

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



Nguyễn Văn Nội

**PHÁT TRIỂN HỆ THỐNG QUẢN LÝ NGƯỜI DÙNG
DỊCH VỤ XE BUÝT TRÊN NỀN TẢNG STRUTS2**

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ thông tin

HÀ NỘI - 2014

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

Nguyễn Văn Nội

**PHÁT TRIỂN HỆ THỐNG QUẢN LÝ NGƯỜI DÙNG
DỊCH VỤ XE BUÝT TRÊN NỀN TẢNG STRUTS2**

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ thông tin

Cán bộ hướng dẫn: TS. Nguyễn Ngọc Hóa

Cán bộ đồng hướng dẫn: ThS. Dư Phương Hạnh

HÀ NỘI - 2014

**VIETNAM NATIONAL UNIVERSITY, HANOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY**

Nguyen Van Noi

**DEVELOPING ACCOUNT MANAGEMENT SYSTEM
IN BUS SERVICE USING STRUTS2**

Major: Information Technology

Supervisor: T.S Nguyen Ngoc Hoa

Co-Supervisor: Th.S Du Phuong Hanh

HA NOI - 2014

TÓM TẮT

Tóm tắt:

Sự phát triển nhanh chóng của công nghệ thông tin kéo theo nhiều tác động đến cuộc sống của con người. Chúng ta có thể thấy qua các ngôi nhà thông minh, xe thông minh ... chúng giúp công việc của con người được hoàn thành một cách dễ dàng và ít tốn công sức nhất. Đó chính là xu thế chính của thời đại.

Tiếp bước theo xu thế đó, dựa vào những công nghệ đã có hoặc mới được phát triển, chúng tôi tiến hành phát triển một hệ thống quản lý người dùng trên xe buýt dựa trên công nghệ NFC đã xuất hiện cách đây không lâu.

Hệ thống được xây dựng gồm 2 thành phần chính là phía Client và Server. Phía Client được xây dựng trên Android với các tính năng cung cấp trực tiếp cho người dùng và các nhân viên soát vé trên xe. Được cung cấp giao diện đơn giản, dễ sử dụng và phù hợp với hầu hết người dùng. Phần này được phát triển trong khóa luận của bạn Quan Tuấn Vũ.

Khóa luận này trình bày phần Server, với mục đích chủ yếu là cung cấp các tiện ích cho hầu hết người dùng và các tiện ích cần thiết cho phía Client thông qua dịch vụ Web. Phía Server được phát triển trên nền Java kết hợp các framework Struts2, Hibernate 3.0 thông qua giao diện hiển thị bằng JSP đã đáp ứng được một số yêu cầu cơ bản của hệ thống đặt ra.

Từ khóa: *Quản lý người dùng, NFC, framework, dịch vụ Web.*

ABSTRACT

Abstract:

The quick development of information technology brings about many effects on people's lives. We can see that through smart houses or smart cars which help people to do their work easily and in a energy-saving way. That is the main trend of the age.

To continue the trend, relying on technologies have had have just developed, we implement to develop a user management system in bus base NFC technology that have appeared for a while.

The system includes two main part which are Client and Server. The Client is built on Android with function providing users and conductors directly. Its interface is simply, easy to use and suitable with most users. This part will be presented in Quan Tuan Vu's thesis.

This thesis presents Server, for the main purpose of providing tools for most users as well as Client via Web service. The Server is developed on Java basis combining Struts2, Hibernate 3.0 frameworks via viewed interface by JSP meeting some basic demands of the system.

Keywords: *User management, NFC, frameworks, Web services*

LỜI CAM ĐOAN

Tôi xin cam đoan kết quả đạt được trong khóa luận này là sản phẩm của riêng cá nhân tôi. Trong toàn bộ nội dung của khóa luận, những điều được trình bày hoặc là của cá nhân hoặc là được tổng hợp từ nhiều nguồn tài liệu. Tất cả những tham khảo từ các nghiên cứu liên quan đều được chỉ rõ trong trích dẫn và danh mục tài liệu tham khảo một cách rõ ràng.

Tôi xin hoàn toàn chịu trách nhiệm theo quy định cho lời cam đoan của mình.

Hà Nội, ngày 16 tháng 05 năm 2014

Sinh viên

Nguyễn Văn Nội

LỜI CẢM ƠN

Trước hết, em xin chân thành gửi lời cảm ơn đến các quý thầy cô Trường Đại học Công Nghệ đã tận tình dạy bảo và truyền đạt cho em những kiến thức nền tảng về Công nghệ thông tin trong suốt quá trình em học tập tại trường.

Em xin gửi lời biết ơn sâu sắc tới thầy Nguyễn Ngọc Hóa, người đã hết lòng chỉ bảo, dành thời gian và tâm huyết hướng dẫn các phương pháp tiếp cận vấn đề, giúp em giải đáp những thắc mắc, giải quyết các khó khăn gặp phải để hoàn thành được khóa luận này.

Em cũng xin gửi lời cảm ơn đến những người thân trong gia đình và bạn bè, những người đã động viên em về mặt tinh thần, những người đã luôn bên cạnh và tạo mọi điều kiện thuận lợi nhất cho em trong quá trình hoàn thành khóa luận.

Trong bài báo cáo không thể tránh khỏi những thiếu sót, em rất mong nhận được những đóng góp quý báu của thầy cô và các bạn để có thể được hoàn thiện hơn.

Em xin chân thành cảm ơn!

Hà Nội, ngày 16 tháng 05 năm 2014

Sinh viên

Nguyễn Văn Nội

MỤC LỤC

TÓM TẮT	i
ABSTRACT	ii
LỜI CAM ĐOAN	iii
LỜI CẢM ƠN	iv
MỤC LỤC	v
BẢNG DANH MỤC CÁC TỪ VIẾT TẮT	vii
DANH MỤC CÁC HÌNH VẼ	viii
DANH MỤC BẢNG BIỂU	x
CHƯƠNG 1	1
MỞ ĐẦU	1
1.1. Thực trạng và yêu cầu đặt ra	1
1.1.1. Thực trạng	1
1.1.2. Yêu cầu đặt ra	1
1.2. Đối tượng nghiên cứu	1
1.3. Mục tiêu và ý nghĩa	1
1.3.1. Mục tiêu	1
1.3.2. Ý nghĩa	1
1.4. Phương pháp nghiên cứu	2
1.5. Tổ chức khóa luận	2
CHƯƠNG 2	4
CƠ SỞ LÝ THUYẾT	4
2.1. Công nghệ thẻ NFC	4
2.2. Mô hình MVC	7
2.3. Framework Struts 2	9
2.4. Framework Hibernate	11

2.5.	Hệ quản trị cơ sở dữ liệu	14
2.6.	Dịch vụ Web và REST	16
2.6.1.	<i>Dịch vụ Web</i>	16
2.6.2.	<i>REST</i>	18
CHƯƠNG 3		20
XÂY DỰNG HỆ THỐNG		20
3.1.	Tổng quan hệ thống	20
3.2.	Phía Client	22
3.3.	Phía Server	23
3.3.1.	<i>Mô hình tổng quan</i>	23
3.3.2.	<i>Lược đồ cơ sở dữ liệu</i>	24
3.3.3.	<i>Mô hình ca sử dụng của hệ thống</i>	25
3.3.4.	<i>Các ca sử dụng</i>	25
3.3.5.	<i>Dịch vụ Web cung cấp cho phía Client</i>	32
3.3.6.	<i>Thiết kế gói dữ liệu</i>	34
3.3.7.	<i>Các giao diện chính</i>	35
CHƯƠNG 4		39
THỰC NGHIỆM		39
4.1.	Môi trường thực nghiệm	39
4.1.1.	<i>Cấu hình máy tính</i>	39
4.1.2.	<i>Các công cụ phần mềm</i>	39
4.2.	Kết quả thực nghiệm	40
4.3.	Đánh giá và bàn luận	40
CHƯƠNG 5		41
KẾT LUẬN CHUNG		41
5.1.	Các kết quả đạt được	41
5.2.	Phương hướng phát triển	41
TÀI LIỆU THAM KHẢO		42

BẢNG DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Ý nghĩa
MVC	Model - View - Controller
CSDL	Cơ sở dữ liệu
DBMS	Database Management System
ORM	Object Relational Mapping
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
JSON	JavaScript Object Notation
XML	eXtensible Markup Language
HTTP	HyperText Transfer Protocol
UUDI	Uniseral Description, Discovery and Intergration
WSDL	Web Services Description Language
URI	Uniform Resource Identifier
HTML	HyperText Mark Language
GUI	Graphic User Interface
JSP	Java Server Pages
W3C	World Wide Web Consortium
SQL	Structured Query Language
LGPL	GNU Lesser General Public License

DANH MỤC CÁC HÌNH VẼ

Hình 2.1.	Mô hình MVC	8
Hình 2.2.	Mô hình kiến trúc Struts 2	10
Hình 2.3.	Vòng đời của Struts 2	11
Hình 2.4.	Mô hình kiến trúc Hibernate.....	13
Hình 3.1.	Mô hình tổng quan hệ thống.....	20
Hình 3.2.	Mô hình tổng quan phía Server	23
Hình 3.3.	Lược đồ cơ sở dữ liệu	24
Hình 3.4.	Mô hình ca sử dụng hệ thống.	25
Hình 3.5.	Mô hình ca sử dụng đăng nhập.....	26
Hình 3.6.	Mô hình ca sử dụng đăng xuất.....	27
Hình 3.7.	Mô hình ca sử dụng quản lý tuyến buýt	28
Hình 3.8.	Mô hình ca sử dụng quản lý thông tin khách hàng.....	29
Hình 3.9.	Mô hình ca sử dụng quản lý giao dịch.....	30
Hình 3.10.	Mô hình ca sử dụng quản lý tài khoản.....	31
Hình 3.11.	Mô hình thiết kế gói dữ liệu	34
Hình 3.12.	Giao diện đăng nhập	35
Hình 3.13.	Giao diện quản lý thông tin khách hàng	36
Hình 3.14.	Giao diện quản lý các loại vé.....	36
Hình 3.15.	Giao diện quản lý các tuyến xe buýt	37
Hình 3.16.	Giao diện quản lý các giao dịch của khách hàng.....	37
Hình 3.17.	Giao diện quản lý thông tin nhân viên.....	38

Hình 3.18.	Giao diện quản lý tài khoản.....	38
------------	----------------------------------	----

DANH MỤC BẢNG BIỂU

Bảng 4.1.	Cấu hình máy tính chạy thực nghiệm.....	39
Bảng 4.2.	Các công cụ phần mềm.....	39

CHƯƠNG 1

MỞ ĐẦU

1.1. Thực trạng và yêu cầu đặt ra

1.1.1. Thực trạng

Hiện nay, dịch vụ xe buýt đang được sử dụng nhiều và ngày càng tiện lợi và thuận tiện hơn, nhất là đối với sinh viên. Việc điều hành và phân phát vé theo ngày hay theo tháng thì cũng đòi hỏi mất rất nhiều công sức và tốn thời gian. Với sự phát triển của công nghệ hiện nay, việc áp dụng các công nghệ mới vào trong quá trình xác thực người dùng và kiểm soát vé xe buýt là tối cần thiết. Vì vậy, chúng tôi xây dựng hệ thống áp dụng công nghệ NFC vào xác thực người dùng sử dụng dịch vụ xe buýt.

1.1.2. Yêu cầu đặt ra

Xây dựng một hệ thống cung cấp các dịch vụ cho phép xác thực người dùng dịch vụ xe buýt kết hợp với công nghệ thẻ NFC.

1.2. Đối tượng nghiên cứu

Người dùng dịch vụ xe buýt và công nghệ NFC.

1.3. Mục tiêu và ý nghĩa

1.3.1. Mục tiêu

Xây dựng một hệ thống xác thực người dùng và cung cấp các tiện ích cho người sử dụng để có thể áp dụng vào thực tiễn đời sống để phục vụ nhu cầu của mọi người.

Hệ thống chỉ tập trung vào việc xác thực người dùng, không liên quan đến các chuyên môn nghiệp vụ như quản lý xe buýt, quản lý nhân viên hay các quy trình bán vé hoặc xé vé trên xe buýt.

1.3.2. Ý nghĩa

Giúp ích cho cuộc sống của con người là ý nghĩa lớn lao nhất mà khóa luận hướng đến. Nâng cao chất lượng cuộc sống, cải thiện đời sống con người và làm tăng năng suất lao động là mục đích hướng đến của khóa luận này.

Ngoài ra, nó cũng có ý nghĩa nhất định về mặt khoa học như việc áp dụng các công nghệ mang tính thời đại mới vào trong quá trình xây dựng và phát triển hệ thống. Mở rộng phạm vi áp dụng các kiến thức công nghệ thông tin trong thực tiễn đời sống.

1.4. Phương pháp nghiên cứu

Tiến hành tìm hiểu các thông tin về quản lý xe buýt cùng các loại vé thông qua thực tế.

Tiến hành nghiên cứu tài liệu để thiết kế hệ thống.

Tìm hiểu các framework cùng các hệ cơ sở dữ liệu để tiến hành xây dựng hệ thống.

Kết hợp các kiến thức sẵn có và kiến thức tìm hiểu được để tiến hành thiết kế cơ sở dữ liệu, xây dựng hệ thống ... cung cấp các tiện ích cho phía người dùng và phía quản trị.

1.5. Tổ chức khóa luận

Khóa luận gồm 5 chương

Chương 1: Mở đầu

Giới thiệu vấn đề đặt ra cùng các phương hướng, mục tiêu cần đạt được, phương pháp nghiên cứu và mô tả tổ chức của khóa luận.

Chương 2: Cơ sở lý thuyết

Nêu các kiến thức nền cơ sở được sử dụng trong khóa luận cùng các kiến thức liên quan cần biết để có thể hoàn thành khóa luận này.

Chương 3: Xây dựng hệ thống

Thể hiện các thiết kế liên quan đến hệ thống tổng quan cùng hệ thống mà khóa luận đảm nhận tìm hiểu, nghiên cứu và phát triển.

Chương 4: Thực nghiệm

Môi trường tiến hành thử nghiệm các kết quả đã đạt được từ bắt đầu cho đến kết thúc khóa luận được nêu ra một cách rõ ràng trong chương này.

Chương 5: Kết luận

Kết quả đạt được trong suốt quá trình tìm hiểu, nghiên cứu và hoàn thành khóa luận này. Đồng thời, nêu lên định hướng phát triển để hoàn thiện các thành phần còn

lại của hệ thống hoặc nâng cấp hệ thống trong tương lai. Với mục đích hiện thực hóa các kết quả của khóa luận vào thực tiễn đời sống.

