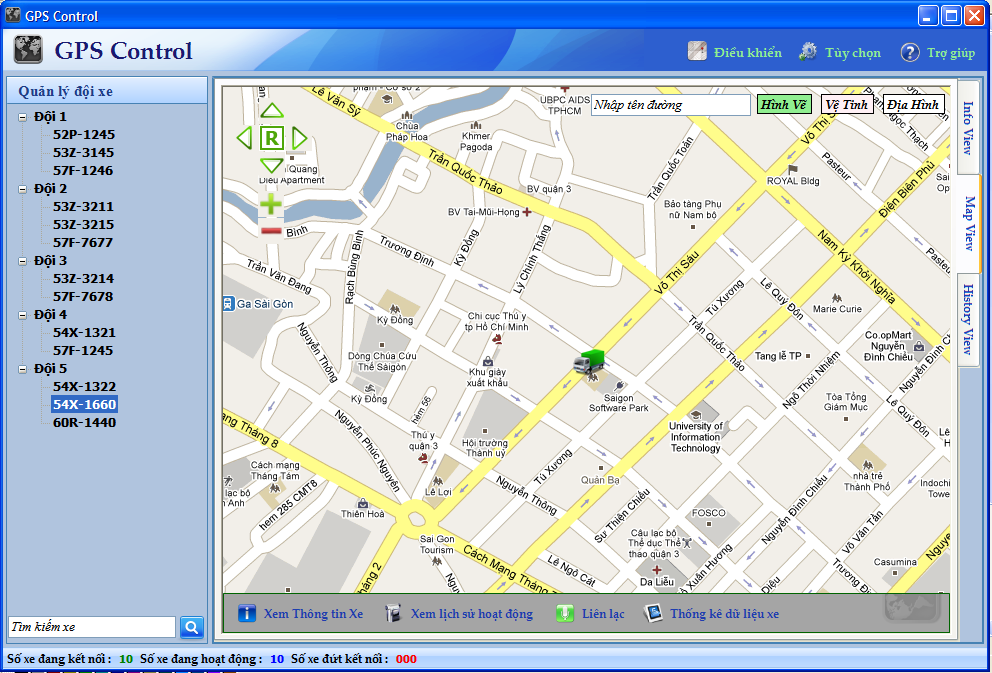
Dữ liệu bản đồ được kế thừa từ dữ liệu của Google với độ chính xác cao và hiển thị một cách trực quan.

Vì dữ liệu bản đồ rất lớn, với một máy tính cơ bản không thể hiển thị toàn bộ bản đồ cùng một lúc. Do đó, nhóm đã sử dụng phương pháp Chia Cắt Bản Đồ (Tile) thành những lát bitmap kích thước 256x256 để hiển thị. Tại một thời điểm bản đồ sẽ tự xác định các Tile cần thiết để hiển thị sao cho phù hợp với kiểu màn hình. Với phương pháp này, phần mềm cải thiện được tối đa tốc độ hiển thị của bản đồ.



Hình 1: Giao diện chương trình GPSContro

Bản đồ phải có các chức năng cơ bản như Pan, Zoom hay hiện tỉ lệ bản đồ, xem ở 3 chế độ: Bản đồ giao thông, Bản đồ Vệ Tinh, Bản Đồ Địa Hình



