ỨNG DỤNG CÔNG NGHỆ JAVA ĐỂ XÂY DỰNG HỆ THỐNG QUẢN LÝ THÔNG TIN ĐĂNG KIỂM

APPLYING JAVA TECHNOLOGY FOR BUILDING REGISTERED TRANSPORT INFOMATION MANAGEMENT SYSTEM

SVTH: Trần Quang Tú

Lớp 05T2, Khoa Công nghệ thông tin, Trường Đại học Bách khoa

GVHD: KS. Nguyễn Võ Quang Đông

Khoa Công nghệ thông tin, Trường Đại học Bách khoa

TÓM TẮT

Công nghệ Java ngày càng được phát triển và sử dụng rỗng rãi để xây dựng những ứng dụng hoạt động trên mạng. Điều này có được do công nghệ java hoàn toàn hỗ trợ hướng đối tượng, miễn phí, bảo mật cao, và độc lập với môi trường thực thi của ứng dụng. Ngoài ra Java còn là một tập hợp các giải pháp dùng để phát triển các ứng dụng web, ứng dụng trên thiết bị cầm tay. Mục đích của bài báo này là nghiên cứu các giải pháp của công nghệ Java: JavaCore, JDBC, J2ME, Java Servlet, Java RMI qua đó vận dụng những giải pháp này để đề xuất một kiến trúc trung gian cho phép các thiết bị di động triệu gọi phương thúc từ máy chủ thông qua mạng wireless. Tiếp đó sử dụng kiến trúc vừa xây dựng để giải quyết bài toán quản lý và tra cứu thông tin đăng kiểm của các phương tiện giao thông đường bộ.

ABSTRACT

Java technology is widely developed and used to build network applications. The reasons for its success are strongly object-oriented, free fee payment, high security, and platform independent. In addition, Java technology is combined with many solutions in order to develop web applications, mobile applications, and so on. The purpose of this article is to do research about Java solutions: Java Core, JDBC, J2ME, Java Servlet, and Java RMI. Then these solutions are applied to propose an intermediate architecture model with the purpose of remote method invocation from Mobile devices to Server via wireless network. Finally, this model is used for building a system which can manage registered transport information.

1. Đặt vấn đề

Cấu hình thiết bị giới hạn được kết nối (Connected Limited Device Configuration: CLDC) cung cấp nền tảng kết nối chung (Generic Connectiong Framework: GCF) có thể được sử dụng để phát triển những ứng dụng nền tảng mạng. Thêm vào đó, MIDP (Mobile Information Device Profile) cung cấp giao diện *HttpConnection* là một phần của gói *java.microedition.io* trong đó gói này định nghĩa những phương thức cho kết nối HTTP. HTTP là giao thức duy nhất mà sự thực thi MIDP cần phải hỗ trợ, tất cả các phương thức khác là tùy chọn. Ví dụ, nó không hỗ trợ cho TCP (Transport Control Protocol) socket hoặc UDP (User Datagram Protocol) datagram.

Thêm vào đó, máy ảo K (K Virtual Machine: KVM) không hỗ trợ tất cả ngôn ngữ Java và những đặc trưng của máy ảo, hoặc bởi vì chúng quá đắt hoặc sự hiện diện của chúng phải đối mặt với vấn đề an toàn. Chúng không hỗ trợ cho tuần tự hóa đối tượng (object serialization), do đó nó không hỗ trợ cho triệu gọi phương thức từ xa (Remote Method Invocation: RMI). Tuy vậy những lợi ích của những ứng dụng di động đã trở nên

