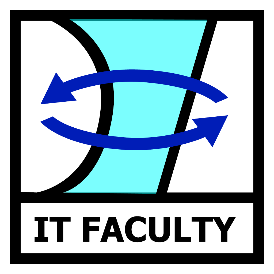


**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙢🕮🙠**



***BÁO CÁO THỰC HÀNH***

**LẬP TRÌNH MẠNG**



|  |  |
| --- | --- |
| ***GVHD:*** | ***ThS. Nguyễn Văn Nguyên*** |
| ***SVTH:*** | ***Phạm Đình Quang*** |
| ***Nhóm HP:*** | ***14.10A*** |
| ***Lớp SH:*** | ***13T2*** |

***Đà Nẵng, tháng 05/2018***

LỜI MỞ ĐẦU

Lập trình mạng là một kiến thức chuyên ngành quan trọng của lĩnh vực công nghệ thông tin, đặc biệt là ngành quản trị mạng. Có thể xem đây là một hướng đi rất rộng mở cho sinh viên và việc nắm bắt những kĩ thuật cơ bản của nó cực kì cần thiết và quan trọng. Sau một loạt các môn học lí thuyết nền tảng như *Mạng máy tính, Lập trình Java, Lập trình mạng*, thì học phần *Thực hành Lập trình mạng* này chính là hội tụ hiện thực các kiến thức học trên sách vở. Nội dung thực hành chủ yếu thực hiện việc lập trình Socket trên họ giao thức TCP/IP và giao thức UDP trên các ứng dụng viết bằng ngôn ngữ Java, kết hợp các kĩ thuật lập trình đa luồng, lập trình web JSP để thực hiện các hạng mục của học phần. Thông qua các bài thực hành với độ khó tăng dần, sinh viên dần thông thạo với công việc lập trình và một phần nào đó làm quen kĩ thuật xây dựng một hệ thống làm việc sao cho hiệu quả.

Báo cáo này là kết quả làm việc của cá nhân em trong thời gian vừa qua, sau 3 buổi thực hành tại phòng máy và mày mò nghiên cứu thêm ở nhà. Mỗi bài thực hành sẽ được chép mã nguồn vào báo cáo cùng với kết quả chạy thử đính kèm bằng hình ảnh.

Dù đã kiểm tra nhiều lần nhưng trong báo cáo này có thể sẽ xuất hiện một số lỗi và sai sót, do đó em rất mong đợi sự góp ý từ các thầy cô. Một lần nữa em xin chân thành cảm ơn.

*Sinh viên Phạm Đình Quang*

**Ý KIẾN ĐÁNH GIÁ CỦA GIẢNG VIÊN**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**MỤC LỤC**

[1 Bài thực hành số 1 5](#_Toc450595651)

[1.1 Bài 1 5](#_Toc450595652)

[1.1.1 Yêu cầu 5](#_Toc450595653)

[1.1.2 Ý tưởng giải quyết vấn đề 5](#_Toc450595654)

[1.1.3 Mã nguồn Server 5](#_Toc450595655)

[1.1.4 Mã nguồn Client 6](#_Toc450595656)

[1.1.5 Kết quả chạy thử 8](#_Toc450595657)

[1.2 Bài 2 10](#_Toc450595658)

[1.2.1 Yêu cầu 10](#_Toc450595659)

[1.2.2 Ý tưởng giải quyết vấn đề 10](#_Toc450595660)

[1.2.3 Mã nguồn Server 10](#_Toc450595661)

[1.2.4 Mã nguồn Client 12](#_Toc450595662)

[1.2.5 Kết quả chạy thử 14](#_Toc450595663)

[1.3 Bài 3 15](#_Toc450595664)

[1.3.1 Yêu cầu 15](#_Toc450595665)

[1.3.2 Ý tưởng giải quyết vấn đề. 15](#_Toc450595666)

[1.3.3 Mã nguồn Server 15](#_Toc450595667)

[1.3.4 Mã nguồn Client 17](#_Toc450595668)

[1.3.5 Kết quả chạy thử 19](#_Toc450595669)

[2 Bài thực hành số 2 20](#_Toc450595670)

[2.1 Bài 1 20](#_Toc450595671)

[2.1.1 Mã nguồn Server 20](#_Toc450595672)

[2.1.2 Mã nguồn Client 21](#_Toc450595673)

[2.1.3 Kết quả chạy thử 23](#_Toc450595674)

[2.2 Bài 2 23](#_Toc450595675)

[2.2.1 Mã nguồn Server 23](#_Toc450595676)

[2.2.2 Mã nguồn Client 26](#_Toc450595677)

[2.2.3 Kết quả chạy thử 28](#_Toc450595678)

[2.3 Bài 3 29](#_Toc450595679)

[2.3.1 Mã nguồn Server 29](#_Toc450595680)

[2.3.2 Mã nguồn Client 30](#_Toc450595681)

[2.3.3 Kết quả chạy thử 33](#_Toc450595682)

[3 Bài Thực hành số 3 34](#_Toc450595683)

[3.1 Yêu cầu 34](#_Toc450595684)

[3.2 Sơ đồ hệ thống 34](#_Toc450595685)

[3.3 Cơ sở dữ liệu 35](#_Toc450595686)

[3.4 Sơ đồ lớp 35](#_Toc450595687)

[3.5 Mã nguồn 37](#_Toc450595688)

[3.5.1 BaseDAO.java 37](#_Toc450595689)

[3.5.2 StaffDAO.java 37](#_Toc450595690)

[3.5.3 DeptDAO.java 41](#_Toc450595691)

[3.5.4 PositionDAO.java 44](#_Toc450595692)

[3.6 Kết quả chạy thử 46](#_Toc450595693)

[4 Kết luận 51](#_Toc450595694)

# Bài thực hành số 1

## Bài 1

### Yêu cầu

Giao thức hoạt động: TCP/IP.

Chương trình Server mở cổng và chờ nhận kết nối từ Client.

Client gửi một chuỗi ký tự đến Server. Server nhận và xử lý gửi trả về cho Client các công việc:

* Đổi chuỗi đã gửi thành chuỗi in hoa.
* Đổi chuỗi đã gửi thành chuỗi thường.
* Đếm số từ của chuỗi đã gửi

### Ý tưởng giải quyết vấn đề

Dùng **Scanner** để lấy một chuỗi từ bàn phím phía Client và gửi lên Server.

Server lập trình đa tuyến. Sau khi Server lấy được chuỗi từ Client gửi lên, dùng phương thức **toUpperCase()** và **toLowerCase()** để chuyển đổi chuỗi thành **HOA** hoặc **thường** sau đó áp dụng các quy tắc của biểu thức chính quy (RegEx) để đếm số từ trong chuỗi và trả kết quả về lại cho Client.

Client nhận lại kết quả và hiển thị ra cho người dùng.

### Mã nguồn Server

package pdq**.**tcpip**;**

**import** java**.**io**.**DataInputStream**;**

**import** java**.**io**.**DataOutputStream**;**

**import** java**.**io**.**IOException**;**

**import** java**.**net**.**ServerSocket**;**

**import** java**.**net**.**Socket**;**

/\*\*

\*/

public class Server **implements** Runnable **{**

Socket socket**;**

ServerSocket serverSocket**;**

DataOutputStream dataOutputStream**;**

DataInputStream dataInputStream**;**

String serverAddress **=** "localhost"**;**

Integer serverPort **=** 8977**;**

public Server**()** **{**

**try** **{**

**this.**serverSocket **=** **new** ServerSocket**(**serverPort**);**

System**.**out**.**println**(**"Server đã khởi động"**);**

**}** **catch** **(**IOException e**)** **{**

System**.**err**.**println**(**"\nLuồng vào/ra bị lỗi: " **+** e**);**

**}**

**}**

public static void main**(**String**[]** args**)** **throws** IOException **{**

Thread thread **=** **new** Thread**(new** Server**());**

thread**.**start**();**

**}**

@Override

public void run**()** **{**

String input**,** output**;**

**while** **(true)** **{**

**try** **{**

**this.**socket **=** **this.**serverSocket**.**accept**();**

System**.**out**.**print**(**"\nĐã mở kết nối với Client."**);**

**this.**dataOutputStream **=** **new** DataOutputStream**(**socket**.**getOutputStream**());**

**this.**dataInputStream **=** **new** DataInputStream**(**socket**.**getInputStream**());**

input **=** **this.**dataInputStream**.**readUTF**();**

System**.**out**.**print**(**"\nServer nhận được chuỗi: " **+** input**);**

output **=** "\t\"" **+** input**.**toUpperCase**()** **+** "\"\n\t\""

**+** input**.**toLowerCase**()** **+** "\""**;**

output **+=** "\n\tSố từ trong chuỗi: " **+** **this.**countWords**(**input**);**

**this.**dataOutputStream**.**writeUTF**(**output**);**

System**.**out**.**print**(**"\nĐã gửi kết quả về Client."**);**

**this.**socket**.**close**();**

System**.**out**.**print**(**"\nĐã đóng socket."**);**

**}** **catch** **(**IOException e**)** **{**

System**.**err**.**println**(**"\nLuồng vào/ra bị lỗi: " **+** e**);**

**new** Server**();**

**continue;**

**}**

**}**

**}**

/\*\*

\* Đếm số lượng từ trong chuỗi, bỏ qua các kí tự đặc biệt.

\* **@param** str chuỗi vào

\* **@return** số từ trong chuỗi

\*/

private int countWords**(**String str**)** **{**

**if** **(**str **==** **null** **||** "\\s\*"**.**equals**(**str**))** **{**

**return** 0**;**

**}** **else** **{**

str **=** str**.**trim**().**replaceAll**(**"\\s+"**,** " "**);**

String splitter1 **=** "\\s\\!\\,\\.\\?\\:\\;\\'\\\"\\)\\(\\[\\]\\{\\}"**;**

String splitter2 **=** "\\@\\#\\$\\%\\^\\&\\\*\\+\\-\\|\\~\\`\\>\\<\\/"**;**

String splitters **=** "[" **+** splitter1 **+** splitter2 **+** "]"**;**

String arr**[]** **=** str**.**split**(**splitters**);**

int num **=** 0**;**

**for** **(**int i **=** 0**;** i **<** arr**.**length**;** i**++)** **{**

**if** **(**arr**[**i**]** **==** **null** **||** ""**.**equals**(**arr**[**i**]))** **{**

**continue;**

**}** **else** **{**

num**++;**

**}**

**}**

**return** num**;**

**}**

**}**

**}**

### Mã nguồn Client

package pdq**.**tcpip**;**

**import** java**.**io**.**DataInputStream**;**

**import** java**.**io**.**DataOutputStream**;**

**import** java**.**io**.**IOException**;**

**import** java**.**net**.**Socket**;**

**import** java**.**net**.**UnknownHostException**;**

**import** java**.**util**.**Scanner**;**

/\*\*

\*/

public class Client **{**

// Thành viên dữ liệu =====================================================

Socket socket**;**

DataOutputStream dataOutputStream**;**

DataInputStream dataInputStream**;**

Scanner scanner**;**

String serverAddress **=** "localhost"**;**

Integer serverPort **=** 8977**;**

public Client**()** **{**

// #1: Khởi tạo client Socket =========================================

**try** **{**

socket **=** **new** Socket**(**serverAddress**,** serverPort**);**

**}** **catch** **(**UnknownHostException e**)** **{**

System**.**err**.**println**(**"\nLỗi khi kết nối máy chủ: " **+** e**);**

System**.**exit**(**1**);**

**}** **catch** **(**IOException e**)** **{**

System**.**err**.**println**(**"\nKhông thể nạp cổng: " **+** serverPort**);**

System**.**exit**(**1**);**

**}**

// #2: Tạo luồng để truyền tin ========================================

**try** **{**

dataOutputStream **=** **new** DataOutputStream**(**socket**.**getOutputStream**());**

dataInputStream **=** **new** DataInputStream**(**socket**.**getInputStream**());**

**}** **catch** **(**IOException e**)** **{**

System**.**err**.**println**(**"\nLuồng vào/ra bị lỗi: " **+** e**);**

System**.**exit**(**1**);**

**}**

// #3: Lấy 1 chuỗi từ người dùng, đẩy lên Server ======================

System**.**out**.**print**(**"Nhập 1 chuỗi: "**);**

**this.**scanner **=** **new** Scanner**(**System**.**in**);**

String outString **=** scanner**.**nextLine**();**

**try** **{**

dataOutputStream**.**writeUTF**(**outString**);**

System**.**out**.**print**(**"\nĐã đẩy chuỗi lên Server"**);**

**}** **catch** **(**IOException e**)** **{**

System**.**err**.**println**(**"\nLuồng ra bị lỗi: " **+** e**);**

**}**

// #4: Lấy kết quả trả về từ Server, in ra màn hình ===================

String inString **=** ""**;**

**try** **{**

inString **=** dataInputStream**.**readUTF**();**

System**.**out**.**print**(**"\nServer trả về:\n" **+** inString**);**

**}** **catch** **(**IOException e**)** **{**

System**.**err**.**println**(**"\nLuồng vào bị lỗi: " **+** e**);**

System**.**exit**(**1**);**

**}**

// #5: Ngắt liên kết, đóng socket =====================================

**try** **{**

dataOutputStream**.**close**();**

dataInputStream**.**close**();**

socket**.**close**();**

**}** **catch** **(**IOException e**)** **{**

System**.**err**.**println**(**"\nLỗi khi đóng kết nối: " **+** e**);**

**}**

System**.**out**.**println**(**"\nĐã đóng kết nối!"**);**

**}**

public static void main**(**String**[]** args**)**

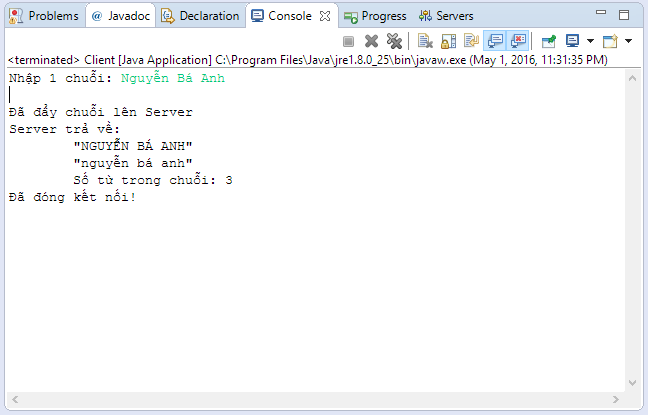
**throws** UnknownHostException**,** IOException **{**

**new** Client**();**

**}**

**}**

### Kết quả chạy thử



Thực hành lập trình mạng

Thực hành lập trình mạng

Figure i:Client gửi 1 chuỗi thông thường

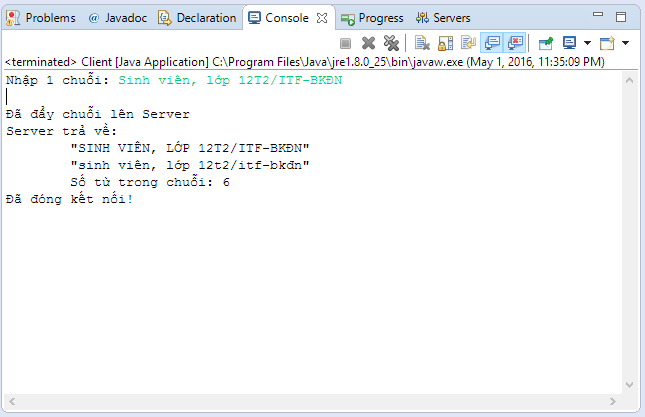


Figure ii: Client gửi 1 chuỗi với dấu câu

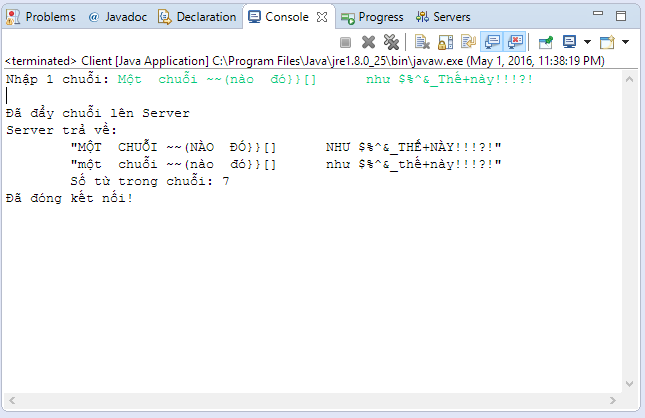


Figure iii: Client gửi 1 chuỗi có lẫn các kí tự đặc biệt

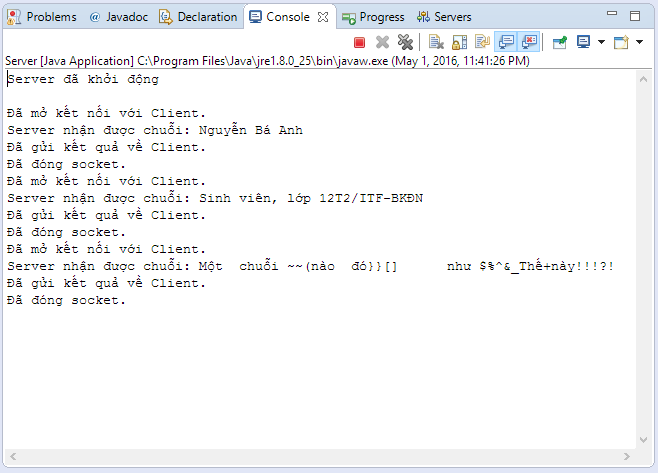


Figure iv: Server tiếp nhận các chuỗi từ Client

## Bài 2

### Yêu cầu

Giao thức hoạt động: TCP/IP.

Chương trình Client cho phép nhập vào từ bàn phím một chuỗi biễu diễn một phép tính gồm các toán tử +, -, (, ). Ví dụ: 5+13-(12-4\*6) –((3+4)-5)

Chương trình Server thực hiện tính toán và trả kết quả về cho Client.

### Ý tưởng giải quyết vấn đề

Lập trình Client lên giao diện mô phỏng một máy tính mini đơn giản, bắt sự kiện nhấn phím Enter để lấy chuỗi sau đó gửi lên Server.

