**Bộ Công Thương**

**Trường Đại Học Công Nghiệp TP.HCM**

**Khoa Công Nghệ Thông Tin**

---🙦🕮🙤---



**MÔN PHÁT TRIỂN ỨNG DỤNG CÓ ĐỒ ÁN**

BÁO CÁO CUỐI KÌ

Đề tài: Topic 7 - Send and received email

Giáo viên hướng dẫn: Huỳnh Thái Học

Thành viên nhóm:

1. Võ Duy lân 14043781
2. Tô Thanh An 14037281
3. Huang Đạt Vinh 15034141

**Mô tả :**

**Lịch phân chia công việc theo tuần:**

|  |  |  |
| --- | --- | --- |
| Tuần | Công việc | Thực hiện |
| 1 | -Cài java (eclipse)  -Tìm hiểu sơ bộ về gửi và nhận email  -Có 3 giao thức để gửi mail: **SMTP,**  POP3, IMAP,Socket tìm hiểu về các giao thức đó | Hoàn thành |
| 2+3 | -Tìm hiểu về cách gửi và nhận mail | Hoàn thành |
| 4+5 | - Viết chương trình về mail server | Hoàn thành |
| 6+7 | -Tạo cơ sở dữ liệu sql  -Code database kết nối java với cơ sở dữ liệu sql server | Hoàn thành |
| 8+9 | Thiết lập các giao diện trên chương trình (các nút để soạn tin, hộp thư đến, hộp thư đã gửi,...) | Hoàn thành |
| 10 | -Thực hiện để gửi mail giữa 2 tài khoản  -Sửa và thêm những cái cần thiết để hoàn thành chương trình | Hoàn thành |

Contents

[**I.** **Email client là gì?** 3](#_Toc501181558)

[**II.** **Database** 3](#_Toc501181559)

[**III.** **Socket** 4](#_Toc501181560)

[**IV.** **Kết nối sql server với java** 5](#_Toc501181561)

[**V.** **DEMO** 15](#_Toc501181562)

[**Tài liệu tham khảo:** 17](#_Toc501181563)

**I.Email client là gì?**

**Email client** (từ kỹ thuật là **Mail User Agent** (MUA), trong tiếng Việt là **trình duyệt mail**) là một [phần mềm máy tính](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m) được dùng để truy cập và quản lý [email](https://vi.wikipedia.org/wiki/Email) của người dùng.

Các mail client phổ biến bao gồm [Microsoft Outlook](https://vi.wikipedia.org/wiki/Microsoft_Outlook), [IBM Lotus Notes](https://vi.wikipedia.org/w/index.php?title=Lotus_Notes&action=edit&redlink=1), [Mozilla](https://vi.wikipedia.org/wiki/Mozilla_Foundation) [Thunderbird](https://vi.wikipedia.org/wiki/Mozilla_Thunderbird),, [Evolution](https://vi.wikipedia.org/w/index.php?title=Evolution_(ph%E1%BA%A7n_m%E1%BB%81m)&action=edit&redlink=1) và [Apple Inc.](https://vi.wikipedia.org/wiki/Apple_Inc.)'s [Mail](https://vi.wikipedia.org/w/index.php?title=Mail_(application)&action=edit&redlink=1).

Một ứng dụng web cung cấp các dịch vụ quản lý tin nhắn, các thành phần, và các tính năng nhận mail đôi khi cũng được em là một trình duyệt mail, nhưng thường được gọi là [webmail](https://vi.wikipedia.org/wiki/Webmail). Các webmail phổ biến gồm [Gmail](https://vi.wikipedia.org/wiki/Gmail), [Lycos Mail](https://vi.wikipedia.org/w/index.php?title=Lycos_Mail&action=edit&redlink=1), [Mail.com](https://vi.wikipedia.org/w/index.php?title=Mail.com&action=edit&redlink=1), [Outlook.com](https://vi.wikipedia.org/w/index.php?title=Outlook.com&action=edit&redlink=1) và [Yahoo! Mail](https://vi.wikipedia.org/wiki/Yahoo!_Mail).

