# LỜI NÓI 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 Phong

**MỤC LỤC**

[LỜI NÓI ĐẦU 2](#_Toc481697374)

[MỤC LỤC 3](#_Toc481697375)

[I. BÀI THỰC HÀNH SỐ 1 HỌC PHẦN LẬP TRÌNH MẠNG 5](#_Toc481697376)

[1. Xây dựng chương trình hội thoại Client/Server hoạt động theo giao thức TCP/IP 5](#_Toc481697377)

[a. Thuật toán: 5](#_Toc481697378)

[b. Mã nguồn: 5](#_Toc481697379)

[c. Demo: 8](#_Toc481697380)

[2. Xây dựng chương trình hội thoại Client/Server hoạt động theo giao thức TCP/IP 8](#_Toc481697381)

[a. Thuật toán: 8](#_Toc481697382)

[b. Mã nguồn: 8](#_Toc481697383)

[c. Demo: 13](#_Toc481697384)

[3. Xây dựng chương trình hội thoại chat room Client/Server hoạt động theo giao thức TCP/IP 14](#_Toc481697385)

[a. Thuật toán: 14](#_Toc481697386)

[b. Mã nguồn: 14](#_Toc481697387)

[c. Demo: 21](#_Toc481697388)

[II. BÀI THỰC HÀNH SỐ 2 HỌC PHẦN LẬP TRÌNH MẠNG 21](#_Toc481697389)

[1. Xây dựng chương trình hội thoại Client/Server hoạt động theo giao thức UDP 21](#_Toc481697390)

[a. Thuật toán: 21](#_Toc481697391)

[b. Mã nguồn: 22](#_Toc481697392)

[c. Demo: 27](#_Toc481697393)

[2. Xây dựng chương trình hội thoại Client/Server hoạt động theo giao thức UDP 27](#_Toc481697394)

[a. Thuật toán: 28](#_Toc481697395)

[b. Mã nguồn: 28](#_Toc481697396)

[c. Demo: 33](#_Toc481697397)

[3. Xây dựng chương trình hội thoại chat room Client/Server hoạt động theo giao thức UDP 33](#_Toc481697398)

[a. Thuật toán: 33](#_Toc481697399)

[b. Mã nguồn: 34](#_Toc481697400)

[c. Demo: 41](#_Toc481697401)

[III. BÀI THỰC HÀNH SỐ 3 HỌC PHẦN LẬP TRÌNH MẠNG 42](#_Toc481697402)

[a. Thuật toán: 43](#_Toc481697403)

[b. Mã nguồn: 44](#_Toc481697404)

[c. Demo: 67](#_Toc481697405)

1. **BÀI THỰC HÀNH SỐ 1 HỌC PHẦN LẬP TRÌNH MẠNG**
2. Xây dựng chương trình hội thoại Client/Server hoạt động theo giao thức 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

1. **Thuật toán:**
   * Sử dụng giao thức TCP/IP để gởi 1 chuỗi từ Client tới Server:
   * Sử dụng socket ở Client và ServerSocket ở Server để thực hiện kết nối. Sau đó sử dụng DataInputStream và DataOutputstrem để gửi và nhận chuỗi.
2. **Mã nguồn:**

**File server.java:**

package bai1;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.net.ServerSocket;

import java.net.Socket;

import java.net.UnknownHostException;

import java.sql.SQLException;

public class Server extends Thread {

public static void main(String[] args) throws UnknownHostException, Exception {

ServerSocket serverSocket = new ServerSocket(888);

System.*out*.print("Server is start............");

Socket socket = serverSocket.accept();

DataInputStream inputStream = new DataInputStream(socket.getInputStream());

DataOutputStream outputStream = new DataOutputStream(socket.getOutputStream());

while (true) {

String snhan = inputStream.readUTF();

String hoa = *doiChuHoa*(snhan);

String thuong = *doiChuThuong*(snhan);

int dem = *demSoTu*(snhan);

outputStream.writeUTF("Chu hoa: "+hoa+" Chu thuong: "+thuong+" So ky tu: "+dem);

}

}

public static String doiChuHoa(String s) {

return s.toUpperCase();

}

public static String doiChuThuong(String s) {

return s.toLowerCase();

}

public static int demSoTu(String s1) {

int dem = 0;

String s = ' ' + s1;

for (int i = 0; i < s.length() - 1; i++) {

if (s.charAt(i) == ' ' && s.charAt(i + 1) != ' ')

dem++;

}

return dem;

}

}

**File client.java**

package bai1;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.net.ServerSocket;

import java.net.Socket;

import java.net.UnknownHostException;

import java.sql.SQLException;

public class Server extends Thread {

public static void main(String[] args) throws UnknownHostException, Exception {

ServerSocket serverSocket = new ServerSocket(888);

System.*out*.print("Server is start............");

Socket socket = serverSocket.accept();

DataInputStream inputStream = new DataInputStream(socket.getInputStream());

DataOutputStream outputStream = new DataOutputStream(socket.getOutputStream());

while (true) {

String snhan = inputStream.readUTF();

String hoa = *doiChuHoa*(snhan);

String thuong = *doiChuThuong*(snhan);

int dem = *demSoTu*(snhan);

outputStream.writeUTF("Chu hoa: "+hoa+" Chu thuong: "+thuong+" So ky tu: "+dem);

}

}

public static String doiChuHoa(String s) {

return s.toUpperCase();

}

public static String doiChuThuong(String s) {

return s.toLowerCase();

}

public static int demSoTu(String s1) {

int dem = 0;

String s = ' ' + s1;

for (int i = 0; i < s.length() - 1; i++) {

if (s.charAt(i) == ' ' && s.charAt(i + 1) != ' ')

dem++;

}

return dem;

}

}

1. **Demo:**

****

1. Xây dựng chương trình hội thoại Client/Server hoạt động theo giao thức 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.

1. **Thuật toán:**

* Gửi dữ liệu từ client lên server thông qua socket TCP/IP.
* Nhập đúng định dạng các phép toán và thực hiện tính toán đơn giản.

1. **Mã nguồn:**

**File serverb2.java**

package bai1;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.net.ServerSocket;

import java.net.Socket;

import java.sql.SQLException;

public class serverb2 extends Thread {

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

ServerSocket serverSocket = new ServerSocket(1234);

System.out.print("Server is start............");

Socket socket = serverSocket.accept();

DataInputStream inputStream = new DataInputStream(socket.getInputStream());

DataOutputStream outputStream = new DataOutputStream(socket.getOutputStream());

while (true) {

String snhan = inputStream.readUTF();

String stra = xulychuoi(snhan);

outputStream.writeUTF(stra);

}

}

public static String xulychuoi(String s) {

float ketqua = 0;

int i = 0;

float giatritam = 0;

while (i < s.length()) {

while ((s.charAt(i) != '+') && ((s.charAt(i) != '-')) && (s.charAt(i) != '\*') && (s.charAt(i) != '/')) {

ketqua = ketqua \* 10 + Integer.parseInt(String.valueOf(s.charAt(i)));

i++;

}

if (s.charAt(i) == '+') {

i++;

while ((i < s.length() && (s.charAt(i) != '+') && ((s.charAt(i) != '-')) && (s.charAt(i) != '\*')

&& (s.charAt(i) != '/'))) {

giatritam = giatritam \* 10 + Integer.parseInt(String.valueOf(s.charAt(i)));

i++;

}

ketqua += giatritam;

giatritam = 0;

}

if ((i < s.length()) && (s.charAt(i) == '-')) {

i++;

while ((i < s.length() && (s.charAt(i) != '+') && ((s.charAt(i) != '-')) && (s.charAt(i) != '\*')

&& (s.charAt(i) != '/'))) {

giatritam = giatritam \* 10 + Integer.parseInt(String.valueOf(s.charAt(i)));

i++;

}

ketqua -= giatritam;

giatritam = 0;

}

if ((i < s.length()) && (s.charAt(i) == '\*')) {

i++;

while ((i < s.length() && (s.charAt(i) != '+') && ((s.charAt(i) != '-')) && (s.charAt(i) != '\*')

&& (s.charAt(i) != '/'))) {

giatritam = giatritam \* 10 + Integer.parseInt(String.valueOf(s.charAt(i)));

i++;

}

ketqua \*= giatritam;

giatritam = 0;

}

if ((i < s.length()) && (s.charAt(i) == '/')) {

i++;

while ((i < s.length() && (s.charAt(i) != '+') && ((s.charAt(i) != '-')) && (s.charAt(i) != '\*')

&& (s.charAt(i) != '/'))) {

giatritam = giatritam \* 10 + Integer.parseInt(String.valueOf(s.charAt(i)));

i++;

}

ketqua /= giatritam;

giatritam = 0;

}

}

System.out.print(ketqua);

return String.valueOf(ketqua);

}

}

File clientb2.java:

package bai1;

import java.awt.GridLayout;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.net.Socket;

import java.net.UnknownHostException;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.JFrame;

public class clientb2 implements ActionListener {

public static Panel pn1, pn2, pn3, pn;

public static TextField tfnhap = new TextField(5);

public static TextField tfkq = new TextField();

public static Label lbnhap = new Label();

public static Label lbkq = new Label();

public static Button bt1, bt2;

public static String s = "";

Socket socket;

DataOutputStream outputStream;

DataInputStream inputStream;

public clientb2() throws UnknownHostException, IOException {

GUI();

}

void GUI() throws UnknownHostException, IOException {

JFrame fr = new JFrame("Client");

fr.setSize(400, 200);

fr.setLayout(new GridLayout());

lbnhap.setText("Nhap vao chuoi can tinh toan");

lbkq.setText("Ket qua");

pn = new Panel(new GridLayout(5, 1));

pn1 = new Panel(new GridBagLayout());

bt1 = new Button("send");

bt2 = new Button("Exit");

pn.add(lbnhap);

pn.add(tfnhap);

pn.add(lbkq);

pn.add(tfkq);

pn1.add(bt1);

pn1.add(bt2);

pn.add(pn1);

fr.add(pn);

bt1.addActionListener(this);

bt2.addActionListener(this);

fr.show();

socket = new Socket("127.0.0.1", 1234);

outputStream = new DataOutputStream(socket.getOutputStream());

inputStream = new DataInputStream(socket.getInputStream());

}

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

clientb2 t = new clientb2();

}

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

if (e.getSource() == bt1) {

try {

String s = tfnhap.getText();

outputStream.writeUTF(s);

String nhan = inputStream.readUTF();

tfkq.setText(nhan);

} catch (UnknownHostException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (IOException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

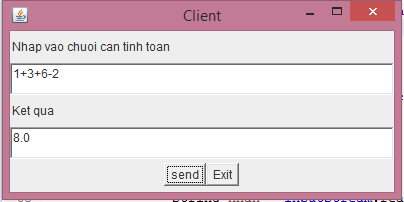
if (e.getSource() == bt2)

System.exit(0);

}

}

1. **Demo:**

****

1. Xây dựng chương trình hội thoại chat room Client/Server hoạt động theo giao thức 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.