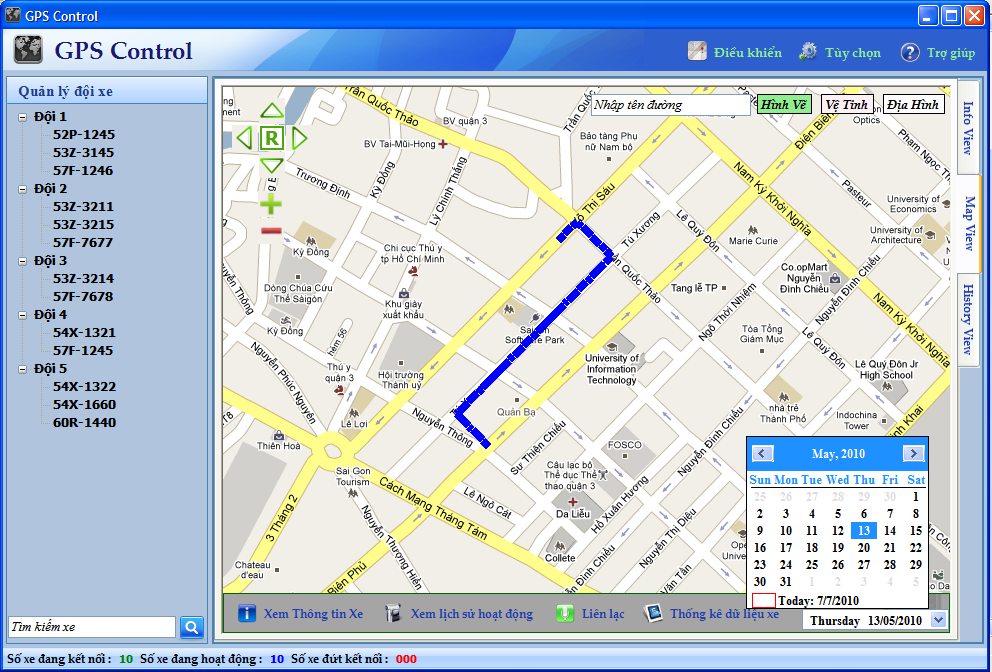
Hình 2: Giao diện Bản Đồ

Hiển thị thông tin: Hiển thị các thông tin về Đội Xe, Xe, Tài Xế giúp người sử dụng dễ dàng quản lý thông tin cũng như theo dõi trạng thái hiện tại của xe.



Hình 3: Thông tin

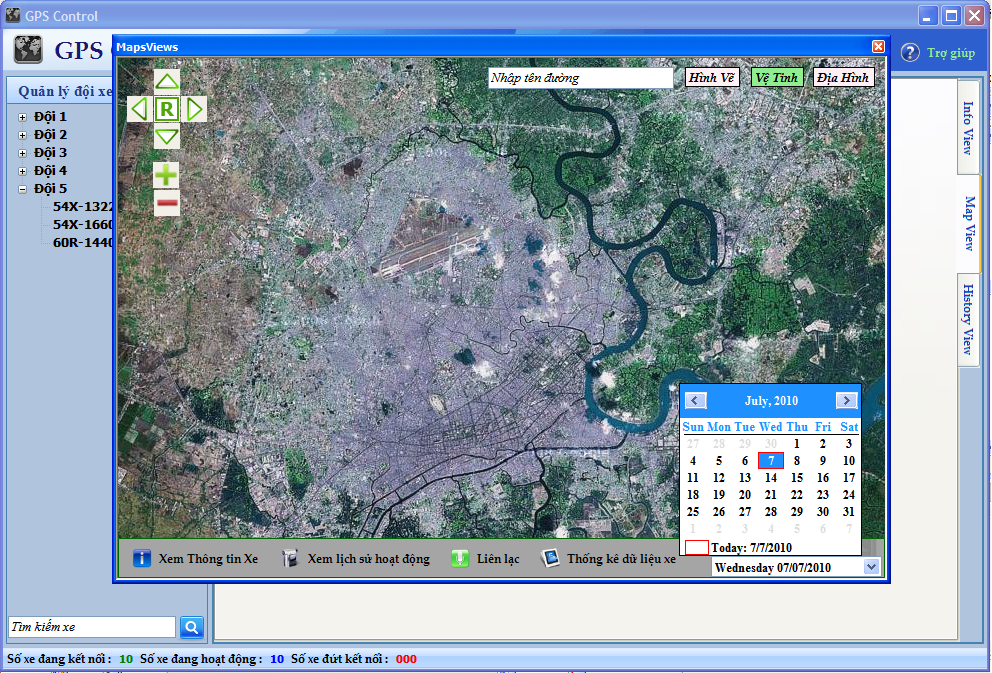
Hiển thị lịch sử hoạt động: nhằm giúp người dùng có thể quản lý hoạt động của xe trong một khoảng thời gian nhất định. Phần mềm sẽ thống kê từ những dữ liệu đã lưu trong khoảng thời gian đó bao gồm: Chi tiết đường đi, các thông tin về lịch sử giao nhận, chi tiết sử dụng nhiên liệu v.v…



Hình 4: Chi tiết đường đi theo ngày

Thống kê dữ liệu: nhằm giúp người dùng có thể quản lý hoạt động của hệ thống, của đội xe bằng biểu đồ từ đó có các tính toán hợp lý nhằm tăng chất lượng hoạt động và giảm chi phí phát sinh.