**Giao thức :**

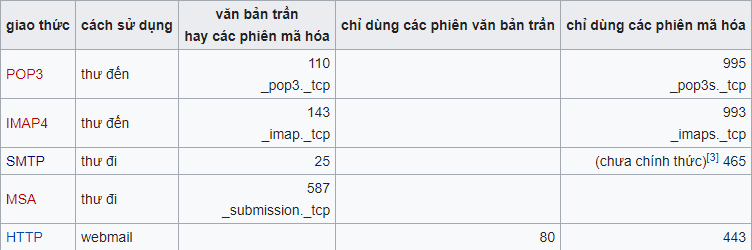
Trong khi các giao thức phổ biến để lấy email bao gồm [POP3](https://vi.wikipedia.org/wiki/POP3) và [IMAP4](https://vi.wikipedia.org/w/index.php?title=IMAP4&action=edit&redlink=1), gửi mail thường được thực hiện bằng cách sử dụng giao thức [SMTP](https://vi.wikipedia.org/wiki/SMTP).

Một tiêu chuẩn quan trọng được hỗ trợ bởi hầu hết các email clients là [MIME](https://vi.wikipedia.org/wiki/MIME), được sử dụng để gửi email đính kèm tập tin nhị phân. File đính kèm là các tập tin không phải là một phần của email, nhưng được gửi đi với email.

Hầu hết các email clients sử dụng một *User-Agent.*

**Số cổng :**

Email servers và clients theo quy ước sử dụng các số [cổng TCP](https://vi.wikipedia.org/wiki/TCP#C.C3.A1c_c.E1.BB.95ng_TCP) trong bảng sau. Đối với MSA, IMAP and POP3, các bảng báo cáo cũng lầcc nhãn mà một mail client sử dụng để truy vấn các bản ghi [SRV](https://vi.wikipedia.org/w/index.php?title=SRV&action=edit&redlink=1) và khám phá cả các tên máy chủ và số cổng của các dịch vụ tương ứng.

****

**II.Database**

**1. Định nghĩa:** là một [tập hợp thông tin](https://vi.wikipedia.org/w/index.php?title=T%E1%BA%ADp_h%E1%BB%A3p_th%C3%B4ng_tin&action=edit&redlink=1) có cấu trúc. Tuy nhiên, thuật ngữ này thường dùng trong [công nghệ thông tin](https://vi.wikipedia.org/wiki/C%C3%B4ng_ngh%E1%BB%87_th%C3%B4ng_tin) và nó thường được hiểu rõ hơn dưới dạng một tập hợp liên kết các [dữ liệu](https://vi.wikipedia.org/wiki/D%E1%BB%AF_li%E1%BB%87u), thường đủ lớn để lưu trên một thiết bị lưu trữ như đĩa hay băng. Dữ liệu này được duy trì dưới dạng một tập hợp các [tập tin](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin) trong [hệ điều hành](https://vi.wikipedia.org/wiki/H%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh) hay được lưu trữ trong các [hệ quản trị cơ sở dữ liệu](https://vi.wikipedia.org/wiki/H%E1%BB%87_qu%E1%BA%A3n_tr%E1%BB%8B_c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u).

**2. Ưu điểm:**

- Giảm sự trùng lặp thông tin xuống mức thấp nhất. Do đó đảm bảo thông tin có tính nhất quán và toàn vẹn dữ liệu.

- Đảm bảo dữ liệu có thể được truy xuất theo nhiều cách khác nhau.

- Nhiều người có thể sử dụng một cơ sở dữ liệu.

**3. Nhược điểm:**

- Tính chủ quyền của dữ liệu:

* Thể hiện ở phương diện an toàn dữ liệu.
* Khả năng biểu diễn mỗi liên hệ ngữ nghĩa của dữ liệu và tính chính xác của dữ liệu.
* Người khai thác cơ sở dữ liệu phải cập nhật cho CSDL những thông tin mới nhất.

- Tính bảo mật và quyền khai thác thông tin của người sử dụng:

* Do ưu điểm CSDL có thể cho nhiều người khai thác đồng thời. nên cần phải có một cơ chế bảo mật phân quyền khai thác CSDL.
* Các hệ điều hành nhiều người sử dụng hay cục bộ đều cung cấp cơ chế này.

- Tranh chấp dữ liệu:

* Khi nhiều người cùng truy nhập CSDL với các mục đích khác nhau. Rất có thể sẽ xảy ra hiện tượng tranh chấp dữ liệu.
* Cần có cơ chế ưu tiên khi truy cập CSDL. Ví dụ: admin luôn có thể truy cập cơ sở dữ liệu.
* Cấp quyền ưu tiên cho từng người khai thác.