1. **Thuật toán:**

* Tạo một server chung.
* Sau đó sử dụng socket tạo ra các client kết nối với nhau qua server chung.

1. **Mã nguồn:**

**Server:**

package server;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.net.Socket;

import java.util.StringTokenizer;

public class ServerThread implements Runnable {

Socket socket;

DataOutputStream dos;

DataInputStream din;

ServerUI main;

boolean isRunning = false;

public ServerThread(Socket socket, ServerUI main) throws IOException {

this.socket = socket;

this.main = main;

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

din = new DataInputStream(socket.getInputStream());

isRunning = true;

}

@Override

public void run() {

while (isRunning) {

try {

String mess = din.readUTF();

StringTokenizer st = new StringTokenizer(mess);

String CMD = st.nextToken();

String from = st.nextToken();

switch (CMD) {

case "LOG\_IN":

if (!main.clients.containsKey(from)) {

main.clients.put(from, this);

main.updateUI(from + " joined chat room!");

sendMessage("SUCCESS");

} else {

sendMessage("ERROR");

}

break;

case "MESSAGE":

String msg = "";

while (st.hasMoreTokens()) {

msg = msg + " " + st.nextToken();

}

main.sendAll(msg, from);

break;

case "LOG\_OUT":

main.clients.remove(from);

updateUI(from + " left chat room!");

isRunning = false;

socket.close();

break;

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

public void updateUI(String mess) {

main.updateUI(mess);

}

public void sendMessage(String mess) throws IOException {

dos.writeUTF(mess);

}

}

package server;

import java.awt.EventQueue;

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 java.util.ArrayList;

import java.util.Enumeration;

import java.util.Hashtable;

import javax.swing.JFrame;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

import javax.swing.JTextField;

import javax.swing.JButton;

import javax.swing.LayoutStyle.ComponentPlacement;

import client.ClientThread;

import javax.swing.JTextArea;

import javax.swing.JScrollPane;

public class ServerUI implements ActionListener {

private JFrame frame;

private JTextField txtMess;

DataInputStream din;

DataOutputStream dos;

ServerSocket serverSocket;

JTextArea ta;

JButton btnSend;

public Hashtable<String, ServerThread> clients = new Hashtable<>();

/\*\*

\* Launch the application.

\* @throws IOException

\*/

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

// ServerUI window = new ServerUI();

// window.frame.setVisible(true);

// window.connect();

//

// }

/\*\*

\* Create the application.

\* @throws IOException

\*/

public ServerUI() throws IOException {

initialize();

connect();

}

private void connect() throws IOException {

serverSocket=new ServerSocket(1000);

ta.append("Server have already! "+"\n");

while(true){

Socket socket=serverSocket.accept();

ServerThread thread=new ServerThread(socket,this);

new Thread(thread).start();

}

}

/\*\*

\* Initialize the contents of the frame.

\*/

private void initialize() {

frame = new JFrame();

frame.setBounds(100, 100, 450, 300);

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

JScrollPane scrollPane = new JScrollPane();

txtMess = new JTextField();

txtMess.setColumns(10);

btnSend = new JButton("Send");

btnSend.addActionListener(this);

GroupLayout groupLayout = new GroupLayout(frame.getContentPane());

groupLayout.setHorizontalGroup(

groupLayout.createParallelGroup(Alignment.LEADING)

.addGroup(Alignment.TRAILING, groupLayout.createSequentialGroup()

.addContainerGap(29, Short.MAX\_VALUE)

.addGroup(groupLayout.createParallelGroup(Alignment.LEADING, false)

.addGroup(groupLayout.createSequentialGroup()

.addComponent(txtMess, GroupLayout.PREFERRED\_SIZE, 292, GroupLayout.PREFERRED\_SIZE)

.addGap(18)

.addComponent(btnSend, GroupLayout.DEFAULT\_SIZE, GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addComponent(scrollPane, Alignment.TRAILING, GroupLayout.PREFERRED\_SIZE, 384, GroupLayout.PREFERRED\_SIZE))

.addContainerGap())

);

groupLayout.setVerticalGroup(

groupLayout.createParallelGroup(Alignment.LEADING)

.addGroup(Alignment.TRAILING, groupLayout.createSequentialGroup()

.addGap(27)

.addComponent(scrollPane, GroupLayout.PREFERRED\_SIZE, 152, GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(ComponentPlacement.RELATED, 20, Short.MAX\_VALUE)

.addGroup(groupLayout.createParallelGroup(Alignment.BASELINE)

.addComponent(txtMess, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(btnSend))

.addContainerGap())

);

ta = new JTextArea();

scrollPane.setViewportView(ta);

frame.getContentPane().setLayout(groupLayout);

}

public void updateUI(String mess){

ta.append(mess+"\n");

}

public void sendAll(String mess, String from) throws IOException{

Enumeration<String> e = clients.keys();

while(e.hasMoreElements()){

String name = e.nextElement();

if(!name.equals(from)){

ServerThread thread = clients.get(name);

thread.sendMessage("MESSAGE "+from+" "+mess);

}

}

}

@Override

public void actionPerformed(ActionEvent e) {

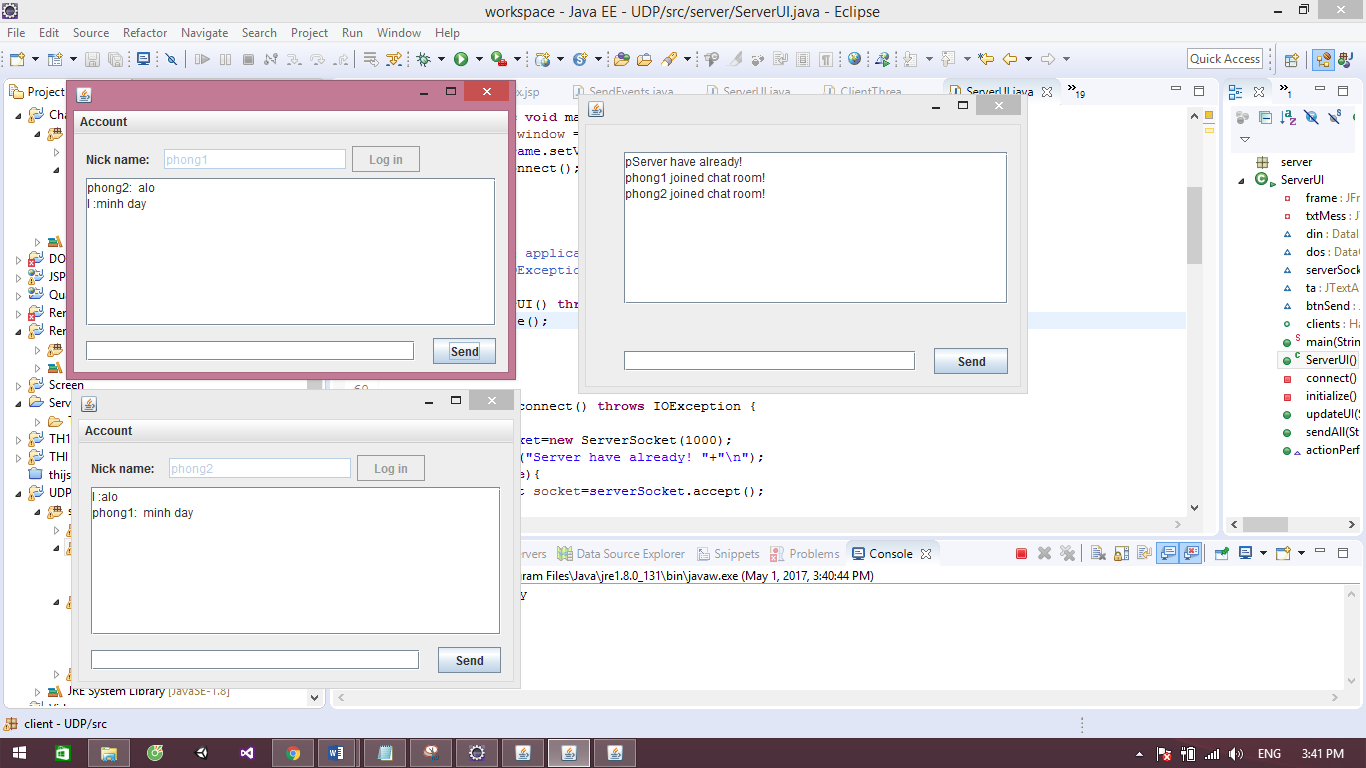
if(e.getSource()==btnSend){

}

}

}

1. **Demo:**

****

1. **BÀI THỰC HÀNH SỐ 2 HỌC PHẦN LẬP TRÌNH MẠNG**
2. Xây dựng chương trình hội thoại Client/Server hoạt động theo giao thức UDP