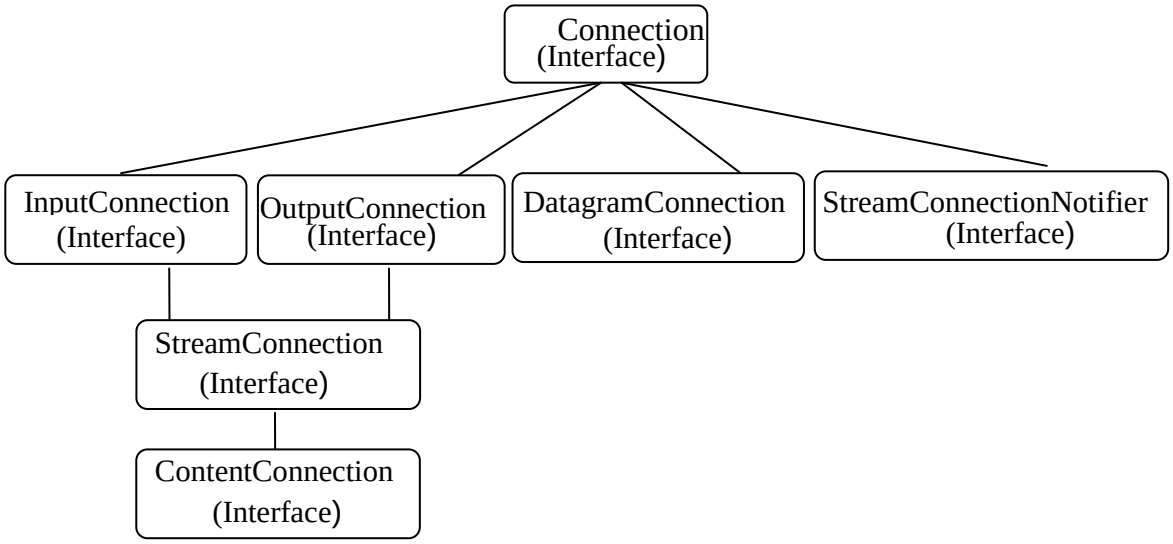
hiện thực và chúng có thể truy xuất tới hạn đến dữ liệu và tài nguyên Internet rất hiệu quả tại bất kỳ nơi đâu. Vì vậy để khắc phục vấn đề cần có một kiến trúc trung gian để thiết bị di động có thể triệu gọi được phương thức từ xa. Từ đó khai thác những đặc điểm tối ưu của thiết bị di động.

2. Phương pháp xây dựng kiến trúc trung gian.

2.1. Tổng quan CLDC và MIDP Networking

CLDC kế thừa từ nhiều lớp trong gói *java.io* nhưng nó không kế thừa liên quan đến tệp I/O cơ bản bởi vì không phải tất cả các thiết bị đều hỗ trợ khái niệm file I/O. J2ME (Java 2 Standard Edition) cung cấp vài lớp cho khả năng kết nối mạng, tuy nhiên, không một lớp nào trong số chúng được kết thừa đơn giản bởi vì không phải tất cả các thiết bị đều yêu cầu TCP/IP hoặc UDP/IP, một số thiết bị có thể không có ngăn xếp IP.

Những thách thức về I/O và kết nối mạng được giải quyết bằng cách định nghĩa một tập hợp những lớp I/O và kết nối mạng. Những lớp này được biết như là GCF. Một nền tảng độc lập không phụ thuộc vào những đăc trưng của một thiết bị. Nó cung cấp một hệ thống cấp bậc của giao diện kết nối, nhưng chúng không thực thi bất cứ điều gì. Sự thực thi được bởi nhưng profile (như là MIDP).



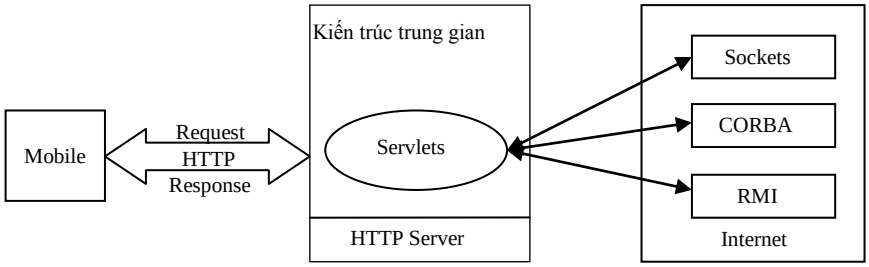
Hình 1. *CLDC Generic Connnection Framework*

Tất cả các kết nối được tạo sử dụng phương pháp tĩnh *open* từ lớp *Connector*. Nếu thành công, phương thức này trả về một đối tượng, đối tượng thực thi trong một giao diện kết nối chung.

MIDP kế thừa khả năng kết nối CLDC cung cấp hỗ trợ cho giao thức HTTP. MIDP cung cấp giao diện *HttpConnection,* lớp con của *ContentConnection.* Lý do đằng sau việc hỗ trợ HTTP sự thật là HTTP có thể thực thi sử dụng những giao thức như IP (như TCP/IP) hoặc những giao thức không IP (như WAP). Một thiết bị hỗ trợ MIDP có thể không tích hợp việc hỗ trợ giao thức IP. Trong trường hợp này, nó sẽ dùng một cổng có thể cho việc giải quyết chuỗi URL để truy xuất Internet.