CHƯƠNG 2

CƠ SỞ LÝ THUYẾT

2.1. Công nghệ thẻ NFC

2.1.1. Sự ra đời của công nghệ NFC

NFC (Near Field Communicate – giao tiếp trường gần) là công nghệ kết nối không dây phạm vi tầm ngắn trong khoảng cách 4cm, sử dụng cảm ứng từ trường để thực hiện kết nối giữa các thiết bị khi có sự tiếp xúc trực tiếp hay gần nhau. NFC hoạt động ở dải băng tần 13,56MHz và tốc độ truyền tối đa 424Kbps. Do khoảng cách truyền dữ liệu khá ngắn nên giao dịch thông qua công nghệ NFC được xem là an toàn.

Công nghệ NFC ra đời từ sự kết hợp công nghệ nhận dạng không tiếp xúc và những công nghệ kết nối truy cập mới.

NFC có 4 định dạng thẻ dựa trên các chuẩn ISO 14443 Type A, 14443 Type B và ISO 18092.

Công nghệ NFC có 2 chế độ truyền dữ liệu: chủ động và thụ động.

Trong chế độ thụ động: Thiết bị nguồn sáng sẽ phát ra từ trường đến nguồn đích. Trong chế độ này, nguồn đích ở trạng thái bị động và chỉ trả lời khi nhận tín hiệu từ nguồn nguồn phát.

Trong chế độ chủ động: Cả thiết bị nguồn phát và thiết bị đích truyền dữ liệu bằng cách tạo ra từ trường riêng.

Hầu hết các ứng dụng hiện nay đều kết hợp cả 2 chế độ chủ động và thụ động, vì sẽ hữu ích cho các thiết bị trong việc truyền dữ liệu giữa các thiết bị không có nguồn điện, ví dụ các thẻ không tiếp xúc.

Một giao dịch diễn ra trên NFC tuân tự theo các bước: phát hiện (Discovery), xác thực (Authentication), trao đổi (Negotiation), truyền dữ liệu (Transfer) và xác nhận (Acknowledgment) từ phía nhận dữ liệu.

Đối với các ứng dụng mang tính nhạy cảm cao, chẳng hạn trong trường hợp muốn tăng cường tính bảo mật trong lúc giao dịch, công nghệ NFC có thêm 2 chuẩn mã hoá:

Chuẩn mã hoá tiên tiến (Advanced Encryption Standard – AES) là thuật toán mã hoá khối được Mỹ áp dụng làm tiêu chuẩn mã hoá.

Chuẩn mã hoá dữ liệu Triple DES (Data Encryption Standard) nghĩa là một thông tin được mã hoá DES 3 lần với khoá khác nhau, do đó chiều dài mã hoá sẽ lớn hơn và an toàn hơn.

2.1.2. Mục đích

Kết nối các thiết bị điện tử

Truy cập nội dung số, người dùng chỉ cần áp điện thoại di động lên áp phích quảng cáo (có gắn thẻ từ tính – RF tag), lập tức người dùng sẽ nhận được các thông tin liên quan.

Giao dịch không tiếp xúc, ví dụ thanh toán, mua vé

Công nghệ NFC sẽ làm thay đổi thói quen sử dụng thiết bị của mọi người. Theo lý thuyết, có khá nhiều ứng dụng dành cho NFC chẳng hạn như NFC được sử dụng trong điện thoại di động camera số, TV hay hệ thống định vị lái xe để thanh toán hoá đơn theo những món hàng muốn mua. Khi muốn mua sắm, người dùng có thể thanh toán bằng cách áp thẻ tín dụng vào màn hình máy tính có gắn NFC. Khi chip NFC chứa thông tin thẻ tín dụng (credit card) trong điện thoại di động, điện thoại di động của bạn sẽ trở thành chìa khoá, vé xem ca nhạc, thẻ lưu thông tin cá nhân... Ngoài ra, người dùng còn có thể biết thông tin của các chương trình giảm giá, khuyến mãi khi dùng điện thoại di động có NFC chạm gần các áp phích quảng cáo thông minh hay chụp hình và gửi ảnh đến màn hình, máy tính hay trao đổi danh thiếp qua điện thoại di động có hỗ trợ NFC. Với điện thoại di động có tích hợp NFC, bạn có thể mua vé, nhận vé và quét thẻ. Sau đó, bạn có thể kiểm tra số tài khoản của mình ngay trên điện thoại di động.

NFC mở ra một xu hướng trao đổi dữ liệu theo dạng mạng ngang hàng (P2P). Do NFC hoạt động dựa trên tần số 13,56MHz nên không ảnh hưởng đến các công nghệ kết nối không dây khác. Khi mạng NFC được kích hoạt, người dùng vẫn có thể sử dụng các công nghệ kết nối khác như Bluetooth, Wifi nếu muốn kết nối tầm xa hơn hay khi cần truyền lượng dữ liệu lớn hơn.

Công nghệ NFC hứa hẹn sẽ tiến xa và có nhiều tiềm năng hơn thẻ tần số vô tuyến (RFID tag), là nhân tố quan trọng trong việc khai thác các dịch vụ tài chính trên thiết

bị di động của các nhà khai thác mạng di động (MNOs-Mobile Network Operators), các nhà thiết kế điện thoại di động.

Mặc dù các giao thức của NFC khác nhau, nhưng vẫn được xem là linh động hơn, bảo mật hơn so với RFID hay thẻ thông minh. Sự khác biệt chủ yếu là NFC được tích hợp trong các thiết bị di động để liên kết đến các dịch vụ bán lẻ hay các giao dịch tài chính.

Thêm vào đó, tính năng bảo mật khi truyền dữ liệu là vô cùng quan trọng, do đó NFC trên các thiết bị di động không chỉ đơn giản là một công nghệ hay ứng dụng mà là một hệ thống nhỏ. NFC trên các thiết bị di động được xác định là nhân tố quan trọng cho các giải pháp thanh toán của Amex, Mastercard và VISA.

Đến nay có khoảng 14 nhà khai thác mạng di động đã cùng nhau phát triển các ứng dụng NFC mà thị phần của 14 nhà khai thác này chiếm đến 40% thị trường di động toàn cầu.

2.1.3. NFC trên di động

Hiệp hội GSM bắt đầu dự án đưa công nghệ NFC vào điện thoại di động từ đầu năm 2006, đồng thời cung cấp tài liệu hướng dẫn kỹ thuật cho các kỹ sư thiết kế. Một thiết bị di động tích hợp NFC gồm thành phần chính sau: ăng-ten được tích hợp vào trong thiết bị di động, chip NFC được tích hợp trong thẻ SIM của điện thoại di động. Phần mềm trên thiết bị sẽ gồm một số ứng dụng cần dung lượng từ 50KB đến 200KB.

Những khó khăn hiện nay của điện thoại di động hỗ trợ NFC là làm thế nào để tạo một hệ thống đáng tin cậy nghĩa là có sự hợp tác, kết nối của các nhà thiết kế thiết bị di động, nhà sản xuất, công ty tài chính, nhà bán lẻ.

Một vài điển hình về các sản phẩm hỗ trợ NFC, hãng Renesas đã ra mắt bộ vi điều khiển RF21S có kết hợp chip đơn NFC hỗ trợ chuẩn ISO/IEC 18092 dành cho các sản phẩm điện tử tiêu dùng như điện thoại thông minh (smartphone), máy tính sổ tay (notebook), và các thiết bị ngoại vi của máy tính. RF21S có chức năng bảo mật nhằm hỗ trợ cho các giao dịch qua thẻ, quét thẻ ở các trạm thu phí đường bộ, thẻ ID.

Với tính năng NFC, người dùng điện thoại di động có thể truy cập các ứng dụng không tiếp xúc để thanh toán qua di động, chia sẻ dữ liệu trực tiếp từ điện thoại Android. Giao diện lập trình ứng dụng NFC nguồn mở của NXP và Trusted Logic hiện đã có tích hợp đầy đủ vào trong khung làm việc (framework) Android.

2.1.4. So sánh với một số công nghệ không dây khác

Bluetooth: Là công nghệ không dây được thiết kế để truyền dữ liệu đến các thiết bị như điện thoại di động, máy tính xách tay và các thiết bị khác trong phạm vi 100m.

Công nghệ Wi-Fi: Dành cho mạng nội bộ (LAN) cho phép mở rộng hệ thống mạng hay thay thế cho mạng có dây trong phạm vi khoảng 100m.

ZigBee: Là công nghệ không dây có khả năng kiểm soát và giám sát các ứng dụng trong ngành công nghiệp và khu nhà ở trong phạm vi hơn 100m.

IrDA: Là chuẩn liên lạc không dây tầm ngắn ($< 1\text{m}$), truyền dữ liệu qua tia hồng ngoại. Giao diện IrDA thường được dùng ở các máy tính và điện thoại di động.

RFID: Là phương thức nhận dạng tự động, cho phép lưu trữ và lấy dữ liệu từ xa dựa vào thẻ nhãn tần số vô tuyến. Thẻ RFID được gắn kèm theo sản phẩm. Hệ thống RFID gồm hai thành phần: thẻ nhãn có gắn chip silicon cùng ăng-ten và phần thứ 2 là bộ đọc giao tiếp với thẻ nhãn và truyền dữ liệu tới hệ thống máy tính trung tâm.

Thẻ thông minh: Không tiếp xúc (contactless smart card) tích hợp trong chip để truyền dữ liệu đến đầu đọc nhờ công nghệ RFID. Ví dụ, thẻ thông truyền dữ liệu qua chuẩn ISO/IEC 14443 và FeliCa trong phạm vi khoảng 10 cm.

2.2. Mô hình MVC

Mô hình MVC là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong kỹ thuật phần mềm.

Sử dụng mô hình MVC chúng ta có thể tách các ứng dụng thành 3 phần khác nhau để có thể tiến hành phát triển đồng thời hoặc phân chia công việc cho các nhóm một cách dễ dàng mà không phải chờ đợi nhau, phụ thuộc nhau vì các phần có chức năng và nhiệm vụ riêng biệt.

Các thành phần của mô hình MVC và chức năng của từng thành phần:

1. Model

Là thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất cơ sở dữ liệu, đối tượng mô tả dữ liệu như các class, hàm xử lý,... cung cấp cho dữ liệu cần thiết cùng các hàm để tương tác với View thông qua sự điều hướng của Controller.

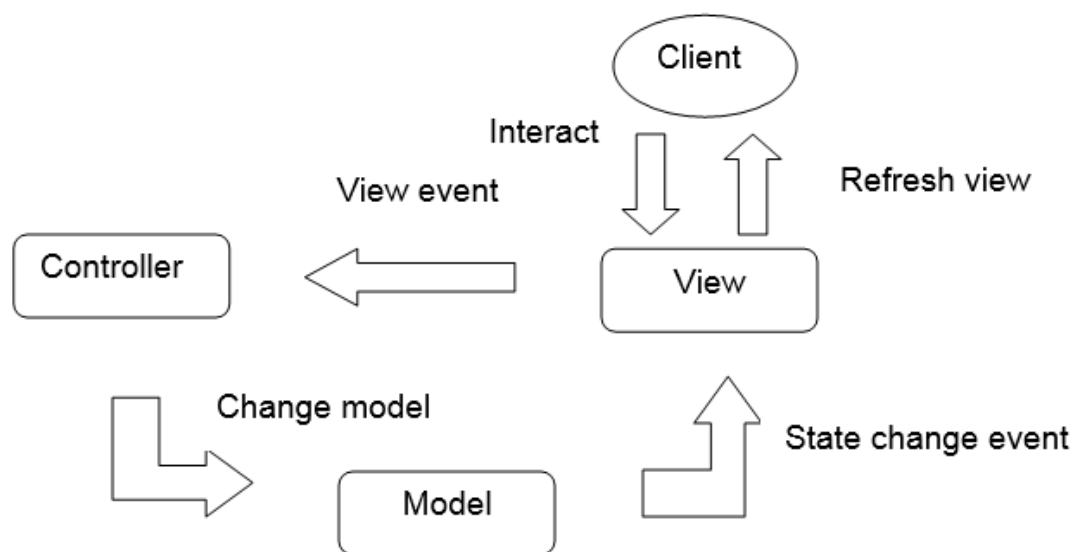
2. View

Đảm nhận công việc hiển thị thông tin, tương tác với người dùng, nơi chứa tất cả các đối tượng GUI như: textbox, images... Tiếp nhận các yêu cầu từ phía người dùng và các dữ liệu trả về từ phía Model thông qua điều hướng của Controller để hiển thị ra màn hình thông qua tập hợp các mã hoặc tương tự các mã HTML.

3. *Controller*

Giữ nhiệm vụ nhận các yêu cầu từ phía người dùng được truyền tới từ View, sau đó gọi các hàm xử lý tương ứng trong Model để xử lý và trả lại các dữ liệu hoặc thông báo theo đúng yêu cầu của người dùng.

Mô hình hoạt động:



Hình 2.1. Mô hình MVC

Ưu điểm:

1. *Phát triển phần mềm*

Mang tính chuyên nghiệp cao, có thể phát triển đồng thời các modul bởi các nhóm lập trình khác nhau. Sử dụng được trong tất cả các công đoạn từ thiết kế hệ thống cho đến lập trình ứng dụng hay tổ chức dữ liệu. Giúp phát triển phần mềm nhanh chóng, đơn giản, dễ bảo trì, nâng cấp...

2. *Bảo trì*

Do các thành phần có thể thay đổi một cách dễ dàng mà không ảnh hưởng đến các thành phần khác của hệ thống hay ứng dụng, sự thay đổi được cô lập trong

từng lớp của hệ thống hoặc chỉ ảnh hưởng nhỏ đến lớp gần kề nên việc bảo trì có thể tiến hành một cách đơn giản, không mất nhiều thời gian, chi phí.

3. Mở rộng

Mô hình MVC do việc phân chia theo các lớp Model – View – Controller giúp chúng ta dễ dàng thêm, xóa hoặc sửa các chức năng cho từng lớp nên việc mở rộng cũng dễ dàng hơn nhiều so với các mô hình phân chia khác như mô hình 3 lớp ...

Nhược điểm:

Việc áp dụng mô hình MVC mang lại nhiều lợi ích nhưng nó cũng tồn tại nhược điểm của nó như chồng kênh, tốn thời gian nếu áp dụng vào dự án nhỏ bởi phải trải qua nhiều công đoạn thiết kế và lắp ghép modul. Ngoài ra, do tồn tại 3 lớp với lớp trung chuyển là Controller nên nó cũng mất thời gian trung chuyển dữ liệu giữa các thành phần[1].