- 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

1. **Thuật toán:**
   * Sử dụng giao thức UDP để giao tiếp giữa Client và Server. Để làm được điều này thì ta sử dụng DatagramSocket ở cả Client và Server giúp giao tiếp với nhau thông qua các cổng ta đã định.
   * Để gửi và nhận thông tin trong giao thức UDP ta dùng Datagrampacket.
2. **Mã nguồn:**

**File server:**

package Bài1;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.net.ServerSocket;

import java.net.Socket;

import java.net.UnknownHostException;

public class Server {

public static void main(String[] args) throws UnknownHostException, Exception {

byte senddata[];

byte receivedata[]=new byte[1024];

DatagramSocket server=new DatagramSocket(8892);

DatagramPacket receive=new DatagramPacket(receivedata, receivedata.length);

server.receive(receive);

InetAddress inet=receive.getAddress();

int port=receive.getPort();

String nhan=new String(receive.getData());

senddata=new byte[nhan.length()];

String chuoihoa=doiChuHoa(nhan);

senddata=chuoihoa.getBytes();

DatagramPacket send=new DatagramPacket(senddata, senddata.length, inet, port);

server.send(send);

String chuoithuong=doiChuThuong(nhan);

senddata=chuoithuong.getBytes();

send=new DatagramPacket(senddata, senddata.length, inet, port);

server.send(send);

int dem=demSoTu(nhan);

String soTu="So tu la:"+dem;

System.out.println(dem);

senddata=soTu.getBytes();

send=new DatagramPacket(senddata, senddata.length, inet, port);

server.send(send);

}

public static String doiChuHoa(String s) {

return s.toUpperCase();

}

public static String doiChuThuong(String s) {

return s.toLowerCase();

}

public static int demSoTu(String s1) {

int dem = 0;

String s = ' ' + s1;

for (int i = 0; i < s.length() - 1; i++) {

if (s.charAt(i) == ' ' && s.charAt(i + 1) != ' ')

dem++;

}

return dem;

}

}

**File client.java:**

package Bài1;

import java.awt.Button;

import java.awt.GridBagLayout;

import java.awt.GridLayout;

import java.awt.Label;

import java.awt.Panel;

import java.awt.TextField;

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 java.net.UnknownHostException;

import javax.swing.JFrame;

public class Client implements ActionListener {

public static Panel *pn1*, *pn2*, *pn3*, *pn*;

public static TextField *tfnhap* = new TextField(5);

public static TextField *tfkq* = new TextField();

public static Label *lbnhap* = new Label();

public static Label *lbkq* = new Label();

public static Button *bt1*, *bt2*;

public static String *s* = "";

DatagramSocket moi;

InetAddress inet;

byte senddata[];

byte receivedata[];

public Client() throws UnknownHostException, IOException {

GUI();

}

void GUI() throws UnknownHostException, IOException {

JFrame fr = new JFrame("Client");

fr.setSize(400, 200);

fr.setLayout(new GridLayout());

*lbnhap*.setText("Nập vào chuỗi cần xử lý");

*lbkq*.setText("Kết quả");

*pn* = new Panel(new GridLayout(5, 1));

*pn1* = new Panel(new GridBagLayout());

*bt1* = new Button("Chuyển");

*bt2* = new Button("Exit");

*pn*.add(*lbnhap*);

*pn*.add(*tfnhap*);

*pn*.add(*lbkq*);

*pn*.add(*tfkq*);

*pn1*.add(*bt1*);

*pn1*.add(*bt2*);

*pn*.add(*pn1*);

fr.add(*pn*);

*bt1*.addActionListener(this);

*bt2*.addActionListener(this);

fr.~~show~~();

inet = InetAddress.*getByName*("localhost");

moi = new DatagramSocket();

}

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

Client t = new Client();

}

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

if (e.getSource() == *bt1*) {

try {

String kq="";

String s = *tfnhap*.getText();

senddata = new byte[s.length()];

receivedata = new byte[s.length()];

senddata = s.getBytes();

DatagramPacket send = new DatagramPacket(senddata, senddata.length, inet, 8892);

moi.send(send);

for(int i=0;i<3;i++){

DatagramPacket receive = new DatagramPacket(receivedata, receivedata.length);

moi.receive(receive);

kq+=new String(receive.getData())+" \*\*\* ";

}

*tfkq*.setText(kq);

} catch (UnknownHostException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (IOException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

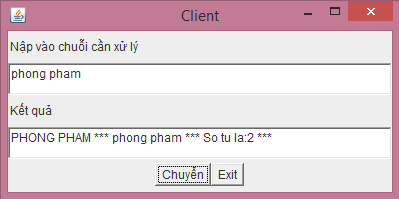
if (e.getSource() == *bt2*)

System.*exit*(0);

}

}

1. **Demo:**

****

1. Xây dựng chương trình hội thoại Client/Server hoạt động theo giao thức UDP

* 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.

1. **Thuật toán:**
   * Sử dụng giao thức UDP để giao tiếp giữa Client và Server. Để làm được điều này thì ta sử dụng DatagramSocket ở cả Client và Server giúp giao tiếp với nhau thông qua các cổng ta đã định.
   * Để gửi và nhận thông tin trong giao thức UDP ta dùng Datagrampacket.
2. **Mã nguồn:**

**File server.java**

package Bài2;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.net.UnknownHostException;

public class Server {

public static void main(String[] args) throws UnknownHostException, Exception {

byte senddata[];

byte receivedata[] = new byte[1024];

DatagramSocket server = new DatagramSocket(8892);

DatagramPacket receive = new DatagramPacket(receivedata, receivedata.length);

server.receive(receive);

InetAddress inet = receive.getAddress();

int port = receive.getPort();

String nhan = new String(receive.getData());

String ketqua = *xulychuoi*(nhan);

senddata = new byte[ketqua.length()];

senddata = ketqua.getBytes();

DatagramPacket send = new DatagramPacket(senddata, senddata.length, inet, port);

server.send(send);

}

public static String xulychuoi(String s) {

float ketqua = 0;

int i = 0;

float giatritam = 0;

while (i < s.length()) {

while ((s.charAt(i) != '+') && ((s.charAt(i) != '-')) && (s.charAt(i) != '\*') && (s.charAt(i) != '/')) {

ketqua = ketqua \* 10 + Integer.*parseInt*(String.*valueOf*(s.charAt(i)));

i++;

}

if (s.charAt(i) == '+') {

i++;

while ((i < s.length() && (s.charAt(i) != '+') && ((s.charAt(i) != '-')) && (s.charAt(i) != '\*')

&& (s.charAt(i) != '/'))) {

giatritam = giatritam \* 10 + Integer.*parseInt*(String.*valueOf*(s.charAt(i)));

i++;

}

ketqua += giatritam;

giatritam = 0;

}

if ((i < s.length()) && (s.charAt(i) == '-')) {

i++;

while ((i < s.length() && (s.charAt(i) != '+') && ((s.charAt(i) != '-')) && (s.charAt(i) != '\*')

&& (s.charAt(i) != '/'))) {

giatritam = giatritam \* 10 + Integer.*parseInt*(String.*valueOf*(s.charAt(i)));

i++;

}

ketqua -= giatritam;

giatritam = 0;

}

if ((i < s.length()) && (s.charAt(i) == '\*')) {

i++;

while ((i < s.length() && (s.charAt(i) != '+') && ((s.charAt(i) != '-')) && (s.charAt(i) != '\*')

&& (s.charAt(i) != '/'))) {

giatritam = giatritam \* 10 + Integer.*parseInt*(String.*valueOf*(s.charAt(i)));

i++;

}

ketqua \*= giatritam;

giatritam = 0;

}

if ((i < s.length()) && (s.charAt(i) == '/')) {

i++;

while ((i < s.length() && (s.charAt(i) != '+') && ((s.charAt(i) != '-')) && (s.charAt(i) != '\*')

&& (s.charAt(i) != '/'))) {

giatritam = giatritam \* 10 + Integer.*parseInt*(String.*valueOf*(s.charAt(i)));

i++;

}

ketqua /= giatritam;

giatritam = 0;

}

}

System.*out*.print(ketqua);

return String.*valueOf*(ketqua);

}

}

**File client.java:**

package Bài2;

import java.awt.Button;

import java.awt.GridBagLayout;

import java.awt.GridLayout;

import java.awt.Label;

import java.awt.Panel;

import java.awt.TextField;

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 java.net.UnknownHostException;

import javax.swing.JFrame;

public class Client implements ActionListener {

public static Panel *pn1*, *pn2*, *pn3*, *pn*;

public static TextField *tfnhap* = new TextField(5);

public static TextField *tfkq* = new TextField();

public static Label *lbnhap* = new Label();

public static Label *lbkq* = new Label();

public static Button *bt1*, *bt2*;

public static String *s* = "";

DatagramSocket moi;

InetAddress inet;

byte senddata[];

byte receivedata[];

public Client() throws UnknownHostException, IOException {

GUI();

}

void GUI() throws UnknownHostException, IOException {

JFrame fr = new JFrame("Client");

fr.setSize(400, 200);

fr.setLayout(new GridLayout());

*lbnhap*.setText("Nập vào chuỗi cần xử lý");

*lbkq*.setText("Kết quả");

*pn* = new Panel(new GridLayout(5, 1));

*pn1* = new Panel(new GridBagLayout());

*bt1* = new Button("Chuyển");

*bt2* = new Button("Exit");

*pn*.add(*lbnhap*);

*pn*.add(*tfnhap*);

*pn*.add(*lbkq*);

*pn*.add(*tfkq*);

*pn1*.add(*bt1*);

*pn1*.add(*bt2*);

*pn*.add(*pn1*);

fr.add(*pn*);

*bt1*.addActionListener(this);

*bt2*.addActionListener(this);

fr.~~show~~();

inet = InetAddress.*getByName*("localhost");

moi = new DatagramSocket();

}

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

Client t = new Client();

}

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

if (e.getSource() == *bt1*) {

try {

String s = *tfnhap*.getText();

senddata = new byte[s.length()];

receivedata = new byte[1024];

senddata = s.getBytes();

DatagramPacket send = new DatagramPacket(senddata, senddata.length, inet, 8892);

moi.send(send);

DatagramPacket receive = new DatagramPacket(receivedata, receivedata.length);

moi.receive(receive);

String kq = new String(receive.getData());

*tfkq*.setText(kq);

} catch (UnknownHostException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (IOException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

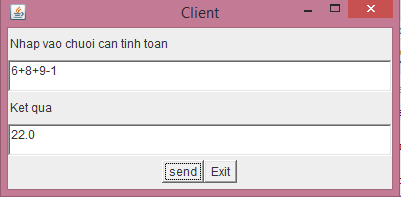
if (e.getSource() == *bt2*)

System.*exit*(0);

}

}

1. **Demo:**

****

1. Xây dựng chương trình hội thoại chat room Client/Server hoạt động theo giao thức UDP

* 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.

