Bean-Managed Persistence Entity Beans

Tạo Database có tên BMP21. Tạo bảng Sinhvien có cấu trúc như hình

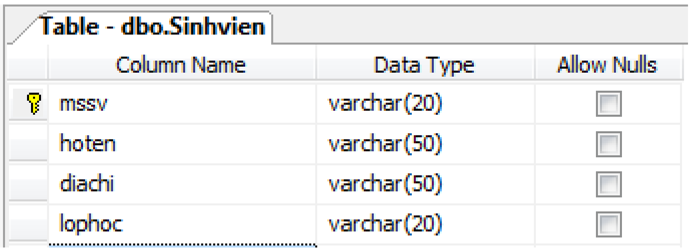


Table - dbo.Sinhvien

Column Name	Data Type	Allow Nulls
mssv	varchar(20)	□
hoten	varchar(50)	□
diachi	varchar(50)	□
lophoc	varchar(20)	□

Dữ liệu thử

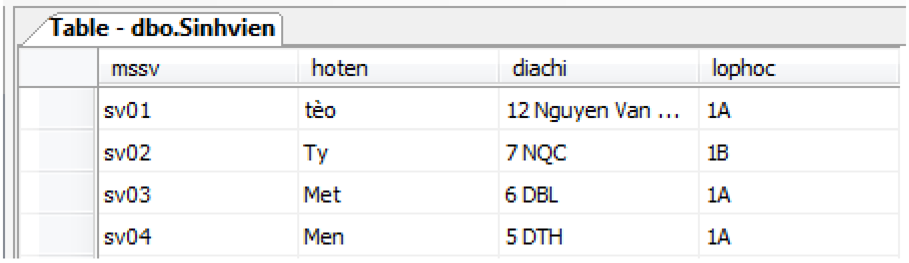
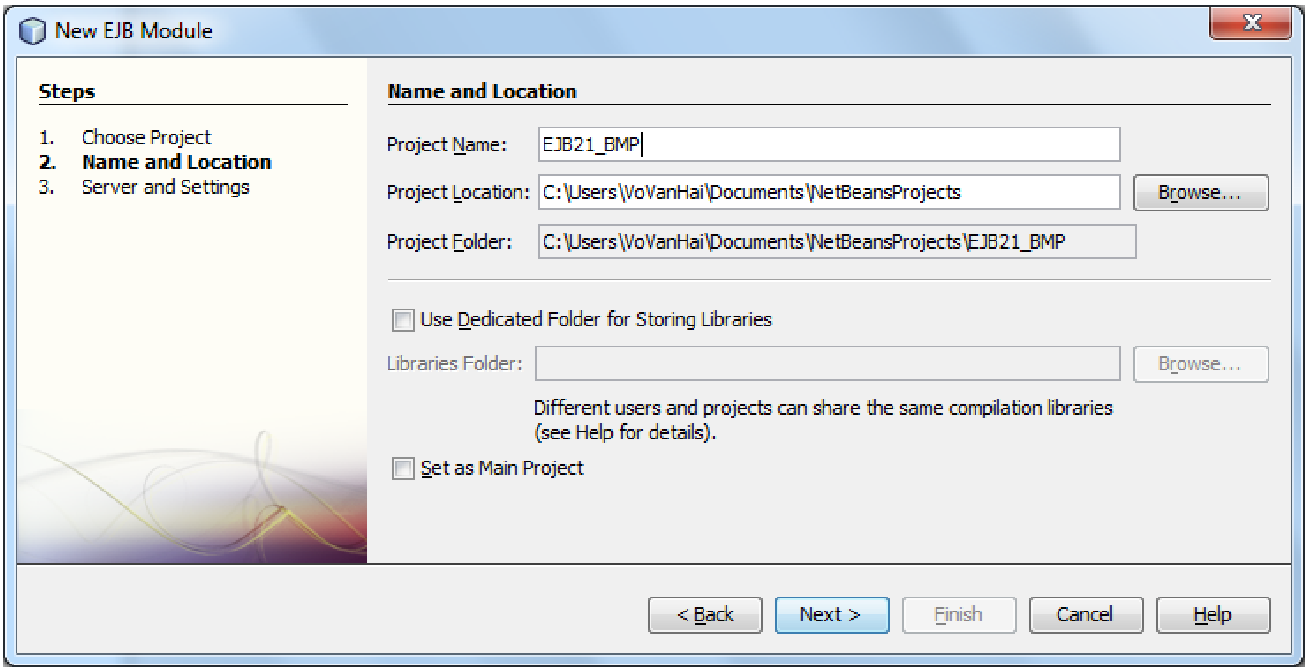