Ngoài ra, phần mềm còn có khả năng tách bản đồ thành một cửa sổ riêng nhằm tạo điều kiện để có thể theo dõi được nhiều xe hơn. Chức năng này đặc biệt hữu ích với việc sử dụng nhiều màn hình.



Hình 5: Bản đồ tách rời

* Thuật toán: **Chuyển đổi giữa tọa độ pixel và tọa độ địa lý:**

Tọa độ pixel được dựa trên các điểm pixel của các tile bitmap bản đồ. Với việc sử dụng sự kiện Mouse, chúng ta có thể xác định được tọa độ của chúng để từ đó chuyển đổi qua lại giữa 2 kiểu tọa độ này.

* Từ tọa độ pixel sang tọa độ địa lý:

// tính kinh độ

x = x + minX \* 256;

mapsize = Math.Pow(2, z) \* 256;

longdpp = 360 / mapsize;

//Tinh kinh do

lng = -180 + x \* longdpp;

return lng;

// tính vĩ độ

y = y + minY \* 256;

mapsize = Math.Pow(2, z) \* 256;

origin = mapsize / 2;

latpprad = mapsize / (Math.PI \* 2);

exp = (y - origin) / (-latpprad);

// Tinh Vi do

if (exp > Math.PI)

{

exp = Math.PI - 0.000001;

}

else if (exp < -Math.PI)

{

exp = 0.000001 - Math.PI;

}

lat = (2 \* Math.Atan(Math.Exp(exp)) - (Math.PI / 2)) \* (180 / Math.PI);

return lat;

* Từ tọa độ địa lý sang tọa độ pixel: Dùng khi xác định vị trí của xe trên bản đồ khi dữ liệu nhận về của thiết bị GPS là tọa độ địa lý

// tính tọa độ x

mapsize = Math.Pow(2, zoom) \* 256;

double xdouble = (mapsize \* (lng + 180)) / 360;

int x = (int)Math.Round(xdouble, 0) - (int)minX\*256;

return x;

// tính tọa độ y

mapsize = Math.Pow(2, zoom) \* 256;

origin = mapsize / 2.0;

latpprad = mapsize / (2.0 \* Math.PI);

exp = Math.Sin(lat \* (Math.PI / 180));

if (exp > 0.9999)

exp = 0.9999;

if (exp < -0.9999)

exp = -0.9999;

double ydouble = origin + 0.5 \* Math.Log((1 + exp) / (1 - exp)) \* (-latpprad);

int y = (int)Math.Round(ydouble, 0) - (int)minY\*256;

return y;

* 1. **Phần mềm GPSDataManager**

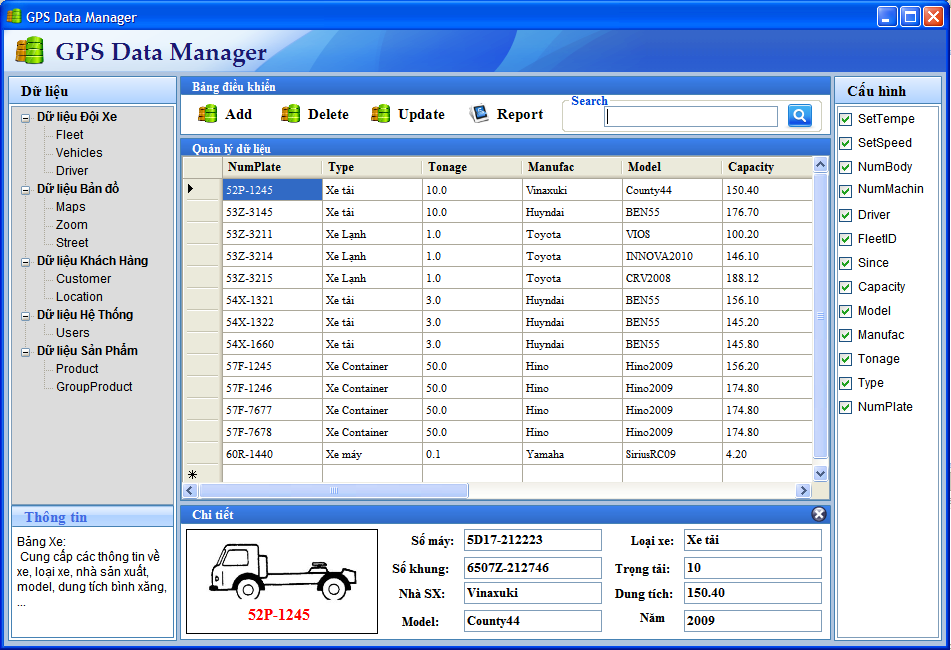
***Chức năng:***

Phần mềm được xây dựng nhằm giúp người quản lý dữ liệu có thể thao tác dễ dàng với cơ sở dữ liệu của hệ thống:

* Quản lý dữ liệu, thao tác với dữ liệu
* Thống kê dữ liệu
* Báo cáo về dữ liệu, in ấn
* Phân quyền User

***Giao diện chương trình***

* Chương trình cũng sử dụng giao diện hình cây (TreeView trong việc hiển thị dữ liệu).
* Một listbox được sử dụng để hiển thị tên của các khóa trong bảng. Người dùng có thể chọn xem một hay nhiêu khóa của một bảng.
* Phần mềm sử dụng công nghệ Report của Microsoft là Crystal Report trong việc thiết lập, xem và in báo cáo.

****

Hình 6: Giao diện GPS Data Manager

**Thuật toán**

***Thuật toán hiển thị dữ liệu trên TreeView:***

Để hiển thị dữ liệu trên TreeView, nhóm sử dụng Dataset để load dữ liệu từ server sau đó sẽ kết nối dữ liệu này và chuyển thành TreeView Item.

ví dụ: Load dữ liệu Đội xe

DataSet ds = PDataset("SELECT \* FROM Fleet");

treeView1.Nodes.Clear();

foreach (DataRow dr in ds.Tables[0].Rows)

{

if ((int)dr["FleetID"] != 0)

{

TreeNode tnParent = new TreeNode();

tnParent.Text = dr["Name"].ToString();

tnParent.Name = dr["FleetID"].ToString();

DataSet ds1 = PDataset("SELECT \* FROM Vehicles WHERE FleetID =" + tnParent.Name);

if (ds1.Tables[0].Rows.Count > 0)

{

foreach (DataRow dr1 in ds1.Tables[0].Rows)

{

TreeNode p = new TreeNode();

p.Name = dr1["NumPlate"].ToString();

p.Text = dr1["NumPlate"].ToString();

tnParent.Nodes.Add(p);

}

}

treeView1.Nodes.Add(tnParent);

}

}

* 1. **GPS Server Service**

Là một service được cài đặt trên server để lắng nghe dữ liệu gửi về từ các thiết bị GPS gắn trên xe.

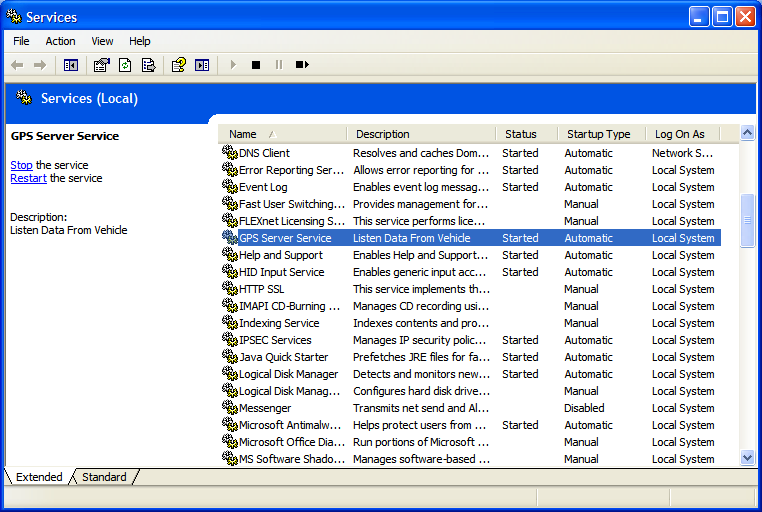
***Chức năng:***

* Lắng nghe dữ liệu từ thiết bị GPS, hạn chế tối đa sự thất thoát dữ liệu khi có nhiều dữ liệu được gửi về từ những thiết bị khác nhau.
* Phân tích dữ liệu và lưu vào Database của hệ thống
* Lưu thông tin của thiết bị GPS đã kết nối để liên lạc khi cần thiết.

***Công nghệ sử dụng:***

Service được thiết kế theo mô hình XYNetServer với việc áp dụng lập trình Thread (đa luồng) trong việc quản lý kết nối nhằm tránh tình trạng tắc nghẽn khi có quá nhiều thiết bị GPS cùng gửi dữ liệu về Server.