2.3. Framework Struts 2

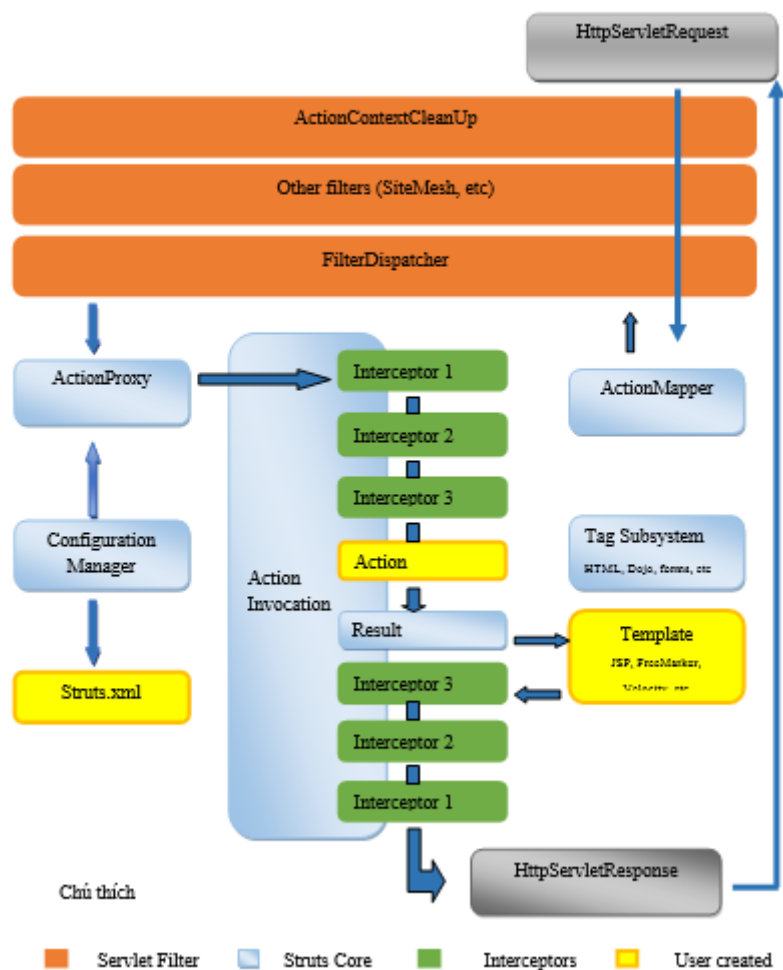
Framework Struts 2 là framework hỗ trợ thiết kế và xây dựng ứng dụng Web theo mô hình MVC bằng cách cung cấp các thư viện và tiện ích để hỗ trợ người dùng trong việc phát triển ứng dụng Web một cách nhanh chóng và dễ dàng.

Được phát triển dựa trên phiên bản Struts 1 theo hướng đơn giản hóa các cấu hình phức tạp đồng thời thể hiện rõ ràng mô hình ứng dụng MVC một cách hoàn chỉnh.

Một số thay đổi được cải tiến trong Struts 2:

- JavaBeans được sử dụng thay thế Action form và có phương thức chỉ định để kích hoạt cơ chế thực thi nhằm tăng khả năng tái sử dụng của các đối tượng. Đồng thời, chúng còn giúp ích trong việc testing các thành phần chức năng một cách dễ dàng, nhanh chóng.
- Sử dụng annotation và tập tin cấu hình XML rút gọn.
- Sử dụng bộ taglib duy nhất cho 4-5 bộ taglib trong Struts 1 và JSTL1.1 trong JSP.

Mô hình kiến trúc:



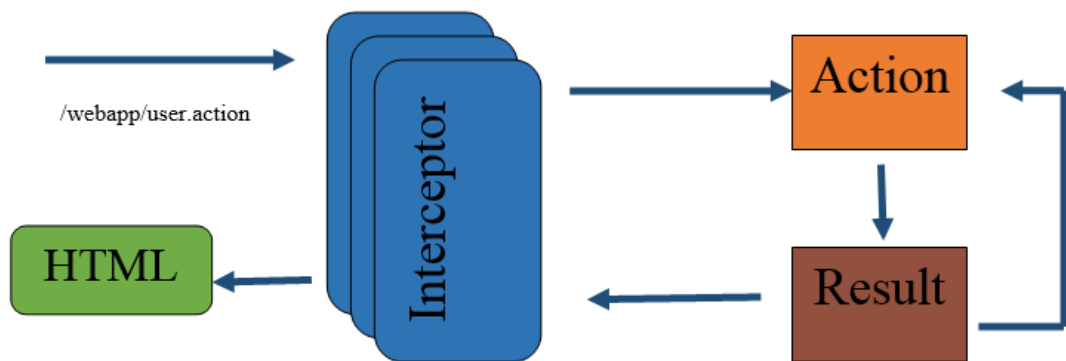
Hình 2.2. Mô hình kiến trúc Struts 2

1. Vòng đời của Struts bắt đầu khi có một yêu cầu được gửi từ phía Client. Yêu cầu này sẽ gửi đến ActionMapper.
2. ActionMapper quyết định xem khi nào thì một Action nên được gọi. Sau đó chuyển kết quả cho FilterDispatcher.
3. FilterDispatcher tiếp nhận kết quả từ ActionMapper và chuyển giao quyền điều khiển cho ActionProxy.
4. ActionProxy đọc file cấu hình chẳng hạn như struts.xml. ActionProxy tạo một thể hiện của lớp ActionInvocation và chuyển giao quyền điều khiển cho lớp này.
5. ActionInvocation chịu trách nhiệm thực hiện các lệnh được cài đặt. Nó gọi Interceptors nếu cần thiết, sau đó sẽ gọi Action.

6. Khi Action trả về, ActionInvocation có trách nhiệm tìm kết quả thích hợp liên quan với mã kết quả được ánh xạ trong struts.xml.

7. Interceptors được thực thi một lần nữa theo thứ tự ngược lại và trả lời đáp ứng cho Filter (Trong hầu hết các trường hợp là FilterDispatcher). Và kết quả được chuyển đến cho Servlet container và gửi trả lại cho Client.

Mô hình vòng đời:



Hình 2.3. Vòng đời của Struts 2

1. Yêu cầu được sinh ra bởi người dùng và được gửi đến Servlet container.
2. Servlet container gọi bộ lọc FilterDispatcher để lựa chọn các action thích hợp.
3. Từng Interceptors được thực hiện trước khi gọi Action. Interceptors thực hiện các công việc như Đăng nhập, Xác thực, Kiểm tra file upload...
4. Action được thực thi và sinh ra kết quả Result.
5. Đầu ra của Action được chuyển sang cho trang View (JSP, Velocity...) và kết quả cuối cùng được chuyển về phía người dùng[5].

2.4. Framework Hibernate

ORM framework là một cơ chế cho phép người lập trình thao tác với database một cách hoàn toàn tự nhiên thông qua các đối tượng. Lập trình viên hoàn toàn không quan tâm đến loại database sử dụng, SQL...

Persistence Layer:

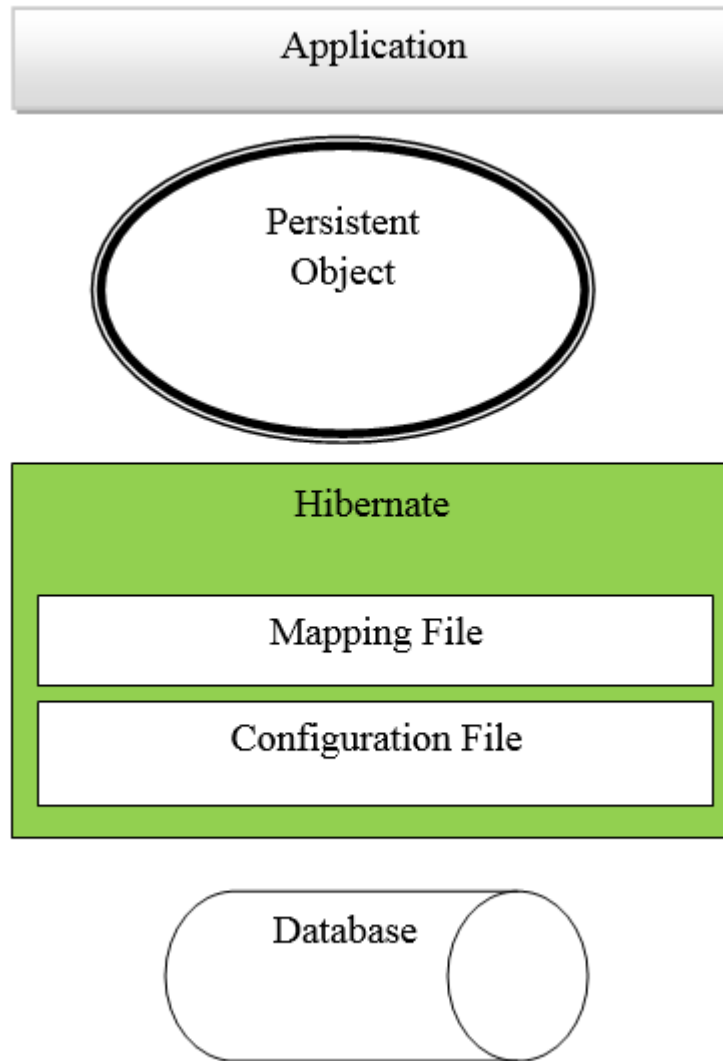
Một ứng dụng có thể được chia thành 3 phần cơ bản là:

- *Presentation layer*: Giao diện người dùng.
- *Business layer*: Phần xử lý nghiệp vụ. Gồm 2 layer con là:
 - + *Business logic layer*: Các tính toán nhằm thỏa mãn yêu cầu người dùng.
 - + *Persistence layer*: Chịu trách nhiệm giao tiếp với data layer (thường là một Relational DBMS), đảm nhiệm các nhiệm vụ mở kết nối, truy xuất và lưu trữ dữ liệu vào các Relational DBMS.
- *Data layer*: Phần chứa dữ liệu.

Hibernate framework là một trong những ORM Framework. Hibernate framework là một framework cho persistence layer. Sử dụng Hibernate framework giúp chúng ta không cần chú tâm vào persistence layer mà chỉ cần quan tâm đến các layer khác. Hibernate giúp lưu trữ và truy vấn dữ liệu quan hệ mạnh mẽ và nhanh. Hibernate cho phép bạn truy vấn dữ liệu bằng ngôn ngữ SQL mở rộng của Hibernate hoặc bằng SQL thuần.

Không giống như các persistence layer khác, Hibernate không ẩn đi sức mạnh của ngôn ngữ SQL thuần túy mà Hibernate còn đảm bảo cho người phát triển đầu tư vào công nghệ và tri thức cơ sở dữ liệu quan hệ luôn chính xác. Mặt khác, Hibernate được license theo LGPL. Vì vậy, chúng ta có thể sử dụng Hibernate trong các dự án open source hoặc các dự án thương mại.

Kiến trúc Hibernate:



Hình 2.4. Mô hình kiến trúc Hibernate.

1. Mỗi table trong database là một object trong Hibernate. Do đó, bạn cần có một java bean cho mỗi table trong database. Các java bean này sẽ có các getters/setters và một số ngoại lệ theo quy ước của Hibernate.

2. Mỗi file mapping (ánh xạ) có dạng *.hbm.xml có nhiệm vụ đặc tả mối quan hệ giữa các thuộc tính của đối tượng và các trường trong bảng CSDL.

3. File Hibernate.cfg.xml

Đây là tập tin được load đầu tiên khi khởi chạy ứng dụng Hibernate. Nó chứa các thông tin sau:

+ Connection JDBC (URL , Driver class, Username, password, pool_size, ...).

- + SQL Dialect.
- + Hibernate configuration (show_sql, format_sql, use_sql_comment, default_schema, order_updates,...).
- + Hibernate cache configuration (class cache, collection cache).
- + Hibernate transaction configuration (factory_class, auto_close_session, manager_lookup_class).
- + Miscellaneous configuration (current_session_context_class, factory_class, ...).
- + Mapping source configuration[2].

2.5. Hệ quản trị cơ sở dữ liệu

Hiện nay có 2 loại hệ quản trị cơ sở dữ liệu chính là: SQL và NoSQL.

SQL là hệ quản trị cơ sở dữ liệu quan hệ, các dữ liệu được tổ chức thành các bảng dữ liệu với các cột là các trường thuộc tính của dữ liệu và các hàng là các bản ghi dữ liệu riêng biệt. Các bảng phân biệt nhau bởi tên bảng, các bản ghi phân biệt với nhau bằng khoá và các bảng khác nhau liên kết với nhau thông qua khoá ngoài.

NoSQL là từ viết tắt của Non-Relational – không ràng buộc. Tuy nhiên, thuật ngữ này ít phổ dụng và ngày nay người ta thường dịch NoSQL thành Not Only SQL. NoSQL là những cơ sở dữ liệu không dùng mô hình dữ liệu quan hệ để quản lý dữ liệu trong lĩnh vực phần mềm.

Trong phạm vi khóa luận này tôi sử dụng hệ quản trị cơ sở dữ liệu loại SQL là MySQL.

MySQL

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng.

MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyên, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. MySQL miễn phí hoàn toàn cho nên có thể sử dụng thoải mái mà không lo vấn đề về chi phí.

MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS, ...

MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL được sử dụng cho việc hỗ trợ PHP, Perl, và nhiều ngôn ngữ khác, nó làm nơi lưu trữ những thông tin trên các trang Web viết bằng PHP hay Perl...

2.6. Dịch vụ Web và REST

2.6.1. Dịch vụ Web

Theo định nghĩa của W3C, dịch vụ Web là một hệ thống phần mềm được thiết kế để hỗ trợ khả năng tương tác giữa các ứng dụng trên các máy tính khác nhau thông qua mạng Internet, giao diện chung và sự gắn kết của nó được mô tả bằng XML. Dịch vụ Web là tài nguyên phần mềm có thể xác định bằng địa chỉ URL, thực hiện các chức năng và đưa ra các thông tin người dùng yêu cầu. Một dịch vụ Web được tạo nên bằng cách lấy các chức năng và đóng gói chúng sao cho các ứng dụng khác dễ dàng nhìn thấy và có thể truy cập đến những dịch vụ mà nó thực hiện, đồng thời có thể yêu cầu thông tin từ dịch vụ Web khác. Nó bao gồm các mô-đun độc lập cho hoạt động của khách hàng và doanh nghiệp và bản thân nó được thực thi trên server.