2.2. Kiến trúc trung gian

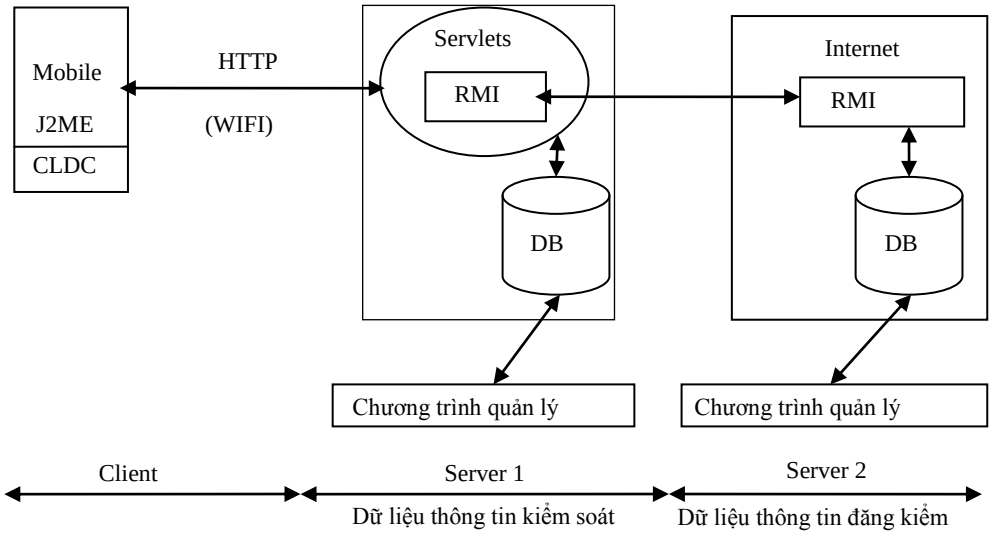
CLDC và MIDP không hỗ trợ cho những kết nối datagram. Hơn nữa, như đã đề cập trước đó, nó không hỗ trợ RMI. Vì vậy, nó không trực tiếp gọi những ứng dụng dựa trên RMI. Như là giải pháp, kiến trúc trung gian gọi một Servlet ở giữa, nơi chấp nhận những yêu cầu của client (nếu cần thiết), xử lý chúng, và trả về kết quả trở lại client. Kiến trúc này được biểu diễn trong hình 2.



Hình 2. *Kiến trúc trung gian*

3. Xây hệ thống quản lý thông tin đăng kiểm

Dựa trên mô hình kiến trúc trung gian, có thể giải quyết vấn đề của việc gọi phương thức từ xa (RMI) từ thiết bị di động. Việc phát triển mô hình để có thể xây dựng một hệ thống quản lý thông tin đăng kiểm cần có.



Hình 3. *Hệ thống tra cứu thông tin đăng kiểm*

Xây dựng hệ thống theo mô hình Client/Server trong đó thiết bị đi động (mobile phone) có hỗ trợ wireless đóng vai trò là Client. Server là một máy deskop có cấu hình mạnh để phục vụ được nhiều client.

- Tại Client là thiết bị di động sử dụng công nghệ J2ME, Servlet. Chương trình ứng dụng trên client gồm tra cứu thông tin phương tiện đăng kiểm và quản lý người dùng.
- Tại Server (Có 2 Server kết nối với nhau thông qua RMI) gồm:
  - Server1: Java Servlet của lực lượng kiểm soát.
  - Server2: Database Server của Cục đăng kiểm.