Server kết nối với thiết bị qua giao thức TCP/IP và Service sẽ lắng nghe dữ liệu từ những port TCP/IP bất kì.



Hình 7: GPS Server Service

* 1. **Cơ sở dữ liệu**

Được xây dựng trên cơ sở yêu cầu thực tiễn. Các bảng dữ liệu có sự liên kết chặt chẽ với nhau nhằm tạo sự dễ dàng cho việc quản lý.

***Chức năng:***

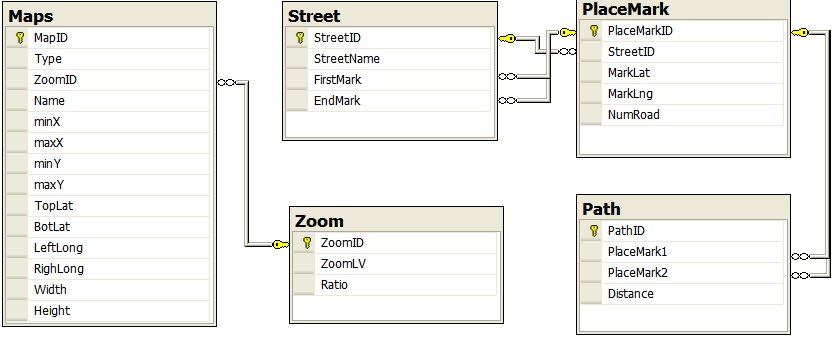
Lưu trữ dữ liệu của chương trình

***Mô hình thiết kế:***

*Dữ liệu bản đồ*

Là nơi lưu trữ những giá trị tọa độ, những điểm giao cắt, các địa danh, mức độ phóng đại và tỉ lệ của bản đồ….

Là dữ liệu chính cho phần mềm Bản Đồ truy xuất trong việc hiển thị và đánh dấu bản đồ

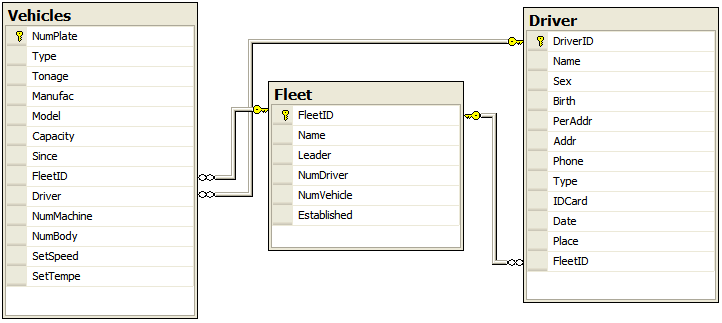


Hình 8: Dữ liệu bản đồ

*Dữ liệu Đội Xe*

Lưu trữ mọi thông tin về Đội Xe, Xe, Tài Xế, Đội trưởng.

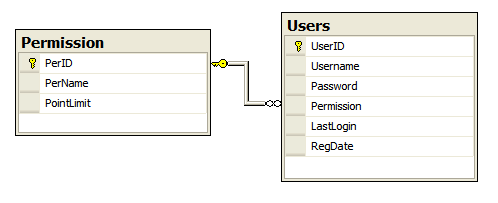
Mọi thao tác với Quản lý Đội Xe sẽ được truy xuất tới Dữ liệu này.



Hình 9: Dữ liệu Đội Xe

*Dữ liệu Hệ Thống*

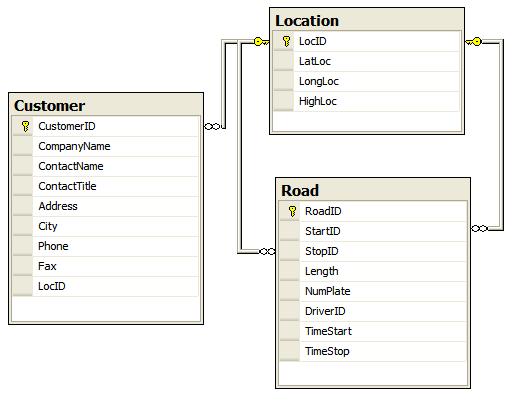
Lưu trữ những thông tin của hệ thống bao gồm các User và phân quyền của chúng. Nó giúp cho người quản trị quản lý dễ dàng các User cũng như hạn chế các thao tác quá sâu vào hệ thống



Hình 10: Dữ liệu hệ thống

*Dữ liệu Khách Hàng*

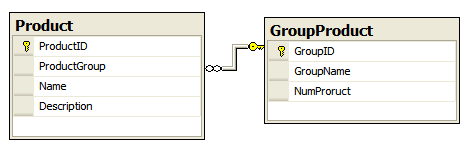
Lưu trữ những thông tin cơ bản của khách hàng: địa chỉ, tọa độ, tên khách hàng. Vị trí của khách hàng trên bản đồ và đường đi ngắn nhất để tới được vị trí đó. Bên cạnh đó là dữ liệu sản phẩm mà khách hàng vận chuyển nhiều nhất.



