MỤC LỤC

[MỤC LỤC 1](#_Toc312242859)

[**I. GIỚI THIỆU ĐỀ TÀI** 1](#_Toc312242860)

[**1. Nội dung đề tài** 2](#_Toc312242861)

[**2. Lý do chọn đề tài:** 2](#_Toc312242862)

[**II. PHÂN TÍCH HỆ THỐNG** 4](#_Toc312242863)

[**1. Kiến trúc thệ thống:** 4](#_Toc312242864)

[**2. Class diagram:** 7](#_Toc312242865)

[**3. Giải thuật:** 9](#_Toc312242866)

[a. Truy xuất lộ trình của một tuyến xe bus: 9](#_Toc312242867)

[b. Cho biết các tuyến xe bus đi qua một điểm nào đó: 10](#_Toc312242868)

[c. Tìm tuyến xe bus giữa hai điểm A và B: 10](#_Toc312242869)

[**III. CÔNG NGHỆ** 11](#_Toc312242870)

[**1. Cơ sở dữ liệu:** 11](#_Toc312242871)

[**2. Công nghệ dùng thiết kế hệ thống website:** 11](#_Toc312242872)

[**3. Công nghệ dùng thiết kế trên mobile:** 13](#_Toc312242873)

[**IV. CHƯƠNG TRÌNH:** 15](#_Toc312242874)

[1. **Website**: Gồm 2 phần 15](#_Toc312242875)

[A. **Admin**: 15](#_Toc312242876)

[B. **Client**: 20](#_Toc312242877)

[**2.** **Mobile:** 23](#_Toc312242878)

[**V. KẾT LUẬN** 24](#_Toc312242879)

[1. **Tổng kết**: 24](#_Toc312242880)

[2. **Hướng phát triển**: 24](#_Toc312242881)

[**VI. TÀI LIỆU THAM KHẢO :** 25](#_Toc312242882)

1. **GIỚI THIỆU ĐỀ TÀI**
   1. **Nội dung đề tài**

Xây dựng một website và phần mềm trên mobile hỗ trợ cho việc tìm kiếm thông tin các tuyến xe bus. Các tính năng bao gồm:

1. Truy xuất lộ trình của một tuyến xe bus
2. Cho biết các tuyến xe bus đi qua một điểm nào đó
3. Tìm tuyến xe bus giữa hai điểm A và B.

Dự án được thực hiện thông qua những phase sau:

**Phase 1**: Xây dựng một website có đầy đủ những tính năng như trên

**Phase 2**: Cung cấp Web Services và lập trình một ứng dụng trên mobile (sử dụng Web Services đó) để có thể mở rộng đối tượng sử dụng

**Phase 3** (Optional): Hiện thực những tính năng nâng cao như:

* Thêm thông tin về khoảng cách, chi phí và thời gian ước tính cho các tuyến.
* Đưa ra lộ trình tối ưu nhất cho tính năng(c).
* Khi không có tuyến đi qua 2 điểm A và B (trong tính năng c) thì phải tìm lộ trình thay thế (như đi 2, 3 tuyến khác nhau) tốt nhất.
  1. **Lý do chọn đề tài:**

Giao thông vốn là một vấn đề rất được quan tâm trong các thành phố lớn. Nhất là trong tình trạng hiện nay ở nước ta nói chung, thành phố Hồ Chí Minh nói riêng, tình hình giao thông đang diễn ra ngày càng phức tạp, nạn kẹt xe trong các giờ cao điểm diễn ra như một căn bệnh cấp tính. Việc phổ biến thông tin và hướng dẫn người dân tìm đường đi thích hợp với các phương tiện giao thông công cộng đang là một việc làm cần thiết. Với điều kiện cụ thể ở nước ta hiện nay, xe buýt sẽ là một giải pháp giúp cho nạn kẹt xe giảm xuống. Bên cạnh việc phải nâng cao chất lượng của xe buýt thì chúng ta cũng cần phải nâng cao các dịch vụ liên quan cũng như các thông tin tra cứu hỗ trợ cho người sử dụng về phương tiện này.