Server lập trình đa tuyến, công nghệ Java mới hiện nay cho phép sử dụng một **ScriptEngine** nào đó (ở đây có thể chọn *Javascript engine*) để đoán nhận chuỗi và tính toán kết quả, bắt biệt lệ nếu chuỗi nhập không đúng quy tắc toán học, trả kết quả về cho Client.

Client hiển thị kết quả lên giao diện.

### Mã nguồn Server

package pdq**.**tcpip**;**

**import** java**.**awt**.**GridLayout**;**

**import** java**.**awt**.**event**.**ActionEvent**;**

**import** java**.**awt**.**event**.**ActionListener**;**

**import** java**.**io**.**DataInputStream**;**

**import** java**.**io**.**DataOutputStream**;**

**import** java**.**io**.**IOException**;**

**import** java**.**net**.**ServerSocket**;**

**import** java**.**net**.**Socket**;**

**import** javax**.**script**.**ScriptEngine**;**

**import** javax**.**script**.**ScriptEngineManager**;**

**import** javax**.**script**.**ScriptException**;**

**import** javax**.**swing**.**JButton**;**

**import** javax**.**swing**.**JFrame**;**

**import** javax**.**swing**.**JLabel**;**

**import** javax**.**swing**.**JOptionPane**;**

**import** javax**.**swing**.**JTextField**;**

/\*\*

\*\*/

public class Server **implements** Runnable **{**

static Socket socket**;**

static ServerSocket server**;**

static int port**;**

static String input**,** output**;**

static DataOutputStream dos**;**

static DataInputStream dis**;**

static ScriptEngineManager sem**;**

static ScriptEngine engine**;**

static boolean isRunning **=** **false;**

static JLabel activeThread**;**

static JButton switchBtn**;**

static JTextField portTf**;**

static JLabel portLabel**;**

public Server**()** **{**

sem **=** **new** ScriptEngineManager**();**

engine **=** sem**.**getEngineByName**(**"JavaScript"**);**

**}**

public static void main**(**String**[]** args**)** **throws** IOException **{**

JFrame frame **=** **new** JFrame**(**"Bài TH số 2 - Server - Nguyễn Bá Anh"**);**

frame**.**setBounds**(**200**,** 200**,** 400**,** 60**);**

frame**.**setResizable**(false);**

frame**.**setLayout**(new** GridLayout**(**1**,** 4**));**

frame**.**setDefaultCloseOperation**(**JFrame**.**EXIT\_ON\_CLOSE**);**

portLabel **=** **new** JLabel**(**"Cổng"**);**

portTf **=** **new** JTextField**();**

activeThread **=** **new** JLabel**(**"0"**);**

switchBtn **=** **new** JButton**();**

switchBtn**.**setText**(**"Start"**);**

frame**.**add**(**portLabel**);**

frame**.**add**(**portTf**);**

frame**.**add**(**activeThread**);**

frame**.**add**(**switchBtn**);**

frame**.**setVisible**(true);**

Thread thread **=** **new** Thread**(new** Server**());**

switchBtn**.**addActionListener**(new** ActionListener**()** **{**

@Override

public void actionPerformed**(**ActionEvent arg0**)** **{**

**if** **(**isRunning**)** **{**

isRunning **=** **false;**

thread**.**interrupt**();**

**try** **{**

server**.**close**();**

JOptionPane**.**showMessageDialog**(null,** "Đã ngừng Server"**);**

**}** **catch** **(**IOException e**)** **{**

JOptionPane**.**showMessageDialog**(null,** "Lỗi khi ngừng Server"**);**

**}**

**}** **else** **{**

port **=** Integer**.**parseInt**(**portTf**.**getText**());**

**try** **{**

server **=** **new** ServerSocket**(**port**);**

JOptionPane**.**showMessageDialog**(null,** "Đã khởi động Server"**);**

**}** **catch** **(**IOException e2**)** **{**

JOptionPane**.**showMessageDialog**(null,** "Lỗi khi khởi động Server"**);**

**}**

isRunning **=** **true;**

thread**.**start**();**

**}**

portTf**.**setEditable**(true);**

switchBtn**.**setText**(**"Start"**);**

activeThread**.**setText**(**"Luồng: " **+** Thread**.**activeCount**());**

**}**

**});**

**}**

@Override

public void run**()** **{**

**while** **(**isRunning**)** **{**

switchBtn**.**setText**(**"Stop"**);**

portTf**.**setEditable**(false);**

activeThread**.**setText**(**"Luồng: " **+** Thread**.**activeCount**());**

**try** **{**

socket **=** server**.**accept**();**

dos **=** **new** DataOutputStream**(**socket**.**getOutputStream**());**

dis **=** **new** DataInputStream**(**socket**.**getInputStream**());**

input **=** dis**.**readUTF**();**

System**.**out**.**print**(**"\nServer nhận được chuỗi: " **+** input**);**

output **=** String**.**valueOf**(**engine**.**eval**(**input**));**

System**.**out**.**print**(**"\n=======================> " **+** output**);**

dos**.**writeUTF**(**output**);**

dos**.**close**();**

dis**.**close**();**

socket**.**close**();**

**}** **catch** **(**ScriptException e**)** **{**

output **=** "Lỗi cú pháp"**;**

**try** **{**

dos**.**writeUTF**(**output**);**

**}** **catch** **(**IOException e1**)** **{**

**}**

System**.**out**.**print**(**"\n=======================> " **+** output**);**

**continue;**

**}** **catch** **(**IOException e**)** **{**

output **=** "Lỗi kết nối"**;**

**try** **{**

dos**.**writeUTF**(**output**);**

**}** **catch** **(**IOException e1**)** **{**

**}**

System**.**out**.**print**(**"\n=======================> " **+** output**);**

**continue;**

**}**

**}**

**}**

**}**

### Mã nguồn Client

package pdq**.**tcpip**;**

**import** java**.**awt**.**BorderLayout**;**

**import** java**.**awt**.**Font**;**

**import** java**.**awt**.**GridLayout**;**

**import** java**.**awt**.**event**.**ActionEvent**;**

**import** java**.**awt**.**event**.**ActionListener**;**

**import** java**.**awt**.**event**.**WindowEvent**;**

**import** java**.**awt**.**event**.**WindowListener**;**

**import** java**.**io**.**DataInputStream**;**

**import** java**.**io**.**DataOutputStream**;**

**import** java**.**io**.**IOException**;**

**import** java**.**net**.**Socket**;**

**import** java**.**net**.**UnknownHostException**;**

**import** javax**.**swing**.**JButton**;**

**import** javax**.**swing**.**JFrame**;**

**import** javax**.**swing**.**JLabel**;**

**import** javax**.**swing**.**JOptionPane**;**

**import** javax**.**swing**.**JPanel**;**

**import** javax**.**swing**.**JTextField**;**

/\*\*

\*

\*

\*\*/

public class Client **{**

static String serverAddress**;**

static int serverPort**;**

static Socket socket**;**

static DataInputStream inStream**;**

static DataOutputStream outStream**;**

static JFrame splashFrame**,** frame**;**

JPanel outputPanel**,** inputPanel**;**

JTextField output**,** input**;**

public Client**()** **{**

initGUI**();**

addListeners**();**

**}**

private void initGUI**()** **{**

frame **=** **new** JFrame**(**"Bài TH số 2 - Client - "   
 **+** serverAddress **+** ":" **+** serverPort**);**

frame**.**setLayout**(new** GridLayout**(**2**,** 1**));**

frame**.**setBounds**(**200**,** 200**,** 500**,** 200**);**

frame**.**setDefaultCloseOperation**(**JFrame**.**DISPOSE\_ON\_CLOSE**);**

inputPanel **=** **new** JPanel**();**

inputPanel**.**setLayout**(new** GridLayout**());**

outputPanel **=** **new** JPanel**();**

outputPanel**.**setLayout**(new** GridLayout**());**

input **=** **new** JTextField**();**

input**.**setFont**(new** Font**(**"Consolas"**,** Font**.**BOLD**,** 30**));**

input**.**setHorizontalAlignment**(**JTextField**.**RIGHT**);**

output **=** **new** JTextField**();**

output**.**setFont**(new** Font**(**"Consolas"**,** Font**.**BOLD**,** 30**));**

output**.**setHorizontalAlignment**(**JTextField**.**RIGHT**);**

output**.**setEditable**(false);**

inputPanel**.**add**(**input**);**

outputPanel**.**add**(**output**);**

frame**.**add**(**output**);**

frame**.**add**(**input**);**

frame**.**setVisible**(true);**

**}**

private void addListeners**()** **{**

input**.**addActionListener**(new** ActionListener**()** **{**

@Override

public void actionPerformed**(**ActionEvent arg0**)** **{**

String expresion **=** input**.**getText**();**

**try** **{**

socket **=** **new** Socket**(**serverAddress**,** serverPort**);**

socket**.**setSoTimeout**(**1000**);**

inStream **=** **new** DataInputStream**(**socket**.**getInputStream**());**

outStream **=** **new** DataOutputStream**(**socket**.**getOutputStream**());**

outStream**.**writeUTF**(**expresion**);**

String result **=** inStream**.**readUTF**();**

output**.**setText**(**result**);**

inStream**.**close**();**

outStream**.**close**();**

socket**.**close**();**

**}** **catch** **(**UnknownHostException e**)** **{**

JOptionPane**.**showMessageDialog**(null,** "Không tìm thấy Server."**);**

e**.**printStackTrace**();**

**}** **catch** **(**IOException e**)** **{**

JOptionPane**.**showMessageDialog**(null,** "Lỗi kết nối vào ra khi truyền dữ liệu."**);**

e**.**printStackTrace**();**

**}**

**}**

**});**

frame**.**addWindowListener**(new** WindowListener**()** **{**

@Override

public void windowOpened**(**WindowEvent arg0**)** **{**

splashFrame**.**setVisible**(false);**

**}**

@Override

public void windowIconified**(**WindowEvent arg0**)** **{}**

@Override

public void windowDeiconified**(**WindowEvent arg0**)** **{}**

@Override

public void windowDeactivated**(**WindowEvent arg0**)** **{}**

@Override

public void windowClosing**(**WindowEvent arg0**)** **{}**

@Override

public void windowClosed**(**WindowEvent arg0**)** **{**

splashFrame**.**setVisible**(true);**

frame**.**setVisible**(false);**

**}**

@Override

public void windowActivated**(**WindowEvent arg0**)** **{}**

**});**

**}**

/\*\*

\* **@param** args

\*/

public static void main**(**String**[]** args**)** **{**

splashFrame **=** **new** JFrame**(**"Nhập thông tin"**);**

splashFrame**.**setBounds**(**200**,** 200**,** 200**,** 100**);**

splashFrame**.**setResizable**(false);**

splashFrame**.**setDefaultCloseOperation**(**JFrame**.**EXIT\_ON\_CLOSE**);**

splashFrame**.**setLayout**(new** BorderLayout**());**

splashFrame**.**add**(new** JLabel**(**"Nhập địa chỉ máy chủ và cổng"**),** BorderLayout**.**NORTH**);**

JTextField host**,** port**;**

host **=** **new** JTextField**(**"localhost"**);**

port **=** **new** JTextField**(**"12345"**);**

splashFrame**.**add**(**host**,** BorderLayout**.**CENTER**);**

splashFrame**.**add**(**port**,** BorderLayout**.**EAST**);**

JButton enterBtn **=** **new** JButton**(**"Enter"**);**

splashFrame**.**add**(**enterBtn**,** BorderLayout**.**SOUTH**);**

splashFrame**.**setVisible**(true);**

enterBtn**.**addActionListener**(new** ActionListener**()** **{**

@Override

public void actionPerformed**(**ActionEvent arg0**)** **{**

serverAddress **=** host**.**getText**();**

serverPort **=** Integer**.**parseInt**(**port**.**getText**());**

**new** Client**();**

**}**

**});**

**}**

**}**

### Kết quả chạy thử

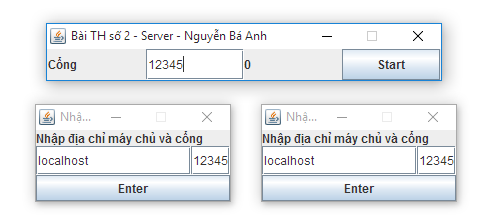


Figure v: Màn hình khởi động của Server và Client

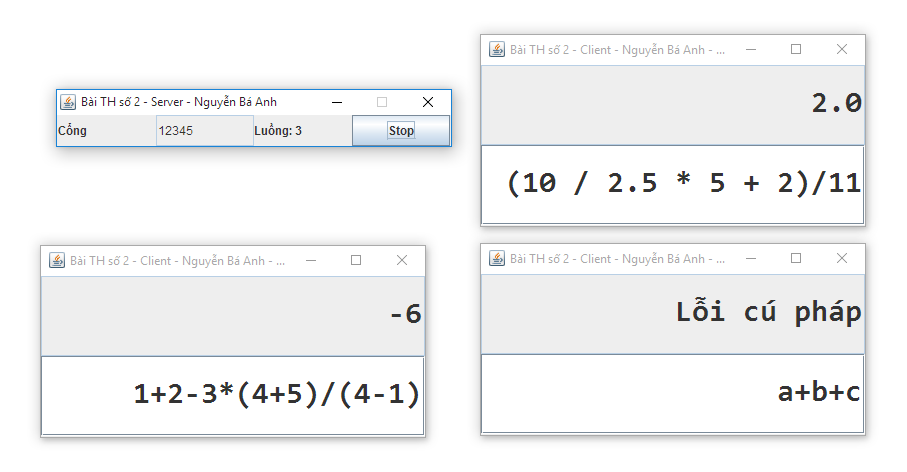


Figure vi: Thực hiện một số phép toán trên hệ 1 server - 3 clients

## Bài 3

### Yêu cầu

Xây dựng chương trình hội thoại chatroom Client /Server.

Giao thức hoạt động: TCP/IP.

Chương trình Server mở cổng chờ nhận kết nối từ Client.

Chương trình Client kết nối và thực hiện trao đổi với chương trình Server.

### Ý tưởng giải quyết vấn đề.

Mỗi client cần đăng kí 1 bí danh (nickname) với Server và xin gia nhập room.

Server lập trình đa tuyến, mỗi người dùng gửi thông điệp đầu tiên lên thì Server xem đó là thao tác đăng kí nick, phía sau là thông điệp. Cứ mỗi khi có client nào gửi 1 thông điệp lên thì Server phát thông điệp đó đến tất cả client đang tham gia.

### Mã nguồn Server

Package pdq**.**tcpip**;**

**import** java**.**io**.**DataInputStream**;**

**import** java**.**io**.**DataOutputStream**;**

**import** java**.**net**.**ServerSocket**;**

**import** java**.**net**.**Socket**;**

**import** java**.**util**.**Enumeration**;**

**import** java**.**util**.**Hashtable**;**

**import** javax**.**swing**.**JOptionPane**;**

public class Server **{**

ServerSocket serverSocket**;**

public Server**(**int port**)** **{**

**try** **{**

System**.**out**.**println**(**port**);**

serverSocket **=** **new** ServerSocket**(**port**);**

**while** **(true)** **{**

Socket clientSocket **=** serverSocket**.**accept**();**

**if** **(**clientSocket**.**isConnected**())** **{**

ClientConnection connectionThread

**=** **new** ClientConnection**(**serverSocket**,** clientSocket**);**

connectionThread**.**start**();**

**}**

**}**

**}** **catch** **(**Exception e**)** **{**

e**.**printStackTrace**();**

**}**

**}**

public static void main**(**String**[]** args**)** **{**

int port **=** Integer**.**parseInt**(**

JOptionPane**.**showInputDialog**(**"Nhập cổng: "**,** "12345"**));**

**new** Server**(**port**);**

**}**

static class ClientConnection **extends** Thread **{**

private Socket clientSocket**;**

private ServerSocket serverSocket**;**

private int connectionCount**;**

private DataInputStream inStream**;**

private DataOutputStream outStream**;**

private static Hashtable**<**Socket**,** ClientConnection**>** connectionList

**=** **new** Hashtable**<**Socket**,** ClientConnection**>();**

public ClientConnection**(**ServerSocket server**,** Socket client**)** **{**

**this.**connectionCount **=** connectionList**.**size**();**

connectionList**.**put**(**client**,** **this);**

**this.**serverSocket **=** server**;**

**this.**clientSocket **=** client**;**

**}**

public void run**()** **{**

**try** **{**

inStream **=** **new** DataInputStream**(**clientSocket**.**getInputStream**());**

outStream **=** **new** DataOutputStream**(**clientSocket**.**getOutputStream**());**

String user **=** inStream**.**readUTF**();**

**this.**sendToAll**(**user **+** " đã tham gia phòng chat."**);**

String data**;**

**while** **((**data **=** inStream**.**readUTF**())** **!=** **null)** **{**

**this.**sendToAll**(**user **+** ": " **+** data**);**

**}**

**}** **catch** **(**Exception e**)** **{**

e**.**printStackTrace**();**

**}**

**}**

/\*\*

\* Phát thông điệp đến tất cả client đang kết nối

\* **@param** message nội dung

\*/

public void sendToAll**(**String message**)** **{**

synchronized **(**connectionList**)** **{**

**for** **(**Enumeration**<?>** element **=** connectionList**.**elements**();**

element**.**hasMoreElements**();)** **{**

ClientConnection cc **=** **(**ClientConnection**)** element**.**nextElement**();**