1. **Thuật toán:**

Bước 1: Nhập vào IP của máy Server và cổng để kết nối đến Server.

Bước 2: Tạo nên 1 giao diện chat room, khi đó Client sẽ gửi 1 thông báo lên server rằng client này đã tham gia phòng chat.

Bước 3: Khi tạo giao diện xong thì sẽ tạo ra 1 luồng nhận tin nhắn từ server thông qua socket (DataInputStream dùng hàm readUTF()) và nối tin nhắn nhận được vào bảng hiển thị.

Bước 4: Khi client nhập vào tin nhắn và nhấn nút Send thì nội dung tin nhắn này sẽ được gửi lên server thông qua socket (DataOutputStream dùng hàm writeUTF()).

1. **Mã nguồn:**

#### Login.java

public class Login extends JFrame implements ActionListener{

JTextField tfUser;

public Login(String title) {

super(title);

JPanel p1 = new JPanel();

JPanel p2 = new JPanel();

JPanel p3 = new JPanel();

JPanel p4 = new JPanel();

JLabel lbTitle = new JLabel("Chat Room TCP");

lbTitle.setFont(new Font(lbTitle.getName(), Font.*PLAIN*, 40));

JLabel lbUser = new JLabel("User");

tfUser = new JTextField(30);

JButton btnLogin = new JButton("Login");

JButton btnReset = new JButton("Reset");

JButton btnExit = new JButton("Exit");

btnLogin.addActionListener(this);

btnReset.addActionListener(this);

btnExit.addActionListener(this);

p1.add(lbTitle);

p2.add(lbUser);

p2.add(tfUser);

p4.add(btnLogin);

p4.add(btnReset);

p4.add(btnExit);

add(p1);

add(p2);

add(p3);

add(p4);

setSize(400, 300);

setLocation(200, 50);

setLayout(new GridLayout(5, 1));

setVisible(true);

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

}

@Override

public void actionPerformed(ActionEvent e) {

String btn=e.getActionCommand();

if(btn.equals("Login")){

String user = tfUser.getText();

String ip = JOptionPane.*showInputDialog*("Nhập vào IP máy chủ", "localhost");

int port = Integer.*parseInt*(JOptionPane.*showInputDialog*("Nhập vào port máy chủ", "1221"));

new Room(user,ip,port);

dispose();

}

if(btn.equals("Reset")){

tfUser.setText("");

}

if(btn.equals("Exit")){

System.*exit*(0);

}

}

public static void main(String[] args) {

new Login("Đăng nhập");

}

}

#### Room.java

(Chương trình client để nhận và gửi tin nhắn)

public class Room extends JFrame implements Runnable{

JTextArea taMessage, taAllMessage;

JList<String> listUser=new JList<String>();

String user, ip; int port;

private Socket socket;

private DataInputStream din;

private DataOutputStream dout;

public Room(String user, String ip, int port){

super("Chat Room TCP:");

this.user = user;

this.ip = ip;

this.port = port;

createUI();

try {

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

socket = new Socket(ip, port);

din = new DataInputStream(socket.getInputStream());

dout = new DataOutputStream(socket.getOutputStream());

dout.writeUTF(this.user);

new Thread(this).start();

} catch (Exception e) {

e.printStackTrace();

}

}

public void createUI(){

setDefaultCloseOperation(*EXIT\_ON\_CLOSE*);

JPanel p0= new JPanel();

JLabel lbTitle = new JLabel("----"+ user+" : "+ip+" ----");

p0.add(lbTitle);

add(p0,"North");

JPanel p1= new JPanel();

p1.setLayout(new BorderLayout());

taAllMessage = new JTextArea();

taAllMessage.setWrapStyleWord(true);

taAllMessage.setLineWrap(true);

taAllMessage.setEditable(false);

JScrollPane spAllMessage = new JScrollPane(taAllMessage);

spAllMessage.setVerticalScrollBarPolicy(JScrollPane.*VERTICAL\_SCROLLBAR\_ALWAYS*);

p1.add(spAllMessage);

add(p1);

JPanel p2= new JPanel();

p2.setLayout(new BorderLayout());

taMessage = new JTextArea(3,35);

taMessage.setWrapStyleWord(true);

taMessage.setLineWrap(true);

JScrollPane spMessage = new JScrollPane(taMessage);

p2.add(spMessage);

JButton btnSend = new JButton("Send");

btnSend.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

try {

dout.writeUTF(taMessage.getText());

taMessage.setText("");

} catch (IOException e1) {

e1.printStackTrace();

}

}

});

p2.add(btnSend, "East");

add(p2, "South");

setSize(350, 300);

setLocation(400, 100);

setVisible(true);

}

@Override

public void run() {

try {

while (true) {

String message = din.readUTF();

taAllMessage.append(message + "\n");

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

* Server.java (Nhận kết nối từ Client)

public class Server{

ServerSocket server;

public Server(int port) {

listen(port);

}

public void listen(int port){

try {

System.*out*.println(port);

server = new ServerSocket(port);

while(true){

Socket client = server.accept();

if(client.isConnected()){

ConnectionThread connectionThread = new ConnectionThread(server, client);

connectionThread.start();

}

}

} catch (Exception e) {

e.printStackTrace();

}

}

public static void main(String[] args) {

int port = Integer.*parseInt*(JOptionPane.*showInputDialog*("Nhập port máy chủ: ","1221"));

new Server(port);

}

}

#### ConnectionThread.java

(Luồng làm việc trên server)

public class ConnectionThread extends Thread{

private Socket client;

ServerSocket server;

private DataInputStream din;

private DataOutputStream dout;

private static Hashtable<Socket,ConnectionThread> *allConnections* = new Hashtable<Socket,ConnectionThread>();

public ConnectionThread(ServerSocket server, Socket client) {

this.connectionNumber = *allConnections*.size();

*allConnections*.put(client,this);

this.server= server;

this.client=client;

}

public void run(){

try {

din = new DataInputStream(client.getInputStream());

dout = new DataOutputStream(client.getOutputStream());

String user=din.readUTF();

messageAllSockets(user+" đã tham gia vào phòng chat.");

String receiveData;

while(( receiveData = din.readUTF())!=null){

messageAllSockets(user+": "+receiveData);

}

} catch (Exception e) {

e.printStackTrace();

}

}

public void messageAllSockets(String message){

synchronized (*allConnections*) {

for (Enumeration<?> e = *allConnections*.elements(); e.hasMoreElements();) {

ConnectionThread ct = (ConnectionThread) e.nextElement();

try {

ct.dout.writeUTF(message);

} catch (Exception e2) {

e2.printStackTrace();

}

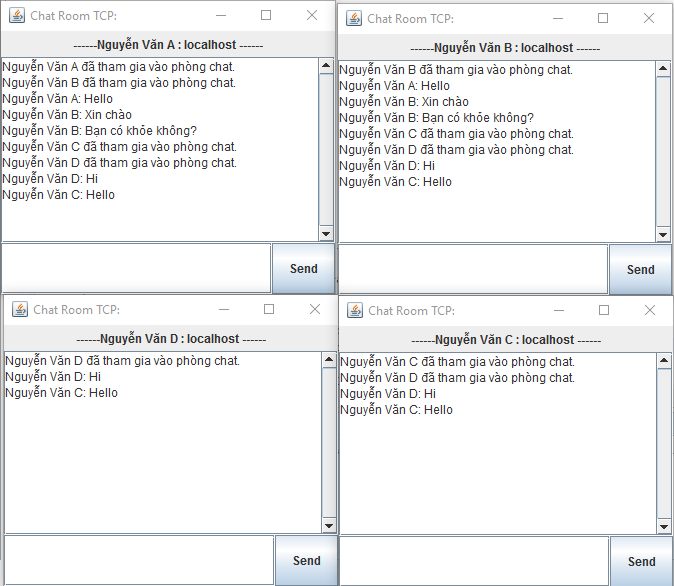
}

}

}

}

1. **Demo:**



1. **BÀI THỰC HÀNH SỐ 3 HỌC PHẦN LẬP TRÌNH MẠNG**

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



1. **Thuật toán:**

**Đăng nhập**

Bước 1: Kiểm tra xem tài khoản và mật khẩu đã đúng định dạng chưa, đúng định dạng là không được để trống và nhỏ hơn 30 kí tự

Bước 2: Kiểm tra tài khoản và mật khẩu nhập vào có tồn tại trong cơ sở dữ liệu hay không. Dùng câu lệnh truy vấn “SELECT \* FROM admin WHERE taikhoan =? AND matkhau=?” để kiểm tra. Nếu tài khoản và mật khẩu có tồn tại trong cơ sở dữ liệu thì vào trang “trang-chu.jsp”, nếu sai thì chuyển về trang “login.jsp” với thông báo lỗi “Tài khoản hoặc mật khẩu không đúng”.

**Xem nhân viên**

Lấy danh sách sinh viên trong cơ sở dữ liệu. Dùng câu lệnh truy vấn “SELECT \* FROM nhansu”. Hiển thị ra cho người dùng.