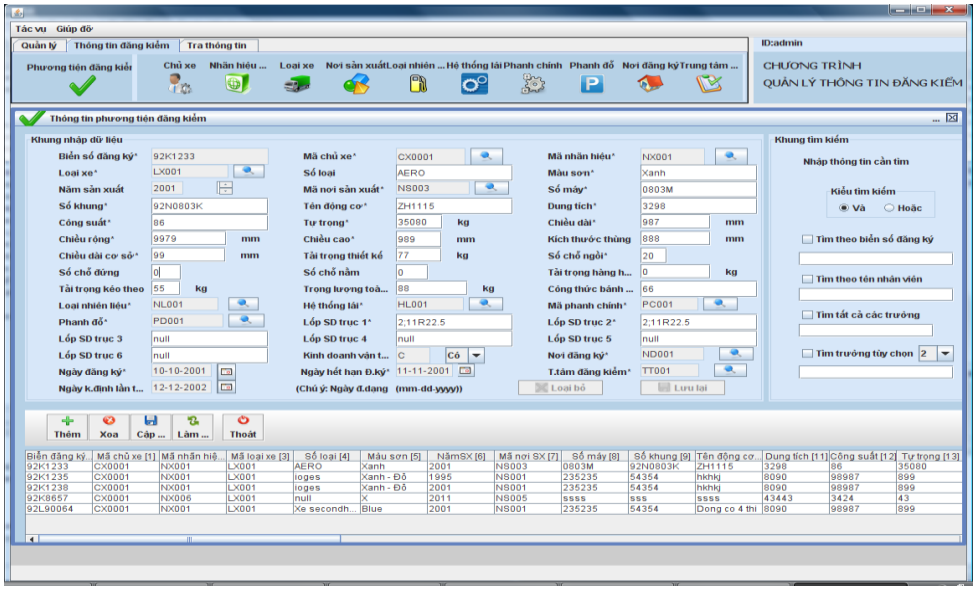
Tại server sử dụng công nghệ : Java Core, Java Servlet, Java RMI. Ứng dụng bao

gồm quản lý người dùng và thống kê cho hệ thống. Với Servlet Server đóng vai trò làm một client kết nối đến server của Cục đăng kiểm sử dụng java RMI.

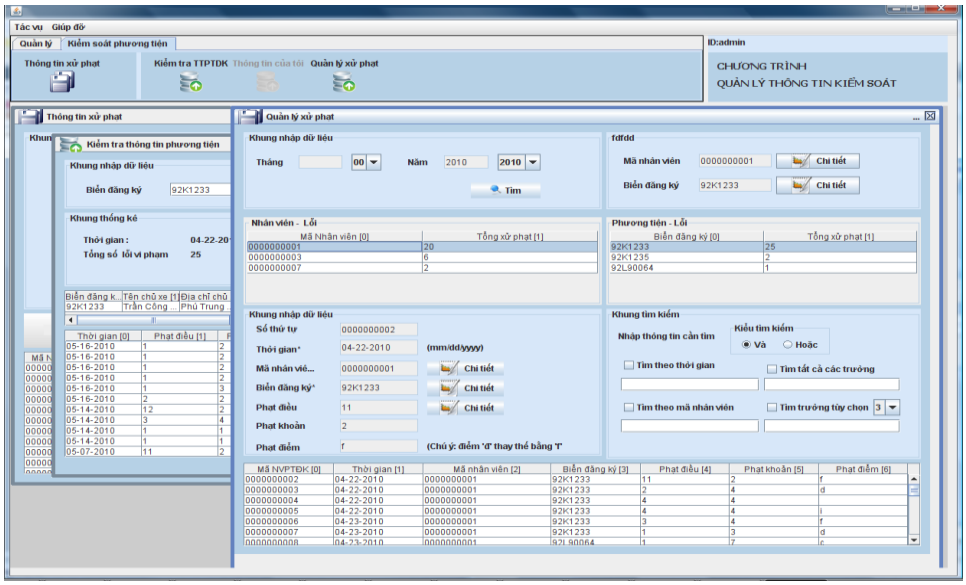
Trong quá trình hoạt động, khi kiểm soát viên (cảnh sát giao thông) sử dụng thiết bị di động được kết nối vào mạng (WIFI) sau khi thực hiện thao tác đăng nhập thì người kiểm soát có thể xem thông tin của phương tiện được lưu trữ dữ liệu trên Server của cục đăng kiểm (Server2). Người kiểm soát có thể lập biên bản xử phạt (nếu người kiểm soát là cảnh sát giao thông), và xem thông tin lich sử vi phạm của phương tiện. Tại client và server có chương trình quản lý thực hiện quản lý người dùng và thông kê số liệu tổng hợp về thông tin xe, số lần vi phạm…

4. Kết quả và đánh giá

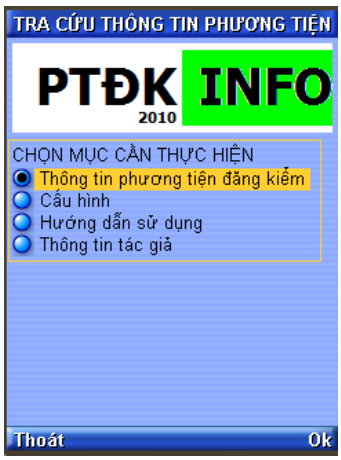
Xây dựng hoàn chỉnh ứng dụng quản lý thông tin đăng kiểm và áp dụng các công nghệ đã nguyên cứu.