Đặc điểm:

- Dịch vụ Web cho phép client và server tương tác được với nhau ngay cả trong những môi trường khác nhau. Ví dụ, đặt Web server cho ứng dụng trên một máy chủ chạy hệ điều hành Linux trong khi người dùng sử dụng máy tính chạy hệ điều hành Windows, ứng dụng vẫn có thể chạy và xử lý bình thường mà không cần thêm yêu cầu đặc biệt để tương thích giữa hai hệ điều hành này.
- Phần lớn kĩ thuật của dịch vụ Web được xây dựng dựa trên mã nguồn mở và được phát triển từ các chuẩn đã được công nhận, ví dụ như XML.
- Một dịch vụ Web bao gồm có nhiều mô-đun và có thể công bố lên mạng Internet.
- Là sự kết hợp của việc phát triển theo hướng từng thành phần với những lĩnh vực cụ thể và cơ sở hạ tầng Web, đưa ra những lợi ích cho cả doanh nghiệp, khách hàng, những nhà cung cấp khác và cả những cá nhân thông qua mạng Internet.
- Một ứng dụng khi được triển khai sẽ hoạt động theo mô hình client-server. Nó có thể được triển khai bởi một phần mềm ứng dụng phía server ví dụ như PHP, Oracle Application server hay Microsoft.Net...
- Ngày nay dịch vụ Web đang rất phát triển, những lĩnh vực trong cuộc sống có thể áp dụng và tích hợp dịch vụ Web là khá rộng lớn như dịch vụ chọn lọc và phân loại tin tức (hệ thống thư viện có kết nối đến web portal để tìm kiếm các thông tin cần thiết); ứng dụng cho các dịch vụ du lịch (cung cấp giá vé, thông tin về địa

điểm...), các đại lý bán hàng qua mạng, thông tin thương mại như giá cả, tỉ giá hối đoái, đấu giá qua mạng...hay dịch vụ giao dịch trực tuyến như đặt vé máy bay, thông tin thuê xe...

- Các ứng dụng có tích hợp dịch vụ Web đã không còn là xa lạ, đặc biệt trong điều kiện thương mại điện tử đang bùng nổ và phát triển không ngừng cùng với sự lớn mạnh của Internet. Bất kì một lĩnh vực nào trong cuộc sống cũng có thể tích hợp với dịch vụ Web, đây là cách thức kinh doanh và làm việc có hiệu quả bởi thời đại ngày nay là thời đại của truyền thông và trao đổi thông tin qua mạng. Do vậy, việc phát triển và tích hợp các ứng dụng với dịch vụ Web đang được quan tâm phát triển là điều hoàn toàn dễ hiểu.

Ưu điểm:

1. Dịch vụ Web cung cấp khả năng hoạt động rộng lớn với các ứng dụng phần mềm khác nhau chạy trên những nền tảng khác nhau.
2. Sử dụng các giao thức và chuẩn mở. Giao thức và định dạng dữ liệu dựa trên văn bản (text), giúp các lập trình viên dễ dàng hiểu được.
3. Nâng cao khả năng tái sử dụng.
4. Thúc đẩy đầu tư các hệ thống phần mềm đã tồn tại bằng cách cho phép các tiến trình/chức năng nghiệp vụ đóng gói trong giao diện dịch vụ Web.
5. Tạo mối quan hệ tương tác lẫn nhau và mềm dẻo giữa các thành phần trong hệ thống, dễ dàng cho việc phát triển các ứng dụng phân tán.
6. Thúc đẩy hệ thống tích hợp, giảm sự phức tạp của hệ thống, hạ giá thành hoạt động, phát triển hệ thống nhanh và tương tác hiệu quả với hệ thống của các doanh nghiệp khác.

Nhược điểm:

1. Những thiệt hại lớn sẽ xảy ra vào khoảng thời gian chết của Dịch vụ Web, giao diện không thay đổi, có thể lỗi nếu một máy khách không được nâng cấp, thiếu các giao thức cho việc vận hành.
2. Có quá nhiều chuẩn cho dịch vụ Web khiến người dùng khó nắm bắt.
3. Phải quan tâm nhiều hơn đến vấn đề an toàn và bảo mật.

Nền tảng Web services bao gồm có:

- XML – eXtensible Markup Language

- SOAP – Simple Object Access Protocol
- UDDI – Uniseral Description, Discovery and Intergration
- WSDL – Web Services Description Language [7]

2.6.2. REST

REST đã được chọn sử dụng rộng rãi thay cho Web service dựa trên SOAP và WSDL.

REST định nghĩa các quy tắc kiến trúc để thiết kế Web service chú trọng vào tài nguyên hệ thống, bao gồm các trạng thái tài nguyên được định dạng như thế nào và được chuyển tải qua HTTP thông qua số lượng lớn người dùng và được viết bởi những ngôn ngữ khác nhau.

Nếu tính theo số dịch vụ mạng sử dụng, REST đã nổi lên trong vài năm qua như một mô hình thiết kế dịch vụ chiếm ưu thế. Trong thực tế, REST đã có những ảnh hưởng lớn và gần như thay thế SOAP và WSDL vì nó đơn giản và dễ sử dụng hơn rất nhiều.

REST tuân thủ 4 quy tắc thiết kế:

1. Sử dụng phương thức HTTP một cách rõ ràng.

REST yêu cầu các nhà phát triển sử dụng phương thức HTTP một cách rõ ràng theo cách tương thích với giao thức chuẩn. Nguyên lý thiết kế REST cơ bản này thiết lập một ánh xạ 1-1 giữa các hành động tạo, đọc, cập nhật và xóa các phương thức HTTP.

Theo cách ánh xạ này thì:

- Để tạo một tài nguyên trên máy chủ, bạn cần sử dụng phương thức POST.
- Để truy xuất một tài nguyên, sử dụng GET.
- Để đổi trạng thái một tài nguyên hoặc cập nhật nó, sử dụng PUT.
- Để hủy bỏ hoặc xóa một tài nguyên, sử dụng DELETE.

2. Phi trạng thái.

Phi trạng thái theo nghĩa là nâng cao tính hiệu quả của dịch vụ Web, đơn giản hóa các thiết kế và sự thi hành của các thành phần của máy chủ vì khi máy chủ không có trạng thái sẽ hủy bỏ nhu cầu để đồng bộ hóa các mảng dữ liệu với một ứng dụng bên ngoài.

Nó giúp cho yêu cầu hoàn chỉnh, độc lập hơn vì không đòi hỏi máy chủ thu thập thông tin về ngữ cảnh hoặc trạng thái của ứng dụng trong lúc xử lý yêu cầu. Đồng thời, kéo theo đó là quy mô đáp ứng các yêu cầu của dịch vụ Web REST sẽ được điều chỉnh tốt hơn để phục vụ nhu cầu ngày càng cao về chất lượng thực hiện.

3. Hiện thị cấu trúc thư mục như URIs.

Các địa chỉ dịch vụ Web REST cần dễ dàng đối với người dùng. Có thể hiểu một đường dẫn như một giao diện tự đóng gói mà không cần nhiều lý giải hoặc tham chiếu. Cấu trúc một địa chỉ cần rõ ràng, có thể dự đoán được và dễ hiểu. Điều này được hiện thực hóa thông qua cấu trúc thư mục giống URIs.

Khi đó, một URI sẽ không chỉ là một chuỗi cắt không giới hạn mà còn là một cây với các nhánh chính và nhánh dọc nối với nhau tại các nút.

4. Chuyển đổi JSON và XML hoặc cả hai.

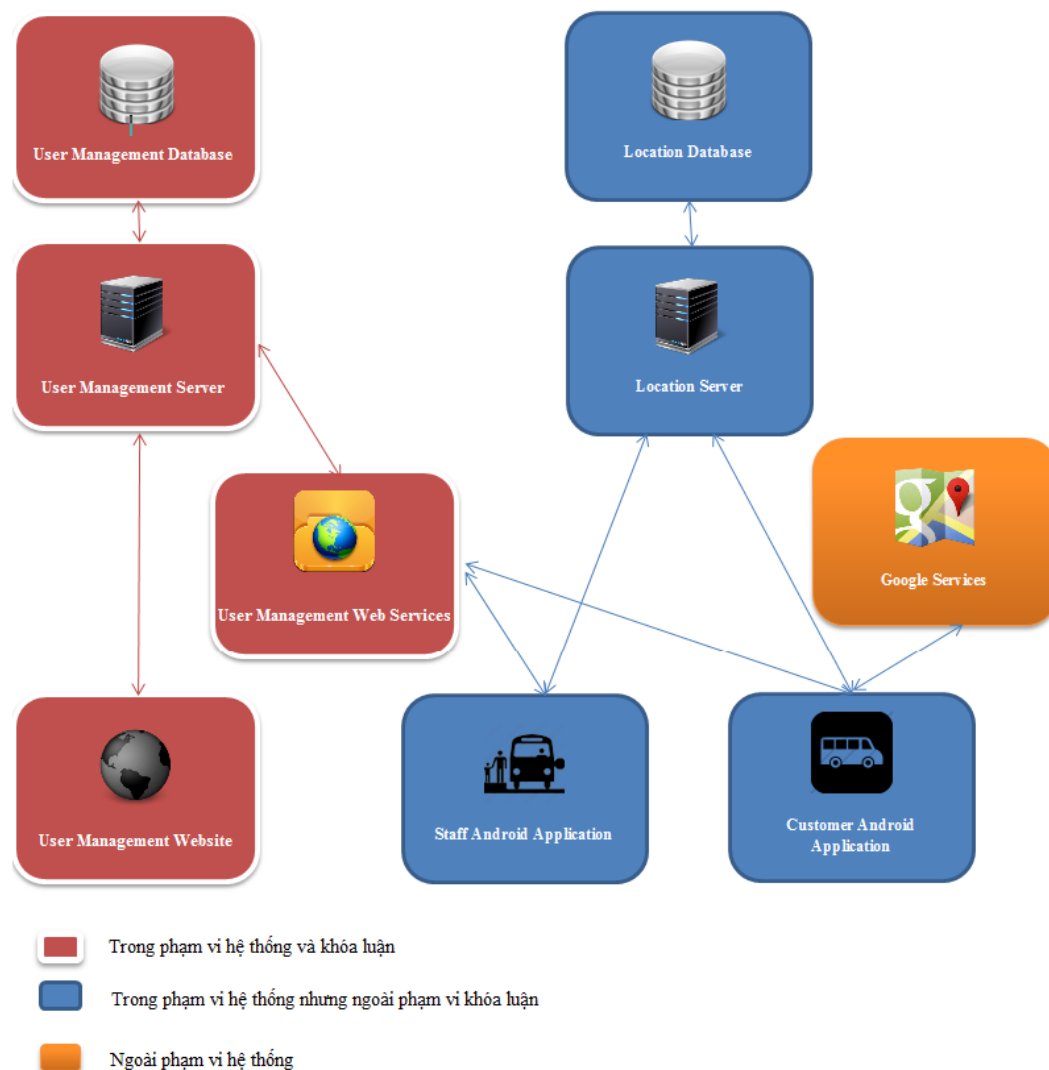
JSON và XML là 2 trong số những định dạng mà REST sử dụng, chúng có thể chuyển đổi qua lại với nhau nhằm phục vụ mục đích đa dạng của người dùng cũng như phù hợp với các trường hợp khác nhau[4].

CHƯƠNG 3

XÂY DỰNG HỆ THỐNG

3.1. Tổng quan hệ thống

Mô hình tổng quan hệ thống xác thực người dùng trên xe buýt.



Hình 3.1. Mô hình tổng quan hệ thống

Hệ thống được thiết kế dựa trên 2 thành phần chính là phía Client và phía Server.

Phía Client

1. Location Database: Lưu trữ các thông tin về tọa độ các điểm dừng đỗ xe buýt.

2. Location Server: Máy chủ nhận các truy vấn từ ứng dụng xác thực người dùng (Staff Android Application) và hỗ trợ người dùng (Customer Android Application) nhằm lưu trữ hoặc lấy thông tin về các tọa độ từ Location Database.

3. Staff Android Application: Ứng dụng Android được cài đặt trên các thiết bị Android cố định trên xe buýt. Có chức năng chính là cung cấp cho nhân viên cách xác thực người dùng thông qua thẻ NFC.

4. Customer Android Application: Ứng dụng Android được cài đặt trên thiết bị của khách hàng. Có chức năng chính là cung cấp thông tin tài khoản, lịch sử giao dịch cho khách hàng và cung cấp một bản đồ các tuyến đường xe buýt và vị trí hiện tại của các xe buýt cho khách hàng.

Phía Server

1. User Management Database: Lưu trữ các thông tin về khách hàng, các tuyến xe buýt hiện nay, thông tin các loại thẻ cơ bản và các giao dịch của khách hàng.

2. User Management Server: Máy chủ cho phép truy cập từ xa, cung cấp các tiện ích cơ bản cho khách hàng, nhân viên, kiểm soát viên và quản lý, đồng thời cung cấp các chức năng cơ bản để User Management Web Services có thể thao tác với CSDL.

3. User Management Website: Cung cấp giao diện trực quan cho người dùng có thể xem thông tin, chỉnh sửa thông tin hoặc xóa thông tin tùy thuộc vào quyền tương ứng của người dùng.

4. User Management Web Services: Sử dụng các chức năng mà User Management Server đã định nghĩa và cài đặt để thao tác với CSDL nhằm cung cấp các dịch vụ Web cho phía client nhằm xử lý các yêu cầu của hệ thống.

Ngoài ra, hệ thống tích hợp thêm Google Service để hiển thị các điểm dừng đỗ đã thực hiện trước đó trong một phạm vi nhất định theo ngày để người dùng có thể xem xét, tham khảo hoặc kiểm soát hoạt động của mình.

Quy trình thực hiện

Gồm 3 tiến trình cơ bản

Tiến trình 1: Thiết bị quét NFC được gắn trên các công cụ trên xe hoặc trên điện thoại tiến hành quét để lấy dữ liệu từ thẻ NFC.

Tiến trình 2: Dữ liệu sau khi được lấy ra sẽ được gửi lên Server để tiến hành xác thực và nhận các thông tin chi tiết về khách hàng tương ứng với mã thẻ cùng

các giao dịch liên quan. Ngoài ra, thiết bị cũng tự động gửi các thông tin về tuyến xe, nhân viên lên Server để tiến hành ghi lại các phiên giao dịch của khách hàng nếu thông tin khách hàng đã được xác thực là chính xác.

Tiến trình 3: Server tiến hành theo các quá trình được định nghĩa sẵn xác thực và ghi lại các phiên giao dịch của khách hàng. Đồng thời, gửi lại các thông tin cho phía Client có thể xem xét một cách trực quan. Thông tin trả về bao gồm toàn bộ thông tin đầy đủ của khách hàng như số dư tài khoản, họ tên, hình ảnh nhận diện... Khi đó, nếu đủ điều kiện sử dụng thì tiến hành cập nhật lại số dư tài khoản của mã thẻ tương ứng.

3.2. Phía Client

Do bạn Quan Tuấn Vũ tiến hành xây dựng các ứng dụng trên nền Android dựa trên các dịch vụ Web được cung cấp từ phía Server.

Được xây dựng thành hai ứng dụng phục vụ cho hai loại đối tượng chính là nhân viên kiểm soát vé và người dùng dịch vụ xe buýt:

1. Ứng dụng cho nhân viên kiểm soát vé.