* **Xem phòng ban**

Lấy danh sách sinh viên trong cơ sở dữ liệu. Dùng câu lệnh truy vấn “SELECT \* FROM phongban”. Hiển thị ra cho người dùng.

* **Tìm kiếm**

Bước 1: Kiểm tra thông tin tìm kiếm nếu chưa nhập trường nào thì thông báo “Phải nhập vào ít nhất 1 trường để tìm kiếm”

Bước 2: Lấy thông tin tìm kiếm và kiểm tra nếu có điều kiện tìm kiếm thì thêm 1 phần câu truy vấn.

* 1. Nếu có nhập id phòng ban (idpb) thì thêm “AND idpb="+idpb”
  2. Nếu có nhập mô tả (mota) thì thêm “AND mota LIKE '%"+moTa+"%'" “
  3. Nếu có nhập tên phòng ban (tenphongban) thì thêm “AND tenphongban LIKE '%"+tenPhongBan+"%'"”
  4. Rồi ghép vào sau với câu lệnh “SELECT \* FROM phongban WHERE 1” để tìm kiếm
* **Cập nhật thông tin phòng ban**

Bước 1: Khi kích chọn sửa thông tin của 1 phòng ban thì dựa vào id phòng ban (idphongban) để lấy thông tin hiện tại về phòng ban. Dùng câu lệnh truy vấn “SELECT \* FROM phongban WHERE idpb =?”. Hiển thị thông tin phòng ban lên màn hình. Không cho phép sửa id phòng ban, chỉ sửa các thông tin khác.

Bước 2: Kiểm tra thông tin sửa, nếu xóa trống thông tin thì sẽ thông báo “Mô tả không để trống” hoặc “Tên phòng ban không để trống”.

Bước 3: Lấy thông tin sửa. Dùng câu lệnh truy vấn “UPDATE phongban SET tenphongban =?, mota=? WHERE idpb=?” để cập nhật lại dữ liệu phòng ban. Sau đó hiển thị danh sách phòng ban đã cập nhật

* **Xóa thông tin phòng ban**

Khi kích chọn sửa thông tin của 1 phòng ban thì dựa vào id phòng ban (idphongban) để xóa thông tin về phòng ban. Dùng câu lệnh truy vấn “DELETE FROM phongban WHERE idpb ="+idpb”. Sau đó hiển thị danh sách phòng ban đã cập nhật. Nếu xóa thành công thì hiển thị thông báo “Xóa thành công”, nếu không thì “Xóa thất bại”.

* **Xóa tất cả phòng ban**

Kích nút xóa tất cả thông tin về phòng ban. Dùng câu lệnh truy vấn “DELETE FROM phongban”. Sau đó hiển thị danh sách phòng ban trống với thông báo “Không tồn tại dữ liệu”. Nếu xóa thành công thì hiển thị thông báo “Xóa thành công”, nếu không thì “Xóa thất bại”.

1. **Mã nguồn:**

**reset.css**

/\*Reset CSS\*/

**html,** **body,** **div,** **span,** **applet,** **object,** **iframe,**

**h1,** **h2,** **h3,** **h4,** **h5,** **h6,** **p,** **blockquote,** **pre,**

**a,** **abbr,** **acronym,** **address,** **big,** **cite,** **code,**

**del,** **dfn,** **em,** **img,** **ins,** **kbd,** **q,** **s,** **samp,**

**small,** **strike,** **strong,** **sub,** **sup,** **tt,** **var,**

**b,** **u,** **i,** **center,**

**dl,** **dt,** **dd,** **ol,** **ul,** **li,**

**fieldset,** **form,** **label,** **legend,**

**table,** **caption,** **tbody,** **tfoot,** **thead,** **tr,** **th,** **td,**

**article,** **aside,** **canvas,** **details,** **embed,**

**figure,** **figcaption,** **footer,** **header,** **hgroup,**

**menu,** **nav,** **output,** **ruby,** **section,** **summary,**

**time,** **mark,** **audio,** **video** {

margin: *0*;

padding: *0*;

}

/\* HTML5 display-role reset for older browsers \*/

**article,** **aside,** **details,** **figcaption,** **figure,**

**footer,** **header,** **hgroup,** **menu,** **nav,** **section** {

display: *block*;

}

**body** {

line-height: *1*;

}

**ol,** **ul** {

list-style: *none*;

}

**blockquote,** **q** {

quotes: *none*;

}

**blockquote***:before***,** **blockquote***:after***,**

**q***:before***,** **q***:after* {

content: *''*;

content: *none*;

}

**table** {

border-collapse: *collapse*;

border-spacing: *0*;

}

* **style.css**

*.center* **ul** **li**{

text-align: *left*;

line-height: *30px*;

font-size: *20px*;

padding-left: *155px*;

padding-top: *20px*;

}

*.center* **table,** *.center* **form**{

margin: *auto*;

margin-top: *30px*;

}

*.center* **form** *.content-form* **input**{

margin: *15px*;

}

**label**{

font-size: *20px*;

}

**input**[type='text']{

width: *200px*;

font-size: *19px*;

}

*.center* **form** *.content-form* **label**{

display: *block*;

text-align: *left*;

padding-left: *350px*;

}

*.center* **table** **tr** **td,** *.center* **table** **tr** **th**{

padding: *10px*;

width: *150px*;

font-size: *20px*;

}

*.loi***,** *#loi*{

color: *red*;

padding-top: *10px*;

padding-bottom: *10px*;

display: *block*;

}

*.clear*{

clear: *both*;

}

*.wrapper*{

margin:*0px* *auto*;

}

*.wrapper* *.header*{

text-align:*center*;

border-bottom:*1px* *solid* *black*;

padding-bottom:*30px*;

background-color:*#d3d3d3*;

}

*.wrapper* *.header* **table**{

margin:*0px* *auto*;

width:*100%*;

border:*3px* *double* *black*;

}

*.wrapper* *.header* **table** **tr**{}

*.wrapper* *.header* **table** **tr** **td**{}

*.wrapper* *.header* **table** **tr** **td** **img**{

width:*200px*;

height:*100px*;

}

*.wrapper* *.header* **table** **tr** **td** **p**{

text-align:*center*;

font-size:*30px*;

}

*.wrapper* *.left-bar*{

float:*left*;

width: *200px*;

}

*.wrapper* *.left-bar* **ul**{

border: *1px* *solid* *black*;

padding:*10px*;

background-color:*#d3d3d3*;

min-height:*450px*;

}

*.wrapper* *.left-bar* **ul** **li**{

border:*1px* *solid* *#6c6cdb*;

border-radius:*10px*;

margin-bottom:*10px*;

text-align:*center*;

background-color: *white*;

font-size: *19px*;

}

*.wrapper* *.left-bar* **ul** **li***:HOVER*{

background-color: *#669eb0*;

}

*.wrapper* *.left-bar* **ul** **li** **a**{

text-decoration:*none*;

}

*.wrapper* *.center*{

padding-top:*50px*;

text-align: *center*;

margin: *auto*;

height: *450px*;

width: *900px*;

}

*.wrapper* *.right-bar*{

float:*right*;

border:*1px* *solid* *#000000*;

min-height:*470px*;

background-color:*#d3d3d3*;

width: *200px*;

}

*.wrapper* *.right-bar* **img**{

display:*block*;

margin:*5px*;

border:*1px* *dotted* *#66ec2d*;

width: *200px*;

height: *140px*;

}

**header.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<link href=*"*<%=request.getContextPath()%>*/templates/public/css/reset.css"* type=*"text/css"* rel=*"stylesheet"*/>

<link href=*"*<%=request.getContextPath()%>*/templates/public/css/style.css"* type=*"text/css"* rel=*"stylesheet"*/>

</head>

<body>

<div class=*"wrapper"*>

<div class=*"header"*>

<table border=*"1"*>

<tr>

<td>

<img alt=*"1.jpg"* src=*"*<%=request.getContextPath() %>*/files/1.jpg"*/>

</td>

<td>

<marquee width=*"940px"*><p>CHƯƠNG TRÌNH QUẢN LÝ PHÒNG BAN</p></marquee>

</td>

<td>

<img alt=*"2.jpg"* src=*"*<%=request.getContextPath() %>*/files/2.jpg"*>

</td>

</tr>

</table>

</div>

**left-bar.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<div class=*"left-bar"*>

<ul>

<li><a href=*"trang-chu.jsp"*>Trang chủ</a></li>

<li><a href=*"xem-nhan-vien.jsp"*>Xem nhân viên</a></li>

<li><a href=*"xem-phong-ban.jsp"*>Xem phòng ban</a></li>

<li><a href=*"tim-kiem-phong-ban.jsp"*>Tìm kiếm phòng ban</a></li>

<li><a href=*"xem-phong-ban.jsp"*>Cập nhật phòng ban</a></li>

<li><a href=*"xem-phong-ban.jsp"*>Xóa phòng ban</a></li>

<li><a href=*"xem-phong-ban.jsp"*>Xóa tất cả phòng ban</a></li>

<li><a href=*"tro-giup-chuong-trinh.jsp"*>Trợ giúp chương trình</a></li>

</ul>

</div>

**right-bar.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<div class=*"right-bar"*>

<img alt=*"1.jpg"* src=*"*<%=request.getContextPath() %>*/files/4.jpg"*/>

<img alt=*"2.jpg"* src=*"*<%=request.getContextPath() %>*/files/5.jpg"*/>

<img alt=*"3.jpg"* src=*"*<%=request.getContextPath() %>*/files/3.jpg"*/>

</div>

**tro-giup-chuong-trinh.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<title>Trợ giúp chương trình</title>

<%@include file=*"/templates/public/inc/header.jsp"* %>

<%@include file=*"/templates/public/inc/left-bar.jsp"* %>

<%@include file=*"/templates/public/inc/right-bar.jsp"* %>

<div class=*"center"*>

<h1>Trợ giúp chương trình</h1>

<ul>

<li>

Chọn menu 1 khác bất kì để đăng nhập vào hệ thống và sử dụng các chức năng khác

</li>

<li>

Sau khi đăng nhập:

</li>

<li>

Chọn menu "Trang chủ" để xem trang chủ của hệ thống

</li>

<li>

Chọn menu "Xem nhân viên" để xem danh sách nhân viên

</li>

<li>

Chọn menu "Xem phòng ban" để xem danh sách phòng ban

<ul>

<li>

Chỉnh sửa phòng ban.