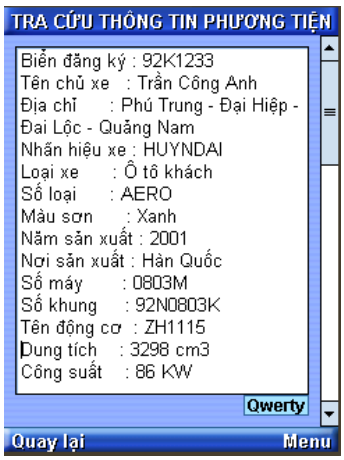
***Hình 4**. Chương trình quản lý thông tin phương tiện đăng kiểm (Server2)*



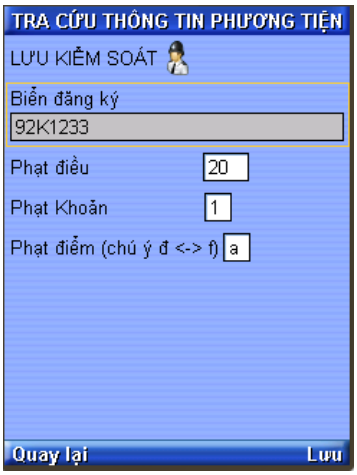
***Hình 5**. Chương trình quản lý thông tin kiểm soát (Server1)*



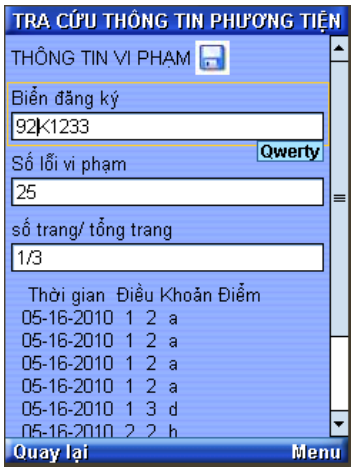
***Hình 6**. Màn hình chính (Mobile)*



***Hình 7**. Màn hình thông tin đăng kiểm(Mobile)*



***Hình 8**. Màn hình lập biên bản xử phạt (Mobile)*



***Hình 9**. Màn hình thông tin vi phạm (Mobile)*

Khi đáp ứng các yêu cấp thiết kế, hệ thống có các ưu điểm của mô hình client-server: dữ liệu tập trung, dữ liệu không còn là gánh nặng phía client. Hệ thống được cải thiện tốc độ nhờ việc xử lý thông tin tại server.

5. Kết luận

Việc xây dựng được thành công kiến trúc trung gian đã giải quyết về mặt kết nối cho thiết bị di động gọi được các phương thức từ xa (RMI) tạo hiệu quả lớn cho việc sử dụng thiết bị di động. Từ đó kết hợp với các công nghệ của Java đã xây dựng hoàn chỉnh ứng dụng có liên quan trong lĩnh vực quản lý thông tin phương tiện đăng kiểm. Góp phần tin học hóa trong quản lý trong lĩnh vực giao thông.

TÀI LIỆU THAM KHẢO

[1] John W. Muchow (2001), *Core J2ME Technology & MIDP*, Prentice Hall PTR, USA.

[2] Cay S. Horstmann, Gary Cornell (2004), *Core Java(TM) 2 Volume I – Fundamentals, Seventh Edition*, Prentice Hall PTR, USA.

[3] Cay S. Horstmann, Gary Cornell (2004), *Core Java(TM) 2 Volume II – Advamced*

Features Seventh Edition, Prentice Hall PTR, USA.

[4] Y. Daniel Liang (2007), *Introduction to Java programming*, Prentice Hall, USA.

[5] Qusay Mahmoud (2002), *Advanced MIDP Networking, Accessing Using Sockets and RMI from MIDP-enable Devices*, Oracle Sun Developer Network (SDN), http://developers.sun.com.