Table - dbo.Sinhvien

mssv	hoten	diachi	lophoc
sv01	tèo	12 Nguyen Van ...	1A
sv02	Ty	7 NQC	1B
sv03	Met	6 DBL	1A
sv04	Men	5 DTH	1A

Phần 1: EJB

Tạo project trong Netbeans



Name and Location

Project Name: EJB21_BMP

Project Location: C:\Users\VoVanHai\Documents\NetBeansProjects

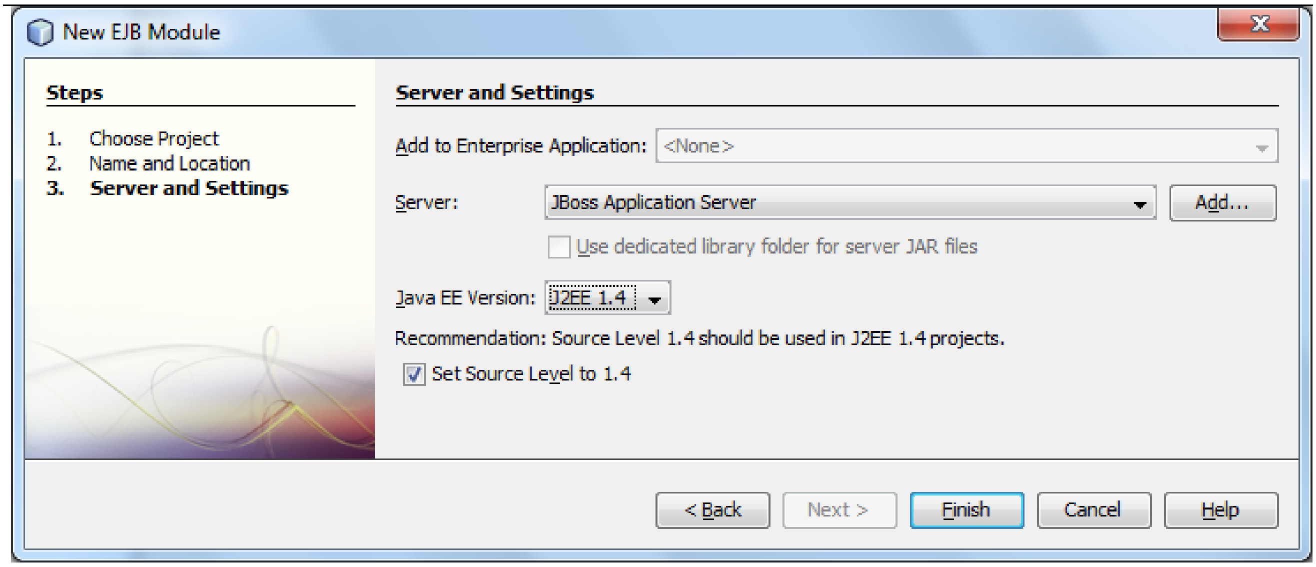
Project Folder: C:\Users\VoVanHai\Documents\NetBeansProjects\EJB21_BMP

□ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

□ Set as Main Project



New EJB Module

Steps

1. Choose Project
2. Name and Location
3. **Server and Settings**

Server and Settings

Add to Enterprise Application: <None>

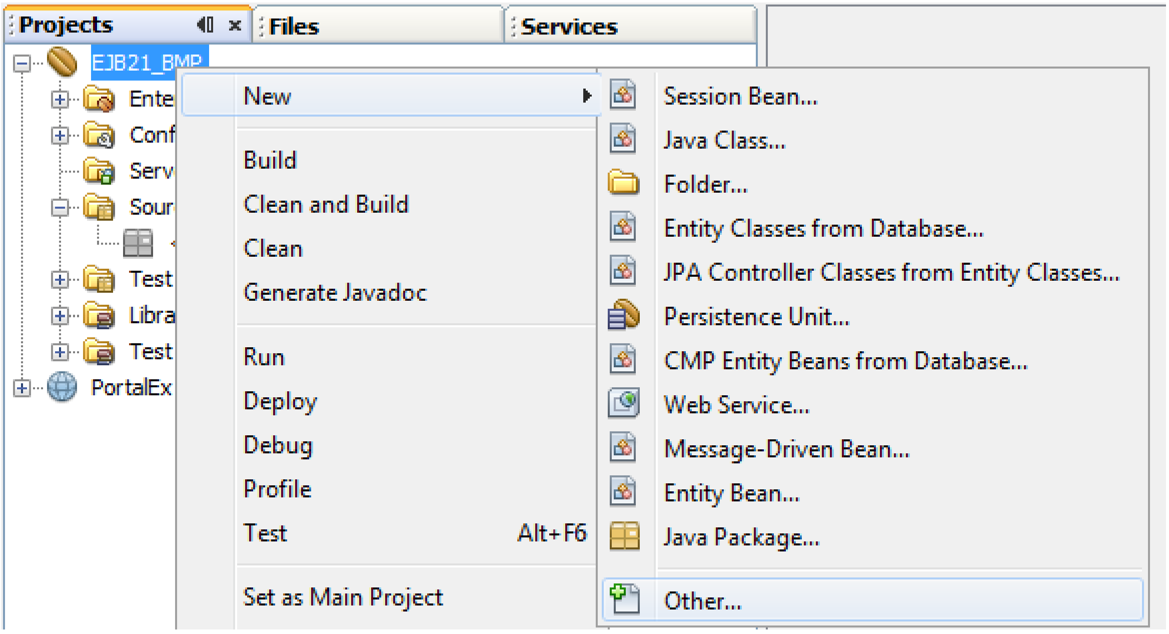
Server: JBoss Application Server

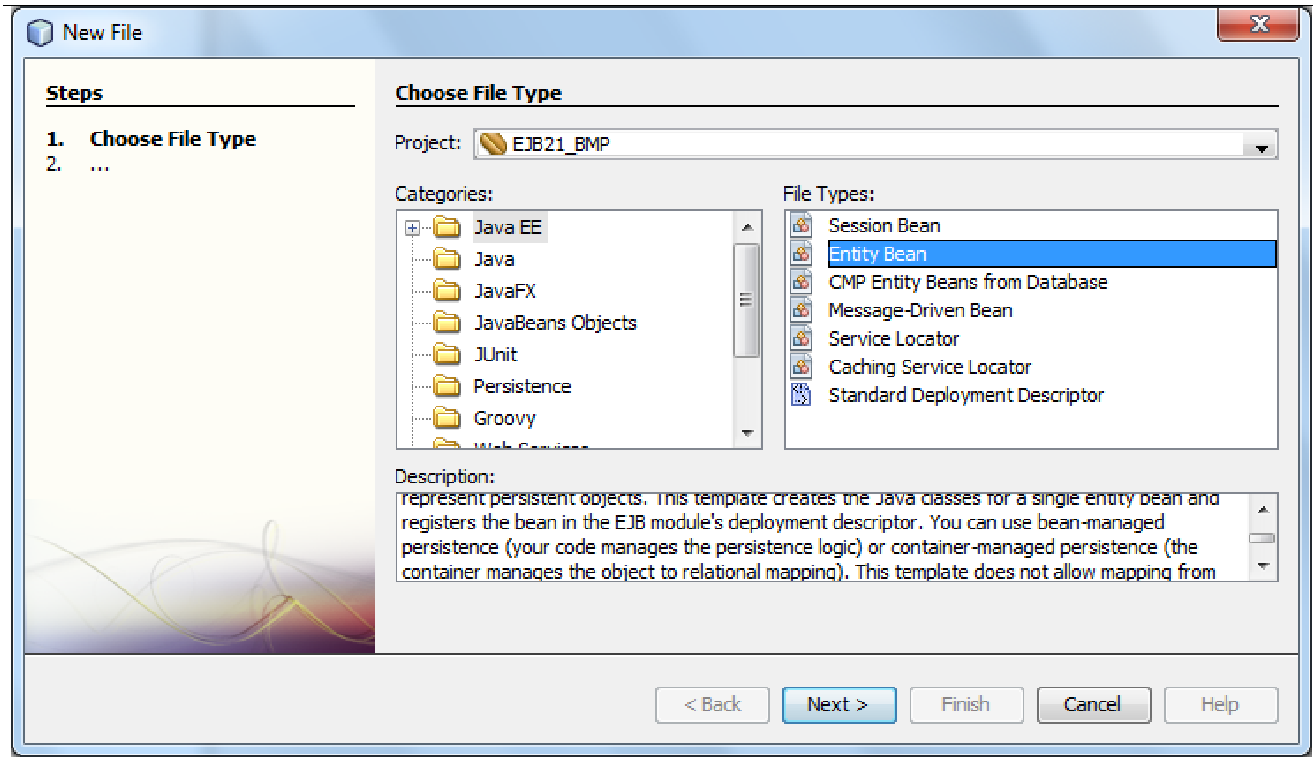
□ Use dedicated library folder for server JAR files

Java EE Version: J2EE 1.4

Recommendation: Source Level 1.4 should be used in J2EE 1.4 projects.