</li>

<li>

Chức năng tìm kiếm phòng ban.

</li>

<li>

Xóa 1 phòng ban.

</li>

<li>

Xóa tất cả phòng ban

</li>

</ul>

</li>

</ul>

</div>

</body>

</html>

**login.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<title>Đăng nhập</title>

<%@include file=*"/templates/public/inc/header.jsp"* %>

<%@include file=*"/templates/public/inc/left-bar.jsp"* %>

<%@include file=*"/templates/public/inc/right-bar.jsp"* %>

<%

String msg = (String)request.getAttribute("msg");

%>

<div class=*"center"*>

<form method=*"post"* action=*"checklogin.jsp"* name=*"f1"* onsubmit="return checkLogin();">

<h1>Đăng nhập hệ thống quản lý phòng ban</h1>

<span class=*"loi"*><%**if**(msg!=**null**) out.print(msg);%></span>

<span id=*"loi"*></span>

<label>Tài khoản: </label>

<input type=*"text"* class=*""* name=*"taiKhoan"* style="width: *130px*;"/>

<br/>

<br/>

<label>Mật khẩu: </label>

<input type=*"password"* class=*""* name=*"matKhau"* style="width: *130px*;" />

<br/>

<br/>

<input type=*"submit"* name=*"dangNhap"* value=*"Đăng nhập"* class=*""* />

<input type=*"reset"* name=*"nhapLai"* value=*"Nhập lại"* class=*""* />

</form>

</div>

<script type=*"text/javascript"*>

**function** checkLogin(){

**var** taiKhoan,matKhau,msg="";

taiKhoan = document.getElementsByName("taiKhoan")[0];

matKhau = document.getElementsByName("matKhau")[0];

**switch**(msg){

**case** taiKhoan.value:

msg="Tài khoản không được để trống";

taiKhoan.focus();

**break**;

**case** matKhau.value:

msg="Mật khẩu không được để trống";

matKhau.focus();

**break**;

}

**if**(taiKhoan.value.length>30){

msg="Tài khoản phải nhỏ hơn 30 ký tự";

taiKhoan.focus();

}**else**

**if**(matKhau.value.length>30){

msg="Mật khẩu phải nhỏ hơn 30 ký tự";

matKhau.focus();

}

**if**(msg!=""){

document.getElementById("loi").innerHTML=msg;

**return** **false**;

}

}

</script>

</div>

</body>

</html>

**check-login.jsp**

<%@page import=*"java.sql.ResultSet"*%>

<%@page import=*"java.sql.PreparedStatement"*%>

<%@page import=*"java.sql.DriverManager"*%>

<%@page import=*"java.sql.Connection"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%

String taiKhoan = request.getParameter("taiKhoan");

String matKhau = request.getParameter("matKhau");

**if**(taiKhoan==**null**){

response.sendRedirect("login.jsp");

}**else**{

Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/data?useUnicode=true&characterEncoding=UTF-8", "root", "");

String sql="SELECT \* FROM admin WHERE taikhoan =? AND matkhau=?";

PreparedStatement pstm = conn.prepareStatement(sql);

pstm.setString(1, taiKhoan);

pstm.setString(2, matKhau);

ResultSet rs = pstm.executeQuery();

**if**(rs.next()){

rs.close();

pstm.close();

conn.close();

session.setAttribute("userSession", taiKhoan);

response.sendRedirect("trang-chu.jsp");

}**else**{

rs.close();

pstm.close();

conn.close();

request.setAttribute("msg", "Tài khoản hoặc mật khẩu không đúng");

request.getRequestDispatcher("login.jsp").forward(request, response);

}

}

%>

**trang-chu.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<title>Trang trình quản lý phòng ban</title>

<%

String userSession = (String)session.getAttribute("userSession");

**if**(userSession==**null**) response.sendRedirect("login.jsp");

**else**{

%>

<%@include file=*"/templates/public/inc/header.jsp"* %>

<%@include file=*"/templates/public/inc/left-bar.jsp"* %>

<%@include file=*"/templates/public/inc/right-bar.jsp"* %>

<div class=*"center"* style=*"background: url('*<%=request.getContextPath()%>*/files/book.png') no-repeat fixed center;"*>

<h1 style="padding-top: *100px*; font-size: *60px*">Xin chào <%=userSession %> đã đăng nhập vào hệ thống quản lý phòng ban</h1>

</div>

</div>

</html>

<%}%>

**xem-nhan-vien.jsp**

<%@page import=*"java.sql.PreparedStatement"*%>

<%@page import=*"java.sql.ResultSet"*%>

<%@page import=*"java.sql.Statement"*%>

<%@page import=*"java.sql.DriverManager"*%>

<%@page import=*"java.sql.Connection"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<title>Xem nhân viên</title>

<%

String userSession = (String)session.getAttribute("userSession");

**if**(userSession==**null**) {

response.sendRedirect("login.jsp");

}

**else**{

Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/data?useUnicode=true&characterEncoding=UTF-8", "root", "");

String sql="SELECT \* FROM nhansu";

Statement stm = conn.createStatement();

ResultSet rs = stm.executeQuery(sql);

%>

<%@include file=*"/templates/public/inc/header.jsp"* %>

<%@include file=*"/templates/public/inc/left-bar.jsp"* %>

<%@include file=*"/templates/public/inc/right-bar.jsp"* %>

<div class=*"center"*>

<h1>Xem thông tin nhân sự</h1>

<table border=*"1px"*>

<tr>

<th>Mã số</th>

<th>Họ tên</th>

<th>Địa chỉ</th>

<th>Ghi chú</th>

</tr>

<%

**while**(rs.next()){

%>

<tr>

<td><%=rs.getInt(1) %></td>

<td><%=rs.getString(2) %></td>

<td><%=rs.getString(3) %></td>

<td><%=rs.getString(4) %></td>

</tr>

<%

}

rs.close();

stm.close();

conn.close();

%>

</table>

</div>

</div>

</body>

</html>

<%}%>

**xem-phong-ban.jsp**

<%@page import=*"java.sql.PreparedStatement"*%>

<%@page import=*"java.sql.ResultSet"*%>

<%@page import=*"java.sql.Statement"*%>

<%@page import=*"java.sql.DriverManager"*%>

<%@page import=*"java.sql.Connection"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<title>Xem phòng ban</title>

<%

String userSession = (String)session.getAttribute("userSession");

**if**(userSession==**null**) {

response.sendRedirect("login.jsp");

}

**else**{

Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/data?useUnicode=true&characterEncoding=UTF-8", "root", "");

String submit = request.getParameter("submit");

String sql="";

Statement stm=**null**;

**boolean** search=**false**;

**if**(submit!=**null**){

submit = **new** String(request.getParameter("submit").getBytes("ISO-8859-1"),"UTF-8");

String idpb = request.getParameter("idpb");

String tenPhongBan = **new** String(request.getParameter("tenPhongBan").getBytes("ISO-8859-1"),"UTF-8");

String moTa = **new** String(request.getParameter("moTa").getBytes("ISO-8859-1"),"UTF-8");

//Sửa

**if**(submit.equals("Sửa")){

sql = "UPDATE phongban SET tenphongban =?, mota=? WHERE idpb=?";

PreparedStatement pstm = conn.prepareStatement(sql);

pstm.setString(1, tenPhongBan);

pstm.setString(2, moTa);

pstm.setInt(3, Integer.parseInt(idpb));

pstm.executeUpdate();

pstm.close();

}

//Tìm kiếm

**if**(submit.equals("Tìm kiếm")){

sql = "SELECT \* FROM phongban WHERE 1";

**if**(!idpb.equals("")){

sql+= " AND idpb="+idpb;

}

**if**(!moTa.equals("")){

sql+= " AND mota LIKE '%"+moTa+"%'";

}

**if**(!tenPhongBan.equals("")){

sql+= " AND tenphongban LIKE '%"+tenPhongBan+"%'";

}

search=**true**;

}

}

//Lấy thông tin

**if**(!search){

sql="SELECT \* FROM phongban";

}

stm = conn.createStatement();

ResultSet rs = stm.executeQuery(sql);

//Thông báo msg khi xóa

String msg = (String)request.getAttribute("msg");

%>

<%@include file=*"/templates/public/inc/header.jsp"* %>

<%@include file=*"/templates/public/inc/left-bar.jsp"* %>

<%@include file=*"/templates/public/inc/right-bar.jsp"* %>

<div class=*"center"*>

<h1>Xem thông tin phòng ban</h1>

<span class=*"loi"*>

<%

**if**(msg!=**null**) out.println(msg);

%>

</span>

<form action=*"xoa-phong-ban.jsp"*>

<table border=*"1px"*>

<tr>

<th>ID phòng ban</th>

<th>Tên phòng ban</th>

<th>Mô tả</th>

<th>Sửa</th>

<th>Xóa<br>

<%**if**(!search) {%>

<input type=*"submit"* name=*"vvxoaTatCa"* value=*"Xóa tất cả"* class=*"delAll"*>

<%} %>

</th>

</tr>

<%

**int** soDong=0;

**while**(rs.next()){

soDong++;

%>

<tr>

<td><%=rs.getInt(1) %></td>

<td><%=rs.getString(2) %></td>

<td><%=rs.getString(3) %></td>

<td>

<a href=*"cap-nhat-phong-ban.jsp?idpb=*<%=rs.getInt(1)%>*"*>Sửa</a>

</td>

<td>

<a href=*"xoa-phong-ban.jsp?idpb=*<%=rs.getInt(1)%>*"*>Xóa</a>

</td>

</tr>

<%

}

//Thông báo lỗi khi không tìm thấy dữ liệu hoặc ko có dl trong bảng

String loi="";