Việc tra cứu thông tin và đặc biệt là tìm các tuyến thích hợp để đi lại bằng bản đồ giấy thường rất phức tạp, tốn thời gian và sẽ càng khó khăn hơn khi giữa các tuyến đi không liên thông với nhau giữa điểm xuất phát và điểm đích hay khi hệ thống các tuyến trở nên ngày một chằng chịt, một xu hướng ắt sẽ diễn ra trong tương lai không xa. Tuy nhiên với sự trợ giúp của công nghệ máy tính thì công việc đó sẽ trở nên dễ dàng và tiện lợi. Với ý tưởng đó, chúng tôi đã tiến hành tìm hiểu và hiện thực một hệ thống hỗ trợ cho người dùng có thể truy xuất nhanh và dễ dàng các thông tin về tuyến xe buýt, trạm xe buýt và đường đi giữa hai trạm bất kỳ…

1. **PHÂN TÍCH HỆ THỐNG**
   1. **Kiến trúc thệ thống:**

Client

Client

Admin

Mobile

Website

Brower

Web Server

Mobile Service

UI Components

UI Process Components

Business Component

Application Facade

Business Entities

Data Helper/ Utilities

Data Access Components

Data Sources

Kiến trúc hệ thống của chương trình chúng tôi được xây dựng theo như hình vẽ ở trên. Đó là mô hình 3 lớp phổ biến nhất hiện nay

* **Lớp Presentation**:

Lớp này làm nhiệm vụ giao tiếp với người dùng cuối để thu thập dữ liệu và hiển thị kết quả/dữ liệu thông qua các thành phần trong giao diện người sử dụng. Lớp này sẽ sử dụng các dịch vụ do lớp Business Logic cung cấp. Trong .NET thì có thể dùng Windows Forms, ASP.NET hay Mobile Forms để hiện thực lớp này.

Trong lớp này có 2 thành phần chính là *User Interface Components và User Interface Process Components*.

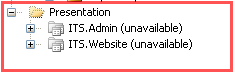
* *User Interface Components***:**

Là những phần tử chịu trách nhiệm thu thập và hiển thị thông tin cho người dùng cuối. Trong ASP.NET thì những thành phần này có thể là các TextBox, các Button, DataGrid…

* *User Interface Process Components***:**

Là thành phần chịu trách nhiệm quản lý các qui trình chuyển đổi giữa các UI Components. Ví dụ chịu trách nhiệm quản lý các màn hình nhập dữ liệu trong một loạt các thao tác định trước như các bước trong một Wizard…

Trong phần hiện thực có 2 thực thể thuộc lớp này:



* **Lớp Business Logic**:

Lớp này thực hiện các nghiệp vụ chính của hệ thống, sử dụng các dịch vụ do lớp **Data Access** cung cấp, và cung cấp các dịch vụ cho lớp **Presentation.**

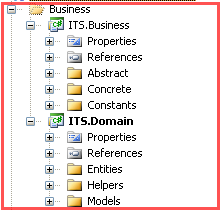
Trong lớp này có các thành phần chính là *Business Components, Business Entities*

* *Business Components***:**

Là những thực thể mô tả những đối tượng thông tin mà hệ thống xử lý. Các business entities này cũng được dùng để trao đổi thông tin giữa lớp **Presentation**và lớp **Data Access.**

* *Business Entities***:**

Là những thành phần chính thực hiện các dịch vụ mà Service Interface cung cấp, chịu trách nhiệm kiểm tra các ràng buộc logic(constraints), các qui tắc nghiệp vụ(business rules), sử dụng các dịch vụ bên ngoài khác để thực hiện các yêu cầu của ứng dụng.



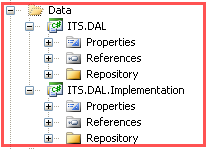
* **Lớp Data Access**:

Lớp này thực hiện các nghiệp vụ liên quan đến lưu trữ và truy xuất dữ liệu của ứng dụng. Lớp này sẽ sử dụng các dịch vụ của các hệ quản trị cơ sở dữ liệu PostgreSQL. Trong lớp này có các thành phần chính là *Data Access Logic Component, Data Helper.*

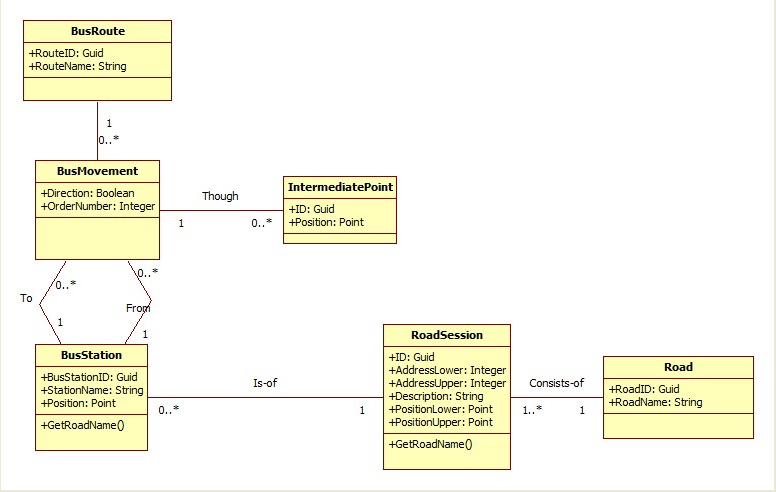
* *Data Access Logic Component (DALC)***:**

Là thành phần chính chịu trách nhiệm lưu trữ vào và truy xuất dữ liệu từ các nguồn dữ liệu – Data Sources như RDMBS, XML, File systems…. Trong .NET Các **DALC**này thường được hiện thực bằng cách sử dụng thư viện ADO.NET để giao tiếp với các hệ cơ sở dữ liệu hoặc sử dụng các O/R Mapping Frameworks để thực hiện việc ánh xạ các đối tượng trong bộ nhớ thành dữ liệu lưu trữ trong CSDL. Chúng ta sẽ tìm hiểu các thư viện O/R Mapping này trong một bài viết khác.