☑ Set Source Level to 1.4





Choose File Type

Project: EJB21_BMP

Categories:

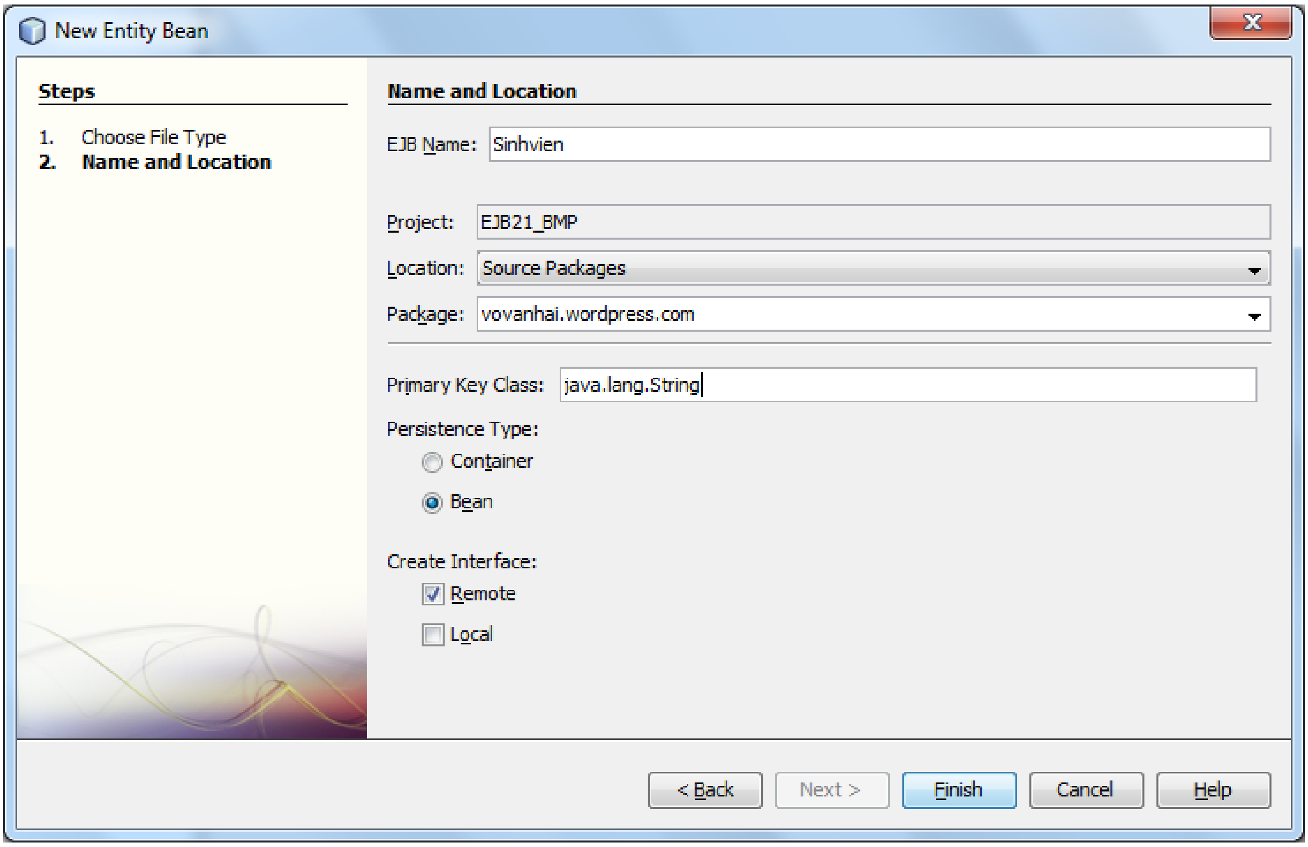
- Java EE
- Java
- JavaFX
- JavaBeans Objects
- JUnit
- Persistence
- Groovy
- Web Services

File Types:

- Session Bean
- Entity Bean
- CMP Entity Beans from Database
- Message-Driven Bean
- Service Locator
- Caching Service Locator
- Standard Deployment Descriptor

Description:

represent persistent objects. This template creates the Java classes for a single entity bean and registers the bean in the EJB module's deployment descriptor. You can use bean-managed persistence (your code manages the persistence logic) or container-managed persistence (the container manages the object to relational mapping). This template does not allow mapping from