**if**(soDong==0){

out.println("<h3 style='color:red;'>Không tồn tại dữ liệu</h3>");

}

rs.close();

stm.close();

conn.close();

%>

</table>

</form>

</div>

</body>

</html>

<%}%>

**cap-nhat-phong-ban.jsp**

<%@page import=*"java.sql.ResultSet"*%>

<%@page import=*"java.sql.PreparedStatement"*%>

<%@page import=*"java.sql.DriverManager"*%>

<%@page import=*"java.sql.Connection"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<%

String userSession = (String)session.getAttribute("userSession");

**if**(userSession==**null**) {

response.sendRedirect("login.jsp");

}

**else**{

String id = request.getParameter("idpb");

**if**(id!=**null**){

**int** idpb = Integer.parseInt(id);

Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/data?useUnicode=true&characterEncoding=UTF-8", "root", "");

String sql="SELECT \* FROM phongban WHERE idpb =?";

PreparedStatement pstm = conn.prepareStatement(sql);

pstm.setInt(1, idpb);

ResultSet rs = pstm.executeQuery();

%>

<%@include file=*"/templates/public/inc/header.jsp"* %>

<%@include file=*"/templates/public/inc/left-bar.jsp"* %>

<%@include file=*"/templates/public/inc/right-bar.jsp"* %>

<div class=*"center"*>

<h1>Cập nhật thông tin phòng ban</h1>

<form action=*"xem-phong-ban.jsp?idpb=*<%=idpb %>*"* method=*"post"* name=*"f1"* onsubmit="return checkCapNhat()">

<%

**if**(rs.next()){

%>

<div class=*"content-form"*>

<span id=*"loi"*></span>

<label>ID phòng ban: <%=idpb %></label>

<br/>

<label>Tên phòng ban: </label>

<input type=*"text"* value=*"*<%=rs.getString(2) %>*"* class=*""* name=*"tenPhongBan"* />

<br/>

<label>Mô tả: </label>

<input type=*"text"* value=*"*<%=rs.getString(3) %>*"* class=*""* name=*"moTa"*/>

</div>

<input type=*"submit"* value=*"Sửa"* name=*"submit"*>

<input type=*"reset"* name=*"reset"* value=*"Nhập lại"* class=*""* />

<%} %>

</form>

</div>

</body>

<%}} %>

<script type=*"text/javascript"*>

**function** checkCapNhat(){

**var** moTa, tenPhongBan;

**var** msg="";

moTa = document.getElementsByName("moTa")[0];

tenPhongBan = document.getElementsByName("tenPhongBan")[0];

**switch**(msg){

**case** moTa.value:

msg="Mô tả không để trống";

moTa.focus();

**break**;

**case** tenPhongBan.value:

msg="Tên phòng ban không để trống";

tenPhongBan.focus();

**break**;

}

**if**(msg!=""){

document.getElementById("loi").innerHTML=msg;

**return** **false**;

}

}

</script>

</html>

**xoa-phong-ban.jsp**

<%@page import=*"java.sql.Statement"*%>

<%@page import=*"java.sql.ResultSet"*%>

<%@page import=*"java.sql.PreparedStatement"*%>

<%@page import=*"java.sql.DriverManager"*%>

<%@page import=*"java.sql.Connection"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%

String userSession = (String)session.getAttribute("userSession");

**if**(userSession==**null**) {

response.sendRedirect("login.jsp");

}

**else**{

System.out.println("12");

String msg;

String id = request.getParameter("idpb");

String xoaTatCa = request.getParameter("xoaTatCa");

Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/data?useUnicode=true&characterEncoding=UTF-8", "root", "");

String sql="";

**if**(id!=**null**){

**int** idpb = Integer.parseInt(id);

sql="DELETE FROM phongban WHERE idpb ="+idpb;

}**else** **if**(xoaTatCa!=**null**){

sql="DELETE FROM phongban";

}

Statement stm = conn.createStatement();

**int** delete = stm.executeUpdate(sql);

conn.close();

stm.close();

**if**(delete!=0){

msg = "Xóa thành công";

}**else** {

msg="Xóa thất bại";

}

request.setAttribute("msg", msg);

request.getRequestDispatcher("xem-phong-ban.jsp").forward(request, response);

}

%>

**tim-kiem-phong-ban.jsp**

<%@page import=*"java.sql.ResultSet"*%>

<%@page import=*"java.sql.PreparedStatement"*%>

<%@page import=*"java.sql.DriverManager"*%>

<%@page import=*"java.sql.Connection"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<title>Tìm kiếm phòng ban</title>

<%

String userSession = (String)session.getAttribute("userSession");

**if**(userSession==**null**) {

response.sendRedirect("login.jsp");

}

**else**{

%>

<%@include file=*"/templates/public/inc/header.jsp"* %>

<%@include file=*"/templates/public/inc/left-bar.jsp"* %>

<%@include file=*"/templates/public/inc/right-bar.jsp"* %>

<div class=*"center"*>

<h1>Tìm kiếm thông tin phòng ban</h1>

<form action=*"xem-phong-ban.jsp"* method=*"post"* name=*"f1"* onsubmit="return checkSearch()">

<div class=*"content-form"*>

<span id=*"loi"*></span>

<label>ID phòng ban:</label>

<input type=*"text"* value=*""* class=*""* name=*"idpb"*/>

<label>Tên phòng ban: </label>

<input type=*"text"* value=*""* class=*""* name=*"tenPhongBan"*/>

<label>Mô tả: </label>

<input type=*"text"* value=*""* class=*""* name=*"moTa"* />

<br />

<input type=*"submit"* value=*"Tìm kiếm"* name=*"submit"*>

<input type=*"reset"* value=*"Nhập lại"* name=*"reset"* />

</div>

</form>

</div>

</body>

<script type=*"text/javascript"*>

**function** checkSearch(){

**var** idpb,moTa, tenPhongBan;

**var** msg="";

idpb = document.getElementsByName("idpb")[0].value;

moTa = document.getElementsByName("moTa")[0].value;

tenPhongBan = document.getElementsByName("tenPhongBan")[0].value;

**if**(idpb==""&&moTa==""&&tenPhongBan==""){

document.getElementById("loi").innerHTML= "Phải nhập vào ít nhất 1 trường để tìm kiếm";

**return** **false**;

}

}

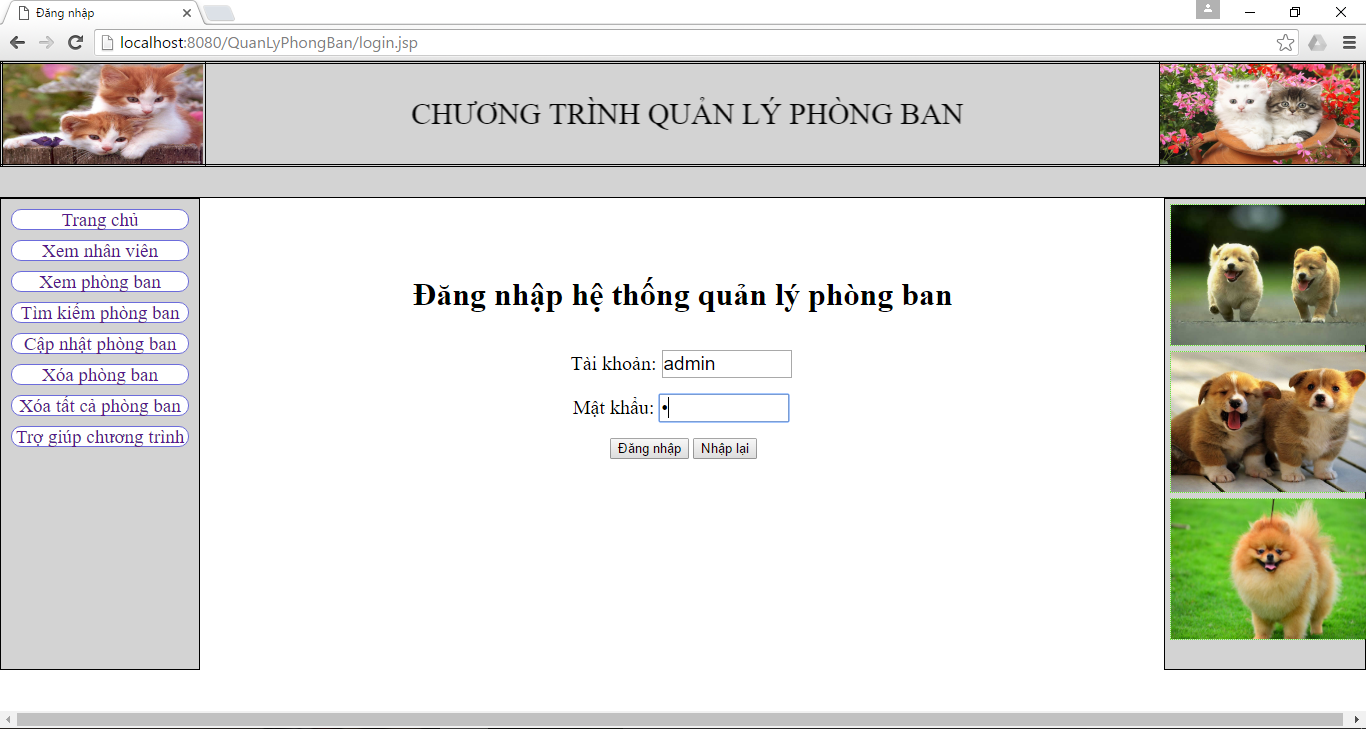
</script>

</html>

<%} %>

1. **Demo:**

* Màn hình đăng nhập.