**try** **{**

cc**.**outStream**.**writeUTF**(**message**);**

**}** **catch** **(**Exception e**)** **{**

e**.**printStackTrace**();**

**}**

**}**

**}**

**}**

**}**

**}**

### Mã nguồn Client

package pdq**.**tcpip**;**

**import** java**.**awt**.**BorderLayout**;**

**import** java**.**awt**.**GridLayout**;**

**import** java**.**awt**.**event**.**ActionEvent**;**

**import** java**.**awt**.**event**.**ActionListener**;**

**import** java**.**io**.**DataInputStream**;**

**import** java**.**io**.**DataOutputStream**;**

**import** java**.**io**.**IOException**;**

**import** java**.**net**.**Socket**;**

**import** javax**.**swing**.**JButton**;**

**import** javax**.**swing**.**JFrame**;**

**import** javax**.**swing**.**JLabel**;**

**import** javax**.**swing**.**JList**;**

**import** javax**.**swing**.**JOptionPane**;**

**import** javax**.**swing**.**JPanel**;**

**import** javax**.**swing**.**JScrollPane**;**

**import** javax**.**swing**.**JTextArea**;**

**import** javax**.**swing**.**JTextField**;**

/\*\*

\*\*/

public class Client **extends** JFrame **implements** ActionListener **{**

private static final long serialVersionUID **=** 1L**;**

JTextField userTf**;**

public Client**(**String title**)** **{**

**super(**title**);**

JPanel userPanel **=** **new** JPanel**();**

JPanel buttonPanel **=** **new** JPanel**();**

JLabel userLb **=** **new** JLabel**(**"Tên"**);**

userTf **=** **new** JTextField**(**30**);**

JButton loginBtn **=** **new** JButton**(**"Login"**);**

JButton resetBtn **=** **new** JButton**(**"Reset"**);**

JButton exitBtn **=** **new** JButton**(**"Exit"**);**

loginBtn**.**addActionListener**(this);**

resetBtn**.**addActionListener**(this);**

exitBtn**.**addActionListener**(this);**

userPanel**.**add**(**userLb**);**

userPanel**.**add**(**userTf**);**

buttonPanel**.**add**(**loginBtn**);**

buttonPanel**.**add**(**resetBtn**);**

buttonPanel**.**add**(**exitBtn**);**

add**(**userPanel**);**

add**(**buttonPanel**);**

setBounds**(**200**,** 200**,** 400**,** 150**);**

setLayout**(new** GridLayout**(**2**,** 1**));**

setVisible**(true);**

setDefaultCloseOperation**(**JFrame**.**EXIT\_ON\_CLOSE**);**

**}**

@Override

public void actionPerformed**(**ActionEvent e**)** **{**

String button **=** e**.**getActionCommand**();**

**if** **(**button**.**equals**(**"Login"**))** **{**

String user **=** userTf**.**getText**();**

String ip **=** JOptionPane

**.**showInputDialog**(**"Nhập địa chỉ máy chủ"**,** "localhost"**);**

int port **=** Integer**.**parseInt**(**JOptionPane**.**showInputDialog**(**"Cổng"**,** "12345"**));**

**new** ChatRoom**(**user**,** ip**,** port**);**

dispose**();**

**}**

**if** **(**button**.**equals**(**"Reset"**))** **{**

userTf**.**setText**(**""**);**

**}**

**if** **(**button**.**equals**(**"Exit"**))** **{**

System**.**exit**(**0**);**

**}**

**}**

public static void main**(**String**[]** args**)** **{**

**new** Client**(**"Đăng nhập"**);**

**}**

class ChatRoom **extends** JFrame **implements** Runnable **{**

private static final long serialVersionUID **=** 1L**;**

JTextArea messageTa**,** allMessagesTa**;**

JPanel titlePanel**,** messagePanel**,** inputPanel**;**

JLabel titleLb**;**

JButton sendBtn**;**

JList**<**String**>** userList **=** **new** JList**<**String**>();**

String user**,** address**;**

int port**;**

private Socket socket**;**

private DataInputStream inStream**;**

private DataOutputStream outStream**;**

public ChatRoom**(**String user**,** String address**,** int port**)** **{**

**super(**"Phòng Chát (TCPIP)"**);**

**this.**user **=** user**;**

**this.**address **=** address**;**

**this.**port **=** port**;**

**this.**initGUI**();**

**try** **{**

System**.**out**.**println**(**address **+** " " **+** port**);**

socket **=** **new** Socket**(**address**,** port**);**

inStream **=** **new** DataInputStream**(**socket**.**getInputStream**());**

outStream **=** **new** DataOutputStream**(**socket**.**getOutputStream**());**

outStream**.**writeUTF**(this.**user**);**

**new** Thread**(this).**start**();**

**}** **catch** **(**Exception e**)** **{**

e**.**printStackTrace**();**

**}**

**}**

public void initGUI**()** **{**

**this.**setDefaultCloseOperation**(**EXIT\_ON\_CLOSE**);**

**this.**setLayout**(new** BorderLayout**());**

titlePanel **=** **new** JPanel**();**

titleLb **=** **new** JLabel**(**address **+** ":" **+** port **+** " >> " **+** user**);**

titlePanel**.**add**(**titleLb**);**

messagePanel **=** **new** JPanel**();**

messagePanel**.**setLayout**(new** BorderLayout**());**

allMessagesTa **=** **new** JTextArea**();**

allMessagesTa**.**setWrapStyleWord**(true);**

allMessagesTa**.**setLineWrap**(true);**

allMessagesTa**.**setEditable**(false);**

JScrollPane scrollPane **=** **new** JScrollPane**(**allMessagesTa**);**

scrollPane**.**setVerticalScrollBarPolicy**(**

JScrollPane**.**VERTICAL\_SCROLLBAR\_AS\_NEEDED**);**

messagePanel**.**add**(**scrollPane**);**

inputPanel **=** **new** JPanel**();**

inputPanel**.**setLayout**(new** BorderLayout**());**

messageTa **=** **new** JTextArea**(**2**,** 35**);**

messageTa**.**setWrapStyleWord**(true);**

messageTa**.**setLineWrap**(true);**

JScrollPane spMessage **=** **new** JScrollPane**(**messageTa**);**

inputPanel**.**add**(**spMessage**);**

sendBtn **=** **new** JButton**(**"Gửi"**);**

sendBtn**.**addActionListener**(new** ActionListener**()** **{**

@Override

public void actionPerformed**(**ActionEvent ae**)** **{**

**try** **{**

outStream**.**writeUTF**(**messageTa**.**getText**());**

messageTa**.**setText**(**""**);**

**}** **catch** **(**IOException e**)** **{}**

**}**

**});**

inputPanel**.**add**(**sendBtn**,** BorderLayout**.**EAST**);**

add**(**titlePanel**,** BorderLayout**.**NORTH**);**

add**(**messagePanel**,**BorderLayout**.**CENTER**);**

add**(**inputPanel**,** BorderLayout**.**SOUTH**);**

setSize**(**400**,** 300**);**

setLocation**(**400**,** 100**);**

setVisible**(true);**

**}**

@Override

public void run**()** **{**

**try** **{**

**while** **(true)** **{**

String message **=** inStream**.**readUTF**();**

allMessagesTa**.**append**(**message **+** "\n"**);**

**}**

**}** **catch** **(**Exception e**)** **{**

e**.**printStackTrace**();**

**}**

**}**

**}**

}

### Kết quả chạy thử

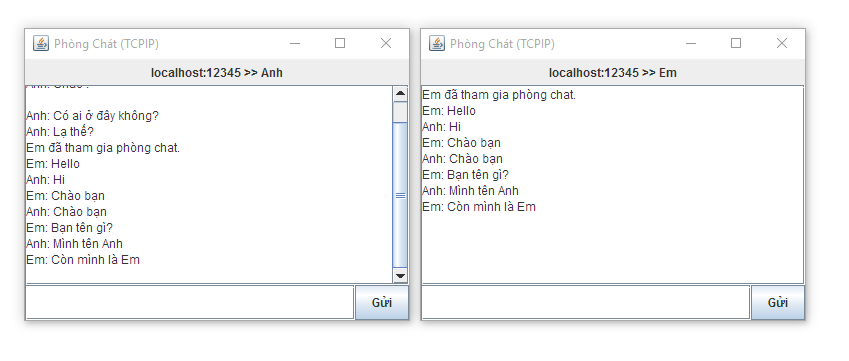


Figure vii: Chạy thử chát-room

# Bài thực hành số 2

Bài thực hành số 2 hoàn toàn tương tự như Bài thực hành số 1, chỉ sử dụng giao thức UDP.

## Bài 1

### Mã nguồn Server

package pdq**.**udp**;**

**import** java**.**io**.**DataInputStream**;**

**import** java**.**io**.**DataOutputStream**;**

**import** java**.**io**.**IOException**;**

**import** java**.**net**.**DatagramPacket**;**

**import** java**.**net**.**DatagramSocket**;**

**import** java**.**net**.**InetAddress**;**

**import** java**.**net**.**ServerSocket**;**

/\*\*

\*\*/

public class ServerUDP **implements** Runnable **{**

private DatagramSocket serverSocket**;**

private DatagramPacket receivePacket**,** sendPacket**;**

private InetAddress IPAddress**;**

private Integer portNumber **=** 9876**;**

private byte**[]** receiveData **=** **new** byte**[**1024**];**

private byte**[]** sendData **=** **new** byte**[**1024**];**

public ServerUDP**()** **{**

**try** **{**

**this.**serverSocket **=** **new** DatagramSocket**(**portNumber**);**

System**.**out**.**println**(**"Server đã khởi động"**);**

**}** **catch** **(**IOException e**)** **{**

System**.**err**.**println**(**"\nLuồng vào/ra bị lỗi: " **+** e**);**

**}**

**}**

public static void main**(**String**[]** args**)** **throws** IOException **{**

Thread thread **=** **new** Thread**(new** ServerUDP**());**

thread**.**start**();**

**}**

@Override

public void run**()** **{**

String input**,** output**;**

**while** **(true)** **{**

// #2: Lấy chuỗi từ Client, in ra, xử lí rồi đẩy về client ========

**try** **{**

receivePacket **=** **new** DatagramPacket**(**receiveData**,** receiveData**.**length**);**

System**.**out**.**println**(**"Đang đợi gói tin..."**);**

serverSocket**.**receive**(**receivePacket**);**

IPAddress **=** receivePacket**.**getAddress**();**

portNumber **=** receivePacket**.**getPort**();**

System**.**out**.**println**(**"Từ: " **+** IPAddress **+** ":" **+** portNumber**);**

///

input **=** **new** String**(**receivePacket**.**getData**()).**trim**();**

System**.**out**.**println**(**"\nServer nhận được chuỗi: " **+** input**);**

output **=** "\t\"" **+** input**.**toUpperCase**()** **+** "\"\n\t\""

**+** input**.**toLowerCase**()** **+** "\""**;**

output **+=** "\n\tSố từ trong chuỗi: " **+** **this.**countWords**(**input**);**

System**.**out**.**println**(**"Kết quả: \n" **+** output**);**

sendData **=** output**.**getBytes**();**

sendPacket **=** **new** DatagramPacket**(**sendData**,** sendData**.**length**,**

IPAddress**,** portNumber**);**

serverSocket**.**send**(**sendPacket**);**

System**.**out**.**println**(**"Đã gửi kết quả về Client.\n"**);**

**}** **catch** **(**IOException e**)** **{**

// TODO Auto-generated catch block

e**.**printStackTrace**();**

**}**

**}**

**}**

/\*\*

\* Đếm số lượng từ trong chuỗi, bỏ qua các kí tự đặc biệt.

\*

\* **@param** str

\* chuỗi vào

\* **@return** số từ trong chuỗi

\*/

private int countWords**(**String str**)** **{**

**if** **(**str **==** **null** **||** "\\s\*"**.**equals**(**str**))** **{**

**return** 0**;**

**}** **else** **{**

str **=** str**.**trim**().**replaceAll**(**"\\s+"**,** " "**);**

String splitter1 **=** "\\s\\!\\,\\.\\?\\:\\;\\'\\\"\\)\\(\\[\\]\\{\\}"**;**

String splitter2 **=** "\\@\\#\\$\\%\\^\\&\\\*\\+\\-\\|\\~\\`\\>\\<\\/"**;**

String splitters **=** "[" **+** splitter1 **+** splitter2 **+** "]"**;**

String arr**[]** **=** str**.**split**(**splitters**);**

int num **=** 0**;**

**for** **(**int i **=** 0**;** i **<** arr**.**length**;** i**++)** **{**

**if** **(**arr**[**i**]** **==** **null** **||** ""**.**equals**(**arr**[**i**]))** **{**

**continue;**

**}** **else** **{**

num**++;**

**}**

**}**

**return** num**;**

**}**

**}**

**}**

### Mã nguồn Client

package pdq**.**udp**;**

**import** java**.**io**.**IOException**;**

**import** java**.**net**.**DatagramPacket**;**

**import** java**.**net**.**DatagramSocket**;**

**import** java**.**net**.**InetAddress**;**

**import** java**.**net**.**SocketTimeoutException**;**

**import** java**.**util**.**Scanner**;**

/\*\*

\*

\*\*/

public class ClientUDP **{**

private DatagramSocket clientSocket **=** **null;**

private String serverHostname **=** "localhost"**;**

private InetAddress IPAddress**,** returnIPAddress**;**

private Integer portNumber **=** 9876**;**

private byte**[]** sendData **=** **new** byte**[**1024**];**

private byte**[]** receiveData **=** **new** byte**[**1024**];**

private DatagramPacket sendPacket**,** receivePacket**;**

private Scanner scanner**;**

public ClientUDP**()** **throws** IOException **{**

// #1: Khởi tạo client DatagramSocket =================================

clientSocket **=** **new** DatagramSocket**();**

IPAddress **=** InetAddress**.**getByName**(**serverHostname**);**

System**.**out**.**print**(**"Nhập 1 chuỗi: "**);**

// #2: Lấy 1 chuỗi từ người dùng, đẩy lên Server ======================

scanner **=** **new** Scanner**(**System**.**in**);**

String line **=** scanner**.**nextLine**();**

sendData **=** line**.**getBytes**();**

System**.**out**.**print**(**"Đang gửi " **+** sendData**.**length

**+** " bytes đến Server..."**);**

sendPacket **=** **new** DatagramPacket**(**sendData**,** sendData**.**length**,**

IPAddress**,** portNumber**);**

clientSocket**.**send**(**sendPacket**);**

System**.**out**.**println**(**"Đã gửi!"**);**

// #3: Nhận phản hồi từ Server ========================================

receivePacket **=** **new** DatagramPacket**(**receiveData**,** receiveData**.**length**);**

System**.**out**.**println**(**"Đang đợi phản hồi..."**);**

clientSocket**.**setSoTimeout**(**10000**);** // chờ 10s tối đa

**try** **{**

clientSocket**.**receive**(**receivePacket**);**

String string **=** **new** String**(**receivePacket**.**getData**());**

returnIPAddress **=** receivePacket**.**getAddress**();**

System**.**out**.**println**(**"Từ Server: " **+** returnIPAddress **+** ":"

**+** receivePacket**.**getPort**());**

System**.**out**.**println**(**"Kết quả: \n\t" **+** string**.**trim**());**

**}** **catch** **(**SocketTimeoutException ste**)** **{**

System**.**out**.**println**(**"Chờ quá lâu: Gói tin đã bị mất!"**);**

**}**

// #5: Đóng ===========================================================

clientSocket**.**close**();**

System**.**out**.**println**(**"Kết thúc."**);**

**}**

public static void main**(**String**[]** args**)** **throws** IOException **{**

**new** ClientUDP**();**

**}**

**}**

### Kết quả chạy thử

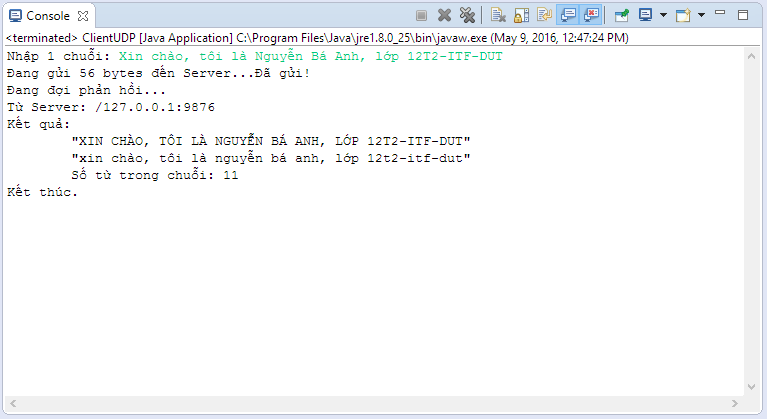


Figure viii: Kết quả ClientUDP

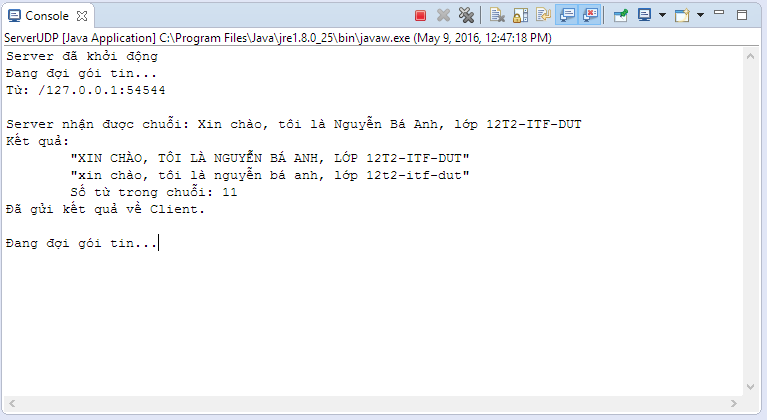


Figure ix: Kết quả ServerUDP

## Bài 2

### Mã nguồn Server

package pdq**.**udp**;**

**import** java**.**awt**.**GridLayout**;**

**import** java**.**awt**.**event**.**ActionEvent**;**

**import** java**.**awt**.**event**.**ActionListener**;**

**import** java**.**io**.**IOException**;**

**import** java**.**net**.**DatagramPacket**;**

**import** java**.**net**.**DatagramSocket**;**

**import** java**.**net**.**InetAddress**;**

**import** javax**.**script**.**ScriptEngine**;**

**import** javax**.**script**.**ScriptEngineManager**;**

**import** javax**.**script**.**ScriptException**;**

**import** javax**.**swing**.**JButton**;**

**import** javax**.**swing**.**JFrame**;**

**import** javax**.**swing**.**JLabel**;**

**import** javax**.**swing**.**JOptionPane**;**

**import** javax**.**swing**.**JTextField**;**

/\*\*

\*\*/

public class ServerUDP **implements** Runnable **{**

private static DatagramSocket serverSocket**;**

private DatagramPacket receivePacket**,** sendPacket**;**

private InetAddress IPAddress**;**

private static Integer portNumber **=** 9876**;**

private byte**[]** receiveData**;**

private byte**[]** sendData**;**

static ScriptEngineManager sem**;**

static ScriptEngine engine**;**

static boolean isRunning **=** **false;**

static JLabel activeThread**;**

static JButton switchBtn**;**

static JTextField portTf**;**

static JLabel portLabel**;**

public ServerUDP**()** **{**

sem **=** **new** ScriptEngineManager**();**

engine **=** sem**.**getEngineByName**(**"JavaScript"**);**

**}**

public static void main**(**String**[]** args**)** **throws** IOException **{**

JFrame frame **=** **new** JFrame**(**"Bài TH số 2 - ServerUDP "**);**

frame**.**setBounds**(**200**,** 200**,** 400**,** 60**);**

frame**.**setResizable**(false);**

frame**.**setLayout**(new** GridLayout**(**1**,** 4**));**

frame**.**setDefaultCloseOperation**(**JFrame**.**EXIT\_ON\_CLOSE**);**

portLabel **=** **new** JLabel**(**"Cổng"**);**

portTf **=** **new** JTextField**(**"12345"**);**

activeThread **=** **new** JLabel**(**"0"**);**

switchBtn **=** **new** JButton**();**

switchBtn**.**setText**(**"Start"**);**

frame**.**add**(**portLabel**);**

frame**.**add**(**portTf**);**

frame**.**add**(**activeThread**);**

frame**.**add**(**switchBtn**);**

frame**.**setVisible**(true);**

Thread thread **=** **new** Thread**(new** ServerUDP**());**

switchBtn**.**addActionListener**(new** ActionListener**()** **{**

@Override

public void actionPerformed**(**ActionEvent arg0**)** **{**

**if** **(**isRunning**)** **{**

isRunning **=** **false;**

thread**.**interrupt**();**

serverSocket**.**close**();**

JOptionPane**.**showMessageDialog**(null,** "Đã ngừng Server"**);**

**}** **else** **{**

portNumber **=** Integer**.**parseInt**(**portTf**.**getText**());**

**try** **{**

serverSocket **=** **new** DatagramSocket**(**portNumber**);**

JOptionPane**.**showMessageDialog**(null,** "Đã khởi động Server"**);**

**}** **catch** **(**IOException e2**)** **{**

JOptionPane**.**showMessageDialog**(null,** "Lỗi khi khởi động Server"**);**

**}**

isRunning **=** **true;**

thread**.**start**();**

**}**

portTf**.**setEditable**(true);**

switchBtn**.**setText**(**"Start"**);**

activeThread**.**setText**(**"Luồng: " **+** Thread**.**activeCount**());**

**}**

**});**

**}**

@Override

public void run**()** **{**

**while** **(**isRunning**)** **{**

switchBtn**.**setText**(**"Stop"**);**

portTf**.**setEditable**(false);**

activeThread**.**setText**(**"Luồng: " **+** Thread**.**activeCount**());**

String input**,** output**;**

**try** **{**

receiveData **=** **new** byte**[**1024**];**

receivePacket **=** **new** DatagramPacket**(**receiveData**,** receiveData**.**length**);**