* *Data Helper***:**



* 1. **Class diagram:**



* Danh sách các đối tượng dữ liệu:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên đối tượng | Phân loại | Diễn giải |
| 1 | BusRoute | Thực thể | Tuyến xe buýt |
| 2 | BusMovement | Quan hệ | Lộ trình của tuyến xe |
| 3 | IntermediatePoint | Thực thể | Điểm trung gian |
| 4 | BusStation | Thực thể | Trạm xe buýt |
| 5 | RoadSession | Thực thể | Đoạn đường |
| 6 | Road | Thực thể | Con đường |

* Mô tả chi tiết từng thành phần:
* Tuyến xe buýt (BusRoute)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Diễn giải | Kiểu | Ràng buộc |
| 1 | RouteID | Mã tuyến xe | Số nguyên | Khóa chính |
| 2 | RouteName | Tên tuyến xe | Chuỗi |  |

* Lộ trình của tuyến xe (BusMovement)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Diễn giải | Kiểu | Ràng buộc |
| 1 | ID | Mã lộ trình | Số nguyên | Khóa chính |
| 2 | RouteID | Mã tuyến xe | Số nguyên | Khóa ngoại |
| 3 | BusStationFrom | Mã trạm xe buýt | Số nguyên | Khóa ngoại |
| 4 | BusStationTo | Mã trạm xe buýt | Số nguyên | Khóa ngoại |
| 5 | Direction | Hướng đi | boolean |  |
| 6 | OrderNumber |  |  |  |

* Điểm trung gian (IntermediatePoint)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Diễn giải | Kiểu | Ràng buộc |
| 1 | ID | Mã điểm trung gian | Số nguyên | Khóa chính |
| 2 | Position | Vị trí | Kiểu điểm |  |

* Trạm xe buýt (BusStation)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Diễn giải | Kiểu | Ràng buộc |
| 1 | BusStationID | Mã trạm | Số nguyên | Khóa chính |
| 2 | StationName | Tên trạm | Chuỗi |  |
| 3 | Position | Vị trí | Kiểu điểm |  |
| 4 | GetRoadName() | Hàm lấy tên đường |  |  |

* Đoạn đường (RoadSession)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Diễn giải | Kiểu | Ràng buộc |
| 1 | ID | Mã lộ trình | Số nguyên | Khóa chính |
| 2 | AddressLower | Địa chỉ thấp | Chuỗi |  |
| 3 | AddressUpper | Địa chỉ cao | Chuỗi |  |
| 4 | Description | Miêu tả thêm | Số nguyên |  |
| 5 | PointLower | Vị trí thấp | Kiểu điểm |  |
| 6 | PointUpper | Vị trí cao | Kiểu điểm |  |
| 7 | GetRoadName() | Hàm lấy tên đường |  |  |

* Con đường (Road)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Diễn giải | Kiểu | Ràng buộc |
| 1 | RoadID | Mã đường | Số nguyên | Khóa chính |
| 2 | RoadName | Tên đường | Chuỗi |  |

* 1. **Giải thuật:**

Với thiết kế cơ sở dữ liệu như lược đồ Lớp ở trên thì nhóm chúng tôi có thể giải quyết được những yêu cầu của đề tài đặt ra.

1. Truy xuất lộ trình của một tuyến xe bus:

Với bảng BusMovement ta có thể truy xuất lộ trình đi qua các trạm của một tuyến xe bất kỳ như sau:

SELECT BusStationFrom, BusStationTo

FROM BusMovement

WHERE RouteID = (SELECT RouteID

FROM BusRoute

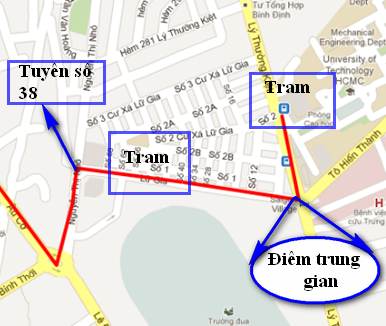
WHERE RouteName = 'số 38')

AND Direction = 0 / / 0 chiều đi, 1 chiều ngược lại

Dùng một danh sách để chứa các trạm trả về gồm :