Cho phép nhân viên xác thực người dùng dịch vụ xe buýt cùng các giao dịch xảy ra trong quá trình sử dụng. Đồng thời, cũng ghi lại một số thông tin cơ bản của nhân viên khi xác thực với từng người dùng cụ thể.

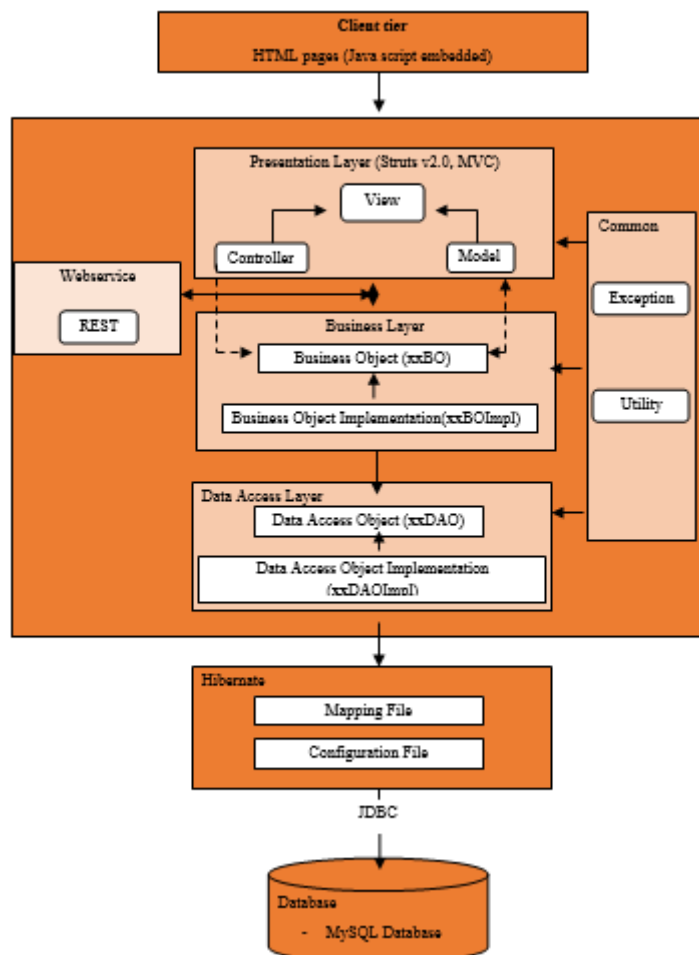
2. Ứng dụng cho người sử dụng dịch vụ xe buýt.

Cho phép người sử dụng có thể xem một số thông tin cơ bản của chính bản thân người dùng. Đồng thời, cho phép xem xét các giao dịch đã xảy ra của bản thân để có thể theo dõi lịch trình bản thân hoặc phục vụ mục đích riêng khác.

Ngoài ra, phía Client còn được tích hợp thêm tiện ích sử dụng Google Maps API để hiển thị lịch sử các địa điểm mà người sử dụng đã đi qua trong một phạm vi nhất định. Người sử dụng dịch vụ có thể thông qua đó để xem xét lịch biểu đã thực hiện để lên kế hoạch hoặc phục vụ cho công việc khác.

3.3. Phía Server

3.3.1. Mô hình tổng quan



Hình 3.2. Mô hình tổng quan phía Server

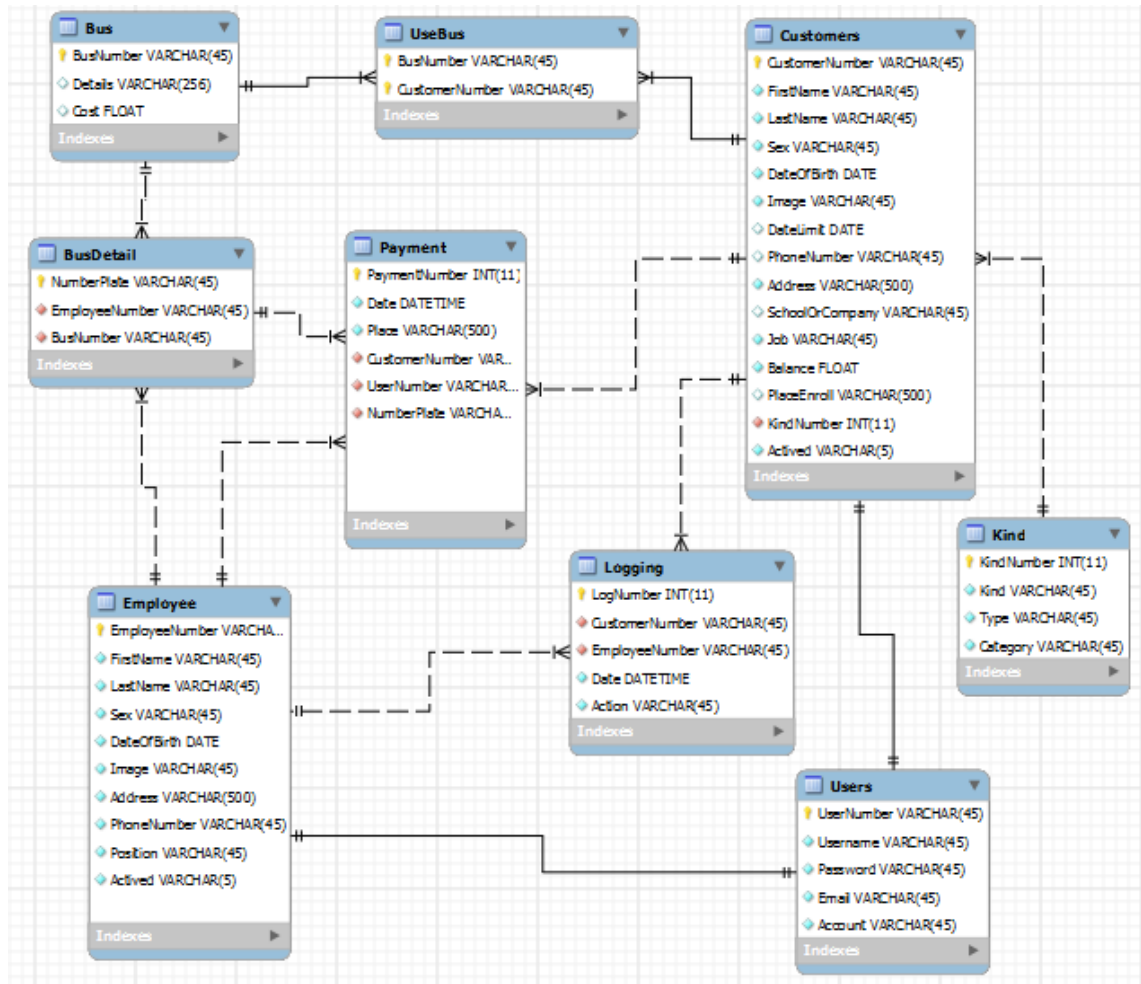
1. Presentational Layer: Sử dụng framework Struts 2 xây dựng theo mô hình MVC 2.0 trên nền Java.
2. Business Layer: Xây dựng dựa theo các Interface và các Implement tương ứng kết hợp với framework Hibernate để thao tác với cơ sở dữ liệu thông qua Persistence Layer.
3. Data Access Layer: Dùng để lấy các dữ liệu trên DBMS thông qua các Interface cùng các Implement tương ứng.
4. Common: Cung cấp một số hàm, hằng số sử dụng trong hệ thống.
5. Database: Cung cấp DBMS để sử dụng lưu trữ dữ liệu của khách hàng.

6. Webservice: Cung cấp các dịch vụ Web thông qua cấu trúc REST.

3.3.2. Lược đồ cơ sở dữ liệu

Cơ sở dữ liệu của hệ thống được xây dựng trên hệ quản trị cơ sở dữ liệu MySQL.

Mô hình CSDL



Hình 3.3. Lược đồ cơ sở dữ liệu

Bảng Users: Lưu trữ thông tin của các tài khoản dùng để đăng nhập vào hệ thống.

Bảng Bus: Lưu trữ tất cả các tuyến xe buýt hiện hành trên địa bàn Hà Nội cùng giá vé tương ứng cho từng tuyến.

Bảng BusDetail: Lưu trữ thông tin của các tài khoản Employee là các lái xe tương ứng với biển số xe cùng tuyến xe tương ứng với xe đó.

Bảng Customers: Lưu trữ thông tin của khách hàng – những người trực tiếp sử dụng dịch vụ xe buýt.

Bảng Payment: Lưu trữ lịch sử giao dịch của khách hàng theo thời gian.

Bảng Employee: Lưu trữ thông tin cơ bản của nhân viên kiểm soát vé trên mỗi tuyến xe buýt.

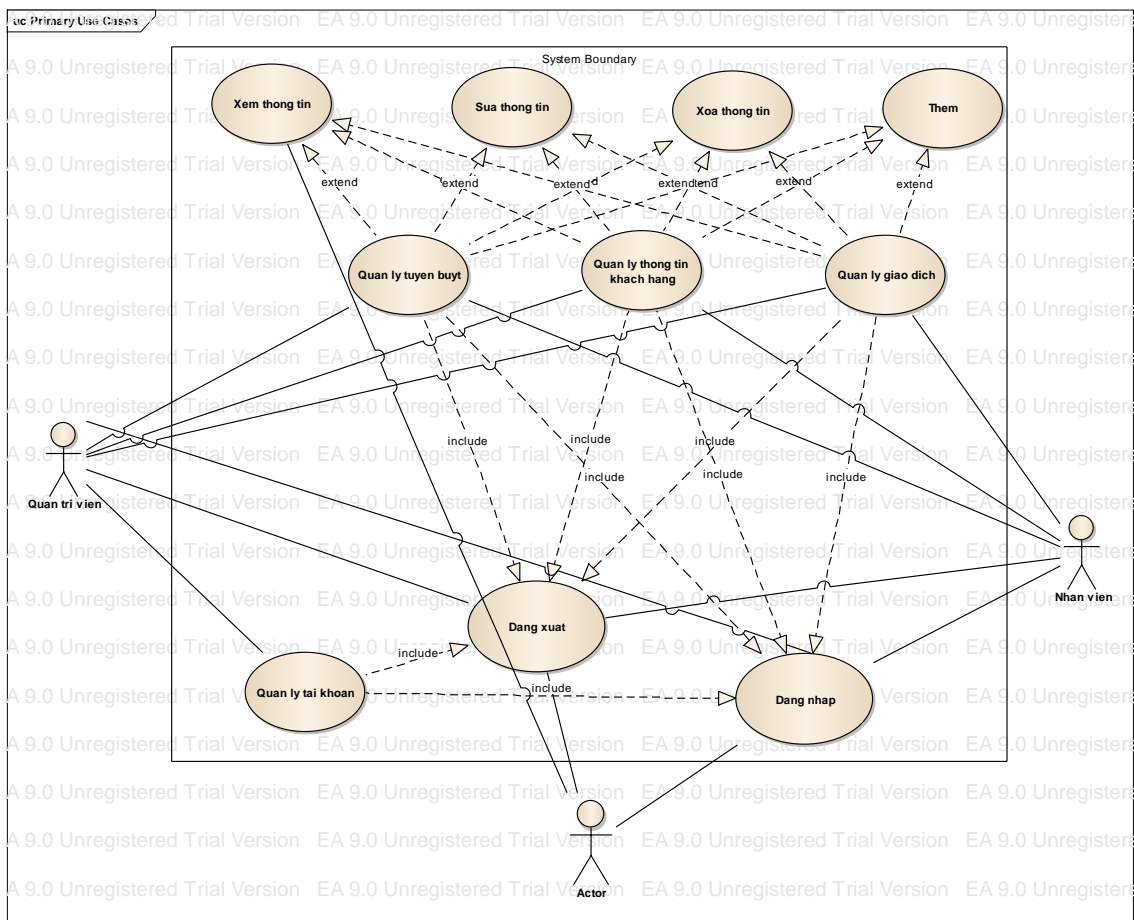
Bảng Logging: Lưu trữ lại một số hành động của các tài khoản Employee đối với khách hàng như xóa, sửa hay kích hoạt hoặc khóa tài khoản.

Bảng Usebus: Lưu trữ các tuyến xe buýt mà khách hàng đăng ký vé tháng theo kiểu một tuyến.

Bảng Kind: Lưu trữ các loại thẻ hiện có mặt trong quản lý xe buýt.

3.3.3. Mô hình ca sử dụng của hệ thống

Hệ thống được thiết kế gồm 6 ca sử dụng chính là đăng nhập, đăng xuất, quản lý tuyến buýt, quản lý thông tin khách hàng, quản lý giao dịch và quản lý tài khoản. Ngoài ra, có thêm 4 ca sử dụng phụ là các thành phần mở rộng hoặc chi tiết hóa cho một số ca sử dụng chính trong hệ thống.

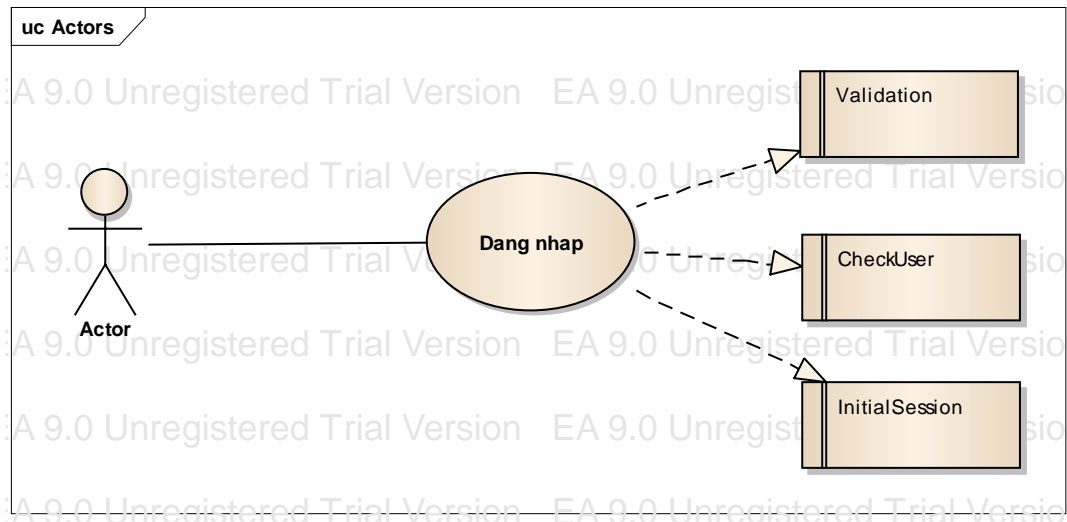


Hình 3.4. Mô hình ca sử dụng hệ thống.