Hình 11: Dữ liệu Khách Hàng

*Dữ liệu Sản Phẩm*

Lưu trữ dữ liệu về hệ thống sản phẩm của công ty bao gồm tất cả các loại sản phẩm, phân loại sản phẩm và đặc điểm của mỗi nhóm sản phẩm đó.



Hình 12: Dữ liệu Sản Phẩm

***Kết nối phần mềm với cơ sở dữ liệu:***

* Sử dụng công nghệ ADO.NET giúp việc quản lý dữ liệu và kết nối trở nên dễ dàng hơn.
* Hiển thị và thao tác trực tiếp với dữ liệu thông qua việc sử dụng DataGridView.
* Kết nối và tính toán các giá trị sử dụng trong phần mềm sử dụng dataset:

public SqlConnection connect =

new SqlConnection(@"Data source=VANTHIEN\SQLEXPRESS;Initial Catalog=GPSControlDatabase;Integrated Security=True");

public string query = null;

connect.Open();

query = "SELECT Name FROM Maps WHERE Type = 'gm' AND ZoomID =

SELECT ZoomID FROM Zoom WHERE ZoomLV = (SELECT min(ZoomLV) FROM Zoom))";

SqlCommand cmd = new SqlCommand(query, connect);

q = cmd.ExecuteScalar().ToString();

## Tìm hiểu cấu trúc phần cứng của thiết bị GPS/GPRS trên xe tải

1. **Chức năng của thiết bị:**

* Thiết bị có nhiệm vụ xác định tọa độ của xe bằng GPS, xác định các thông số khác của xe qua các cảm biến gắn trên xe (cảm biến nhiệt độ, tốc độ, cảm biến đo mức nhiên liệu…)
* Thiết bị kết nối và truyền nhận dữ liệu với Server qua giao thức TCP/IP sử dụng công nghệ GPRS cho mạng di động.

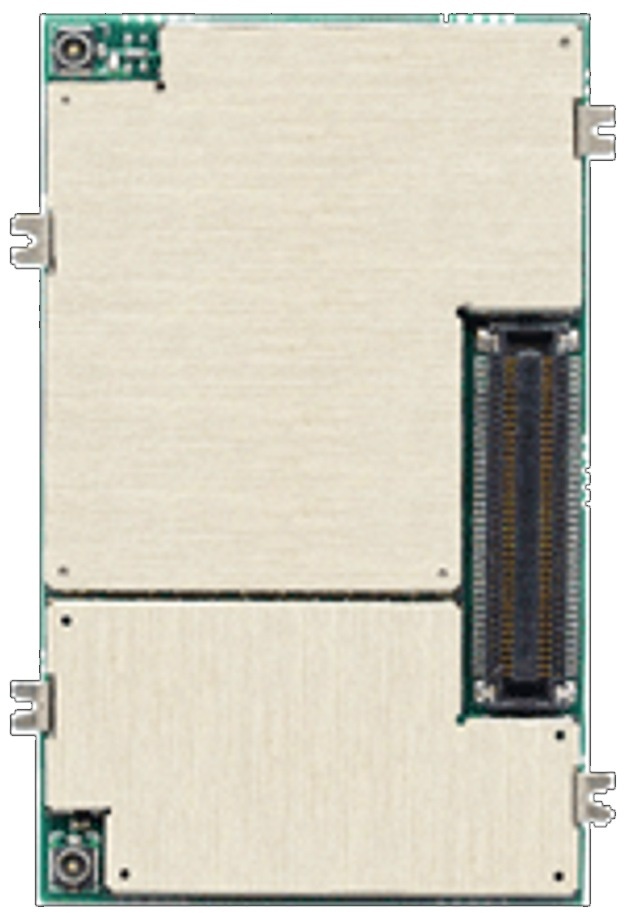
1. **Linh kiện chính của thiết bị:**

Chip SIM548 – Chip chuyên dụng GPS/GPRS

Là sản phẩm được thiết kế bởi SIMCOM tích hợp sẵn 2 module GPS và GPRS.