BusStationFrom\_1 … BusStationFrom\_n + BusStationTo\_n

Với từng BusStationID có thể truy xuất thông tin tên của trạm tương ứng để hiển thị. Đối với việc hiển thị thì cần thêm các điểm trung gian như ví dụ sau:



1. Cho biết các tuyến xe bus đi qua một điểm nào đó:

Input: người dùng sẽ nhập vào một địa chỉ trên một con đường nào đó. Giải thuật của chúng tôi sẽ làm từng bước như sau:

* Trả về thông tin của con đường trùng với tên con đường người nhập vào.
* Trả về thông tin đoạn đường thuộc con đường vừa tìm mà chứa địa chỉ của người nhập : điạ chỉ và tọa độ điểm đầu , địa chỉ và tọa độ điểm cuối của đoạn đường -> tìm được tọa độ của điểm vừa nhập.
* Với tọa độ vừa tìm được, có thể dùng hàm GetNearestStations(Point p,int count) để tìm trạm xe buýt gần nhất.
* Với mã trạm xe buýt vừa tìm được có thể liệt kê những tuyến xe đi qua.

1. Tìm tuyến xe bus giữa hai điểm A và B:

Input: Người dùng nhập vào 2 điểm

* Điểm đầu: địa chỉ và tên đường
* Điểm cuối: địa chỉ và tên đường

Giải thuật từng bước:

: Điểm đầu

: Điểm cuối

: quét tìm 2 trạm gần nhất trong bán kính 500 m

: Trạm xe buýt

: Đoạn đường mà người nhập phải đi bộ

: với kết nối này cần tìm tuyến xe đi qua

Như vậy với hình vẽ trên chúng tôi đã trình bày giải thuật tìm 1 tuyến xe đi qua hai điểm do người dùng nhập vào.

1. **CÔNG NGHỆ**
   1. **Cơ sở dữ liệu:**

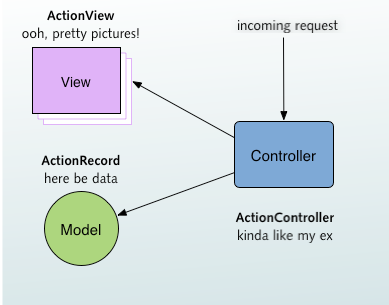
Hệ thống truy xuất thông tin xe bus là một hệ thống thông tin địa lý (GIS). Trong quá trình lưu trữ và xử lý dữ liệu chắc chắn phải cần đến những tính năng liên quan đến vị trí, khoảng cách...

PostGIS là một extension của hệ quản trị cơ sở dữ liệu quan hệ - đối tượng PostgreSQL cho phép lưu trữ các đối tượng GIS. PostGIS hỗ trợ các spatial index GiST-based R-Tree và các function cho việc phân tích và xử lý các đối tượng GIS.

Trong số một vài lựa chọn khác như spatial extension của Oracl, SDE, Microsoft SQL Server 2008 R2, thì PostGIS/PostgreSQL đã được chúng tôi lựa chọn vì tính năng mở của nó - là một opensource.

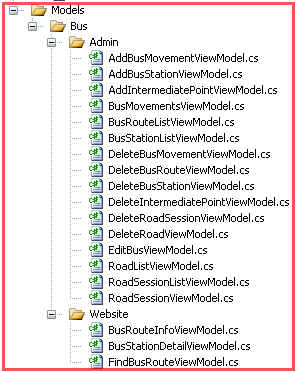
* 1. **Công nghệ dùng thiết kế hệ thống website:**

Để đảm bảo cho việc hiện thực và phát triển hệ thống một cách dễ dàng, dễ đọc và bảo trì và mang tính chuyên nghiệp. Chúng tôi sẽ hiện thực hệ thống theo kiến trúc Model-View-Controller(MVC).

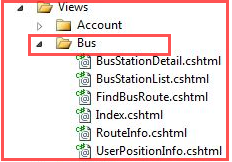


Thật vậy MVC là một phương pháp chia nhỏ một ứng dụng thành ba thành phần để cài đặt, mỗi thành phần đóng một vai trò khác nhau và ảnh hưởng lẫn nhau, đó là models, views, và controllers.

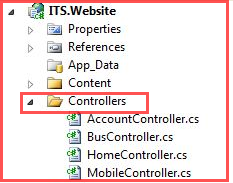
* **Models**: trong các ứng dụng dựa trên MVC là những thành phần có nhiệm vụ lưu trữ thông tin, trạng thái của các đối tượng, thông thường nó là một lớp được ánh xạ từ một bảng trong CSDL.