- Cần đảm bảo an toàn dữ liệu khi có sự cố:

* Khi CSDL nhiều và được quản lý tập trung. Khả năng rủi ro mất dữ liệu rất cao. Các nguyên nhân chính là mất điện đột ngột hoặc hỏng thiết bị lưu trữ.
* Hiện tại có một số hệ điều hành đã có cơ chế tự động sao lưu ổ cứng và fix lỗi khi có sự cố xảy ra.
* Tuy nhiên: cẩn tắc vô áy náy. Chúng ta nên sao lưu dự phòng cho dữ liệu đề phòng trường hợp xấu xảy ra.

**4. Tính chất :**

- Phản ánh 1 phần của thế giới thực (mini world) được cập nhật phản ánh sự thay đổi của thế giới nó biểu diễn.

- 1 CSDL là 1 tập hợp DL liên kết với nhau một cách logic và mang 1 ý nghĩa nào đó.

- Được thiết kế phổ biến cho 1 mục đích riêng có nhóm người sử dụng.

**III.Socket**

* Socket là một cổng logic mà một chương trình sử dụng để kết nối với một chương trình

khác chạy trên một máy tính khác trên Internet. Chương trình mạng có thể sử dụng nhiều

Socket cùng một lúc, nhờ đó nhiều chương trình có thể sử dụng Internet cùng một lúc.

Có 2 loại Socket:

+ Stream Socket : Dựa trên giao thức TCP( Tranmission Control Protocol) việc truyền dữ

liệu chỉ thực hiện giữa 2 quá trình **đã thiết lập kết nối**. Giao thức này đảm bảo dữ

liệu được truyền đến nơi nhận một cách đáng tin cậy, đúng thứ tự nhờ vào cơ chế quản

lý luồng lưu thông trên mạng và cơ chế chống tắc nghẽn.

+ Datagram socket : Dựa trên giao thức UDP( User Datagram Protocol) việc truyền dữ liệu

**không yêu cầu có sự thiết lập kết nối giữa 2 quá trình**. Ngược lại với giao thức

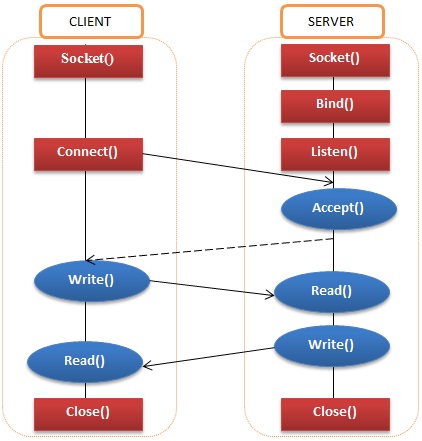
TCP thì dữ liệu được truyền theo giao thức UDP không được tin cậy, có thế không đúng

trình tự và lặp lại. Tuy nhiên vì nó không yêu cầu thiết lập kết nối không phải có

những cơ chế phức tạp nên tốc độ nhanh…ứng dụng cho các ứng dụng truyền

dữ liệu nhanh như chat, game…..

**TCP (Transmission Control Protocol - "Giao thức điều khiển truyền vận")** là một trong các giao thức cốt lõi của bộ giao thức TCP/IP. Sử dụng TCP, các ứng dụng trên các máy chủ được nối mạng có thể tạo các "kết nối" với nhau, mà qua đó chúng có thể trao đổi dữ liệu hoặc các gói tin. Giao thức này đảm bảo chuyển giao dữ liệu tới nơi nhận một cách đáng tin cậy và đúng thứ tự. TCP còn phân biệt giữa dữ liệu của nhiều ứng dụng (chẳng hạn, dịch vụ Web và dịch vụ thư điện tử) đồng thời chạy trên cùng một máy chủ.