serverSocket**.**receive**(**receivePacket**);**

IPAddress **=** receivePacket**.**getAddress**();**

portNumber **=** receivePacket**.**getPort**();**

input **=** **new** String**(**receivePacket**.**getData**()).**trim**();**

System**.**out**.**print**(**"\nServer nhận được chuỗi: " **+** input**);**

output **=** String**.**valueOf**(**engine**.**eval**(**input**));**

System**.**out**.**print**(**"\n=======================> " **+** output**);**

sendData **=** **new** byte**[**1024**];**

sendData **=** output**.**getBytes**();**

sendPacket **=** **new** DatagramPacket**(**sendData**,** sendData**.**length**,**

IPAddress**,** portNumber**);**

serverSocket**.**send**(**sendPacket**);**

**}** **catch** **(**ScriptException e**)** **{**

output **=** "Lỗi cú pháp"**;**

**try** **{**

sendData **=** **new** byte**[**1024**];**

sendData **=** output**.**getBytes**();**

sendPacket **=** **new** DatagramPacket**(**sendData**,** sendData**.**length**,**

IPAddress**,** portNumber**);**

serverSocket**.**send**(**sendPacket**);**

**}** **catch** **(**IOException e1**)** **{**

**}**

System**.**out**.**print**(**"\n=======================> " **+** output**);**

**continue;**

**}** **catch** **(**IOException e**)** **{**

output **=** "Lỗi kết nối"**;**

**try** **{**

sendData **=** **new** byte**[**1024**];**

sendData **=** output**.**getBytes**();**

sendPacket **=** **new** DatagramPacket**(**sendData**,** sendData**.**length**,**

IPAddress**,** portNumber**);**

serverSocket**.**send**(**sendPacket**);**

**}** **catch** **(**IOException e1**)** **{**

**}**

System**.**out**.**print**(**"\n=======================> " **+** output**);**

**continue;**

**}**

**}**

**}**

**}**

### Mã nguồn Client

package pdq**.**udp**;**

**import** java**.**awt**.**BorderLayout**;**

**import** java**.**awt**.**Font**;**

**import** java**.**awt**.**GridLayout**;**

**import** java**.**awt**.**event**.**ActionEvent**;**

**import** java**.**awt**.**event**.**ActionListener**;**

**import** java**.**awt**.**event**.**WindowEvent**;**

**import** java**.**awt**.**event**.**WindowListener**;**

**import** java**.**io**.**IOException**;**

**import** java**.**net**.**DatagramPacket**;**

**import** java**.**net**.**DatagramSocket**;**

**import** java**.**net**.**InetAddress**;**

**import** java**.**net**.**SocketTimeoutException**;**

**import** java**.**net**.**UnknownHostException**;**

**import** javax**.**swing**.**JButton**;**

**import** javax**.**swing**.**JFrame**;**

**import** javax**.**swing**.**JLabel**;**

**import** javax**.**swing**.**JOptionPane**;**

**import** javax**.**swing**.**JPanel**;**

**import** javax**.**swing**.**JTextField**;**

/\*\*

\*

\*

\*\*/

public class ClientUDP **{**

private DatagramSocket clientSocket **=** **null;**

private static String serverHostname **=** "localhost"**;**

private InetAddress IPAddress**,** returnIPAddress**;**

private static Integer portNumber **=** 9876**;**

private byte**[]** sendData**;**

private byte**[]** receiveData**;**

private DatagramPacket sendPacket**,** receivePacket**;**

static JFrame splashFrame**,** frame**;**

JPanel outputPanel**,** inputPanel**;**

JTextField output**,** input**;**

public ClientUDP**()** **{**

initGUI**();**

addListeners**();**

**}**

private void initGUI**()** **{**

frame **=** **new** JFrame**(**"Bài TH số 2 - ClientUDP "

**+** serverHostname **+** ":" **+** portNumber**);**

frame**.**setLayout**(new** GridLayout**(**2**,** 1**));**

frame**.**setBounds**(**200**,** 200**,** 500**,** 200**);**

frame**.**setDefaultCloseOperation**(**JFrame**.**DISPOSE\_ON\_CLOSE**);**

inputPanel **=** **new** JPanel**();**

inputPanel**.**setLayout**(new** GridLayout**());**

outputPanel **=** **new** JPanel**();**

outputPanel**.**setLayout**(new** GridLayout**());**

input **=** **new** JTextField**();**

input**.**setFont**(new** Font**(**"Consolas"**,** Font**.**BOLD**,** 30**));**

input**.**setHorizontalAlignment**(**JTextField**.**RIGHT**);**

output **=** **new** JTextField**();**

output**.**setFont**(new** Font**(**"Consolas"**,** Font**.**BOLD**,** 30**));**

output**.**setHorizontalAlignment**(**JTextField**.**RIGHT**);**

output**.**setEditable**(false);**

inputPanel**.**add**(**input**);**

outputPanel**.**add**(**output**);**

frame**.**add**(**output**);**

frame**.**add**(**input**);**

frame**.**setVisible**(true);**

**}**

private void addListeners**()** **{**

input**.**addActionListener**(new** ActionListener**()** **{**

@Override

public void actionPerformed**(**ActionEvent arg0**)** **{**

String expresion **=** input**.**getText**();**

**try** **{**

System**.**out**.**println**(**"Đang gửi: " **+** expresion**);**

clientSocket **=** **new** DatagramSocket**();**

IPAddress **=** InetAddress**.**getByName**(**serverHostname**);**

sendData **=** **new** byte**[**1024**];**

sendData **=** expresion**.**getBytes**();**

sendPacket **=** **new** DatagramPacket**(**sendData**,** sendData**.**length**,**

IPAddress**,** portNumber**);**

clientSocket**.**send**(**sendPacket**);**

System**.**out**.**println**(**"Đã gửi!"**);**

receiveData **=** **new** byte**[**1024**];**

receivePacket **=** **new** DatagramPacket**(**receiveData**,** receiveData**.**length**);**

System**.**out**.**println**(**"Đang đợi phản hồi..."**);**

clientSocket**.**setSoTimeout**(**10000**);** // chờ 10s tối đa

String result **=** ""**;**

**try** **{**

clientSocket**.**receive**(**receivePacket**);**

result **=** **new** String**(**receivePacket**.**getData**());**

returnIPAddress **=** receivePacket**.**getAddress**();**

System**.**out**.**println**(**"Từ Server: " **+** returnIPAddress **+** ":"

**+** receivePacket**.**getPort**());**

**}** **catch** **(**SocketTimeoutException ste**)** **{**

System**.**out**.**println**(**"Chờ quá lâu: Gói tin đã bị mất!"**);**

**}**

output**.**setText**(**result**.**trim**());**

System**.**out**.**println**(**"Kết quả: " **+** result**.**trim**());**

clientSocket**.**close**();**

**}** **catch** **(**UnknownHostException e**)** **{**

JOptionPane**.**showMessageDialog**(null,** "Không tìm thấy Server."**);**

e**.**printStackTrace**();**

**}** **catch** **(**IOException e**)** **{**

JOptionPane**.**showMessageDialog**(null,** "Lỗi kết nối vào ra."**);**

e**.**printStackTrace**();**

**}**

**}**

**});**

frame**.**addWindowListener**(new** WindowListener**()** **{**

@Override

public void windowOpened**(**WindowEvent arg0**)** **{**

splashFrame**.**setVisible**(false);**

**}**

@Override

public void windowIconified**(**WindowEvent arg0**)** **{**

**}**

@Override

public void windowDeiconified**(**WindowEvent arg0**)** **{**

**}**

@Override

public void windowDeactivated**(**WindowEvent arg0**)** **{**

**}**

@Override

public void windowClosing**(**WindowEvent arg0**)** **{**

**}**

@Override

public void windowClosed**(**WindowEvent arg0**)** **{**

splashFrame**.**setVisible**(true);**

frame**.**setVisible**(false);**

**}**

@Override

public void windowActivated**(**WindowEvent arg0**)** **{**

**}**

**});**

**}**

/\*\*

\* **@param** args

\*/

public static void main**(**String**[]** args**)** **{**

splashFrame **=** **new** JFrame**(**"Nhập thông tin"**);**

splashFrame**.**setBounds**(**200**,** 200**,** 200**,** 100**);**

splashFrame**.**setResizable**(false);**

splashFrame**.**setDefaultCloseOperation**(**JFrame**.**EXIT\_ON\_CLOSE**);**

splashFrame**.**setLayout**(new** BorderLayout**());**

splashFrame**.**add**(new** JLabel**(**"Nhập địa chỉ máy chủ và cổng"**),**

BorderLayout**.**NORTH**);**

JTextField host**,** port**;**

host **=** **new** JTextField**(**"localhost"**);**

port **=** **new** JTextField**(**"12345"**);**

splashFrame**.**add**(**host**,** BorderLayout**.**CENTER**);**

splashFrame**.**add**(**port**,** BorderLayout**.**EAST**);**

JButton enterBtn **=** **new** JButton**(**"Enter"**);**

splashFrame**.**add**(**enterBtn**,** BorderLayout**.**SOUTH**);**

splashFrame**.**setVisible**(true);**

enterBtn**.**addActionListener**(new** ActionListener**()** **{**

@Override

public void actionPerformed**(**ActionEvent arg0**)** **{**

serverHostname **=** host**.**getText**();**

portNumber **=** Integer**.**parseInt**(**port**.**getText**());**

**new** ClientUDP**();**

**}**

**});**

**}**

**}**

### Kết quả chạy thử

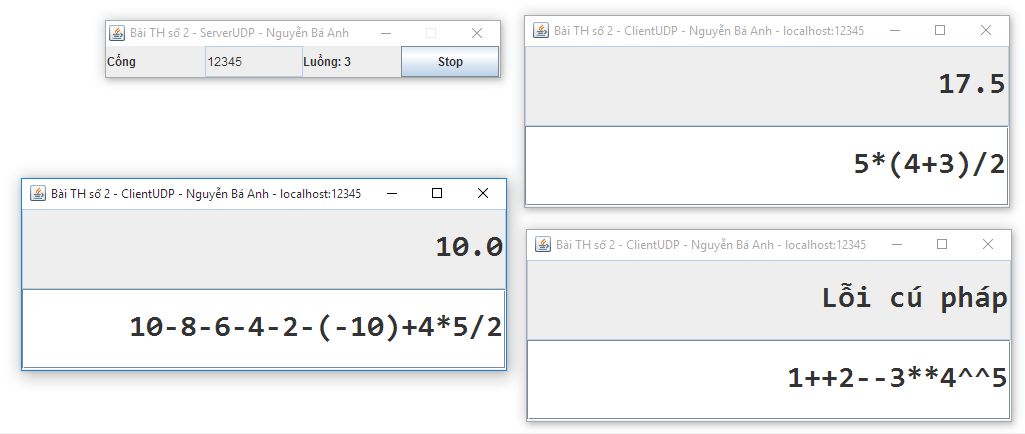


Figure x: Kết quả chạy thử trên hệ 1 Server - 3Clients

## Bài 3

### Mã nguồn Server

package pdq**.**udp**;**

**import** java**.**io**.**IOException**;**

**import** java**.**net**.**DatagramPacket**;**

**import** java**.**net**.**DatagramSocket**;**

**import** java**.**net**.**InetAddress**;**

**import** java**.**util**.**Enumeration**;**

**import** java**.**util**.**Hashtable**;**

**import** javax**.**swing**.**JOptionPane**;**

public class Server **{**

DatagramSocket server**;**

DatagramPacket receivePacket**;**

byte**[]** receiveData**;**

public Server**(**int port**)** **{**

**try** **{**

server **=** **new** DatagramSocket**(**port**);**

**while** **(true)** **{**

receiveData **=** **new** byte**[**1024**];**

receivePacket **=** **new** DatagramPacket**(**receiveData**,** receiveData**.**length**);**

server**.**receive**(**receivePacket**);**

ClientConnection cc **=** **new** ClientConnection**(**server**,** receivePacket**);**

cc**.**start**();**

**}**

**}** **catch** **(**Exception e**)** **{**

e**.**printStackTrace**();**

**}**

**}**

public static void main**(**String**[]** args**)** **{**

int port **=** Integer**.**parseInt**(**

JOptionPane**.**showInputDialog**(**"Nhập cổng: "**,** "12345"**));**

**new** Server**(**port**);**

**}**

static class ClientConnection **extends** Thread **{**

DatagramSocket server**;**

int connectionNumber**;**

DatagramPacket sendPacket**;**

DatagramPacket receivePacket**;**

byte**[]** sendData**;**

byte**[]** receiveData**;**

private static Hashtable**<**Integer**,**InetAddress**>** connectionList

**=** **new** Hashtable**<**Integer**,**InetAddress**>();**

InetAddress address**;**

int port**;**

public ClientConnection**(**DatagramSocket server**,** DatagramPacket receivePacket**)** **{**

**this.**connectionNumber **=** connectionList**.**size**();**

**this.**server**=** server**;**

**this.**receivePacket **=** receivePacket**;**

**}**

public void run**()** **{**

**try** **{**

receiveData **=** **new** byte**[**1024**];**

receiveData **=** receivePacket**.**getData**();**

String message **=** **new** String**(**receiveData**).**trim**();**

address **=** receivePacket**.**getAddress**();**

port **=** receivePacket**.**getPort**();**

**if(!**connectionList**.**containsKey**(**port**)){**

connectionList**.**put**(**port**,**address**);**

**}**

sendToAll**(**message**);**

**}** **catch** **(**Exception e**)** **{**

e**.**printStackTrace**();**

**}**

**}**

/\*\*

\* Phát thông điệp đến tất cả client đang kết nối

\* **@param** message nội dung

\*/

public void sendToAll**(**String message**)** **{**

synchronized **(**connectionList**)** **{**

Enumeration**<**Integer**>** keys**;**

keys **=** connectionList**.**keys**();**

**while(**keys**.**hasMoreElements**()){**

port **=** **(**Integer**)** keys**.**nextElement**();**

address **=** connectionList**.**get**(**port**);**

sendData **=** **new** byte**[**1024**];**

sendData **=** **(**message**).**getBytes**();**

sendPacket **=** **new** DatagramPacket**(**sendData**,** sendData**.**length**,**

address**,** port**);**

**try** **{**

server**.**send**(**sendPacket**);**

**}** **catch** **(**IOException e**)** **{**

e**.**printStackTrace**();**

**}**

**}**

**}**

**}**

**}**

}

### Mã nguồn Client

package pdq**.**udp**;**

**import** java**.**awt**.**BorderLayout**;**

**import** java**.**awt**.**GridLayout**;**

**import** java**.**awt**.**event**.**ActionEvent**;**

**import** java**.**awt**.**event**.**ActionListener**;**

**import** java**.**io**.**IOException**;**

**import** java**.**net**.**DatagramPacket**;**

**import** java**.**net**.**DatagramSocket**;**

**import** java**.**net**.**InetAddress**;**

**import** javax**.**swing**.**JButton**;**

**import** javax**.**swing**.**JFrame**;**

**import** javax**.**swing**.**JLabel**;**

**import** javax**.**swing**.**JList**;**

**import** javax**.**swing**.**JOptionPane**;**

**import** javax**.**swing**.**JPanel**;**

**import** javax**.**swing**.**JScrollPane**;**

**import** javax**.**swing**.**JTextArea**;**

**import** javax**.**swing**.**JTextField**;**

/\*\*

\*

\*\*/

public class Client **extends** JFrame **implements** ActionListener **{**

private static final long serialVersionUID **=** 1L**;**

JTextField userTf**;**

public Client**(**String title**)** **{**

**super(**title**);**

JPanel userPanel **=** **new** JPanel**();**

JPanel buttonPanel **=** **new** JPanel**();**

JLabel userLb **=** **new** JLabel**(**"Tên"**);**

userTf **=** **new** JTextField**(**30**);**

JButton loginBtn **=** **new** JButton**(**"Login"**);**

JButton resetBtn **=** **new** JButton**(**"Reset"**);**

JButton exitBtn **=** **new** JButton**(**"Exit"**);**

loginBtn**.**addActionListener**(this);**

resetBtn**.**addActionListener**(this);**

exitBtn**.**addActionListener**(this);**

userPanel**.**add**(**userLb**);**

userPanel**.**add**(**userTf**);**

buttonPanel**.**add**(**loginBtn**);**

buttonPanel**.**add**(**resetBtn**);**

buttonPanel**.**add**(**exitBtn**);**

add**(**userPanel**);**

add**(**buttonPanel**);**

setBounds**(**200**,** 200**,** 400**,** 150**);**

setLayout**(new** GridLayout**(**2**,** 1**));**

setVisible**(true);**

setDefaultCloseOperation**(**JFrame**.**EXIT\_ON\_CLOSE**);**

**}**

@Override

public void actionPerformed**(**ActionEvent e**)** **{**

String button **=** e**.**getActionCommand**();**

**if** **(**button**.**equals**(**"Login"**))** **{**

String user **=** userTf**.**getText**();**

String ip **=** JOptionPane**.**showInputDialog**(**"Nhập địa chỉ máy chủ"**,**

"localhost"**);**

int port **=** Integer**.**parseInt**(**JOptionPane**.**showInputDialog**(**"Cổng"**,** "12345"**));**