****

* **Views**: nó chính là các thành phần chịu trách nhiệm hiển thị các thông tin lên cho người dùng thông qua giao diện. Thông thường, các thông tin cần hiển thị được lấy từ thành phần **Models**.

****

* **Controllers**: trong các ứng dụng kiểu MVC chịu trách nhiệm xử lý các tác động về mặt giao diện, các thao tác đối với models, và cuối cùng là chọn một view thích hợp để hiển thị ra màn hình. Trong kiến trúc MVC, view chỉ có tác dụng hiển thị giao diện mà thôi, còn điều kiển dòng nhập xuất của người dùng vẫn do **Controllers** đảm trách.



* 1. **Công nghệ dùng thiết kế trên mobile:**

Để hiện thực trên mobile , chúng tôi chọn Android platform để hiện thực bởi vì một số điểm lợi.

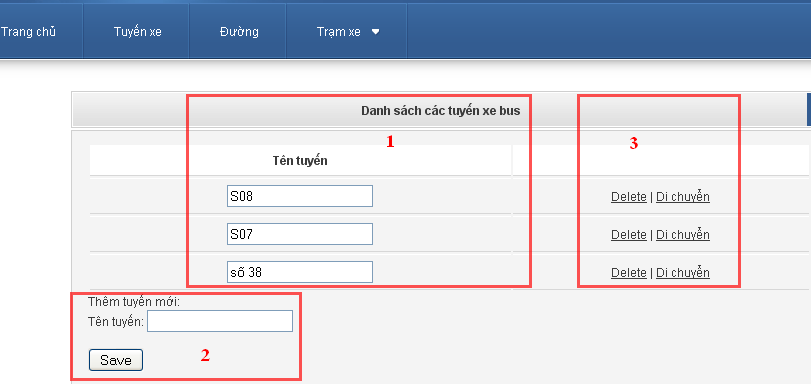
* Tốn ít chi phí để thiết lập một hệ thống phát triển cho người lập trình.
* Tính Năng Mở của hệ điều hành Android:

Android được xây dựng  để cho phép các nhà phát triển để tạo ra các ứng dụng di động hấp dẫn tận dụng tất cả một chiếc điện thoại đã cung cấp. Nó được xây dựng để được thực sự mở. Ví dụ, một ứng dụng có thể kêu gọi bất kỳ chức năng lõi của điện thoại như thực hiện cuộc gọi, gửi tin nhắn văn bản, hoặc bằng cách sử dụng [máy ảnh](http://www.vietnamandroid.com/tag/may-anh), cho phép các nhà phát triển để tạo ra phong phú hơn và nhiều hơn nữa những kinh nghiệm cố kết cho người dùng. Android được xây dựng trên mở Linux Kernel. Hơn nữa, nó sử dụng một máy ảo tuỳ chỉnh được thiết kế để tối ưu hóa bộ nhớ và tài nguyên phần cứng trong một môi trường di động. Android là mã nguồn mở, nó có thể được liberally mở rộng. Nền tảng này sẽ tiếp tục tiến triển như cộng đồng nhà phát triển công việc cùng nhau để xây dựng các ứng dụng di động sáng tạo.

* Với Android tốc độ nhanh và phát triển ứng dụng dễ dàng:

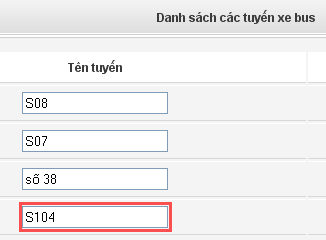
Android cung cấp truy cập đến một loạt các thư viện công cụ hữu ích và có thể được sử dụng để xây dựng các ứng dụng phong phú. Ví dụ, Android cho phép các nhà phát triển để có được vị trí của điện thoại, và cho phép các thiết bị để giao tiếp với nhau tạo điều kiện cho đồng đẳng rich-to-peer ứng dụng xã hội. Ngoài ra, Android bao gồm một tập hợp đầy đủ công cụ đã được xây dựng từ mặt đất lên cùng với việc cung cấp nền tảng phát triển, với năng suất cao và cái nhìn sâu vào các ứng dụng của họ

1. **CHƯƠNG TRÌNH:**
2. **Website**: Gồm 2 phần
3. **Admin**:
4. Quản lý tuyến xe:



Gồm 3 chức năng:

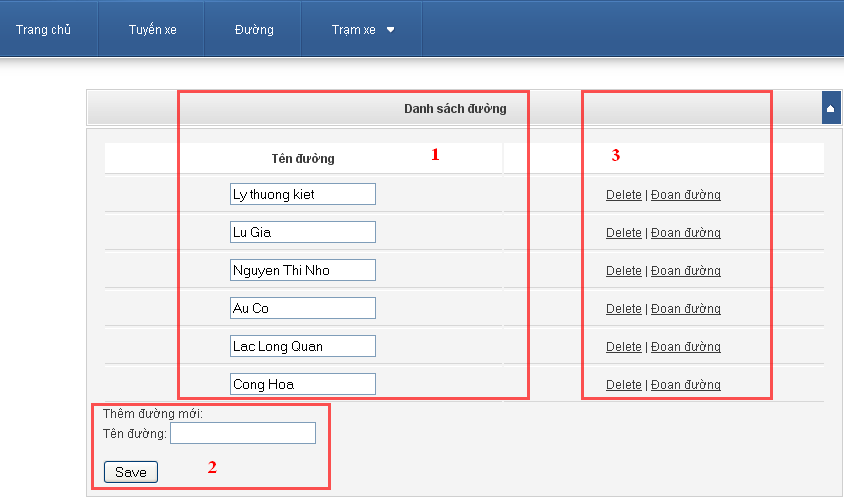
* Chức năng 1: Liệt kê danh sách các tuyến xe bus đang có trong hệ thống.
* Chức năng 2: Thêm tuyến mới. Với chức năng này cho phép cơ sở dữ liệu của hệ thống được bổ xung ví dụ ta có thêm tuyến xe S104. Danh sách tuyến xe sau khi tạo.



* Chức năng 3: Xóa hoặc di chuyển tuyến đường. Ví dụ xóa tuyến đường S104 vừa tạo.

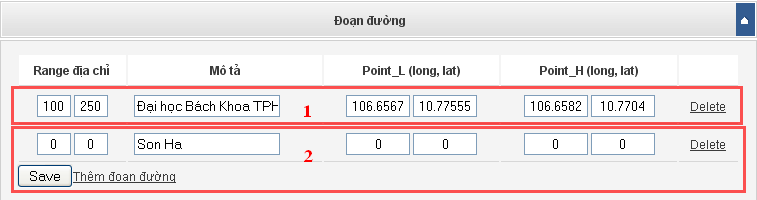


1. Quản lý đường:



Gồm 3 chức năng:

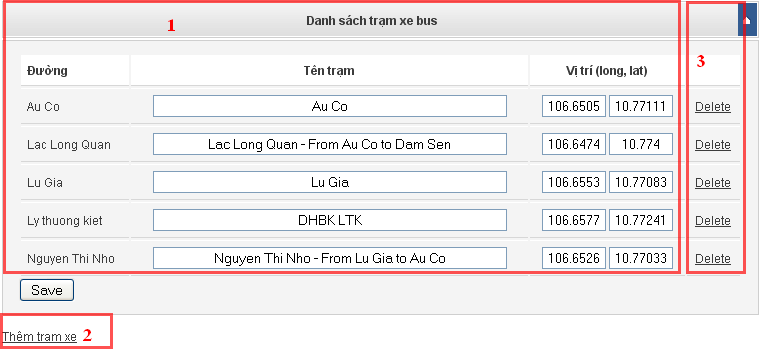
* Chức năng 1: Liệt kê danh sách các đường đang có trong hệ thống.
* Chức năng 2: Thêm một đường mới vào hệ thống. Gần giống với chức năng thêm tuyến trong phần quản lý tuyến xe.
* Chức năng 3:
* Xóa đường: khi dữ liệu người dùng nhập có vấn đề admin có thể sử dụng chức năng này để xóa.
* Nhập đoạn đường: khi nhấp vào link “Đoạn đường” ở khối 3 thì chương trình sẽ link đến màn hình sau.



Gồm hai phần:

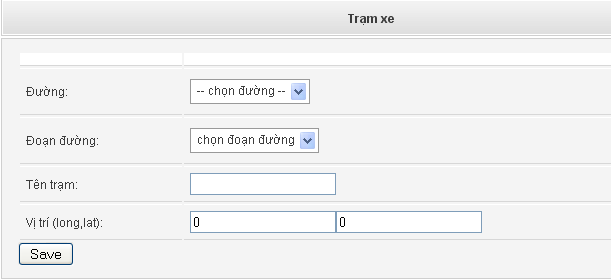
* Phần 1: Liệt kê danh sách tất cả các đoạn đường thuộc đường này
* Phần 2: Thêm mới đoạn đường gồm những trường: địa chỉ, phần mô tả đoạn đường, tọa độ thấp và cao và bấm nút Save để lưu vào hệ thống.