Name and Location

EJB Name: Sinhvien

Project: EJB21_BMP

Location: Source Packages

Package: vovanhai.wordpress.com

Primary Key Class: java.lang.String

Persistence Type:

○ Container

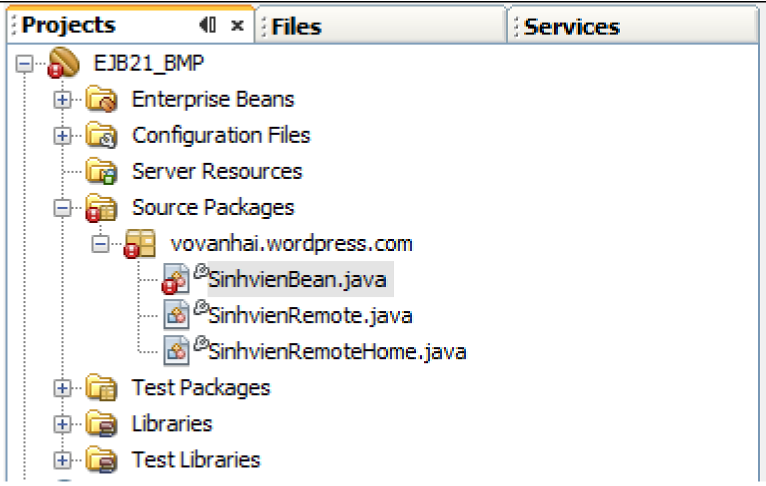
● Bean

Create Interface:

☑ Remote

□ Local

Kết quả ta được

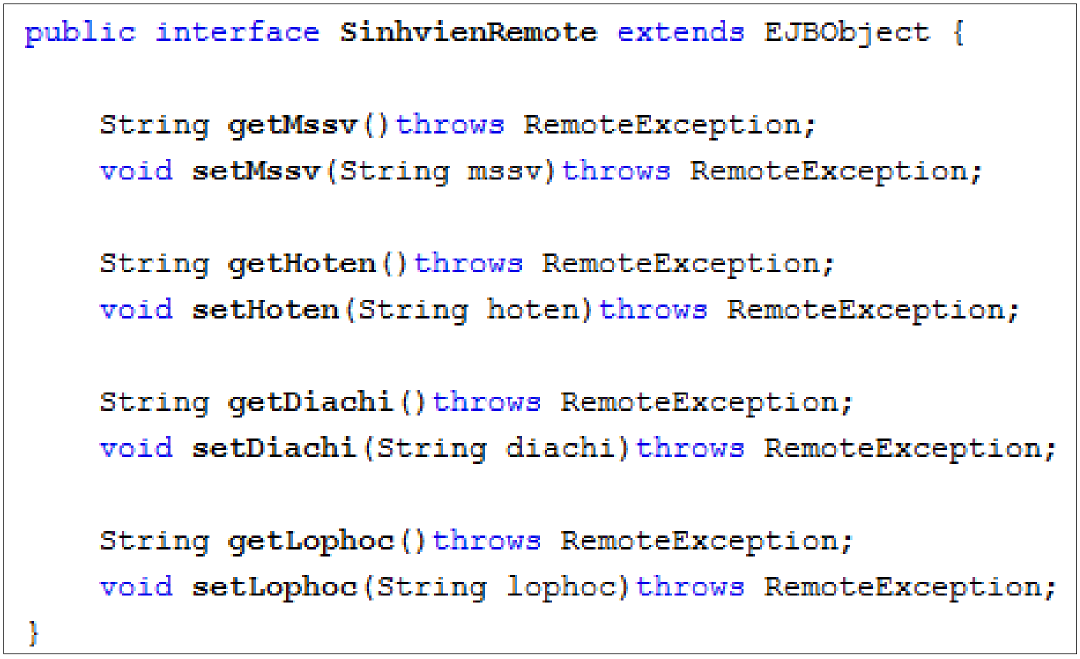


Coding

Lớp **SinhvienRemoteHome.java**

```
public interface SinhvienRemoteHome extends EJBHome {

    SinhvienRemote create(String mssv, String hoten, String diachi,
            String lophoc)throws CreateException, RemoteException;

    SinhvienRemote findByPrimaryKey(String mssv)
            throws FinderException, RemoteException;

    Enumeration findByHoten(String hoten)
            throws FinderException, RemoteException;

    Enumeration findByLophoc(String lophoc)
            throws FinderException, RemoteException;
}
```

Lớp **SinhvienRemote.java.** Chú ý ở đây tương ứng với 1 field trong bảng CSDL ta có 1 cặp set/get trong interface này.