**new** ChatRoom**(**user**,** ip**,** port**);**

dispose**();**

**}**

**if** **(**button**.**equals**(**"Reset"**))** **{**

userTf**.**setText**(**""**);**

**}**

**if** **(**button**.**equals**(**"Exit"**))** **{**

System**.**exit**(**0**);**

**}**

**}**

public static void main**(**String**[]** args**)** **{**

**new** Client**(**"Đăng nhập"**);**

**}**

class ChatRoom **extends** JFrame **implements** Runnable **{**

private static final long serialVersionUID **=** 1L**;**

JTextArea messageTa**,** allMessagesTa**;**

JPanel titlePanel**,** messagePanel**,** inputPanel**;**

JLabel titleLb**;**

JButton sendBtn**;**

JList**<**String**>** userList **=** **new** JList**<**String**>();**

String user**;**

int port**;**

private DatagramSocket socket**;**

private byte**[]** sendData**;**

private byte**[]** receiveData**;**

private DatagramPacket sendPacket**;**

private DatagramPacket receivePacket**;**

private InetAddress IPAddress**;**

public ChatRoom**(**String user**,** String address**,** int port**)** **{**

**super(**"Phòng Chát (UDP)"**);**

**try** **{**

System**.**out**.**println**(**address **+** " " **+** port**);**

**this.**user **=** user**;**

**this.**port **=** port**;**

socket **=** **new** DatagramSocket**();**

IPAddress **=** InetAddress**.**getByName**(**address**);**

sendData **=** **new** byte**[**1024**];**

sendData **=** **(this.**user**+**" đã tham gia phòng chat"**).**getBytes**();**

sendPacket **=** **new** DatagramPacket**(**sendData**,** sendData**.**length**,**

IPAddress**,** **this.**port**);**

socket**.**send**(**sendPacket**);**

**this.**initGUI**();**

**new** Thread**(this).**start**();**

**}** **catch** **(**Exception e**)** **{**

e**.**printStackTrace**();**

**}**

**}**

public void initGUI**()** **{**

**this.**setDefaultCloseOperation**(**EXIT\_ON\_CLOSE**);**

**this.**setLayout**(new** BorderLayout**());**

titlePanel **=** **new** JPanel**();**

titleLb **=** **new** JLabel**(**IPAddress**.**getHostAddress**()**

**+** ":" **+** port **+** " >> " **+** user**);**

titlePanel**.**add**(**titleLb**);**

messagePanel **=** **new** JPanel**();**

messagePanel**.**setLayout**(new** BorderLayout**());**

allMessagesTa **=** **new** JTextArea**();**

allMessagesTa**.**setWrapStyleWord**(true);**

allMessagesTa**.**setLineWrap**(true);**

allMessagesTa**.**setEditable**(false);**

JScrollPane scrollPane **=** **new** JScrollPane**(**allMessagesTa**);**

scrollPane**.**setVerticalScrollBarPolicy**(**

JScrollPane**.**VERTICAL\_SCROLLBAR\_AS\_NEEDED**);**

messagePanel**.**add**(**scrollPane**);**

inputPanel **=** **new** JPanel**();**

inputPanel**.**setLayout**(new** BorderLayout**());**

messageTa **=** **new** JTextArea**(**2**,** 35**);**

messageTa**.**setWrapStyleWord**(true);**

messageTa**.**setLineWrap**(true);**

JScrollPane spMessage **=** **new** JScrollPane**(**messageTa**);**

inputPanel**.**add**(**spMessage**);**

sendBtn **=** **new** JButton**(**"Gửi"**);**

sendBtn**.**addActionListener**(new** ActionListener**()** **{**

@Override

public void actionPerformed**(**ActionEvent ae**)** **{**

**try** **{**

sendData **=** **new** byte**[**1024**];**

sendData **=** **(**user**+**":"**+** messageTa**.**getText**()).**getBytes**();**

messageTa**.**setText**(**""**);**

sendPacket **=** **new** DatagramPacket**(**sendData**,** sendData**.**length**,**

IPAddress**,** port**);**

socket**.**send**(**sendPacket**);**

**}** **catch** **(**IOException e**)** **{**

**}**

**}**

**});**

inputPanel**.**add**(**sendBtn**,** BorderLayout**.**EAST**);**

add**(**titlePanel**,** BorderLayout**.**NORTH**);**

add**(**messagePanel**,** BorderLayout**.**CENTER**);**

add**(**inputPanel**,** BorderLayout**.**SOUTH**);**

setSize**(**400**,** 300**);**

setLocation**(**400**,** 100**);**

setVisible**(true);**

**}**

@Override

public void run**()** **{**

**try** **{**

**while** **(true)** **{**

receiveData **=** **new** byte**[**1024**];**

receivePacket **=** **new** DatagramPacket**(**receiveData**,** receiveData**.**length**);**

socket**.**receive**(**receivePacket**);**

receiveData **=** receivePacket**.**getData**();**

String message **=** **new** String**(**receiveData**).**trim**();**

allMessagesTa**.**append**(**message **+** "\n"**);**

**}**

**}** **catch** **(**Exception e**)** **{**

e**.**printStackTrace**();**

**}**

**}**

**}**

}

### Kết quả chạy thử

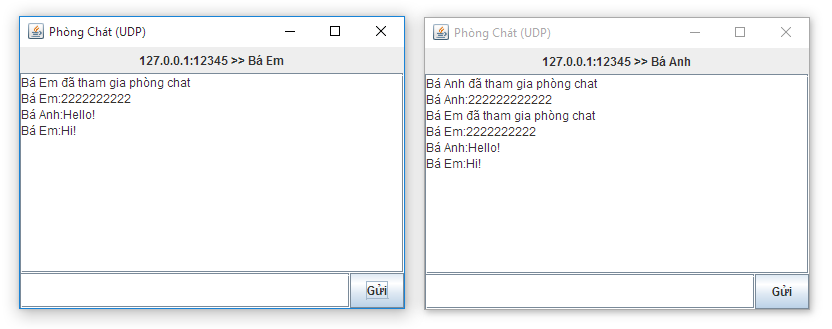


Figure xi: Phòng chat UDP

# Bài Thực hành số 3

## Yêu cầu

Thiết kế một Website quản lý nhân sự phòng ban theo mẫu sau với cơ sở dữ liệu bất kỳ (Gồm các trang: xem thông tin, chèn thông tin, cập nhật thông tin, xóa thông tin, tìm kiếm thông tin). Kết quả minh họa:



## Sơ đồ hệ thống

**Trang chủ**

**Đăng nhập**

**Xem danh sách nhân viên**

**Xem danh sách phòng ban**

**Tìm kiếm**

**Xoá tất cả**

**Xem chi tiết nhân viên**

**Xem chi tiết phòng ban**

**Thêm nhân viên**

**Xóa nhân viên**

**Thêm phòng ban**

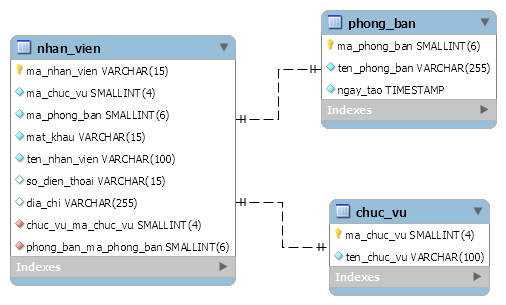
**Xóa phòng ban**

**Cập nhật**

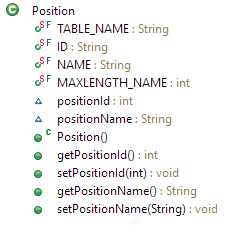
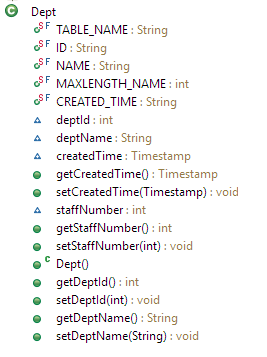
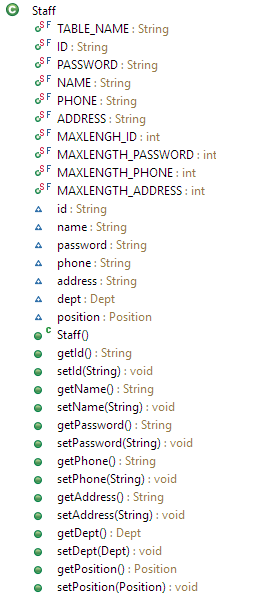
**Cập nhật**

**Đăng xuất**

## Cơ sở dữ liệu

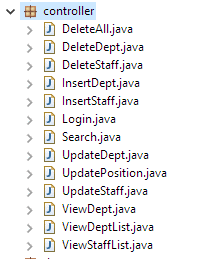
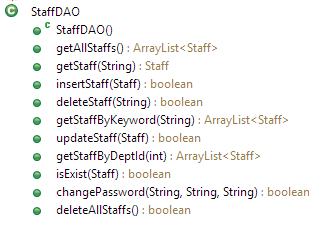
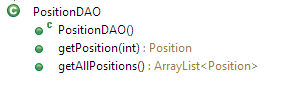
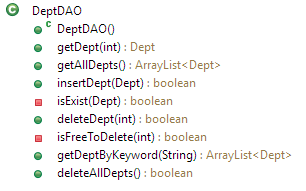
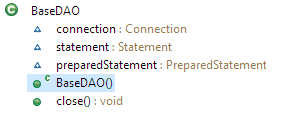


## Sơ đồ lớp



1..1

1..1



## Mã nguồn

Mã nguồn khá dài nên trong phạm vi báo cáo em xin phép chỉ chép vào package dao, mọi mã nguồn khác được đặt trong CD đính kèm báo cáo này.

### BaseDAO.java

package dao**;**

**import** java**.**sql**.**Connection**;**

**import** java**.**sql**.**DriverManager**;**

**import** java**.**sql**.**PreparedStatement**;**

**import** java**.**sql**.**SQLException**;**

**import** java**.**sql**.**Statement**;**

/\*\*

\*

\*

\*\*/

public class BaseDAO **{**

Connection connection **=** **null;**

Statement statement **=** **null;**

PreparedStatement preparedStatement **=** **null;**

public BaseDAO**()** **{**

**try** **{**

Class**.**forName**(**"com.mysql.jdbc.Driver"**);**

String address **=** "jdbc:mysql://localhost:3306/ql\_phongban?useUnicode=true&characterEncoding=utf-8"**;**

connection **=** DriverManager**.**getConnection**(**address**,**"root"**,**""**);**

**}** **catch** **(**Exception e**)** **{**

System**.**err**.**println**(**"[Database constructor] Loi: " **+** e**);**

**}**

**}**

/\*\*

\* Đóng kết nối và statement

\*

\* **@throws** SQLException

\*/

public void close**()** **throws** SQLException **{**

**if** **(**statement **!=** **null)** **{**

statement**.**close**();**

**}**

**if** **(**preparedStatement **!=** **null)** **{**

preparedStatement**.**close**();**

**}**

**if** **(**connection **!=** **null)** **{**

connection**.**close**();**

**}**

**}**

**}**

### StaffDAO.java

package dao**;**

**import** java**.**sql**.**ResultSet**;**

**import** java**.**sql**.**SQLException**;**

**import** java**.**util**.**ArrayList**;**

**import** java**.**sql**.**PreparedStatement**;**

**import** bean**.**Dept**;**

**import** bean**.**Position**;**

**import** bean**.**Staff**;**

/\*\*

\*

\*

\*\*/

public class StaffDAO **extends** BaseDAO **{**

public StaffDAO**()** **{**

**super();**

**}**

public ArrayList**<**Staff**>** getAllStaffs**()** **{**

String query **=** "select " **+** Staff**.**ID **+** " from " **+** Staff**.**TABLE\_NAME**;**

**try** **{**

preparedStatement **=** connection**.**prepareStatement**(**query**);**

System**.**out**.**print**(**">> StaffDAO: Chạy truy vấn :"

**+** preparedStatement**.**toString**());** // FIXME

ResultSet rs **=** preparedStatement**.**executeQuery**();**

ArrayList**<**Staff**>** staffList **=** **new** ArrayList**<**Staff**>();**

**while** **(**rs**.**next**())** **{**

staffList**.**add**(this.**getStaff**(**rs**.**getString**(**1**)));**

**}**

preparedStatement**.**close**();**

**return** staffList**;**

**}** **catch** **(**SQLException e**)** **{**

System**.**err**.**print**(**"Lỗi: " **+** e **+** " > NULL"**);** // FIXME

e**.**printStackTrace**();**

**return** **null;**

**}**

**}**

public Staff getStaff**(**String id**)** **{**

**if** **(**id **==** **null** **||** ""**.**equals**(**id**.**trim**()))** **{**

**return** **null;**

**}**

String query **=** "select \* from " **+** Staff**.**TABLE\_NAME

**+** " where " **+** Staff**.**ID **+** "=?"**;**

**try** **{**

preparedStatement **=** connection**.**prepareStatement**(**query**);**

preparedStatement**.**setString**(**1**,** id**);**

ResultSet rs **=** preparedStatement**.**executeQuery**();**

Staff staff **=** **new** Staff**();**

**if** **(**rs**.**next**())** **{**

staff**.**setId**(**rs**.**getString**(**Staff**.**ID**));**

staff**.**setPassword**(**rs**.**getString**(**Staff**.**PASSWORD**));**

staff**.**setName**(**rs**.**getString**(**Staff**.**NAME**));**

staff**.**setAddress**(**rs**.**getString**(**Staff**.**ADDRESS**));**

staff**.**setPhone**(**rs**.**getString**(**Staff**.**PHONE**));**

staff**.**setDept**(new** DeptDAO**().**getDept**(**rs**.**getInt**(**Dept**.**ID**)));**

staff**.**setPosition**(new** PositionDAO**().**getPosition**(**rs**.**getInt**(**Position**.**ID**)));**

**}**

preparedStatement**.**close**();**

**return** staff**;**

**}** **catch** **(**SQLException e**)** **{**

e**.**printStackTrace**();**

**return** **null;**

**}**

**}**

public boolean insertStaff**(**Staff staff**)** **{**

System**.**out**.**print**(**">> StaffDAO: kiểm tra hợp lệ dữ liệu..."**);** // FIXME

// Kiểm tra sự hợp lệ của thông tin

**if** **(**staff **==** **null** **||** ""**.**equals**(**staff**.**getName**().**trim**())**

**||** ""**.**equals**(**staff**.**getPassword**().**trim**())**

**||** Staff**.**MAXLENGH\_ID **<** staff**.**getId**().**length**()**

**||** Staff**.**MAXLENGTH\_PASSWORD **<** staff**.**getPassword**().**length**()**

**||** Staff**.**MAXLENGTH\_PHONE **<** staff**.**getPhone**().**length**()**

**||** Staff**.**MAXLENGTH\_ADDRESS **<** staff**.**getAddress**().**length**()**

**||** staff**.**getId**().**length**()** **<** 3

**||** staff**.**getDept**()** **==** **null** **||** staff**.**getPosition**()** **==** **null)** **{**

System**.**out**.**print**(**"...! "**);**

**return** **false;**

**}**

System**.**out**.**print**(**"OK "**);** // FIXME

// Cập nhật CSDL

String query **=** "insert into " **+** Staff**.**TABLE\_NAME **+** "(" **+** Staff**.**ID **+** ","

**+** Staff**.**PASSWORD **+** "," **+** Staff**.**NAME **+** "," **+** Staff**.**ADDRESS **+** ","

**+** Staff**.**PHONE **+** "," **+** Dept**.**ID **+** "," **+** Position**.**ID

**+** ") values(?,?,?,?,?,?,?)"**;**

**try** **{**

preparedStatement **=** connection**.**prepareStatement**(**query**);**

preparedStatement**.**setString**(**1**,** staff**.**getId**());**

preparedStatement**.**setString**(**2**,** staff**.**getPassword**());**

preparedStatement**.**setString**(**3**,** staff**.**getName**());**

preparedStatement**.**setString**(**4**,** staff**.**getAddress**());**

preparedStatement**.**setString**(**5**,** staff**.**getPhone**());**

preparedStatement**.**setInt**(**6**,** staff**.**getDept**().**getDeptId**());**

preparedStatement**.**setInt**(**7**,** staff**.**getPosition**().**getPositionId**());**

System**.**out**.**print**(**"> Chạy truy vấn: " **+** preparedStatement**.**toString**()** **+** "..."**);** // FIXME

boolean added **=** preparedStatement**.**executeUpdate**()** **>** 0 **?** **true** **:** **false;**

preparedStatement**.**close**();**

**return** added**;**

**}** **catch** **(**SQLException e**)** **{**

System**.**err**.**print**(**"> Lỗi: " **+** e**);** // FIXME

**return** **false;**

**}**

**}**

public boolean deleteStaff**(**String id**)** **{**

System**.**out**.**println**(**">> StaffDAO: validting... "**);**

**if** **(**id **==** **null** **||** id**.**trim**().**equals**(**""**)** **||** id**.**length**()** **>** Staff**.**MAXLENGH\_ID**)** **{**