3.3.4. Các ca sử dụng

3.3.4.1. Đăng nhập

Mô hình ca sử dụng



Hình 3.5. Mô hình ca sử dụng đăng nhập

Đặc tả ca sử dụng

Actor: Khách hàng, nhân viên và quản lý.

Mô tả: Cho phép actor đăng nhập vào hệ thống.

Luồng sự kiện:

- Luồng cơ bản: Chức năng này được gọi đến khi actor muốn đăng nhập vào hệ thống quản lý.
 - Tác nhân nhập thông tin về username và password.
 - Hệ thống thực hiện việc kiểm tra username và password. Sau đó, hệ thống thiết lập quyền phù hợp với actor và đưa actor vào trong hệ thống.
- Luồng thay thế:
 - Nếu việc kiểm tra username và password có lỗi, hệ thống sẽ thông báo cho actor biết và cho phép đăng nhập lại. Actor có thể bắt đầu lại hoạt động đăng nhập hoặc không.

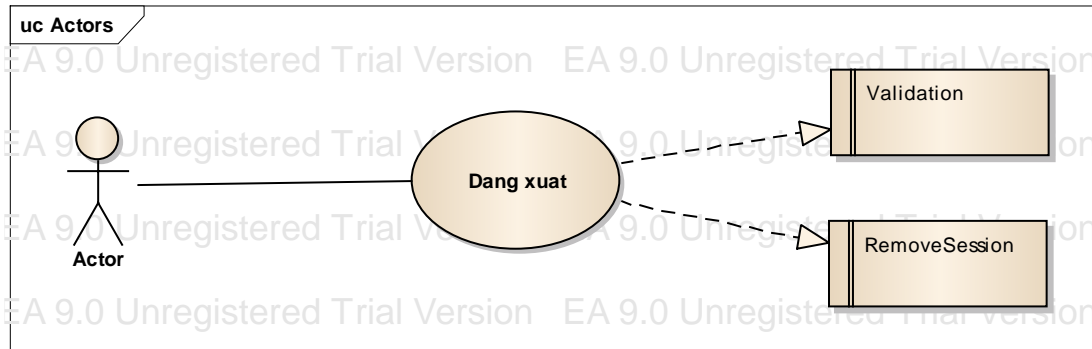
Mối quan hệ: Tất cả các use-case trong mô hình phải “include” use case đăng nhập.

Điều kiện trước: Hệ thống ở trong trạng thái đăng nhập và màn hình đăng nhập được hiển thị lên trình duyệt.

Điều kiện sau: Nếu đăng nhập thành công, actor ngay lập tức được đưa vào hệ thống cùng với các quyền sử dụng của mình. Nếu thất bại, trạng thái của hệ thống không thay đổi.

3.3.4.2. Đăng xuất

Mô hình ca sử dụng



Hình 3.6. Mô hình ca sử dụng đăng xuất

Đặc tả ca sử dụng

Actor: Khách hàng, nhân viên và quản lý.

Mô tả: Cho phép actor thoát khỏi hệ thống.

Luồng sự kiện:

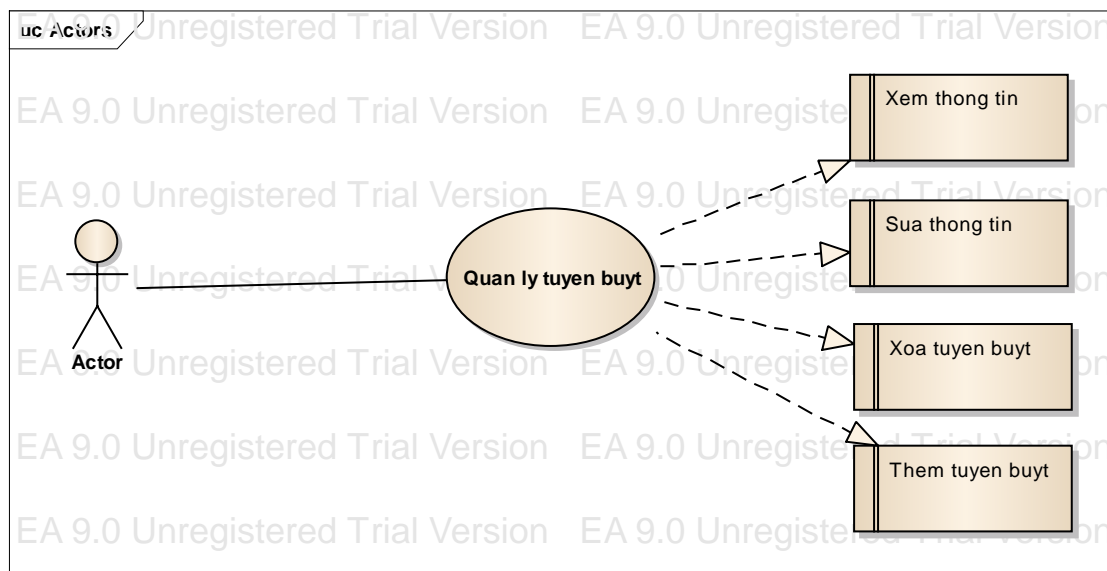
- Luồng cơ bản: Chức năng này được gọi đến khi actor muốn thoát khỏi hệ thống quản lý.
 - Tiến hành xóa bỏ session đã được khởi tạo khi đăng nhập.
 - Đóng tất cả các thao tác đang thực hiện hoặc chuẩn bị thực hiện.
- Luồng thay thế:
 - Nếu còn việc đang trong tiến trình thực hiện thì tiến hành chờ hoặc thông báo cho actor.

Điều kiện trước: Hệ thống đang trong trạng thái đăng nhập.

Điều kiện sau: Nếu đăng xuất thành công thì màn hình chuyển về giao diện đăng nhập ban đầu để chuẩn bị cho phiên làm việc mới.

3.3.4.3. Quản lý tuyển buýt

Mô hình ca sử dụng



Hình 3.7. Mô hình ca sử dụng quản lý tuyển buýt

Đặc tả ca sử dụng

Actor: Khách hàng, nhân viên và quản lý.

Mô tả: Cho phép actor xem các thông tin, sửa thông tin, xóa hoặc thêm các tuyến xe buýt.

Luồng sự kiện:

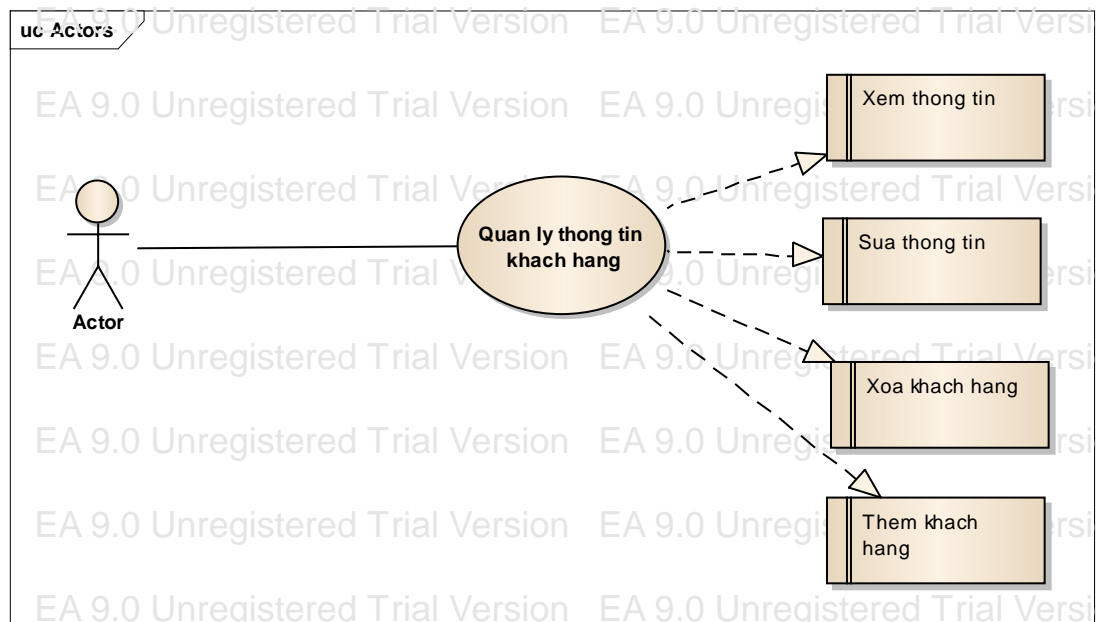
- Luồng cơ bản: Chức năng này được gọi đến khi actor muốn gọi chức năng liên quan tùy theo quyền tương ứng các giao dịch và các tuyến xe buýt.
 - Tiến hành xác nhận yêu cầu từ actor.
 - Xác định xem quyền của actor đối với yêu cầu có thỏa mãn hay không.
 - Hiện thị thông tin ra màn hình cho actor hoặc thông báo tương ứng.
- Luồng thay thế:
 - Nếu có lỗi, cần phải thông báo lại cho actor.

Điều kiện trước: Hệ thống đang trong trạng thái đăng nhập.

Điều kiện sau: Actor nhận được thông tin đúng theo yêu cầu hoặc thông báo tương ứng với các trường hợp xảy ra.

3.3.4.4. Quản lý thông tin khách hàng

Mô hình ca sử dụng



Hình 3.8. Mô hình ca sử dụng quản lý thông tin khách hàng

Đặc tả ca sử dụng

Actor: Khách hàng, nhân viên và quản lý.

Mô tả: Cho phép actor xem thông tin, cập nhật, xóa hoặc thêm người dùng mới.

Luồng sự kiện:

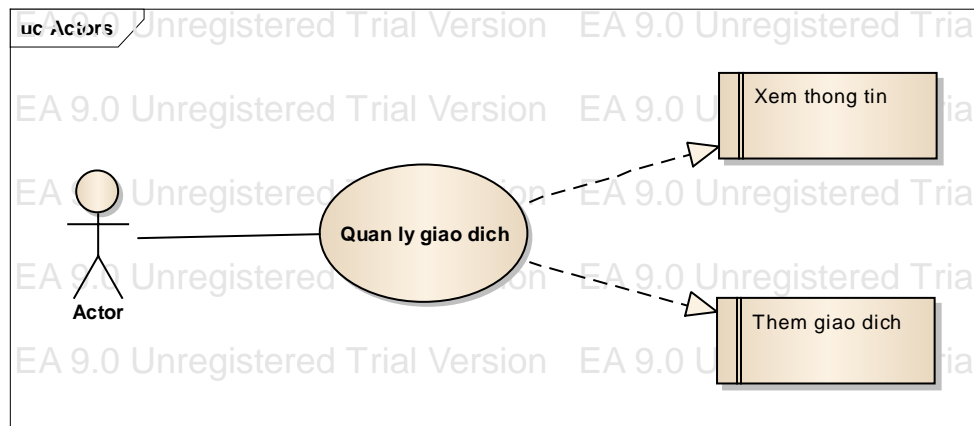
- Luồng cơ bản: Chức năng này được gọi đến khi gọi chức năng tương ứng.
 - Server xác nhận yêu cầu của actor truyền lên.
 - Tiến hành xác thực quyền và thực thi yêu cầu của actor.
 - Trả lại thông tin actor yêu cầu hoặc thông báo tương ứng.
- Luồng thay thế:
 - Nếu có lỗi, cần phải thông báo lại cho actor.

Điều kiện trước: Hệ thống đang trong trạng thái đăng nhập.

Điều kiện sau: Actor nhận được thông tin đúng theo yêu cầu hoặc thông báo tương ứng với các trường hợp xảy ra.

3.3.4.5. Quản lý giao dịch

Mô hình ca sử dụng



Hình 3.9. Mô hình ca sử dụng quản lý giao dịch

Đặc tả ca sử dụng

Actor: Khách hàng, nhân viên và quản lý.

Mô tả: Cho phép actor xem thông tin giao dịch hoặc tiến hành thêm các giao dịch mới.

Luồng sự kiện:

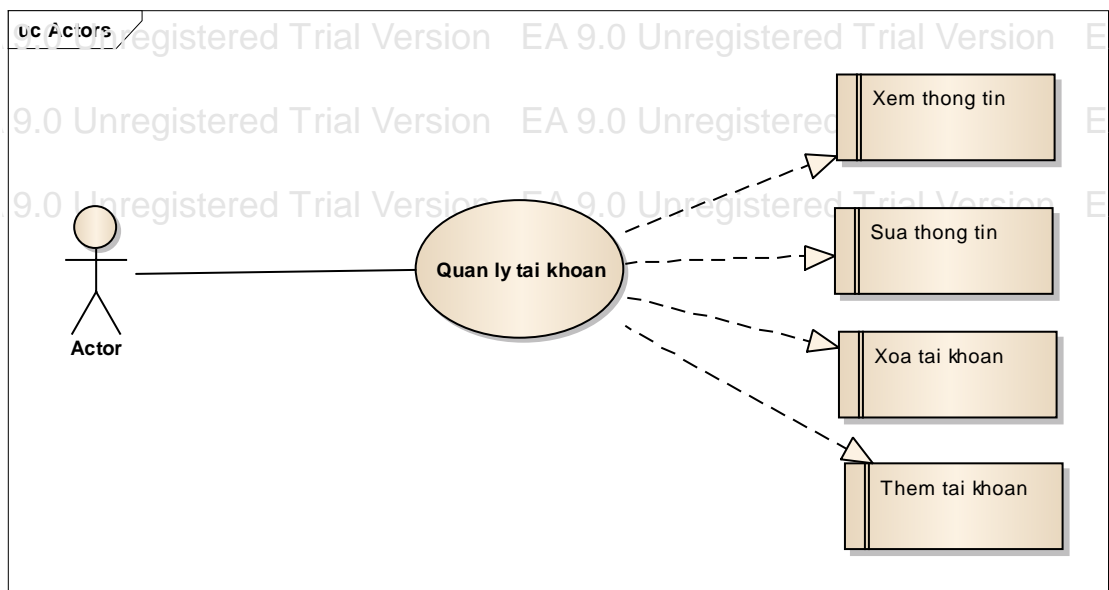
- Luồng cơ bản: Chức năng này được gọi đến khi actor gọi chức năng tương ứng với quyền được cấp.
 - Nhận yêu cầu từ actor.
 - Tiến hành xác nhận yêu cầu và thực thi yêu cầu của actor.
 - Trả lại thông tin actor yêu cầu hoặc thông báo tương ứng.
- Luồng thay thế:
 - Nếu có lỗi, cần phải thông báo lại cho actor.

Điều kiện trước: Hệ thống đang trong trạng thái đăng nhập.

Điều kiện sau: Actor nhận được thông tin đúng theo yêu cầu hoặc thông báo tương ứng với yêu cầu.

3.3.4.6. Quản lý tài khoản

Mô hình ca sử dụng



Hình 3.10. Mô hình ca sử dụng quản lý tài khoản

Đặc tả ca sử dụng

Actor: Quản lý.