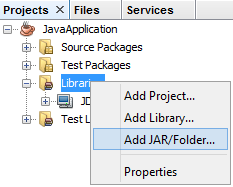
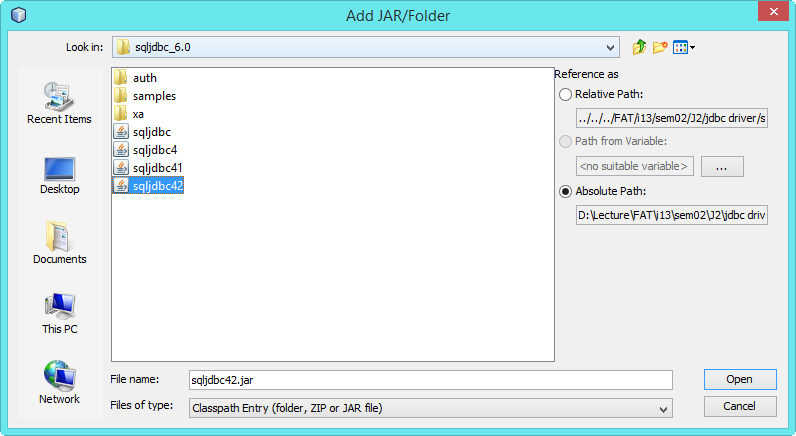


**IV.Kết nối sql server với java**

**Kết nối sql server với java** (Connect to Microsoft SQL Server via JDBC) sẽ hướng dẫn các bạn làm thế nào để có được trình điều khiển JDBC (JDBC Driver) và viết mã (code) cho việc kết nối đến cơ sở dữ liệu được cài đặt trên Microsoft SQL Server từ ứng dụng Java. Nội dung bài học bao gồm tải JDBC Driver, chuỗi kết nối cho Microsoft SQL Server (sql server jdbc connection string), thiết lập kết nối (establish connection) và chương trình mẫu.

**Kết nối sql server với java – Download Microsoft JDBC driver**

Tải [JDBC Driver](https://drive.google.com/file/d/0BzFICpYwyqoab3FTN3p0SkNaeUE/view?usp=sharing) và thêm sqljdbc42.jar vào Java project sử dụng NetBeans theo các bước sau

[](http://giasutinhoc.vn/wp-content/uploads/2016/04/cai-dat-jdbc-driver-5.png)[](http://giasutinhoc.vn/wp-content/uploads/2016/04/cai-dat-jdbc-driver-6.png)Chuột phải **Libraries ->** Chọn **Add JAR/Folder…**

Chỉ định thư mục chứa JDBC Driver và chọn tập tin **sqljdbc42.jar ->** chọn **Open**

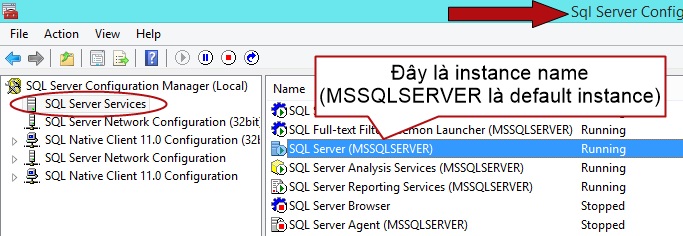
**Kết nối sql server với java – SQL Server JDBC connection string**

Là chuỗi được sử dụng để kết nối đến SQL Server. Mỗi một hệ quản trị cơ sở dữ liệu sẽ có một chuỗi kết nối khác nhau. Sau đây là chuỗi kết nối JDBC dành cho Microsoft SQL Server

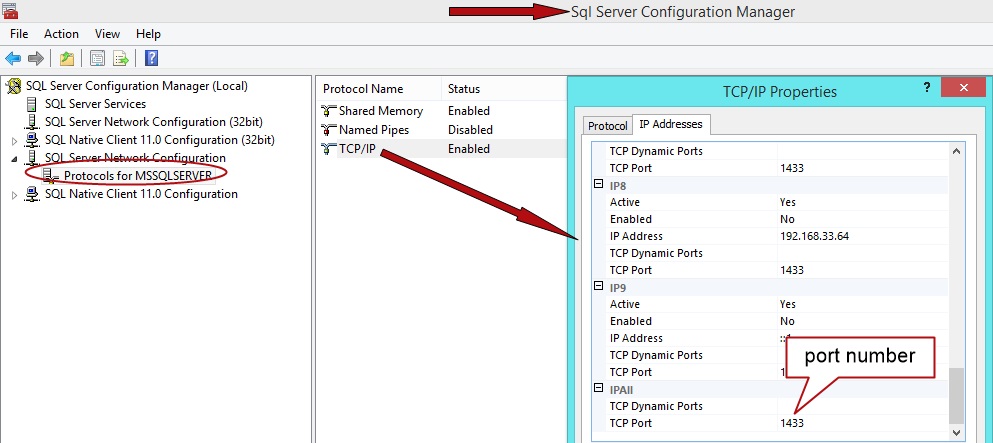
***jdbc:sqlserver://****[serverName[\instanceName][:portNumber]][;property=value[;property=value]]*