*Những tính năng chính:*

* Kích thước: 34mm x 55mm x 3 mm, nặng 12g
* Hoạt động trong -20oC đến +55oC
* Hỗ trợ chuẩn GPRS class 10, tốc độ tối đa 85,6Kbps
* Băng tần GSM/GPRS 850/900/1800/1900MHz
* Hoạt động với điện thế 3,4 – 4,5 V
* Tích hợp TCP/IP
* Được gắn sẵn 2 ăngten riêng biệt cho GSM/GPRS và GPS.
* Hỗ trợ giao diện Dual – Serial cho GPS
* GPS với kiểu tọa độ WGS-84
* Hỗ trợ FAX, SMS, Voice
* Hỗ trợ PBCCH
* Đóng gói với 80 chân.
* Bộ nhớ bao gồm 4Mbit Flash và 1Mbit SDRAM



Hình 13: SIM548

## Tìm hiểu thuật toán tìm đường đi ngắn nhất:

**Bài toán tìm đường đi ngắn nhất**

Bài toán tìm đường đi ngắn nhất là bài toán quan trọng trong Lý thuyết đồ thị, nó được áp dụng để giải quyết rất nhiều bài toán trong thực tế như điều khiển tối ưu, giao thông vận tải, mạng viễn thông ...

Bài toán này có thể chia làm 2 loại:

*Tìm đường đi ngắn nhất giữa một cặp đỉnh:* Cho đồ thị G(V,E) có trọng số cạnh và hai đỉnh u, v thuộc V tìm đường đi ngắn nhất từ đỉnh u đến đỉnh v trên đồ thị G. Các giải thuật được phát triển để giải bài toán dạng này tiêu biểu là các giải thuật: Dijkstra, Bellman-Ford,...

*Tìm đường đi ngắn nhất giữa tất cả các cặp đỉnh:* Cho đồ thị G(V,E) có trọng số cạnh tìm đường đi từ đỉnh u đến đỉnh v, với mọi cặp đỉnh u, v thuộc V. Các giải thuật đã được phát triển để giải bài toán này là: Floyd-Warshall, Johnson,...

Trong thực tế nhiều khi ta không chỉ cần tìm đường đi ngắn nhất giữa hai đỉnh mà còn cần xác định đường đi ngắn nhất giữa một tập đỉnh này đến một tập đỉnh khác. Bài toán đó được phát biểu như sau: Cho đồ thị G(V,E) có trọng số cạnh và hai tập đỉnh A,B ⊂ V tìm đường đi ngắn nhất từ tập đỉnh A đến tập đỉnh B.

Sau đây, nhóm xin giới thiệu một số thuật toán mà nhóm đã nghiên cứu.

1. **Bài toán Đường đi ngắn nhất**

**Bài toán:** Cho đồ thị G = (V, E) và hai đỉnh a, b. Tìm đường đi ngắn nhất (nếu có) đi từ đỉnh a đến đỉnh b trong đồ thị G.

ý nghĩa thực tế: Bài toán này giúp chúng ta chọn các hành trình tiết kiệm nhất

(quãng đường, thời gian, chi phí ...) trong giao thông, lập lịch thi công các công

trình một cách tối ưu, xử lý trong truyền tin ...

Thuật toán duyệt đồ thị theo chiều rộng đã cho ta lời giải của bài toán này.

Song ta có thêm thuật toán sau đây.

***Thuật toán 1:***

*Bước 1*: Lần lượt gán nhãn cho các đỉnh của đồ thị, mỗi đỉnh không quá một lần, như sau:

- Đỉnh a được gán nhãn là số 0.

- Những đỉnh kề với đỉnh a được gán số 1.

- Những đỉnh kề với đỉnh đã được gán nhãn số 1, được gán số 2. ………………………………….

- Tương tự, những đỉnh kề với đỉnh đã được gán số i được gán nhãn là số

i+1.

………………………………….

Thực hiện cho đến khi gán được nhãn cho đỉnh b hoặc không gán nhãn

được nữa.

*Bước 2*: Nếu đỉnh b được gán nhãn nào đó là k thì kết luận có đường đi ngắn nhất từ đỉnh a tới đỉnh b với độ dài k, ngược lại thì trả lời là không có.