1. Quản lý trạm xe:

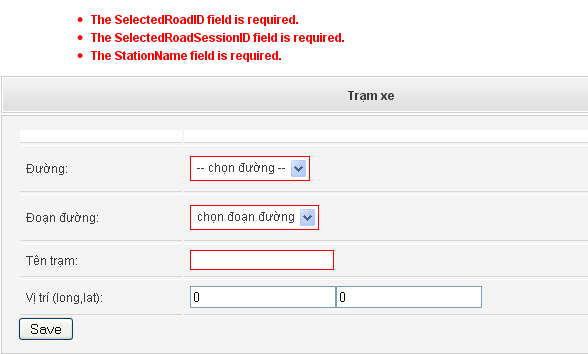


Gồm 3 chức năng:

* Chức năng 1: Liệt kê danh sách các trạm gồm : tên đường, tên trạm, vị trí của trạm.
* Chức năng 2: Thêm trạm xe. Khi nhấp vào link thì chương trình sẽ link tới một màn hình như sau.

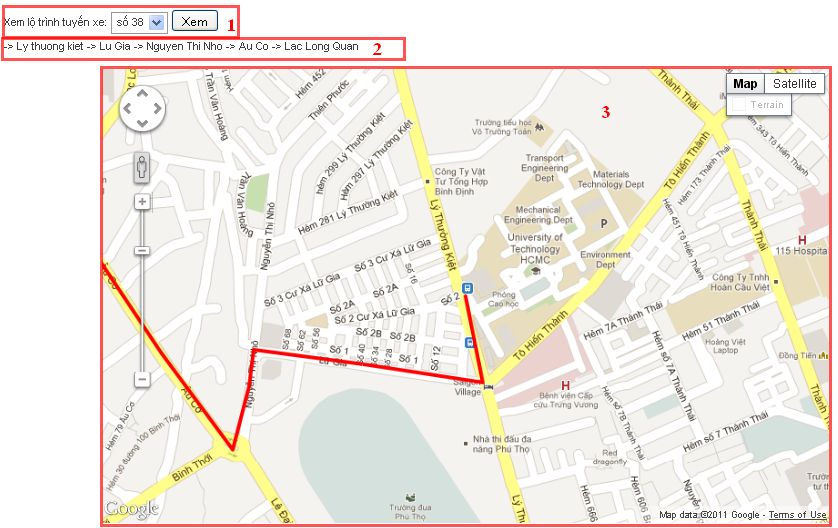


Để tạo một trạm mới người sử dụng phải nhập các thông tin trên như: Đường, Đoạn đường, Tên trạm, Vị trí. Sau đó nhấp nút “Save”, nếu dữ liệu chính xác thì trạm sẽ được lưu vào hệ thống và quay trở về màn hình ban đầu. Nếu dữ liệu sai thì chương trình sẽ yêu cầu nhập lại.



* Chức năng 3: Xóa một trạm trong danh sách Tên trạm.

1. **Client**:
2. Truy xuất lộ trình của một tuyến xe bus:



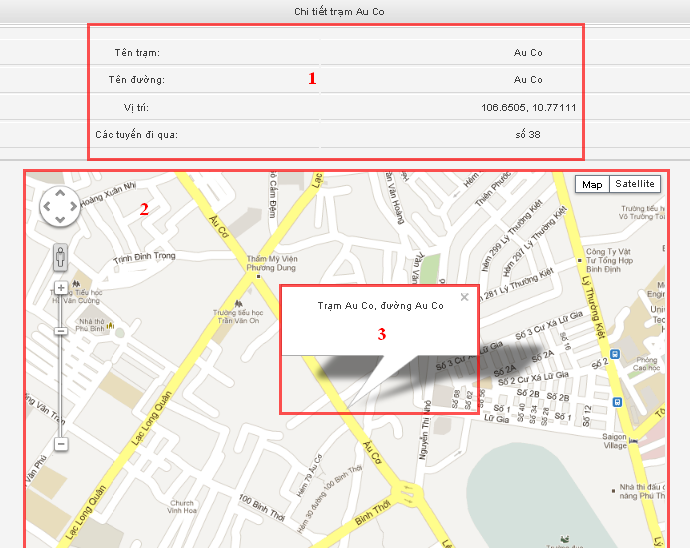
Gồm 3 phần:

* Phần 1: Nhập tuyến xe và nhấn nút xem. Ví dụ “số 38”.
* Phần 2: Hiển thị một lộ trình của tuyến xe số 38.
* Phần 3: Bản đồ và phần vẽ lộ trình của tuyến xe bus số 38.

1. Tìm một tuyến xe đi qua một trạm xe bus:

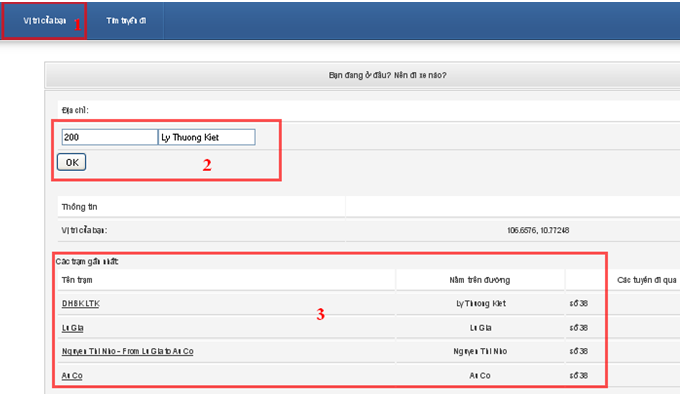


* Phần 1: Hiển thị danh sách của tất cả các trạm của các tuyến xe bus.
* Phần 2: Nút chức năng xem thông tin chi tiết từng trạm.