**return** **false;**

**}**

System**.**out**.**print**(**"OK!\nDeleting... "**);** // FIXME

String query **=** "delete from " **+** Staff**.**TABLE\_NAME **+** " where " **+** Staff**.**ID **+** "=?"**;**

**try** **{**

preparedStatement **=** connection**.**prepareStatement**(**query**);**

preparedStatement**.**setString**(**1**,** id**);**

System**.**out**.**print**(**preparedStatement**.**toString**());** // FIXME

**return** preparedStatement**.**executeUpdate**()** **>** 0**;**

**}** **catch** **(**SQLException e**)** **{**

System**.**out**.**print**(**"Failed [sql exception]."**);** // FIXME

**return** **false;**

**}**

**}**

public ArrayList**<**Staff**>** getStaffByKeyword**(**String searchQuery**)** **{**

String query **=** "select " **+** Staff**.**ID **+** " from " **+** Staff**.**TABLE\_NAME

**+** " where " **+** Staff**.**ID **+** " like ? or "

**+** Staff**.**NAME **+** " like ? or " **+** Staff**.**ADDRESS **+** " like ?"**;**

**try** **{**

preparedStatement **=** connection**.**prepareStatement**(**query**);**

preparedStatement**.**setString**(**1**,** "%" **+** searchQuery **+** "%"**);**

preparedStatement**.**setString**(**2**,** "%" **+** searchQuery **+** "%"**);**

preparedStatement**.**setString**(**3**,** "%" **+** searchQuery **+** "%"**);**

System**.**out**.**print**(**preparedStatement**.**toString**());** // FIXME

ArrayList**<**Staff**>** staffList **=** **new** ArrayList**<**Staff**>();**

ResultSet rs **=** preparedStatement**.**executeQuery**();**

**while** **(**rs**.**next**())** **{**

staffList**.**add**(this.**getStaff**(**rs**.**getString**(**Staff**.**ID**)));**

**}**

**return** staffList**;**

**}** **catch** **(**SQLException e**)** **{**

System**.**out**.**print**(**"Failed [sql exception]."**);** // FIXME

**return** **null;**

**}**

**}**

public boolean updateStaff**(**Staff staff**)** **{**

System**.**out**.**print**(**">> StaffDAO: kiểm tra hợp lệ dữ liệu..."**);** // FIXME

// Kiểm tra sự hợp lệ của thông tin

**if** **(**staff **==** **null** **||** ""**.**equals**(**staff**.**getName**().**trim**())** **||** Staff**.**MAXLENGH\_ID **<** staff**.**getId**().**length**()**

**||** Staff**.**MAXLENGTH\_PHONE **<** staff**.**getPhone**().**length**()**

**||** Staff**.**MAXLENGTH\_ADDRESS **<** staff**.**getAddress**().**length**()** **||** staff**.**getDept**()** **==** **null**

**||** staff**.**getPosition**()** **==** **null)** **{**

System**.**out**.**print**(**"...! "**);**

**return** **false;**

**}**

System**.**out**.**print**(**"OK "**);** // FIXME

// Cập nhật CSDL

String query **=** "update " **+** Staff**.**TABLE\_NAME **+** " set "

**+** Staff**.**NAME **+** "=?, " **+** Staff**.**ADDRESS **+** "=?, "

**+** Staff**.**PHONE **+** "=?, " **+** Dept**.**ID **+** "=?, " **+** Position**.**ID **+** "=? "

**+** " where " **+** Staff**.**ID **+** "=? "**;**

**try** **{**

preparedStatement **=** connection**.**prepareStatement**(**query**);**

preparedStatement**.**setString**(**1**,** staff**.**getName**());**

preparedStatement**.**setString**(**2**,** staff**.**getAddress**());**

preparedStatement**.**setString**(**3**,** staff**.**getPhone**());**

preparedStatement**.**setInt**(**4**,** staff**.**getDept**().**getDeptId**());**

preparedStatement**.**setInt**(**5**,** staff**.**getPosition**().**getPositionId**());**

preparedStatement**.**setString**(**6**,** staff**.**getId**());**

System**.**out**.**print**(**"> Chạy truy vấn: " **+** preparedStatement**.**toString**()** **+** "..."**);** // FIXME

boolean added **=** preparedStatement**.**executeUpdate**()** **>** 0 **?** **true** **:** **false;**

preparedStatement**.**close**();**

**return** added**;**

**}** **catch** **(**SQLException e**)** **{**

System**.**err**.**print**(**"> Lỗi: " **+** e**);** // FIXME

**return** **false;**

**}**

**}**

public ArrayList**<**Staff**>** getStaffByDeptId**(**int id**)** **{**

String query **=** "select " **+** Staff**.**ID **+** " from " **+** Staff**.**TABLE\_NAME

**+** " where " **+** Dept**.**ID **+** "=?"**;**

**try** **{**

preparedStatement **=** connection**.**prepareStatement**(**query**);**

preparedStatement**.**setInt**(**1**,** id**);**

ArrayList**<**Staff**>** staffList **=** **new** ArrayList**<**Staff**>();**

ResultSet rs **=** preparedStatement**.**executeQuery**();**

**while** **(**rs**.**next**())** **{**

staffList**.**add**(this.**getStaff**(**rs**.**getString**(**Staff**.**ID**)));**

**}**

**return** staffList**;**

**}** **catch** **(**SQLException e**)** **{**

System**.**out**.**print**(**"Lỗi: " **+** e**);** // FIXME

**return** **null;**

**}**

**}**

public boolean isExist**(**Staff staff**)** **{**

**if** **(**staff **==** **null** **||** staff**.**getId**()** **==** **null** **||** staff**.**getPassword**()** **==** **null**

**||** staff**.**getPassword**().**trim**().**equals**(**""**)**

**||** staff**.**getId**().**trim**().**equals**(**""**))** **{**

**return** **false;**

**}** **else** **{**

String checkQuery **=** "select \* from " **+** Staff**.**TABLE\_NAME

**+** " where " **+** Staff**.**ID **+** "=? and " **+** Staff**.**PASSWORD **+** "=?"**;**

**try** **{**

PreparedStatement preparedStatement **=**

connection**.**prepareStatement**(**checkQuery**);**

preparedStatement**.**setString**(**1**,** staff**.**getId**());**

preparedStatement**.**setString**(**2**,** staff**.**getPassword**());**

**return** preparedStatement**.**executeQuery**().**next**();**

**}** **catch** **(**SQLException e**)** **{**

System**.**out**.**print**(**"Lỗi: " **+** e**);** // FIXME

**return** **false;**

**}**

**}**

**}**

public boolean changePassword**(**String userId**,** String oldPassword**,** String newPassword**)** **{**

String updateQuery **=** "update " **+** Staff**.**TABLE\_NAME

**+** " set " **+** Staff**.**PASSWORD **+** "=? where " **+** Staff**.**ID **+** "=? and "

**+** Staff**.**PASSWORD **+** "=?"**;**

**try** **{**

PreparedStatement preparedStatement **=**

connection**.**prepareStatement**(**updateQuery**);**

preparedStatement**.**setString**(**1**,** newPassword**);**

preparedStatement**.**setString**(**2**,** userId**);**

preparedStatement**.**setString**(**3**,** oldPassword**);**

**return** preparedStatement**.**executeUpdate**()** **>** 0**;**

**}** **catch** **(**SQLException e**)** **{**

System**.**out**.**print**(**"Lỗi: " **+** e**);** // FIXME

**return** **false;**

**}**

**}**

public boolean deleteAllStaffs**()** **{**

String selectAll **=** "select " **+** Staff**.**ID **+** " from " **+** Staff**.**TABLE\_NAME

**+** " where " **+** Staff**.**ID **+** " <> 'admin'"**;**

String resetAdmin **=** "update " **+** Staff**.**TABLE\_NAME

**+** " set " **+** Staff**.**PASSWORD **+** " ='admin' where " **+** Staff**.**ID **+** "='admin'"**;**

**try** **{**

PreparedStatement preparedStatement **=**

connection**.**prepareStatement**(**selectAll**);**

ResultSet rs **=** preparedStatement**.**executeQuery**();**

**while** **(**rs**.**next**())** **{**

**this.**deleteStaff**(**rs**.**getString**(**Staff**.**ID**));**

**}**

preparedStatement **=** connection**.**prepareStatement**(**resetAdmin**);**

**return** preparedStatement**.**executeUpdate**()** **>** 0**;**

**}** **catch** **(**SQLException e**)** **{}**

**return** **false;**

**}**

**}**

### DeptDAO.java

package dao**;**

**import** java**.**sql**.**PreparedStatement**;**

**import** java**.**sql**.**ResultSet**;**

**import** java**.**sql**.**SQLException**;**

**import** java**.**util**.**ArrayList**;**

**import** bean**.**Dept**;**

**import** bean**.**Staff**;**

public class DeptDAO **extends** BaseDAO **{**

public DeptDAO**()** **{**

**super();**

**}**

public Dept getDept**(**int id**)** **{**

String query **=** "select \* from " **+** Dept**.**TABLE\_NAME **+** " where " **+** Dept**.**ID **+** "=?"**;**

**try** **{**

preparedStatement **=** connection**.**prepareStatement**(**query**);**

preparedStatement**.**setInt**(**1**,** id**);**

ResultSet rs **=** preparedStatement**.**executeQuery**();**

Dept dept **=** **new** Dept**();**

**if** **(**rs**.**next**())** **{**

dept**.**setDeptId**(**rs**.**getInt**(**Dept**.**ID**));**

dept**.**setDeptName**(**rs**.**getString**(**Dept**.**NAME**));**

dept**.**setCreatedTime**(**rs**.**getTimestamp**(**Dept**.**CREATED\_TIME**));**

**}**

preparedStatement**.**close**();**

**return** dept**;**

**}** **catch** **(**SQLException e**)** **{**

e**.**printStackTrace**();**

**return** **null;**

**}**

**}**

public ArrayList**<**Dept**>** getAllDepts**()** **{**

System**.**out**.**print**(**">> DeptDAO: "**);** // FIXME

String query **=** "select " **+** Dept**.**ID **+** " from " **+** Dept**.**TABLE\_NAME**;**

**try** **{**

preparedStatement **=** connection**.**prepareStatement**(**query**);**

System**.**out**.**print**(**"> Chạy truy vấn: " **+** preparedStatement**.**toString**());** // FIXME

ResultSet rs **=** preparedStatement**.**executeQuery**();**

ArrayList**<**Dept**>** deptList **=** **new** ArrayList**<**Dept**>();**

**while** **(**rs**.**next**())** **{**

deptList**.**add**(this.**getDept**(**rs**.**getInt**(**1**)));**

**}**

preparedStatement**.**close**();**

System**.**out**.**print**(**"> Đã lấy danh sách phòng ban. "**);** // FIXME

**return** deptList**;**

**}** **catch** **(**SQLException e**)** **{**

System**.**out**.**print**(**"> Lỗi " **+** e **+** " > NULL "**);** // FIXME

**return** **null;**

**}**

**}**

public boolean insertDept**(**Dept dept**)** **{**

System**.**out**.**print**(**">> DeptDAO: Validating..."**);** // FIXME

**if** **(**dept **==** **null** **||** dept**.**getDeptName**()** **==** **null)** **{**

System**.**out**.**print**(**"Failed [null]."**);** // FIXME

**return** **false;**

**}** **else** **if** **(**""**.**equals**(**dept**.**getDeptName**().**trim**())** **||** Dept**.**MAXLENGTH\_NAME **<** dept**.**getDeptName**().**length**())** **{**

System**.**out**.**print**(**"Failed [empty]."**);** // FIXME

**return** **false;**

**}** **else** **if** **(this.**isExist**(**dept**))** **{**

System**.**out**.**print**(**"Failed [existed before]."**);** // FIXME

**return** **false;**

**}**

System**.**out**.**print**(**"OK!\nInserting... "**);** // FIXME

String query **=** "insert into " **+** Dept**.**TABLE\_NAME **+** "(" **+** Dept**.**NAME **+** ") values(?)"**;**

**try** **{**

preparedStatement **=** connection**.**prepareStatement**(**query**);**

preparedStatement**.**setString**(**1**,** dept**.**getDeptName**());**

System**.**out**.**print**(**preparedStatement**.**toString**());** // FIXME

**return** preparedStatement**.**executeUpdate**()** **>** 0**;**

**}** **catch** **(**SQLException e**)** **{**

System**.**out**.**print**(**"Failed [sql exception]."**);** // FIXME

**return** **false;**

**}**

**}**

private boolean isExist**(**Dept dept**)** **{**

String query **=** "select \* from " **+** Dept**.**TABLE\_NAME **+** " where " **+** Dept**.**NAME **+** "=?"**;**

**try** **{**

preparedStatement **=** connection**.**prepareStatement**(**query**);**

preparedStatement**.**setString**(**1**,** dept**.**getDeptName**().**trim**());**

**return** preparedStatement**.**executeQuery**().**next**();**

**}** **catch** **(**SQLException e**)** **{**

**return** **true;**

**}**

**}**

public boolean deleteDept**(**int id**)** **{**

System**.**out**.**print**(**">> DeptDAO: Validating..."**);** // FIXME

**if** **(**id **<=** 0**)** **{**

System**.**out**.**print**(**"Failed [negative id]."**);** // FIXME

**return** **false;**

**}** **else** **if** **(!this.**isFreeToDelete**(**id**))** **{**

System**.**out**.**print**(**"Failed [not free to delete]."**);** // FIXME

**return** **false;**

**}**

System**.**out**.**print**(**"OK!\nDeleting... "**);** // FIXME

String query **=** "delete from " **+** Dept**.**TABLE\_NAME **+** " where " **+** Dept**.**ID **+** "=?"**;**

**try** **{**

preparedStatement **=** connection**.**prepareStatement**(**query**);**

preparedStatement**.**setInt**(**1**,** id**);**

System**.**out**.**print**(**preparedStatement**.**toString**());** // FIXME

**return** preparedStatement**.**executeUpdate**()** **>** 0**;**

**}** **catch** **(**SQLException e**)** **{**

System**.**out**.**print**(**"Failed [sql exception]."**);** // FIXME

**return** **false;**

**}**

**}**

private boolean isFreeToDelete**(**int id**)** **{**

String query **=** "select \* from " **+** Staff**.**TABLE\_NAME **+** " where " **+** Dept**.**ID **+** "=?"**;**

**try** **{**

preparedStatement **=** connection**.**prepareStatement**(**query**);**

preparedStatement**.**setInt**(**1**,** id**);**

System**.**out**.**print**(**preparedStatement**.**toString**());** // FIXME

**return** **!(**preparedStatement**.**executeQuery**().**next**());**

**}** **catch** **(**SQLException e**)** **{**

System**.**out**.**print**(**"Failed [sql exception]."**);** // FIXME

**return** **false;**

**}**

**}**

public ArrayList**<**Dept**>** getDeptByKeyword**(**String searchQuery**)** **{**

String query **=** "select " **+** Dept**.**ID **+** " from " **+** Dept**.**TABLE\_NAME **+** " where " **+** Dept**.**NAME **+** " like ?"**;**

**try** **{**

preparedStatement **=** connection**.**prepareStatement**(**query**);**

preparedStatement**.**setString**(**1**,** "%" **+** searchQuery **+** "%"**);**

ResultSet rs **=** preparedStatement**.**executeQuery**();**

ArrayList**<**Dept**>** deptList **=** **new** ArrayList**<**Dept**>();**

**while** **(**rs**.**next**())** **{**

deptList**.**add**(this.**getDept**(**rs**.**getInt**(**1**)));**

**}**

**return** deptList**;**

**}** **catch** **(**SQLException e**)** **{**

**return** **null;**

**}**

**}**

public boolean deleteAllDepts**()** **{**

String selectAll **=** "select " **+** Dept**.**ID **+** " from " **+** Dept**.**TABLE\_NAME**;**

**try** **{**

**if** **(new** StaffDAO**().**deleteAllStaffs**())** **{**

PreparedStatement preparedStatement **=** connection**.**prepareStatement**(**selectAll**);**

ResultSet rs **=** preparedStatement**.**executeQuery**();**

**while** **(**rs**.**next**())** **{**

**this.**deleteDept**(**rs**.**getInt**(**Dept**.**ID**));**

**}**

**return** **true;**

**}** **else** **{**

**return** **false;**

**}**

**}** **catch** **(**SQLException e**)** **{}**

**return** **false;**

**}**

**}**

### PositionDAO.java

package dao**;**

**import** java**.**sql**.**ResultSet**;**

**import** java**.**sql**.**SQLException**;**

**import** java**.**util**.**ArrayList**;**

**import** bean**.**Position**;**

/\*\*

\*

\*

\*\*/

public class PositionDAO **extends** BaseDAO **{**

public PositionDAO**()** **{**

**super();**

**}**

public Position getPosition**(**int id**)** **{**

String query **=** "select \* from " **+** Position**.**TABLE\_NAME

**+** " where " **+** Position**.**ID **+** "=?"**;**

**try** **{**

preparedStatement **=** connection**.**prepareStatement**(**query**);**

preparedStatement**.**setInt**(**1**,** id**);**

ResultSet rs **=** preparedStatement**.**executeQuery**();**

Position pos **=** **new** Position**();**

**if** **(**rs**.**next**())** **{**

pos**.**setPositionId**(**rs**.**getInt**(**Position**.**ID**));**

pos**.**setPositionName**(**rs**.**getString**(**Position**.**NAME**));**

**}**

preparedStatement**.**close**();**

**return** pos**;**

**}** **catch** **(**SQLException e**)** **{**

e**.**printStackTrace**();**

**return** **null;**

**}**

**}**

public ArrayList**<**Position**>** getAllPositions**()** **{**

System**.**out**.**print**(**">> PositionDAO: "**);** // FIXME

String query **=** "select " **+** Position**.**ID **+** " from " **+** Position**.**TABLE\_NAME**;**

**try** **{**

preparedStatement **=** connection**.**prepareStatement**(**query**);**

System**.**out**.**print**(**"> Chạy truy vấn: " **+** preparedStatement**.**toString**());**