Mô tả: Cho phép một người quản trị xem thông tin, cập nhật, xóa hoặc thêm các tài khoản mới.

Luồng sự kiện:

- Luồng cơ bản: Chức năng này được gọi đến khi người quản trị gọi chức năng tương ứng.
 - Quản lý chọn chức năng tương ứng như thêm mới, sửa, xóa tài khoản.
 - Thêm hoặc thay đổi các thông tin ban đầu cho phù hợp với thông tin mới.
 - Gửi yêu cầu cùng tập hợp các dữ liệu lên server.
 - Server tiến hành xác định yêu cầu, thực thi và trả lại thông báo theo từng trường hợp.
- Luồng thay thế:
 - Nếu có lỗi, cần phải thông báo lại cho người dùng.

Điều kiện trước: Hệ thống đang trong trạng thái đăng nhập.

Điều kiện sau: Sau khi thêm dữ liệu mới cần hiển thị thông báo tương ứng để người thao tác có thể biết và tiến hành thao tác tiếp theo.

3.3.5. Dịch vụ Web cung cấp cho phía Client

Được cài đặt trên máy chủ của Đại học Quốc gia Hà Nội với địa chỉ truy cập là 112.137.130.45 qua cổng 8080.

Mỗi chức năng được cung cấp theo một phương thức nhất định thông qua kiến trúc REST và được định dạng kiểu trả về là JSON – Một kiểu dữ liệu được REST cung cấp.

3.3.5.1. Đăng nhập

Được cung cấp qua:

Đường dẫn 1

http://112.137.130.45:8080/Webservice/REST/WebService/login_id đăng nhập bằng ID thẻ NFC dành cho người dùng với tham số đầu vào ID thẻ NFC mà họ sử dụng.

Đường dẫn 2

http://112.137.130.45:8080/Webservice/REST/WebService/login_uname đăng nhập thông qua tên đăng nhập và mật khẩu của người dùng khi họ không sử dụng thẻ hay không tiện sử dụng thẻ. Tham số đầu vào là tên đăng nhập và mật khẩu của người dùng đã được cung cấp từ trước.

3.3.5.2. Đăng xuất

Được cung cấp qua:

Đường dẫn 1

http://112.137.130.45:8080/Webservice/REST/WebService/logout_id dành cho người dùng đăng xuất khi tiến hành đăng nhập bằng ID thẻ NFC.

Đường dẫn 2

http://112.137.130.45:8080/Webservice/REST/WebService/logout_uname được gọi đến khi người dùng đăng xuất trong chế độ đăng nhập bằng tên đăng nhập và mật khẩu.

3.3.5.3. Lấy thông tin khách hàng

Được cung cấp qua đường dẫn

<http://112.137.130.45:8080/Webservice/REST/WebService/customer> được gọi đến khi nhân viên muốn xác thực người sử dụng hay khách hàng muốn xem thông tin của mình tùy theo tham số ID thẻ được truyền lên từ phía Client.

3.3.5.4. Lấy các giao dịch khách hàng

Được cung cấp qua đường dẫn

http://112.137.130.45:8080/Webservice/REST/WebService/history_customer được sử dụng khi nhân viên kiểm soát muốn kiểm tra các giao dịch của khách hàng hoặc khách hàng muốn xem lịch sử giao dịch của chính bản thân.

3.3.5.5. Ghi lại các giao dịch của khách hàng với nhân viên

Được cung cấp qua đường dẫn

<http://112.137.130.45:8080/Webservice/REST/WebService/payment> được gọi đến khi nhân viên tiến hành xác thực thông tin khách hàng xong, khách hàng đã đủ yêu cầu sử dụng dịch vụ thì tiến hành gửi các thông tin về ngày giờ, địa điểm cùng mã nhân viên và mã khách hàng lên cho phía Server để tiến hành lưu trữ lại các giao dịch phục vụ cho việc theo dõi, giám sát nhân viên cũng như các nhu cầu tiện ích khác của người dùng.

3.3.5.6. Lấy các giao dịch của nhân viên

Được cung cấp qua đường dẫn

http://112.137.130.45:8080/Webservice/REST/WebService/history_employee được nhân viên sử dụng khi muốn xem thông tin bản thân hoặc các tài khoản người dùng có quyền hạn cao hơn tiến hành xem xét thông tin nhân viên thông qua ID nhân viên được phía Client gửi truy vấn lên Server.

3.3.5.7. Khóa tài khoản

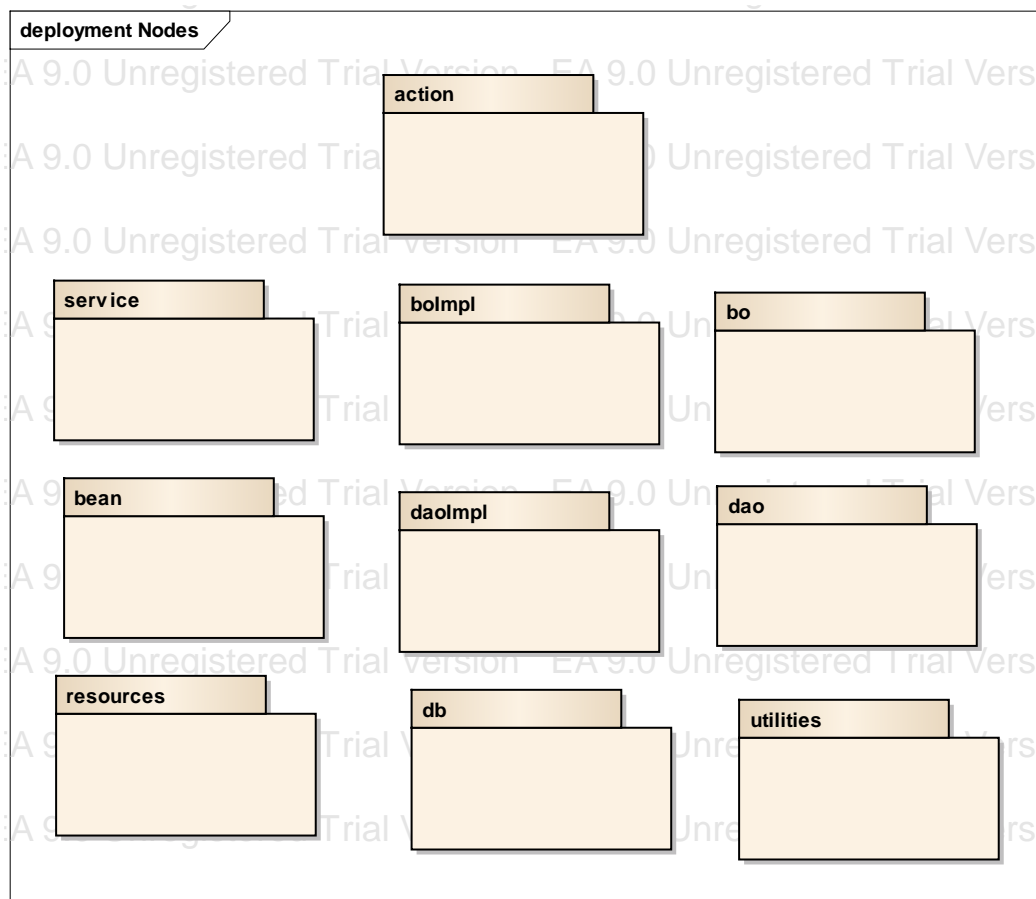
Được cung cấp qua đường dẫn

<http://112.137.130.45:8080/Webservice/REST/WebService/lock> khi kiểm soát viên thấy có sự không trùng khớp trong khâu kiểm soát vé của nhân viên thì kiểm soát viên có thể tiến hành khóa lại tài khoản đang được sử dụng của khách hàng và thông báo vi phạm của nhân viên về phía quản lý. Đồng thời, nhân viên cũng có thể tiến hành khóa

tài khoản của khách hàng khi phát hiện có sự gian lận trong việc xác thực người sử dụng.

3.3.6. Thiết kế gói dữ liệu

Mô hình gói dữ liệu.



Hình 3.11. Mô hình thiết kế gói dữ liệu

Các gói dữ liệu trong hệ thống gồm:

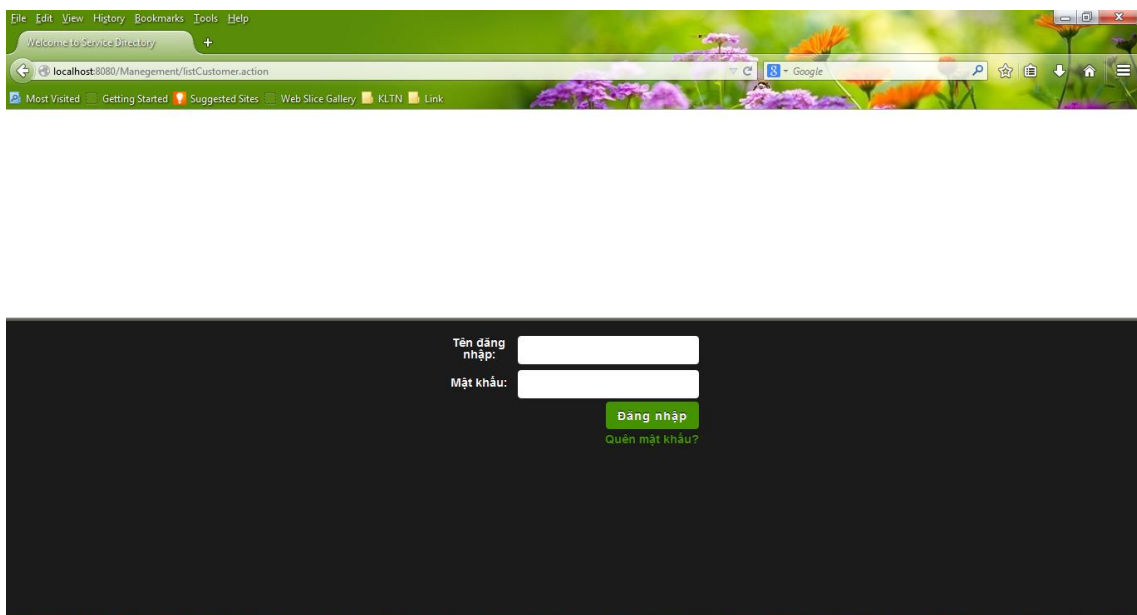
- **Bean:** Chứa các lớp thực thể được sử dụng trong hệ thống bao gồm: Bus, Payment, Kind, Users, Employee, Usebus và Customer.
- **Utilities:** Chứa các hằng, các tham số hoặc các hàm mặc định được sử dụng trong hệ thống.
- **Db:** Chứa class dùng để khởi tạo session truy cập đến CSDL bằng các hàm trong Hibernate framework.
- **Resources:** Chứa các file XML dùng để cấu hình các kết nối đến CSDL và chứa file điều hướng giữa View và Model trong hệ thống.
- **Dao:** Chứa các class dùng khai báo các interface được sử dụng trong kết nối với CSDL theo từng bảng, từng đối tượng.

- **DaoImpl:** Chứa các class thực hiện việc implement các interface được khai báo trong package dao.
- **Bo:** Chứa các class khai báo các hàm có thể sử dụng trong chương trình sử dụng các hàm cơ bản trong package dao.
- **BoImpl:** Chứa các class là các implement của các hàm được khai báo trong package bo.
- **Service:** Chứa class định nghĩa các hàm theo chuẩn REST dùng để cài đặt các dịch vụ Web.
- **Action:** Chứa các class định nghĩa các hàm tương ứng để sử dụng khi có yêu cầu từ phía người dùng. Đây là thể hiện của phần Model trong mô hình MVC.

3.3.7. Các giao diện chính

Giao diện đăng nhập

Cho phép người dùng đăng nhập thông qua “Tên đăng nhập” và “Mật khẩu”.

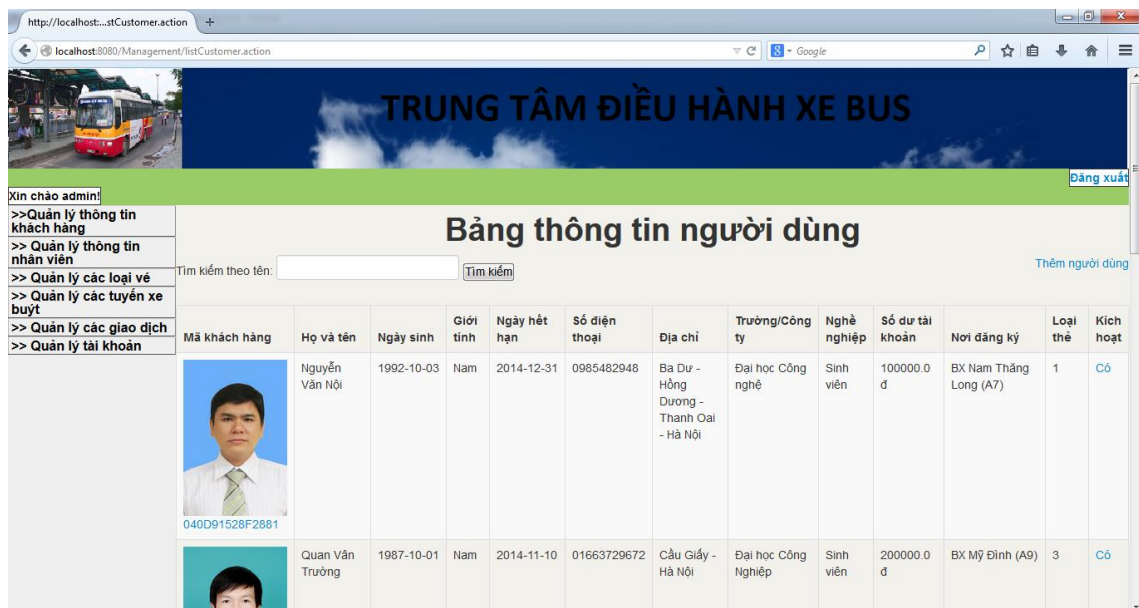


Hình 3.12. Giao diện đăng nhập

Giao diện quản lý thông tin khách hàng

Cho phép người sử dụng có thể xem thông tin của của bản thân.

Cho phép nhân viên có thể xem thông tin của tất cả các khách hàng, sửa thông tin, xóa hoặc thêm khách hàng mới.



Hình 3.13. Giao diện quản lý thông tin khách hàng

Giao diện quản lý các loại vé

Cho phép người sử dụng có thể xem các loại vé.

Cho phép nhân viên xem, xóa, sửa hoặc thêm một loại vé mới.



Hình 3.14. Giao diện quản lý các loại vé

Giao diện quản lý các tuyến xe buýt hiện nay

Cho phép người dùng xem thông tin cơ bản của các tuyến xe buýt hiện nay.