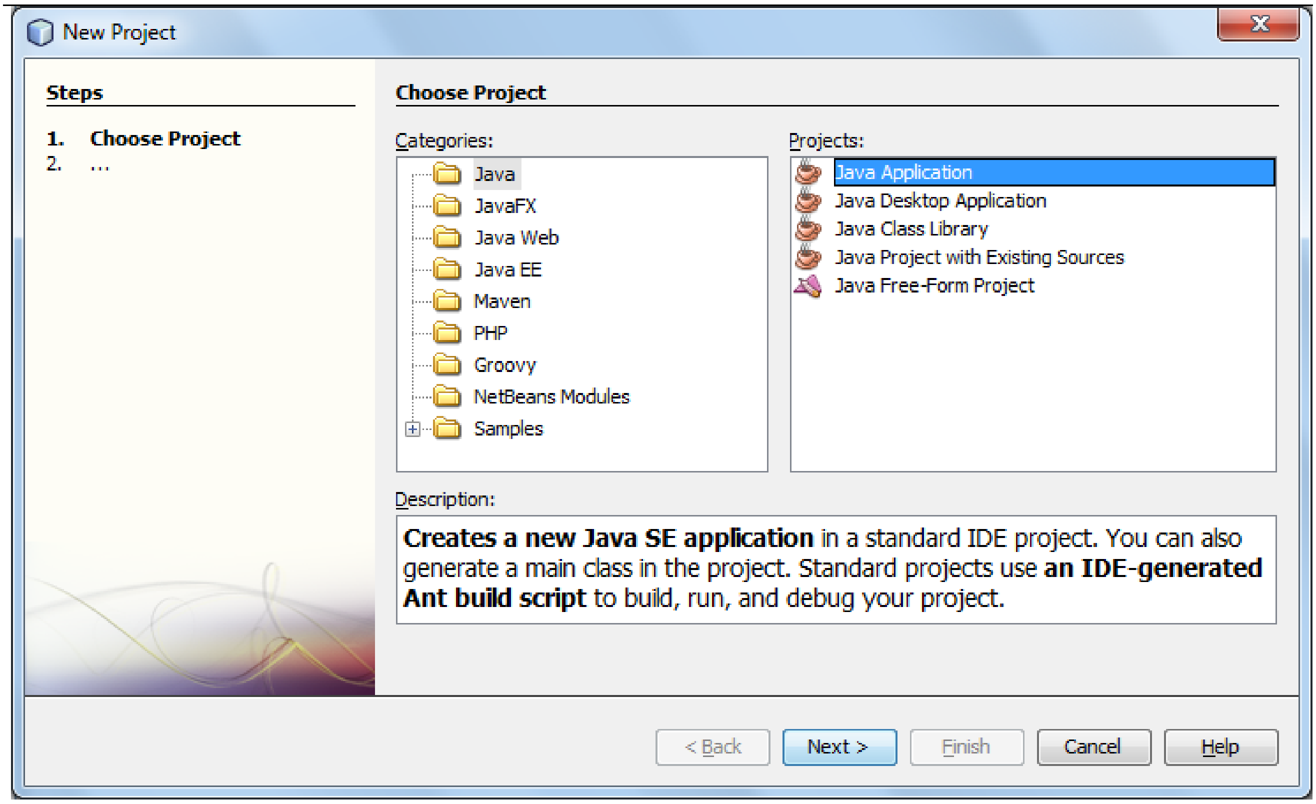
```
public interface SinhvienRemote extends EJBObject {

    String getMssv()throws RemoteException;
    void setMssv(String mssv)throws RemoteException;

    String getHoten()throws RemoteException;
    void setHoten(String hoten)throws RemoteException;

    String getDiachi()throws RemoteException;
    void setDiachi(String diachi)throws RemoteException;

    String getLophoc()throws RemoteException;
    void setLophoc(String lophoc)throws RemoteException;
}
```