ResultSet rs **=** preparedStatement**.**executeQuery**();**

ArrayList**<**Position**>** posList **=** **new** ArrayList**<**Position**>();**

**while** **(**rs**.**next**())** **{**

posList**.**add**(this.**getPosition**(**rs**.**getInt**(**1**)));**

**}**

preparedStatement**.**close**();**

System**.**out**.**print**(**"> Đã lấy danh sách chức vụ "**);** // FIXME

**return** posList**;**

**}** **catch** **(**SQLException e**)** **{**

e**.**printStackTrace**();**

System**.**out**.**print**(**"> Lỗi: " **+** e **+** " > NULL "**);** // FIXME

**return** **null;**

**}**

**}**

**}**

## Kết quả chạy thử

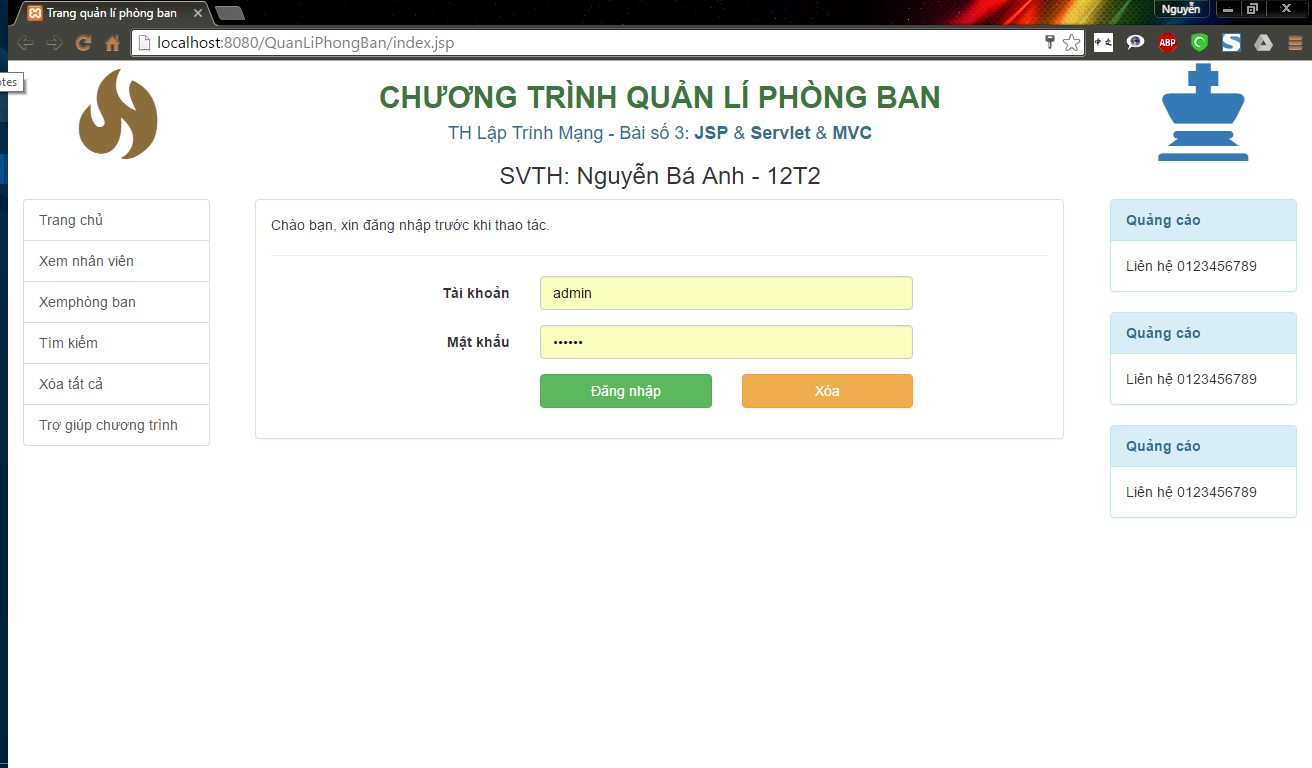


Figure xii: Giao diện trang chủ

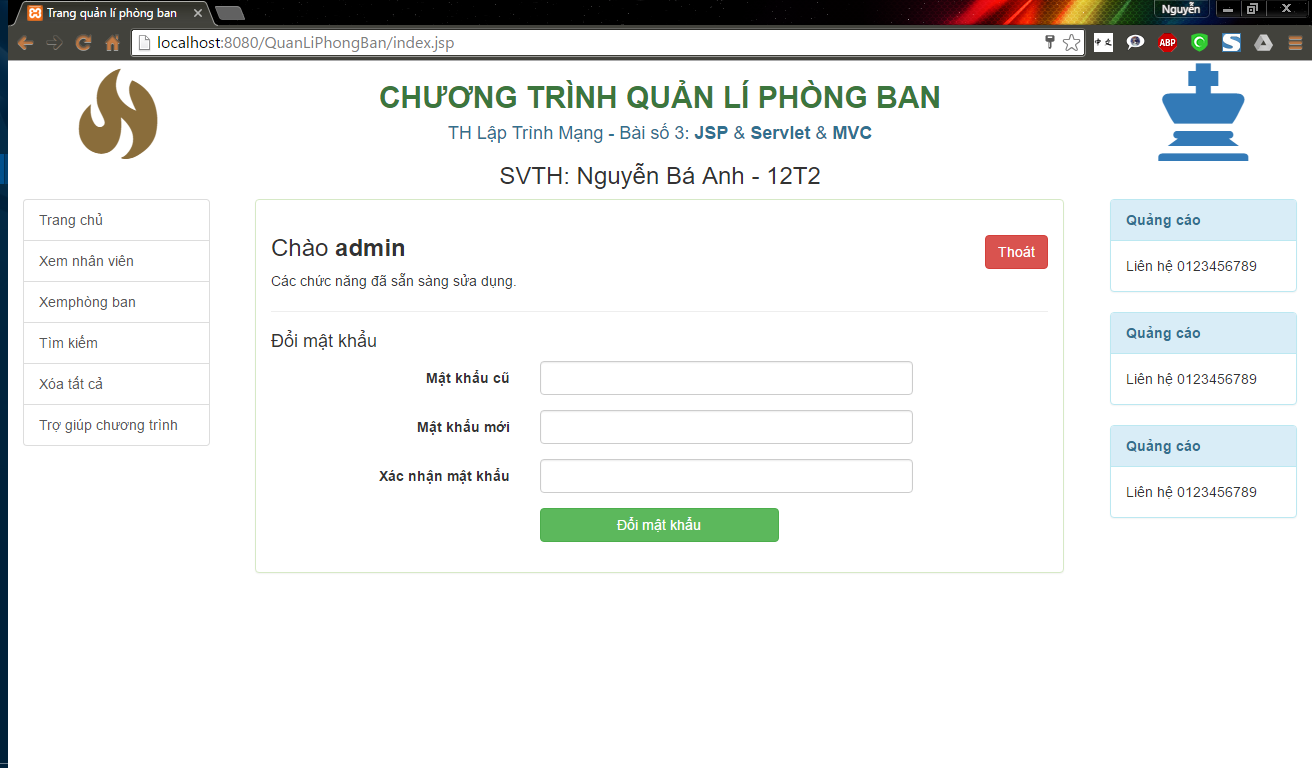


Figure xiii: Giao diện trang chủ sau khi đăng nhập

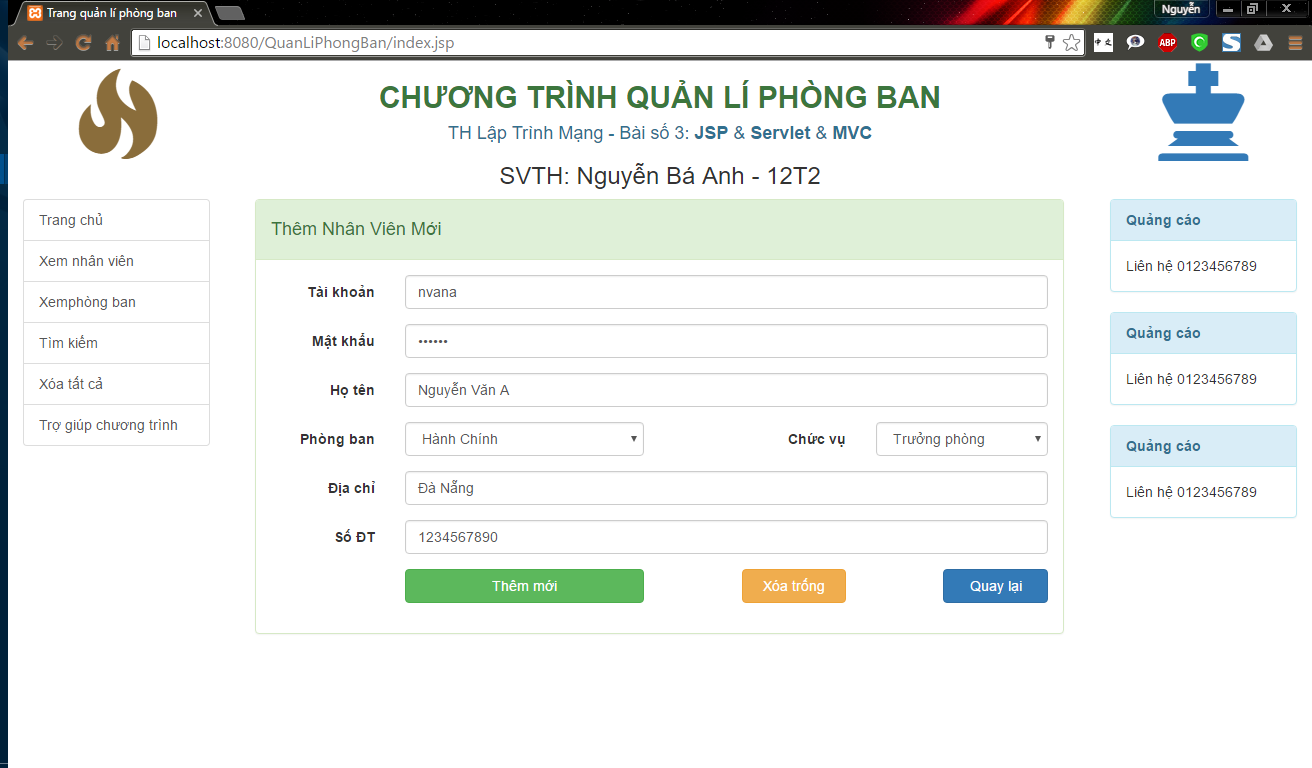


Figure xiv: Thêm nhân viên mới