Cho phép nhân viên có thể xem các tuyến buýt, sửa, xóa hoặc thêm một tuyến buýt.

Tuyến xe bus	Chi tiết	Giá vé	Xóa
01	Bến xe Gia Lâm - Bến xe Yên Nghĩa	7000.0 đ	Xóa
02	Bắc Cỗ - Bến xe Yên Nghĩa	7000.0 đ	Xóa
03	Bến xe Giáp Bát - Bến xe Gia Lâm	7000.0 đ	Xóa
03B	Bến xe Giáp Bát - Vinhomes Riverside (Vincom Village)	7000.0 đ	Xóa
04	Long Biên - Bến xe Nước Ngầm	7000.0 đ	Xóa
05	Khu đô thị Linh Đàm - Phú Diễn (Trại Gà)	7000.0 đ	Xóa
06	Bến xe Giáp Bát - Cầu Giẽ	9000.0 đ	Xóa
07	Cầu Giấy - Sân bay Nội Bài	8000.0 đ	Xóa
08	Long Biên - Đồng Mỹ	7000.0 đ	Xóa
09	Bờ Hồ - Công viên Thống Nhất - Cầu Giấy - Bờ Hồ	7000.0 đ	Xóa

Hình 3.15. Giao diện quản lý các tuyến xe buýt

Giao diện quản lý các giao dịch của khách hàng

Cho phép người dùng xem lịch sử các giao dịch mà bản thân đã thực hiện.

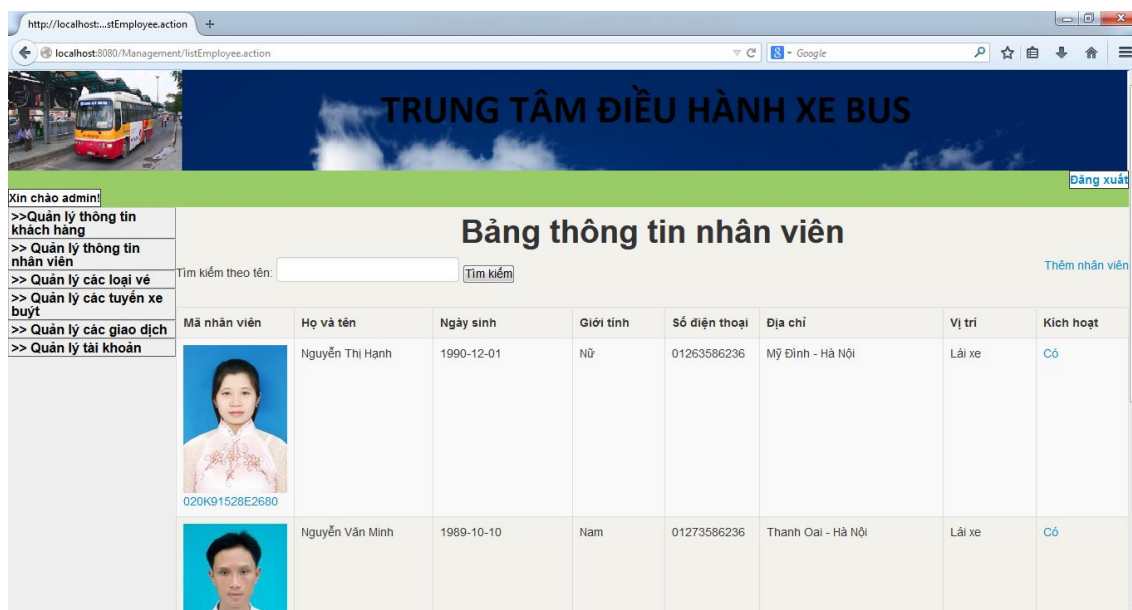
Cho phép nhân viên xem hoặc thêm các giao dịch thông qua phía Client.

Mã giao dịch	Ngày giao dịch	Địa điểm	Khách hàng	Nhân viên	Biển số xe	Xóa
1	2014-05-01 07:34:15.0	Bến xe Mỹ Đình, Phạm Hùng, Hà Nội, Hà Nội, Việt Nam	Quan Tuấn Vũ	Nguyễn Văn Minh	29A 00123	Xóa
2	2014-05-01 10:34:15.0	Sân vận động Quốc Gia, Lê Đức Thọ, Hà Nội, Hà Nội, Việt Nam	Quan Tuấn Vũ	Nguyễn Văn Minh	29A 00123	Xóa
3	2014-05-02 11:34:15.0	Sân vận động Quốc Gia, Lê Đức Thọ, Hà Nội, Hà Nội, Việt Nam	Nguyễn Văn Nội	Nguyễn Thị Hạnh	29A 00328	Xóa
4	2014-05-04 07:34:15.0	Bến xe Mỹ Đình, Phạm Hùng, Hà Nội, Hà Nội, Việt Nam	Quan Tuấn Vũ	Nguyễn Văn Minh	29A 00123	Xóa
5	2014-05-06 10:34:15.0	Sân vận động Quốc Gia, Lê Đức Thọ, Hà Nội, Hà Nội, Việt Nam	Quan Tuấn Vũ	Nguyễn Văn Minh	29A 00123	Xóa
6	2014-05-10 11:34:15.0	Bến xe Kim Mã, Kim Mã, Hà Nội, Việt Nam	Nguyễn Văn Nội	Nguyễn Thị Hạnh	29A 00328	Xóa

Hình 3.16. Giao diện quản lý các giao dịch của khách hàng

Giao diện quản lý thông tin nhân viên

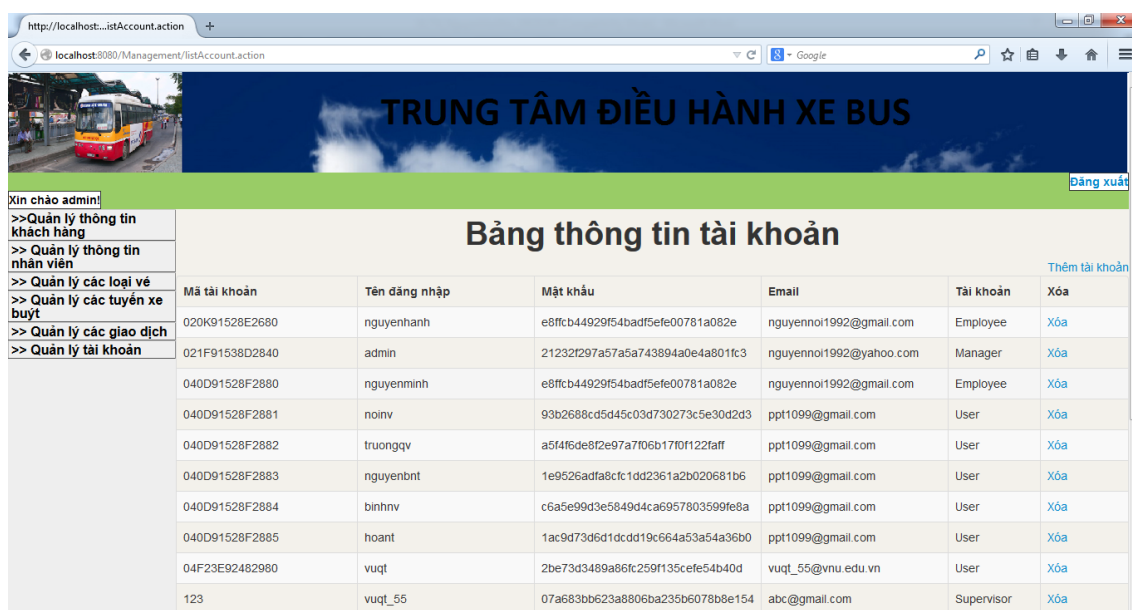
Cho phép quản lý có thể quản lý các thông tin của nhân viên cùng các chức năng thêm, sửa, kích hoạt hoặc khóa tài khoản nhân viên.



Hình 3.17. Giao diện quản lý thông tin nhân viên

Giao diện quản lý tài khoản

Cho phép quản lý có thể quản lý tất cả các tài khoản người dùng trong hệ thống với các chức năng xóa và thêm tài khoản người dùng.



Hình 3.18. Giao diện quản lý tài khoản

CHƯƠNG 4

THỰC NGHIỆM

4.1. Môi trường thực nghiệm

4.1.1. Cấu hình máy tính

Máy tính sử dụng trong thực nghiệm có cấu hình như sau:

Bảng 4.1. Cấu hình máy tính chạy thực nghiệm

TOSHIBA Satellite L40-A	
CPU	Intel(R) Core(TM) i5-3337U CPU @ 1.8GHz(4CPUs), ~1.8GHz
Memory	4GB
OS	Windows 7 Ultimate 64-bit (6.1, Build 7601)

4.1.2. Các công cụ phần mềm

Bảng sau trình bày về các công cụ, gói phần mềm, phiên bản của chúng mà tôi đã sử dụng để xây dựng hệ thống.

Bảng 4.2. Các công cụ phần mềm

STT	Tên công cụ	Ghi chú
1	Eclipse Java EE IDE for Web Developers.	Môi trường phát triển ứng dụng.
2	MySQL server 5.5	Hệ quản trị cơ sở dữ liệu MySQL.
3	Struts 2	Là framework hỗ trợ xây dựng ứng dụng Web.
4	Hibernate 3.0	Là framework hỗ trợ xây dựng ứng

		dụng Web để kết nối CSDL.
5	Apache Tomcat v7.0	Là server được sử dụng trong việc chạy thử nghiệm.

4.2. Kết quả thực nghiệm

Tạo được cơ sở dữ liệu quản lý người dùng.

Cung cấp giao diện thao tác cho người quản trị hệ thống và người dùng xem xét một số thông tin liên quan.

Kết nối thành công 2 thành phần của hệ thống để tiến hành xác thực thông tin khách hàng sử dụng dịch vụ xe buýt.

Cung cấp các dịch vụ Web cho phía Client để tiến hành hoàn thiện hệ thống phía người dùng.

4.3. Đánh giá và bàn luận

Phạm vi khóa luận chỉ nằm trong việc xác thực người dùng dịch vụ xe buýt là chủ yếu nên hệ thống có thể phát triển thêm để phù hợp với thực tế trong quản lý xe buýt hiện nay. Nhưng do thời gian có hạn nên hệ thống chỉ tiến hành một số thao tác cơ bản phục vụ mục đích chính của việc xác thực người dùng dịch vụ xe buýt.

Trong quá trình thiết kế và phát triển hệ thống còn nhiều thiếu sót, một số chức năng chưa hoàn thiện, giao diện tương tác người dùng thiết kế không hoàn thiện, chưa phù hợp hoàn toàn với hiện thực nên cần có nâng cấp, hoàn thiện để có thể áp dụng vào thực tiễn đời sống theo đúng mục đích đã đề ra.

CHƯƠNG 5

KẾT LUẬN CHUNG

5.1. Các kết quả đạt được

Dựa trên những tiêu chí định ra từ trước, chúng tôi đã xây dựng được hệ thống cơ bản, có thể hỗ trợ hầu hết các tính năng cơ bản trong xác thực người dùng sử dụng dịch vụ xe buýt.

Trong quá trình tìm hiểu các kiến thức liên quan để hoàn thành khóa luận này, tôi đã tiếp thu và học hỏi thêm được nhiều kiến thức mà trước đó chưa biết như hiểu cách sử dụng một số framework, kiến thức về dịch vụ Web, các công nghệ mới như NFC ... Ngoài ra, hệ thống mà tôi xây dựng được mang ý nghĩa rất thực tiễn, tuy rằng chưa thật hoàn chỉnh nhưng có thể sẽ áp dụng được vào thực tiễn đời sống, phục vụ cho chính mục đích của con người.

5.2. Phương hướng phát triển

Có thể thay thế thẻ NFC bằng thẻ RFID bởi công nghệ NFC cũng hỗ trợ đọc được các định dạng thẻ của RFID. Nên có thể mở rộng ứng dụng đối với hầu hết các loại thẻ từ (tag) hiện nay.

Trong việc sử dụng công nghệ NFC để tiến hành xác thực người dùng, ta có thể tiến hành thêm tiện ích cho người sử dụng như bản đồ phân bố, vị trí hiện tại, vị trí các điểm xe bus, ngân hàng, cây ATM... để phục vụ, cung cấp các dịch vụ ngày càng hoàn thiện hơn cho người sử dụng.

Có thể phát triển ứng dụng phức tạp hơn thông qua việc xác thực qua sim điện thoại mà người dùng đang sử dụng. Việc tích hợp này sẽ đem lại nhiều tiện ích cho người sử dụng như giao dịch thuận tiện, dễ dàng, nhanh chóng thông qua mạng điện thoại di động.

Việc xác thực thẻ cũng như thanh toán qua thẻ cần được tiến hành một cách nhanh chóng, tự động, hạn chế thấp nhất lỗi phát sinh khác quan cũng như chủ quan, đảm bảo người sử dụng luôn có được điều kiện phục vụ tốt nhất, an toàn và nhanh nhất có thể. Đồng thời, thường xuyên nâng cấp, cải thiện hệ thống cũng như công nghệ để có thể theo kịp hoặc đón đầu các công nghệ mới nhất trong lĩnh vực công nghệ cao.

TÀI LIỆU THAM KHẢO

- [1] "TỔNG QUAN VỀ MÔ HÌNH MODEL-VIEW-CONTROLLER (MVC)," [Online]. Available:
http://docs.4share.vn/docs/39350/TONG_QUAN_VE_MO_HINH_MODEL_VIEW_CONTROLLER_MVC_.html.
- [2] "Giới thiệu Hibernate Framework," 1 February 2014. [Online]. Available:
<http://cafeitvn.com/blog/kien-truc-hibernate-framework-tong-quan/>.
- [3] "Tìm hiểu web service," 21 October 2013. [Online]. Available:
<http://www.slideshare.net/thieumao/tm-hiu-web-service>.
- [4] A. Rodriguez, "Căn bản về RESTful Web services," 07 March 2013. [Online]. Available: <http://www.ibm.com/developerworks/vn/library/ws-restful/>.
- [5] L. M. Hùng, "Giới thiệu về Framework Struts 2," 28 March 2011. [Online]. Available:
<http://tieuhungphong.blogspot.com/2011/03/introduction-to-struts-2-framework.html>.
- [6] "TÌM HIỂU STRUTS TRONG JAVA(J2EE)," 3 December 2011. [Online]. Available:
<http://quangnet.blogspot.com/2011/12/tim-hieu-struts-trong-javaj2ee.html>.
- [7] Đ. M. Nhật, "NGHIÊN CỨU WEB SERVICE VÀ ỨNG DỤNG TRONG THƯƠNG MẠI ĐIỆN TỬ," Khóa luận tốt nghiệp, Trường Đại học Công Nghệ, Đại học Quốc gia Hà Nội, 2009.