**Hoc vien:**

**Client:**package hocvien;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.rmi.registry.LocateRegistry;

import java.rmi.registry.Registry;

public class Client extends JFrame {

private JTextField studentIDField;

private JTextArea resultArea;

public Client () {

super("Student Information Lookup");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(400, 200);

setLocationRelativeTo(null);

JPanel panel = new JPanel();

panel.setLayout(new GridLayout(3, 1));

JLabel label = new JLabel("Enter Student ID:");

panel.add(label);

studentIDField = new JTextField();

panel.add(studentIDField);

JButton searchButton = new JButton("Search");

searchButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

searchStudentInfo();

}

});

panel.add(searchButton);

resultArea = new JTextArea();

resultArea.setEditable(false);

JScrollPane scrollPane = new JScrollPane(resultArea);

panel.add(scrollPane);

add(panel);

setVisible(true);

}

private void searchStudentInfo() {

try {

Registry registry = LocateRegistry.getRegistry("localhost", 1099);

StudentInfoProcessor processor = (StudentInfoProcessor) registry.lookup("StudentInfoProcessor");

String studentID = studentIDField.getText();

String result = processor.getStudentInfo(studentID);

resultArea.setText(result);

} catch (Exception e) {

resultArea.setText("Error: " + e.getMessage());

}

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

new Client();

}

});

}

}

**Server**package hocvien;

import java.rmi.registry.LocateRegistry;

import java.rmi.registry.Registry;

public class Server {

public static void main(String[] args) {

try {

StudentInfoProcessor processor = new StudentInfoProcessorImpl();

Registry registry = LocateRegistry.createRegistry(1099);

registry.rebind("StudentInfoProcessor", processor);

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

} catch (Exception e) {

System.err.println("Server exception: " + e.toString());

e.printStackTrace();

}

}

}

**interface StudentInfoProcessor**

package hocvien;

import java.rmi.Remote;

import java.rmi.RemoteException;

public interface StudentInfoProcessor extends Remote {

String getStudentInfo(String studentID) throws RemoteException;

}

**StudentInfoProcessorImpl**

package hocvien;

import java.rmi.RemoteException;

import java.rmi.server.UnicastRemoteObject;

import java.util.HashMap;

public class StudentInfoProcessorImpl extends UnicastRemoteObject implements StudentInfoProcessor {

private HashMap<String, String> studentInfoMap;

protected StudentInfoProcessorImpl() throws RemoteException {

super();

studentInfoMap = new HashMap<>();

studentInfoMap.put("123", "Student ID: 123\nName: John Doe\nMajor: Computer Science");

studentInfoMap.put("456", "Student ID: 456\nName: Jane Smith\nMajor: Biology");

}

@Override

public String getStudentInfo(String studentID) throws RemoteException {

// Lấy thông tin của học viên từ map

String info = studentInfoMap.get(studentID);

if (info != null) {

return info;

} else {

return "Không tìm thấy thông tin cho học viên có mã số " + studentID;

}

}

}