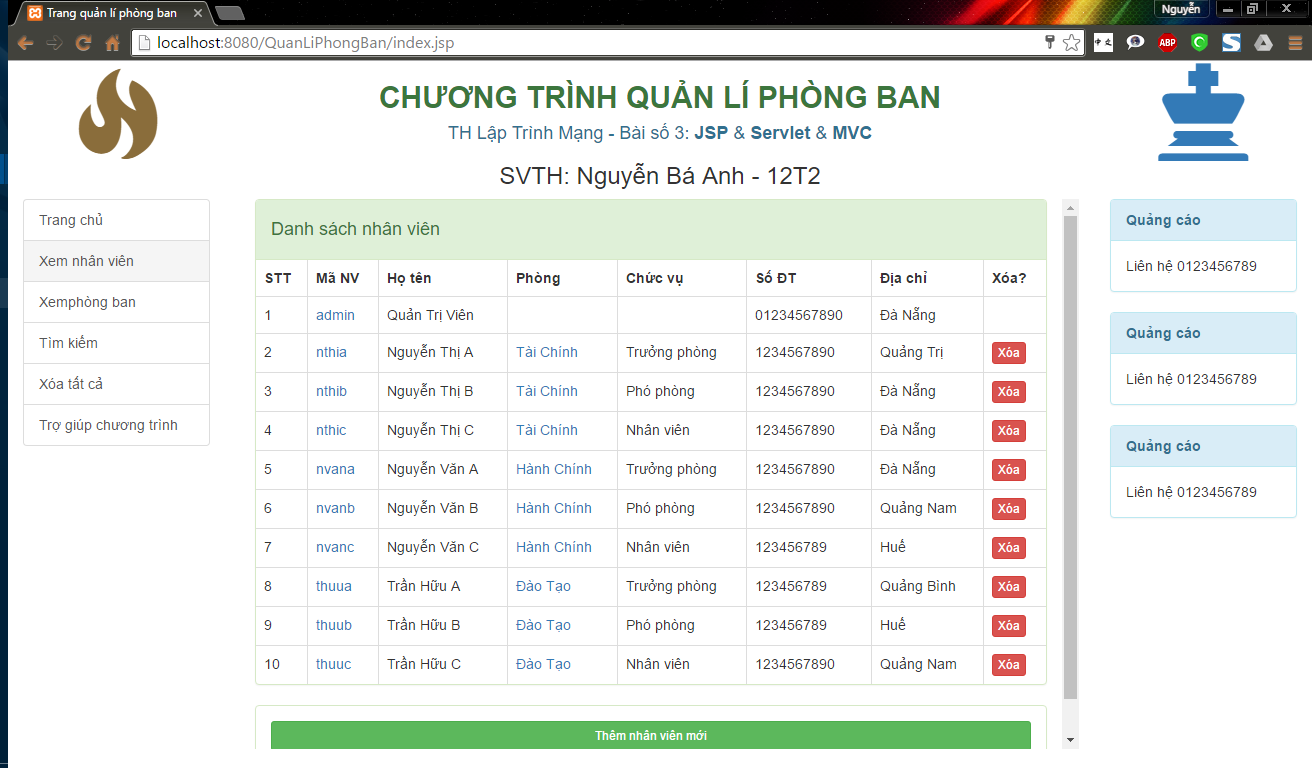


Figure xv: Xem danh sách nhân viên

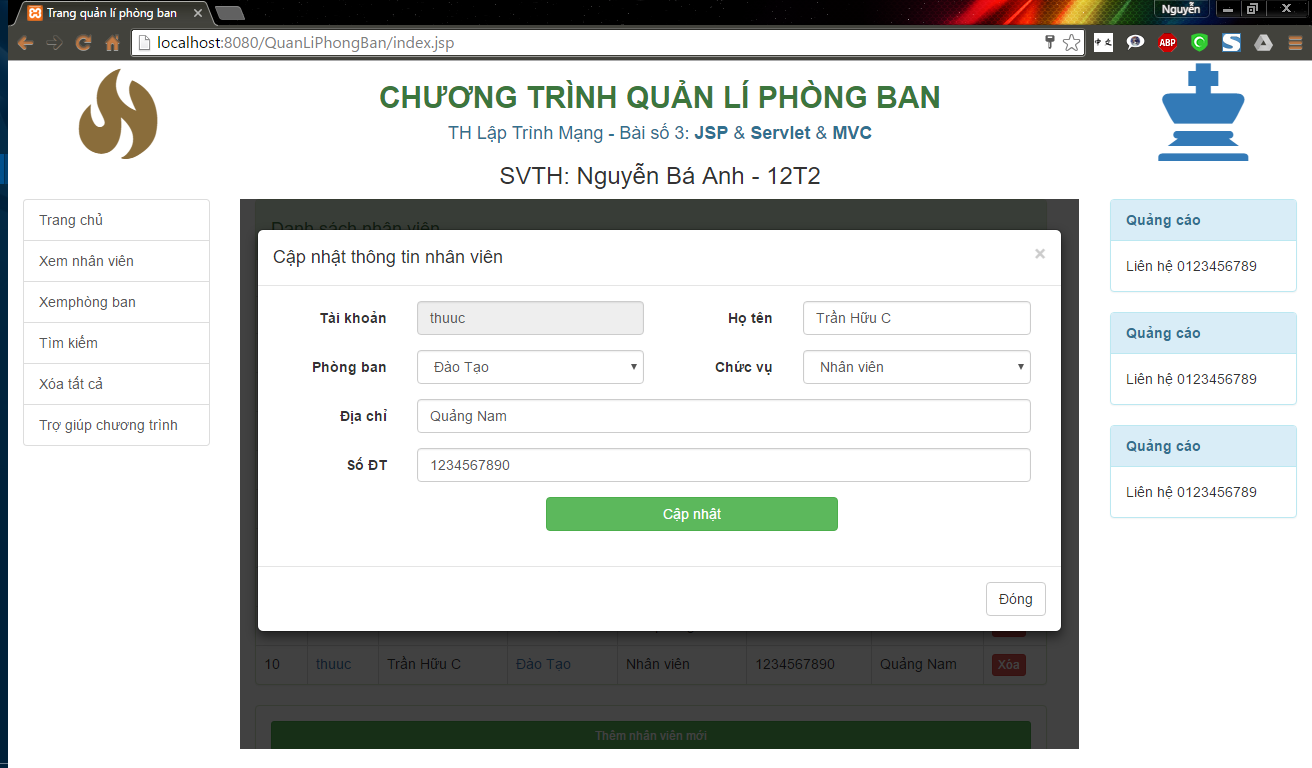


Figure xvi: Giao diện Cập nhật nhân viên

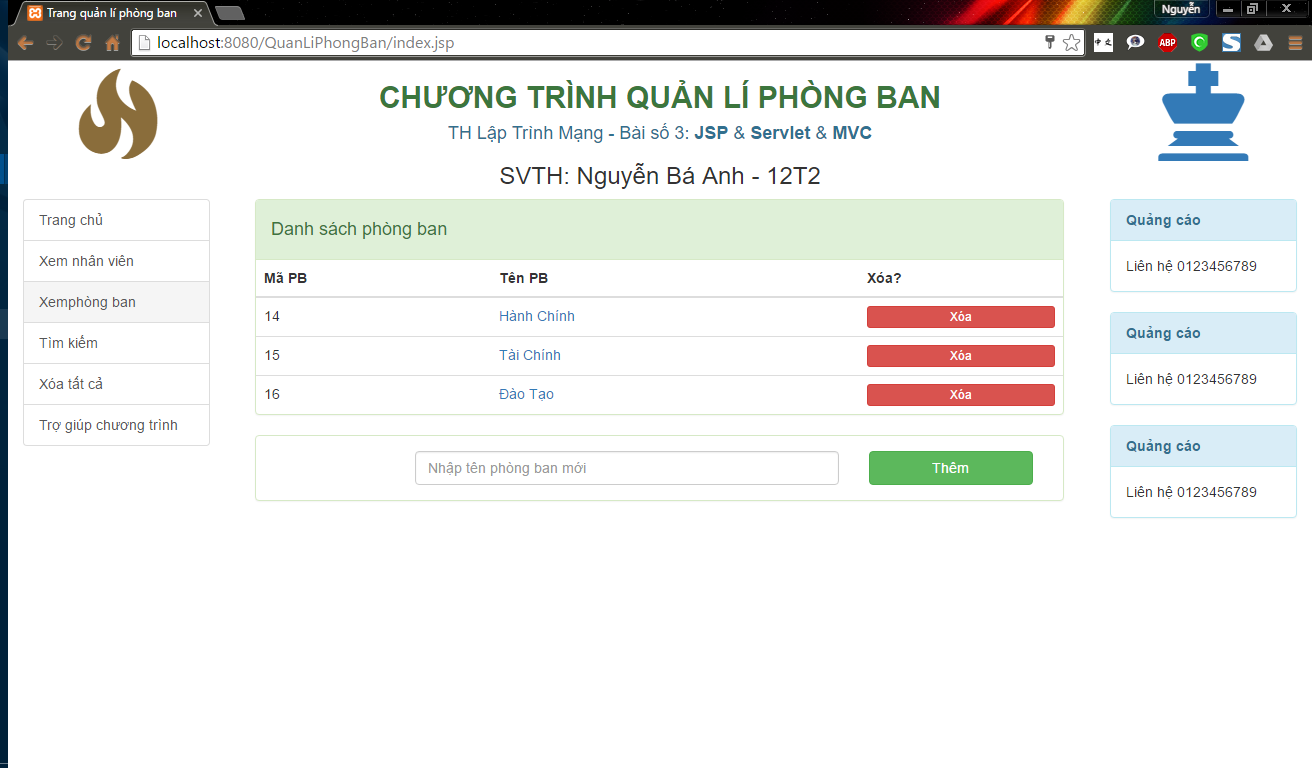


Figure xvii: Xem danh sách phòng ban

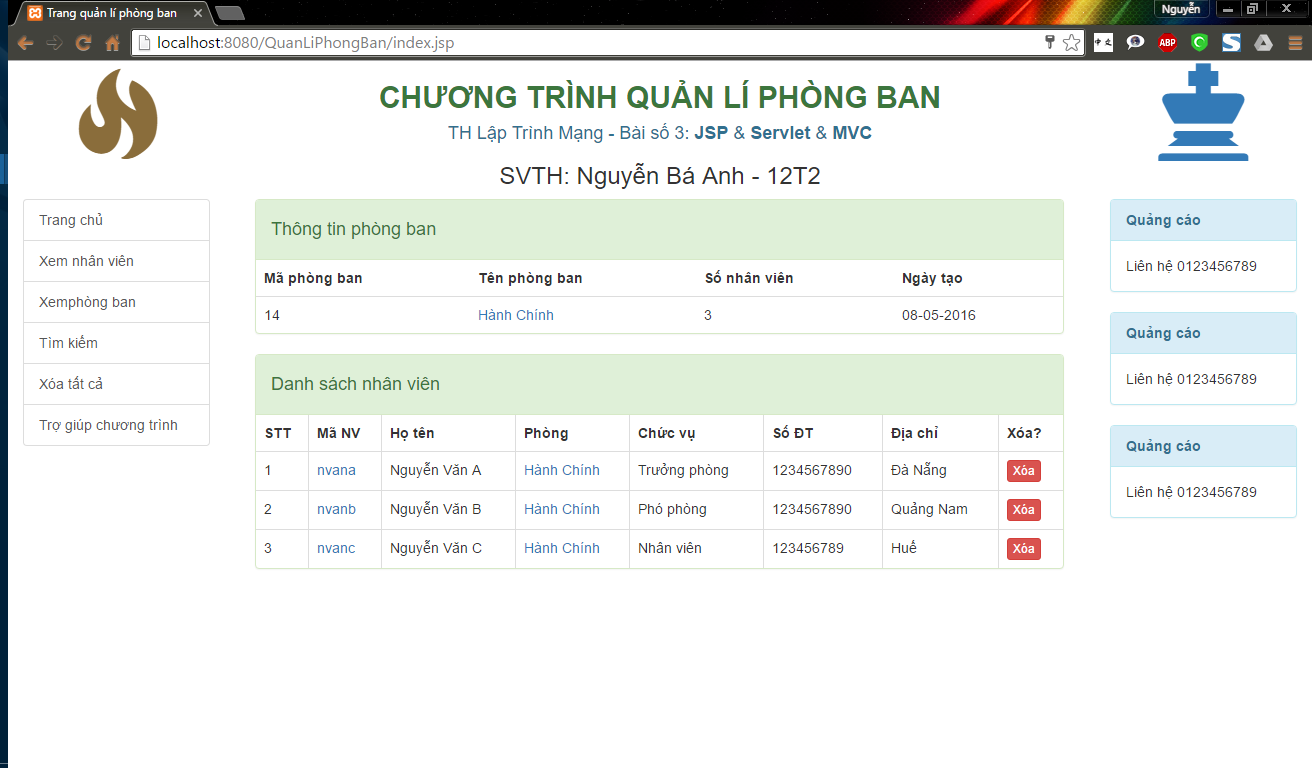


Figure xviii: Xem chi tiết phòng ban

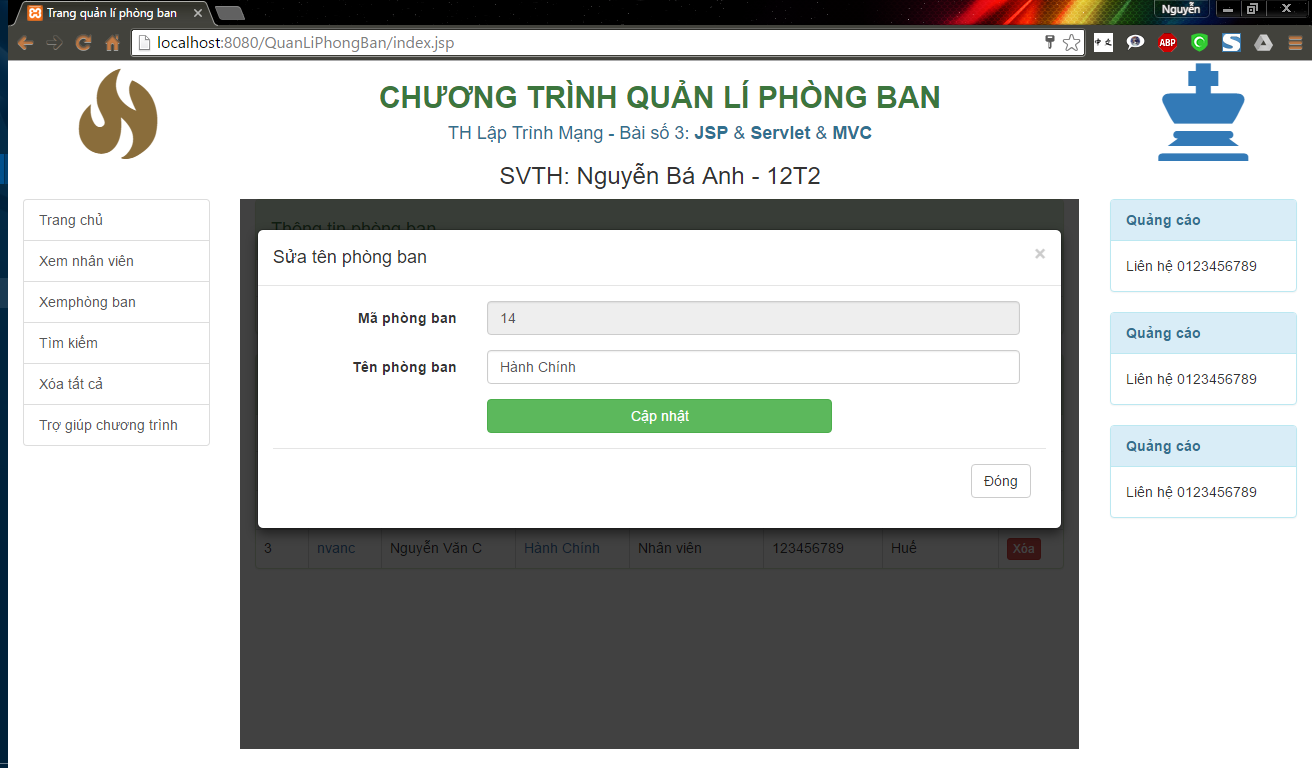


Figure xix: Sửa tên phòng ban

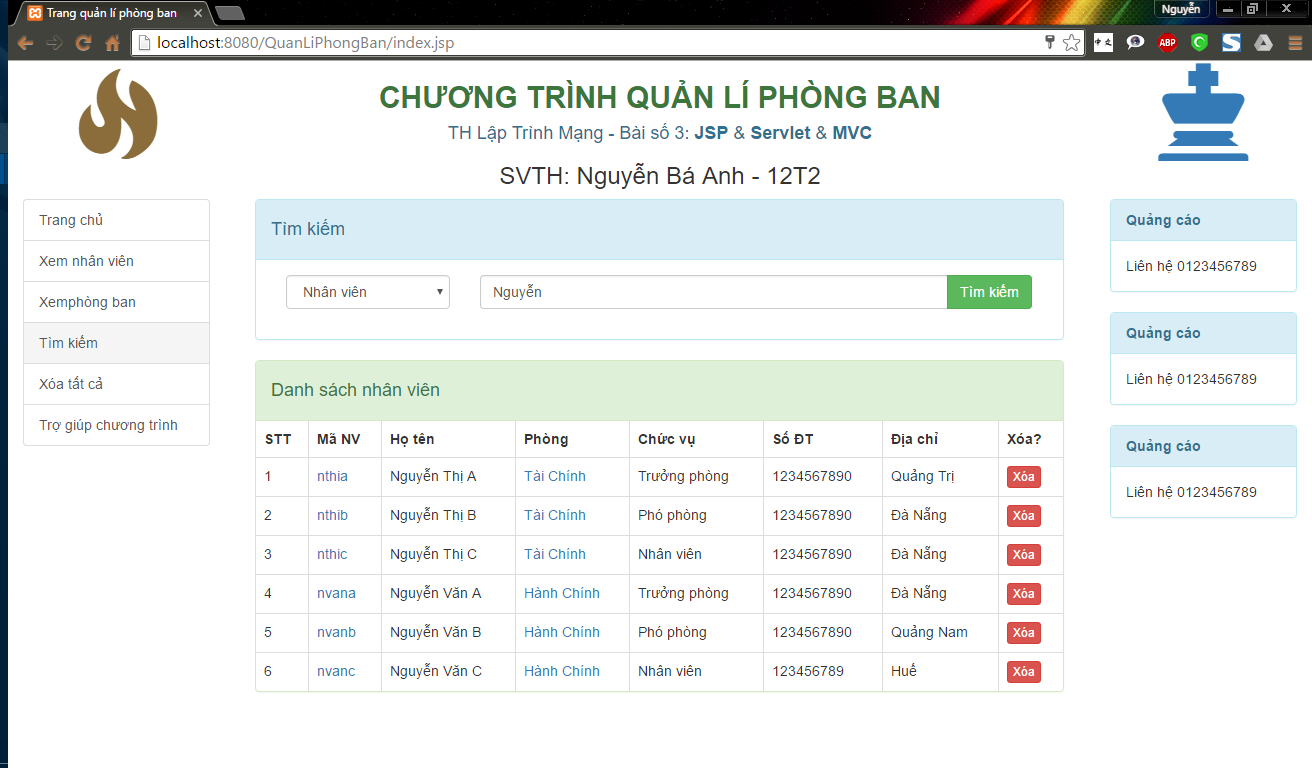


Figure xx: Tìm kiếm nhân viên

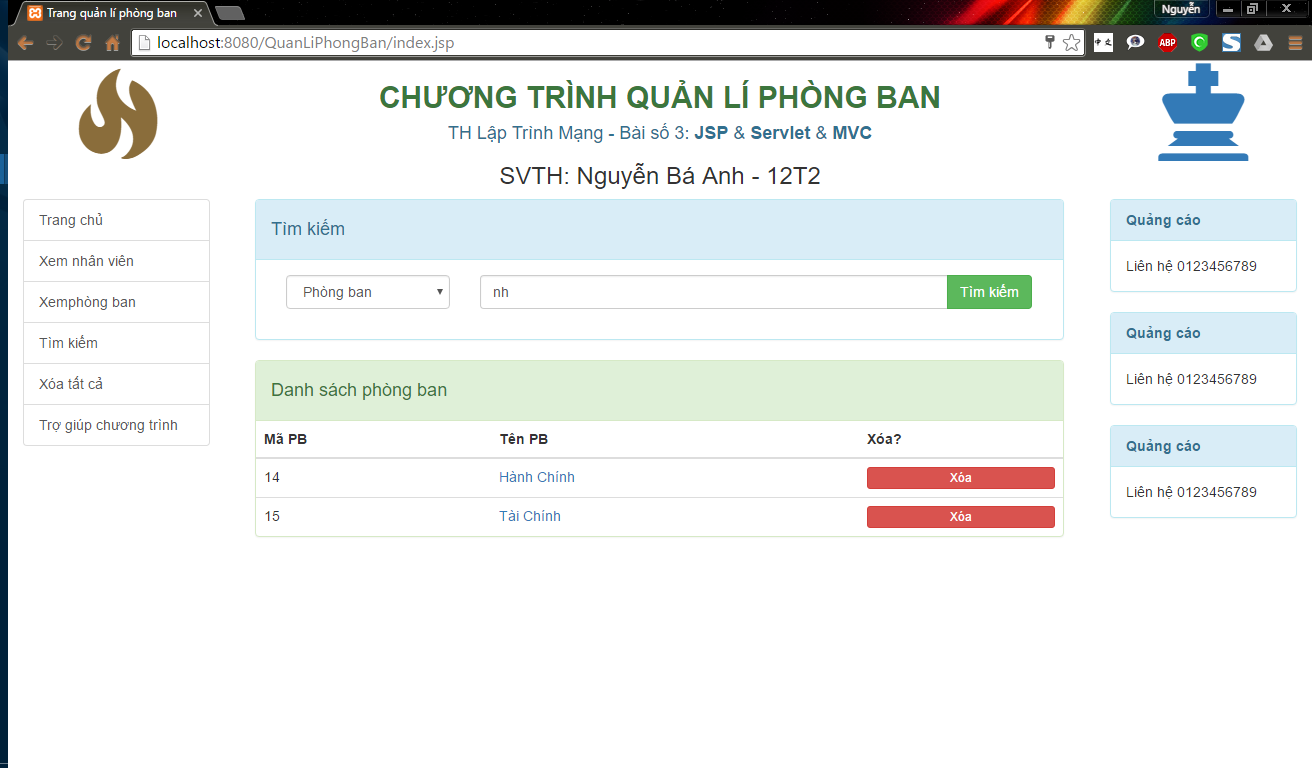


Figure xxi: Tìm kiếm phòng ban

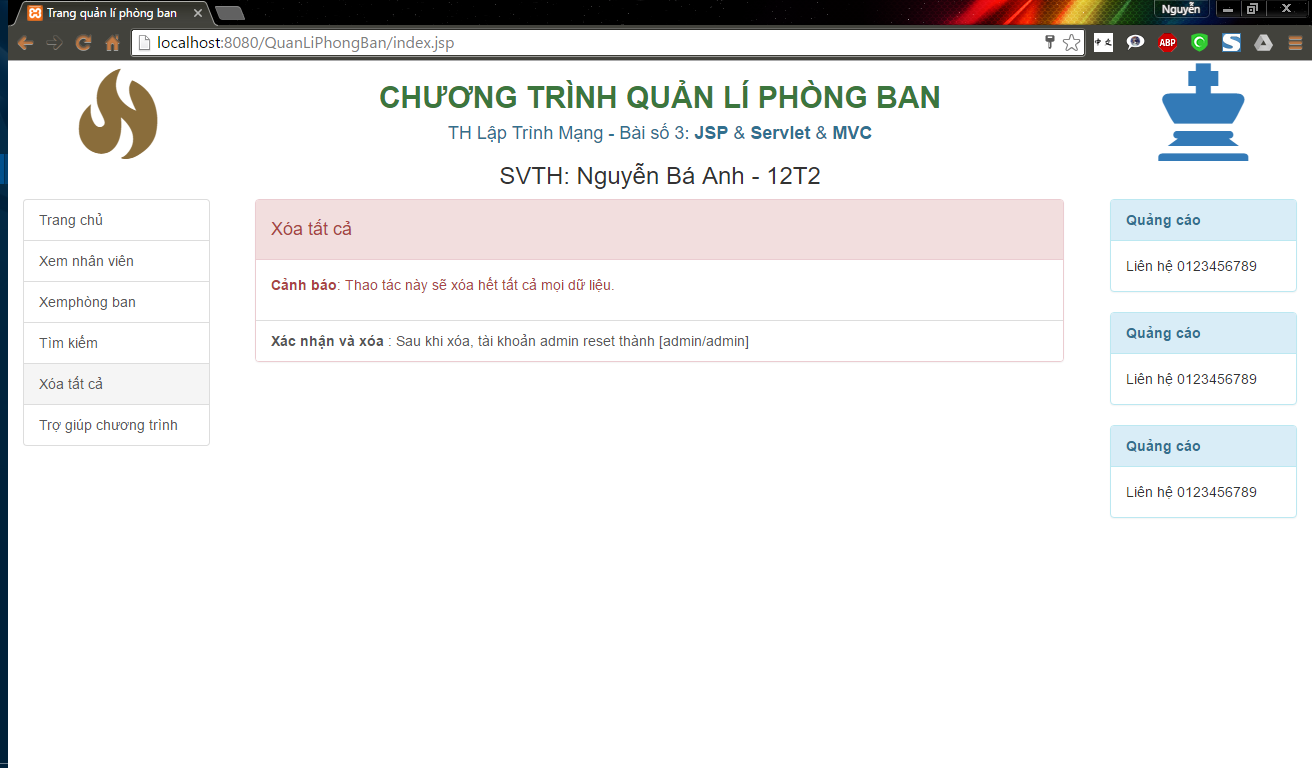


Figure xxii: Xóa tất cả

# Kết luận

Về cơ bản báo cáo thỏa mãn các yêu cầu của môn học và bản thân em cũng rút ra được nhiều kinh nghiệm khi lập trình mạng.

Hi vọng đây sẽ là bước đầu để em có thể tự nghiên cứu được những kiến thức mới hơn có liên quan. Cảm ơn thầy đã dành thời gian duyệt bài cho em.