* Phần 1: Thông tin chi tiết của trạm xe: Tên trạm, tên đường, vị trí và các tuyến xe bus đi qua trạm.
* Phần 2: Hiện thị bản đồ.
* Phần 3: cho thấy tọa độ của trạm trong bản đồ.

1. Tìm các tuyến xe bus đi qua vị trí mà bạn đang đứng:

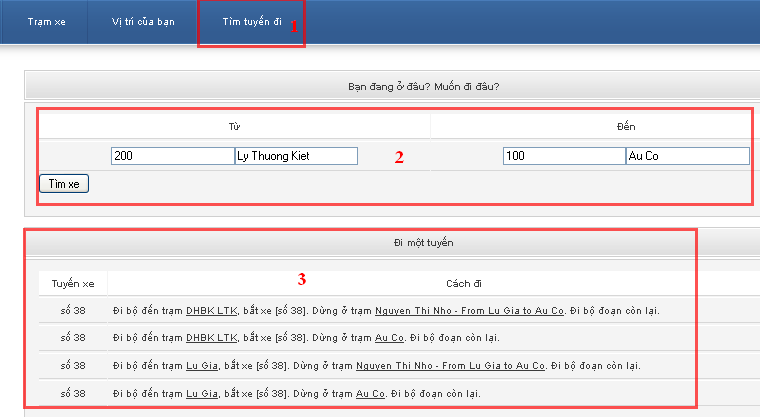


Gồm 4 phần:

* Phần 1: Nhấp vào tap “vị trí của bạn”.
* Phần 2: Nhập địa chỉ và tên đường của vị trí mà bạn đáng đứng và nhấp vào nút “OK”.
* Phần 3: Liệt kê danh sách của các trạm gần vị trí của bạn nhất.
* Phần 4: Hiển thị vị trí trên map.



1. Tìm tuyến xe bus giữa hai địa điểm:



Gồm 3 phần:

* Phần 1: nhấp “Tìm tuyến đi” tab.
* Phần 2: nhập địa điểm đi (200, Ly Thuong Kiet) , nhập địa điểm đến (100, Au Co) và nhấn nút “Tìm xe”.
* Phần 3: hiển thị thông tin về tuyến xe và cách đi.

1. **Mobile:**
2. **KẾT LUẬN**
3. **Tổng kết**:

Sau quá trình làm đề tài thì chúng tôi đã thu được một số kết quả nhất định :

* Về Phía bản thân:
* Nắm thêm các kiến thức về công nghệ : .NET framework, mô hình 3 lớp để hiện thực web và kiến trúc MVC.
* Đồng thời cũng hiểu nền tảng để phát triển một ứng dụng trên mobile với platform Android.
* Ngoài ra cũng học thêm về cách thức làm việc nhóm để tăng tính hiệu quả trong công việc.
* Về phía đề tài:
* Website hỗ trợ tìm đường đi xe buýt với giao diện trực quan.
* Một ứng dụng tìm kiếm đường đi xe buýt trên gPhone.

1. **Hướng phát triển**:

Với những đề xuất trong đề tài chúng tôi đề xuất một số hướng dể phát triển đề tài như sau:

* Mở rộng database với hệ thống giao thông gồm các đường trong thành phố.
* Xây dựng thêm tính năng tìm kiếm và cải tiến giải thuật để tăng độ chính xác khi tiềm kiếm.

1. **TÀI LIỆU THAM KHẢO :**
2. Microsoft Corporation, Microsoft Application Architecture Guid, 2nd ed, 2009.
3. Refractions Research, PostGIS 1.5.3 Manua, 2005.
4. Stephen R. Galati, Geographic Information Systems Demystified, Artech House, 2006.
5. S. Sophat, Fundamentals of geographic information systems, Royal University of Phnom Penh, December 2007.
6. Google Maps Javascript API V3 Basics, http://code.google.com/apis/maps/documentation/javascript/basics.html.
7. Giới thiệu ASP.NET MVC Framework, <http://bachkhoa-software.com/News/1/88/gioi-thieu-aspnet-mvc-framework.aspx>.
8. ASP.NET MVC Overview, <http://www.asp.net/mvc/tutorials/overview/asp-net-mvc-overview>.