Lớp SinhvienBean.java

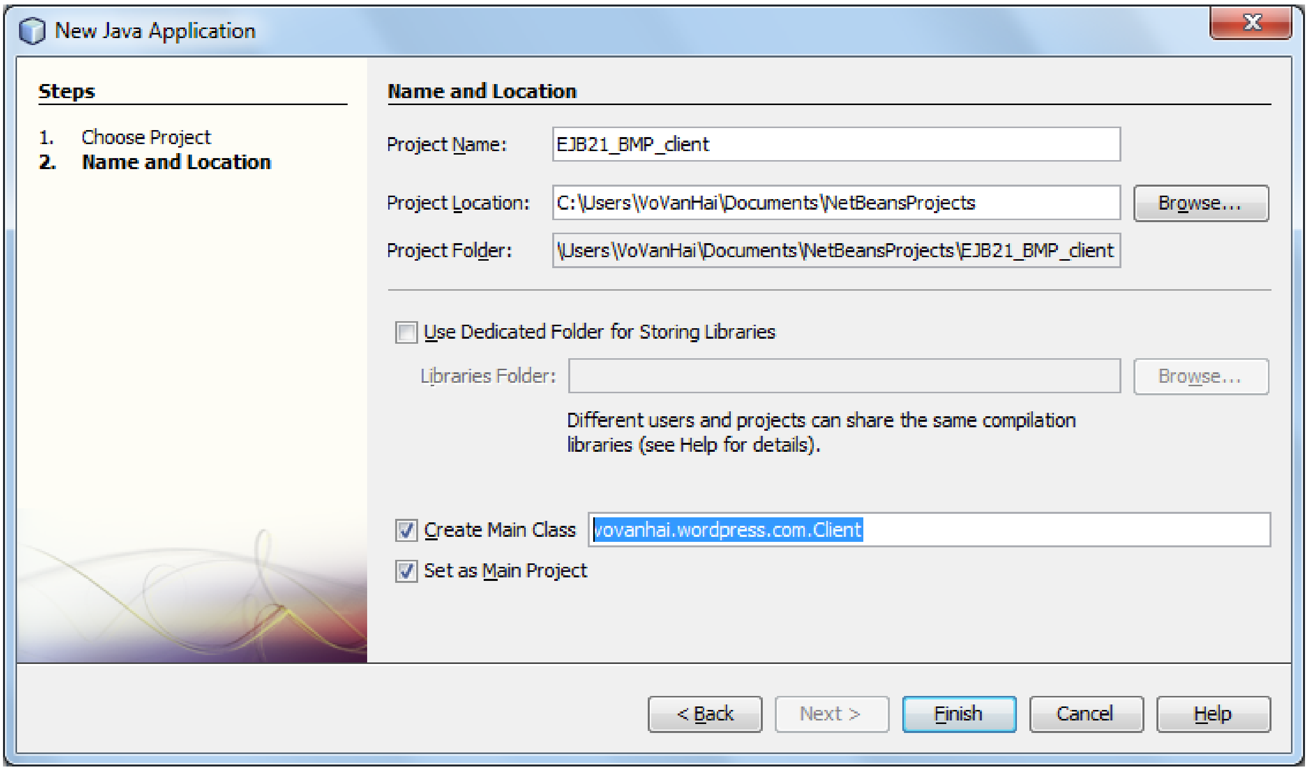
```
package vovanhai.wordpress.com;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Enumeration;
import java.util.Vector;
import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.ejb.EntityBean;
import javax.ejb.EntityContext;
import javax.ejb.FinderException;
import javax.ejb.ObjectNotFoundException;

/**
 *
 * @author VoVanHai
 */
public class SinhvienBean implements EntityBean {

    private EntityContext context;
    private String mssv;
    private String hoten;
    private String diachi;
    private String lophoc;

    public String getDiachi() {
        return diachi;
    }

    public void setDiachi(String diachi) {
        this.diachi = diachi;
    }

    public String getHoten() {
        return hoten;
    }

    public void setHoten(String hoten) {
        this.hoten = hoten;
    }

    public String getLophoc() {
        return lophoc;
    }

    public void setLophoc(String lophoc) {
        this.lophoc = lophoc;
    }

    public String getMssv() {
        return mssv;
    }

    public void setMssv(String mssv) {
        this.mssv = mssv;
    }

    public void setEntityContext(EntityContext aContext) {
        context = aContext;
    }

    public String ejbCreate(String mssv, String hoten, String diachi,
            String lophoc) throws CreateException {
        try {
            this.mssv = mssv;
            this.hoten = hoten;
            this.diachi = diachi;
            this.lophoc = lophoc;
            insertStudent(mssv, hoten, diachi, lophoc);
        } catch (Exception ex) {
            throw new EJBException("ejbCreate: " + ex.getMessage());
        }
        return mssv;
    }

    public void ejbPostCreate(String mssv, String hoten, String diachi,
            String lophoc) {
    }

    public void ejbActivate() {
        this.mssv = (String) context.getPrimaryKey();
    }

    public void ejbPassivate() {
        this.mssv = null;
    }

    public void ejbRemove() {
        try {
            deleteStudent(mssv);
        } catch (Exception ex) {
            throw new EJBException("ejbRemove: " + ex.getMessage());
        }
    }

    public void unsetEntityContext() {
        context = null;
    }

    public void ejbLoad() {
        try {
            loadStudent();
        } catch (Exception ex) {
            throw new EJBException("ejbLoad: " + ex.getMessage());
        }
    }

    public void ejbStore() {
        try {
            storeStudent();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public String ejbFindByPrimaryKey(String mssv) throws FinderException
{
        boolean result = false;
        try {
            result = findByPrimaryKey(mssv);
        } catch (Exception ex) {
            throw new EJBException("ejbFindByPrimaryKey" +
ex.getMessage());
        }
        if (!result) {
            throw new ObjectNotFoundException("row for id " + mssv + "
not found!");
        }
        return mssv;
    }

    public Enumeration ejbFindByHoten(String hoten) throws
FinderException {
        Enumeration a = null;
        try {
            a = findByHoten(hoten);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return a;
    }

    public Enumeration ejbFindByLophoc(String lophoc) throws
FinderException {
        Enumeration a = null;
        try {
            a = findByLophoc(lophoc);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return a;
    }
```

```
//========================================================================