* Màn hình trang chủ

Màn hình xem nhân viên

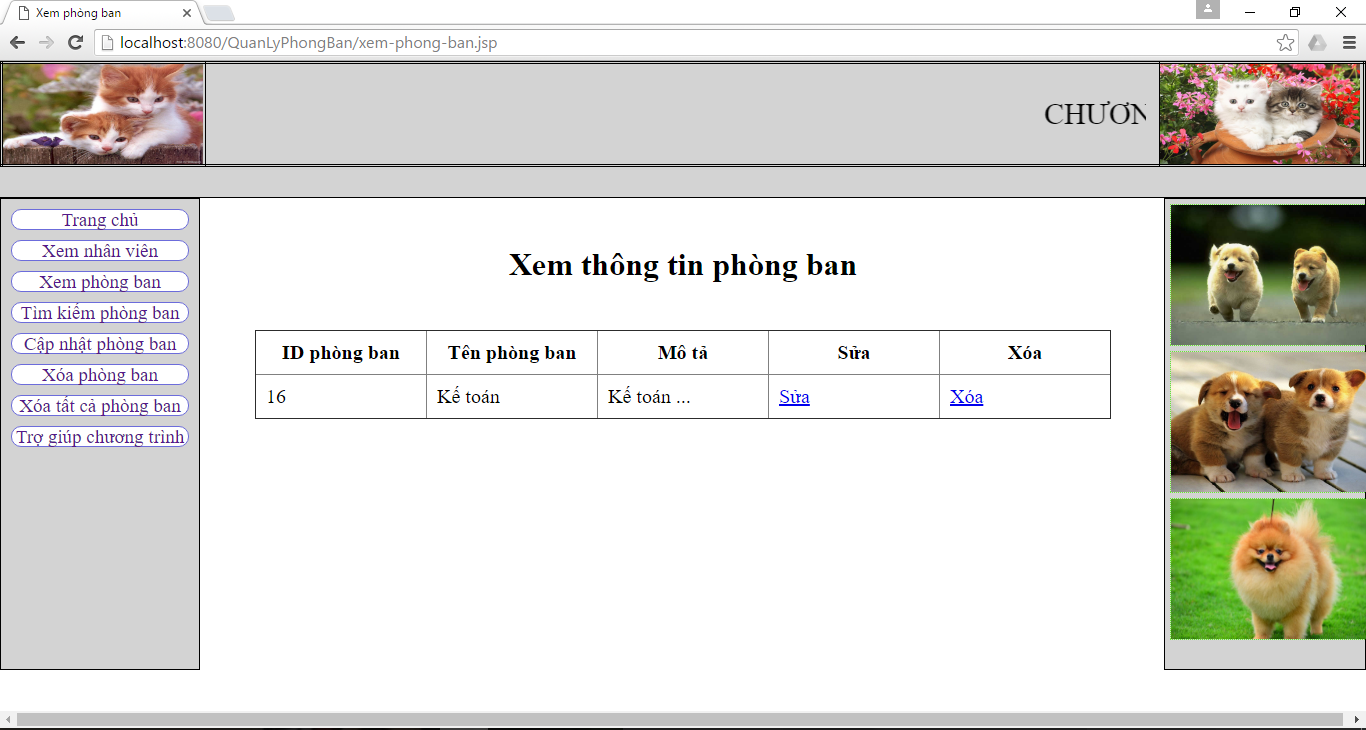


* Màn hình xem phòng ban



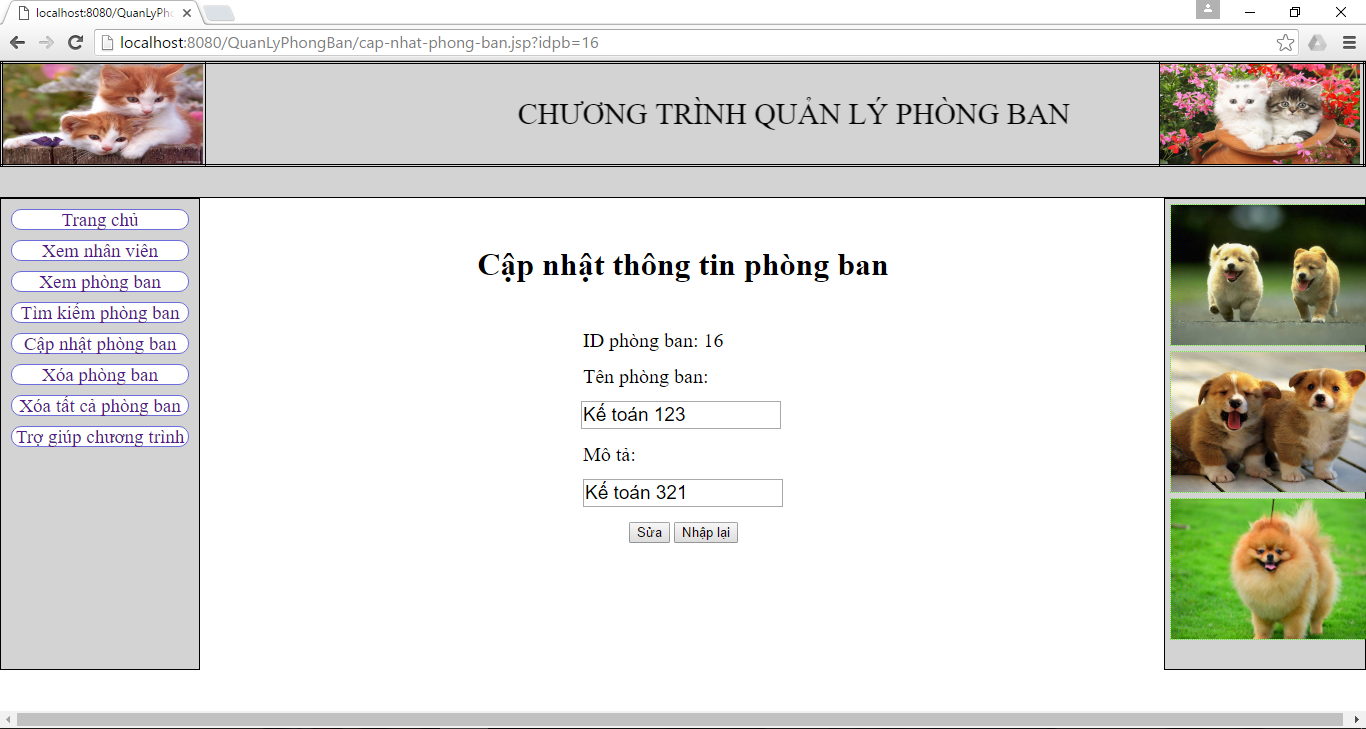
-Màn hình tìm kiếm phòng ban

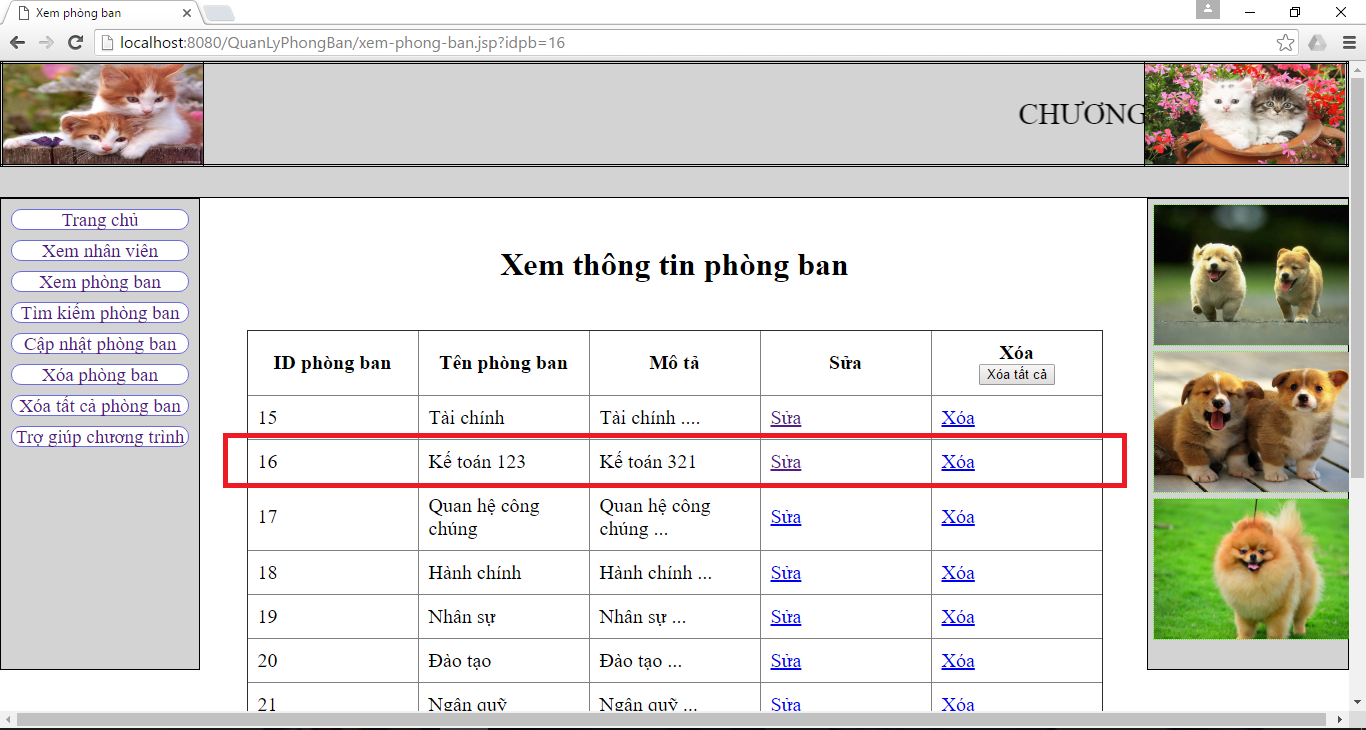




-Cập nhật phòng ban







* Xóa phòng ban





* Xóa tất cả phòng ban



-Trợ giúp chương trình