Trong đó

* serverName là tên máy chủ hoặc địa chỉ IP của máy đang cài đặt Microsoft SQL Server
* instanceName tên của một thể hiện để kết nối đến serverName. Nếu tham số này không được chỉ định thì default instance sẽ được sử dụng
* portNumber là cỗng ứng dụng của SQL Server, mặc định là 1433

[](http://giasutinhoc.vn/wp-content/uploads/2016/04/cai-dat-jdbc-driver-7.jpg)

* property=value chỉ định tên cơ sở dữ liệu, tên và mật khẩu đăng nhập vào SQL Server

[](http://giasutinhoc.vn/wp-content/uploads/2016/04/cai-dat-jdbc-driver-8.jpg)

*databaseName=EMPDB; user=sa; password=sa*

**Kết nối sql server với java – Thiết lập kết nối (establish connection)**

Để tạo một kết nối, gọi phương thức getConnection() của lớp DriverManager. Sau đây là đoạn code được sử dụng để kết nối đến localhost sử dụng user là sa, mật khẩu là sa và kết nối vào cơ sở dữ liệu tên EMPDB

**Kết nối sql server với java – Chương trình mẫu (Example program)**

package jdbc;

import java.sql.Connection;

import java.sql.DatabaseMetaData;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*\*

\*

\* @author giasutinhoc.vn

\*/

public class ConnectionTest {

public static void main(String[] args) {

try {

String dbURL = "jdbc:sqlserver://localhost;databaseName=EMPDB;user=sa;password=sa";

Connection conn = DriverManager.getConnection(dbURL);

if (conn != null) {

System.out.println("Connected");

DatabaseMetaData dm = (DatabaseMetaData) conn.getMetaData();

System.out.println("Driver name: " + dm.getDriverName());

System.out.println("Driver version: " + dm.getDriverVersion());

System.out.println("Product name: " + dm.getDatabaseProductName());

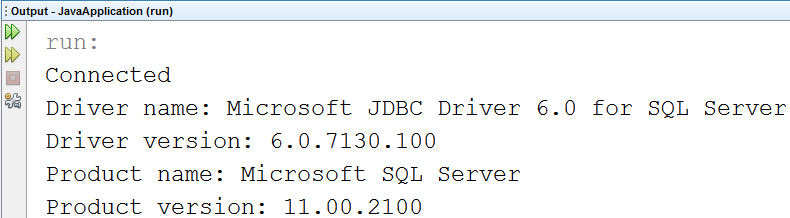
System.out.println("Product version: " + dm.getDatabaseProductVersion());

}

} catch (SQLException ex) {

System.err.println("Cannot connect database, " + ex);

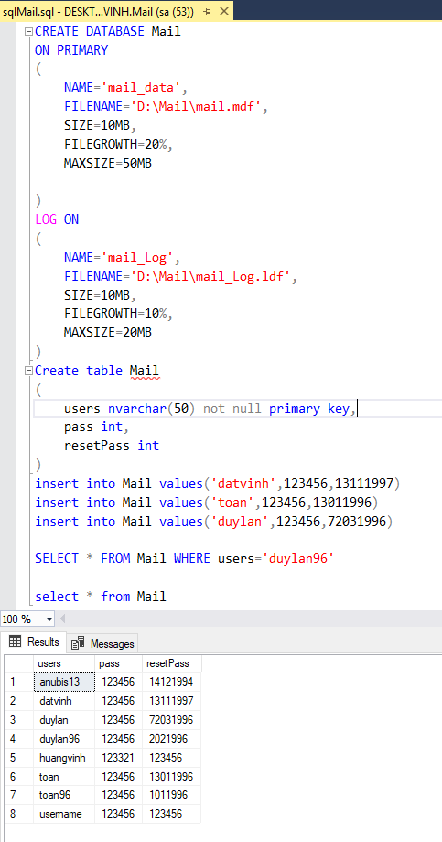
}}}

[](http://giasutinhoc.vn/wp-content/uploads/2016/04/cai-dat-jdbc-driver-15.png)Khi chạy chương trình trên, chúng ta sẽ nhận được kết quả sau