*Bước 3*: Khôi phục đường đi: Nếu ở bước 2. chỉ ra b được gán nhãn k nào đó thì ta đi ngược lại theo quy tắc sau đây: Nếu đỉnh y được gán nhãn j với j ≥ 1 thì sẽ có đỉnh x được gãn nhãn j-1 sao cho có cạnh đi từ x tới y. Đi ngược lại cho đến khi gặp đỉnh a, ta nhận được đường đi ngắn nhất cần tìm.

1. **Thuật toán Dijkstra tìm đường đi ngắn nhất**

Năm 1959 E. W. Dijkstra đưa ra một thuật toán rất hiệu quả để giải bài toán đường đi ngắn nhất.

Thuật toán thực hiện việc gán và giảm giá trị của nhãn l(i) tại mỗi đỉnh i của đồ thị G như sau:

*Thuật toán 2: (Tìm đường đi ngắn nhất):*

Bước 1. Với đỉnh xuất phát a, gán nhãn l(a) := 0.

Bước 2. Nếu có cạnh (i,j) mà đỉnh i đã được gán nhãn và đỉnh j chưa được gán nhãn hoặc đỉnh j đã được gán nhãn nhưng l(i) + c(i,j) < l(j) thì giảm nhãn l(j) := l(i) + c(i,j).

Bước 3. Lặp lại bước 2. cho đến khi không gán hoặc giảm nhãn được nữa.

1. **Thuật toán Floyd**

Ta sử dụng ma trận Dn x n để tính độ dài đường đi ngắn nhất giữa tất cả các

cặp đỉnh.

Bước 1. Bắt đầu gán D := C - ma trận trọng số.

Bước 2. Thực hiện n lần lặp trên D. Sau bước lặp thứ k, D[i,j] chứa độ dài đường đi ngắn nhất từ đỉnh i đến đỉnh j mà chỉ đi qua các đỉnh có chỉ số không vượt quá k. Vậy trong bước lặp thứ k ta thực hiện theo công thức sau đây:



với k = 1, 2, ... , n.

# Kết quả đạt được qua đợt thực tập

Trong suốt quá trình thực tập, dựa trên những vấn đề và nhiệm vụ thực tiễn, nhóm đã được củng cố lại những kiến thức lý thuyết, được tiếp cận với một môi trường làm việc thực sự chuyên nghiệp với nhiều kỹ năng mới. Qua đó tích lũy cho mình những kinh nghiệm bổ ích cho công việc sau này.

## Những lý thuyết được củng cố:

* Lập trình Windows Form với C#
* Lập trình Windows Service
* Lập trình nhúng cho vi điều khiển PIC
* Thiết kế mạch với OrCad
* Lý thuyết đồ thị và những ứng dụng trong việc tìm đường đi ngắn nhất
* Lập trình đa luồng (Multi Thread) với C# .Net 3.5

## Những kỹ năng thực hành đã được học thêm:

* Kỹ năng xây dựng UserControl & Custom Control trong lập trình giao diện với C#
* Lập trình lắng nghe dữ liệu qua giao thức TCP/IP sử dụng Thread.
* Kỹ năng làm việc nhóm được củng cố.
* Kỹ năng phân tích và lập kế hoạch làm việc
* Kỹ năng thiết kế mạch in với OrCad
* Kỹ năng thử nghiệm một sản phẩm hay một hệ thống

## Những kinh nghiệm thực tiễn đã tích lũy được:

* Kinh nghiệm lập trình đồ họa: thay vì sử dụng đồ họa bitmap thông thường, việc sử dụng GDI+ và DirectX làm chương trình trở nên mạnh mẽ và tối ưu hơn
* Kinh nghiệm tối ưu code trong lập trình phần mềm và lập trình nhúng
* Kinh nghiệm phân tích vấn đề: cần xác định rõ từng bước trong quá trình thực hiện đề tài, từ đó chia nhỏ những vấn đề để giải quyết một cách dễ dàng hơn.
* Kinh nghiệm họp và báo cáo nhiệm vụ: không chỉ tập trung vào vấn đề của mình, cần lắng nghe những vấn đề của người khác và cùng nhau đưa ra ý tưởng giải quyết vấn đề.
* Kinh nghiệm nghiên cứu đề tài: Cần tìm hiểu rõ tình hình phát triển của đề tài ở trong và ngoài nước. Những vấn đề mà những đề tài trước chưa giải quyết được từ đó chọn hướng đi phù hợp cho việc phát triển đề tài.