**V.DEMO**

CODE DEMO: Sử dụng thư viện sqljdbc4-2.0.jar

**CSDL:**



* 1. **Client :**
* **private** JButton login\_JButton = **new** JButton("Đăng nhập");
* **private** JButton logout\_JButton = **new** JButton("Đăng xuất");
* **private** JButton compose\_JButton = **new** JButton("Soạn thư");
* **private** JButton send\_JButton = **new** JButton("Gửi");
* **private** JButton send1\_JButton = **new** JButton("Gưi");
* **private** JButton save\_JButton = **new** JButton("Lưu");
* **private** JButton cancel\_JButton = **new** JButton("Hủy");
* **private** JButton delete\_JButton = **new** JButton("Xóa");
* **private** JButton reply\_JButton = **new** JButton("Trả lời");
* **private** JButton forward\_JButton = **new** JButton("Chuyển tiếp");
* **private** JButton resetPassForm\_JButton = **new** JButton("Lấy lại mật khẩu");
* **private** JButton form\_Register\_JButton = **new** JButton("Tạo mới tàikhoản");
* **private** JButton form\_Login\_JButton = **new** JButton("Quay lại Đăng nhập");
* **private** JButton form\_Login1\_JButton = **new** JButton("Quay lại Đăng nhập");
* **private** JButton resetPass\_JButton = **new** JButton("Ok");
* **private** JButton register\_JButton = **new** JButton("Đăng ký");
* **private** JButton changePass\_JButton = **new** JButton("Đổi mật khẩu");
* **private** Label inbox\_Label = **new** Label(" Hộp thư đến");
* **private** Label outbox\_Label = **new** Label(" Thư đã gửi");
* **private** Label drafts\_Label = **new** Label(" Thư nháp");
* **private** Label trash\_Label = **new** Label(" Thùng rác");
* **private** Label contact\_Label = **new** Label(" Liên hệ?");

**2.Database :**

* **import** java.sql.Connection;
* **import** java.sql.DriverManager;
* **import** java.sql.PreparedStatement;
* **import** java.sql.ResultSet;
* **import** java.sql.SQLException;
* **public** **class** Database {
* **public** **static** Connection *con*= **null**;
* PreparedStatement stmt=**null**;
* **public** Database() {
* connect();
* }
* **public** **void** connect() {
* String url="jdbc:sqlserver://localhost:62881;databasename=Mail";
* String user= "sa";
* String password ="sapassword";
* **try** {
* *con*=DriverManager.*getConnection*(url, user, password);
* }**catch**(SQLException e) {
* e.printStackTrace();
* }
* }

**3.Server :**

**public** **class** Server {

DatagramSocket server\_DatagramSocket;

InetAddress server\_ip;

**final** **int** server\_port = 2222;

String server\_sendData;

String server\_receivedData;

**final** **int** size\_of\_packet = 4096;

Database db = **new** Database();

**private** JTextArea listMail = **new** JTextArea(20,15);

**private** Label listMail\_Array[] = **new** Label[15];

**private** String listMailTime\_Array[];

**private** String listMailView\_Array[];

**private** **int** listMailStatus\_Array[];

**private** **int** num\_inbox=0;

**public** Server() {

listMailTime\_Array = **new** String[100];

listMailView\_Array = **new** String[100];

listMailStatus\_Array = **new** **int**[100];

**try** {

server\_ip = InetAddress.*getByName*("localhost");

server\_DatagramSocket = **new** DatagramSocket(server\_port,server\_ip);

System.***out***.println("Server is running");

JOptionPane.*showMessageDialog*(**new** Panel(),"Server đã chạy. Bấm ok để tiếp tục","Thông báo",JOptionPane.***PLAIN\_MESSAGE***);

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

// Nhận gói tin từ Client bất kỳ

**byte**[] server\_receivedByte = **new** **byte**[size\_of\_packet];

DatagramPacket server\_receivedPacket = **new** DatagramPacket(server\_receivedByte, server\_receivedByte.length);

server\_DatagramSocket.receive(server\_receivedPacket);

server\_receivedData = **new** String(server\_receivedPacket.getData()).trim();

// Xử lý gói tin

server\_sendData = process(server\_receivedData);

//System.out.printf(server\_sendData + " " +server\_receivedData + "\n");

// Hồi đáp cho Client tương ứng

InetAddress client\_ip = server\_receivedPacket.getAddress();

**int** client\_port = server\_receivedPacket.getPort();

**byte**[] server\_sendByte = **new** **byte**[size\_of\_packet];

server\_sendByte = server\_sendData.getBytes();

DatagramPacket server\_sendPacket = **new** DatagramPacket(server\_sendByte, server\_sendByte.length,client\_ip,client\_port);

server\_DatagramSocket.send(server\_sendPacket);

}

}

**catch** (Exception e) {

JOptionPane.*showMessageDialog*(**new** Panel(),"Server đã chạy rồi","Cảnh báo",JOptionPane.***WARNING\_MESSAGE***);

System.*exit*(0);

e.printStackTrace();

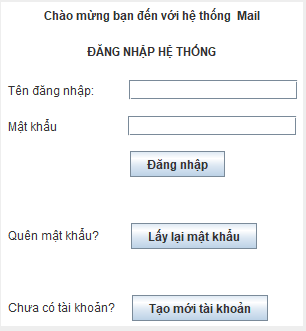
}

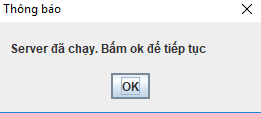
}

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

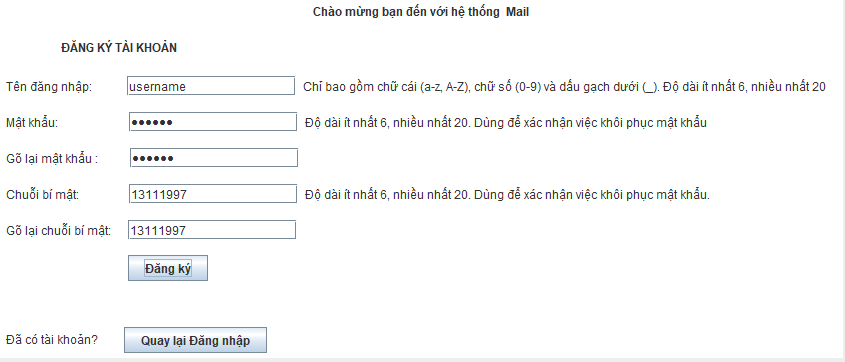
**new** Server();

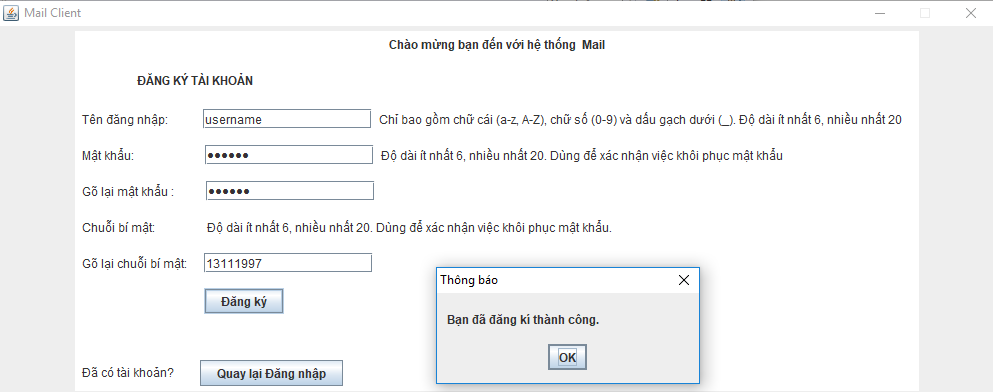
}

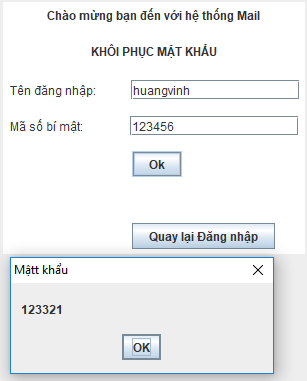
1. Đăng nhập :

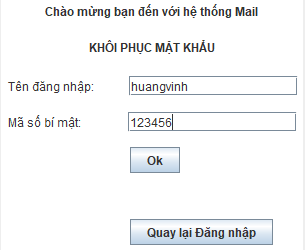


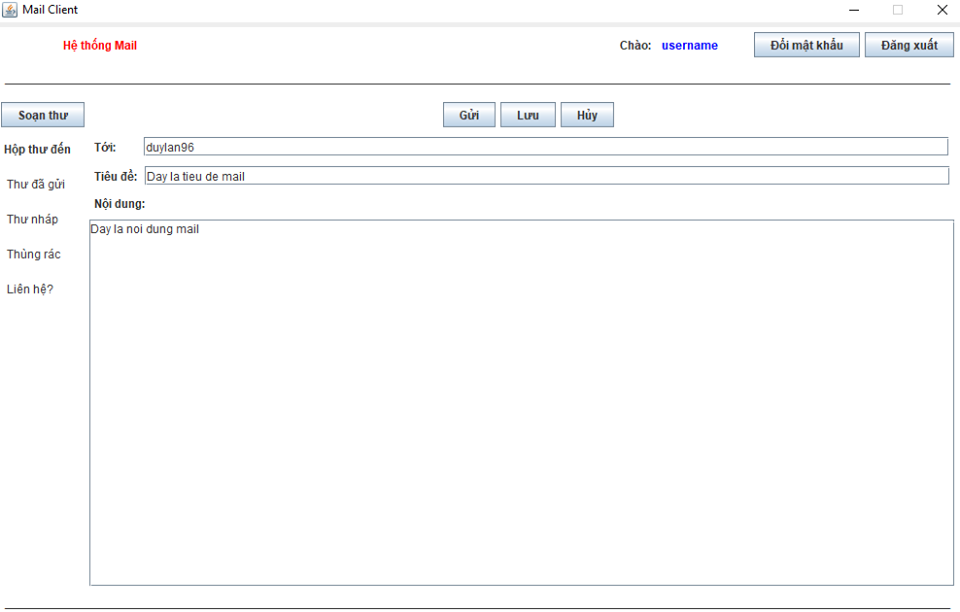
1. Đăng kí :

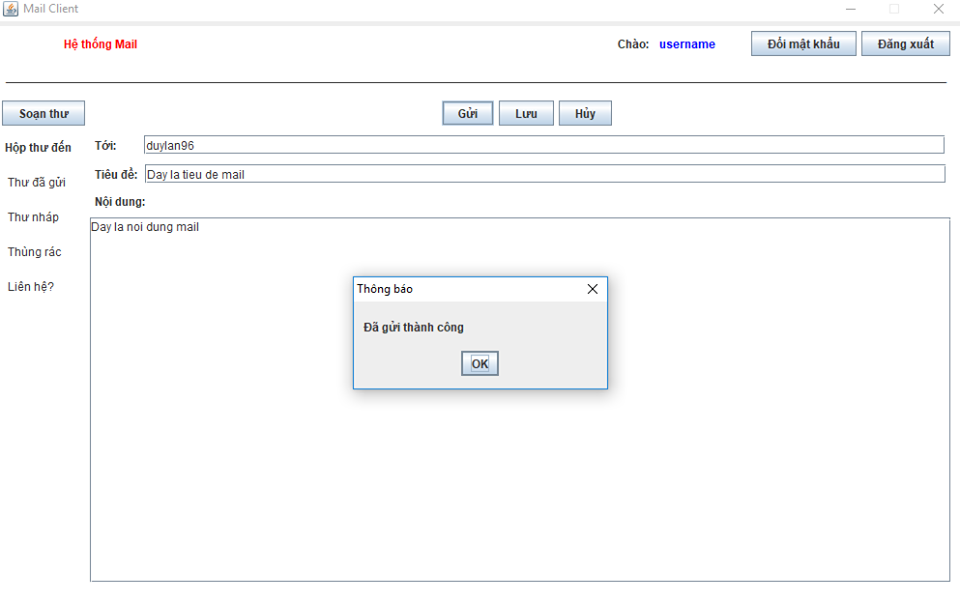




1. Lấy lại mật khẩu :



4.Gửi mail:



1. Nhận mail :

**Tài liệu tham khảo:**

<https://vi.wikipedia.org/wiki/Wikipedia>

<http://giasutinhoc.vn/lap-trinh-co-so-du-lieu-voi-jdbc/ket-noi-sql-server-voi-java-bai-3/>