=

    private Connection makeConnection() throws Exception {
        Connection con = null;
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        String url = 
"jdbc:sqlserver://localhost:1433;databaseName=BMP21";
        con = DriverManager.getConnection(url,"sa","");
        return con;
    }

    private void insertStudent(String mssv, String hoten, String diachi,
            String lophoc) throws Exception {
        Connection con = makeConnection();
        String sql = "insert into Sinhvien values(?,?,?,?)";
        PreparedStatement pSt = con.prepareStatement(sql);
        pSt.setString(1, mssv);
        pSt.setString(2, hoten);
        pSt.setString(3, diachi);
        pSt.setString(4, lophoc);
        pSt.executeUpdate();
        con.close();
    }

    private void storeStudent() throws Exception {
        Connection con = makeConnection();
        String sql = "update Sinhvien set hoten=?,diachi=?,lophoc=? where 
mssv=?";
        PreparedStatement pSt = con.prepareStatement(sql);
        pSt.setString(1, mssv);
        pSt.setString(2, hoten);
        pSt.setString(3, diachi);
        pSt.setString(4, lophoc);
        pSt.executeUpdate();
        con.close();
    }

    private void deleteStudent(String mssv) throws Exception {
        Connection con = makeConnection();
        String sql = "delete from Sinhvien where mssv='" + mssv + "'";
        con.createStatement().executeUpdate(sql);
        con.close();
    }

    private void loadStudent() throws Exception {
        Connection con = makeConnection();
        String sql = "select * from Sinhvien where mssv='" + mssv + "'";
        ResultSet rs = con.createStatement().executeQuery(sql);
        if (rs.next()) {
            this.hoten = rs.getString("hoten");
            this.diachi = rs.getString("diachi");
            this.lophoc = rs.getString("lophoc");
        } else {
            throw new Exception("Row for student id:" + mssv + " not 
found in database");
        }
    }
    private boolean findByPrimaryKey(String mssv) throws Exception {
        Connection con = makeConnection();
        String sql = "select * from Sinhvien where mssv='" + mssv + "'";
        ResultSet rs = con.createStatement().executeQuery(sql);
        boolean ret = rs.next();
        con.close();
        return ret;
    }

    private Enumeration findByHoten(String hoten) throws Exception {
        Vector vec = new Vector();
        Connection con = makeConnection();
        String sql = "select * from Sinhvien where hoten='" + hoten +
"'";
        ResultSet rs = con.createStatement().executeQuery(sql);
        while (rs.next()) {
            vec.add(rs.getString("mssv"));
        }
        con.close();
        return vec.elements();
    }

    private Enumeration findByLophoc(String lophoc) throws Exception {
        Vector vec = new Vector();
        Connection con = makeConnection();
        String sql = "select * from Sinhvien where lophoc='" + lophoc +
"'";
        ResultSet rs = con.createStatement().executeQuery(sql);
        while (rs.next()) {
            vec.add(rs.getString("mssv"));
        }
        con.close();
        return vec.elements();
    }
}
```

Phần 2: Client



Creates a new Java SE application in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.



Project Name: EJB21_BMP_client

Project Location: C:\Users\VoVanHai\Documents\NetBeansProjects

Project Folder: \Users\VoVanHai\Documents\NetBeansProjects\EJB21_BMP_client

□ Use Dedicated Folder for Storing Libraries

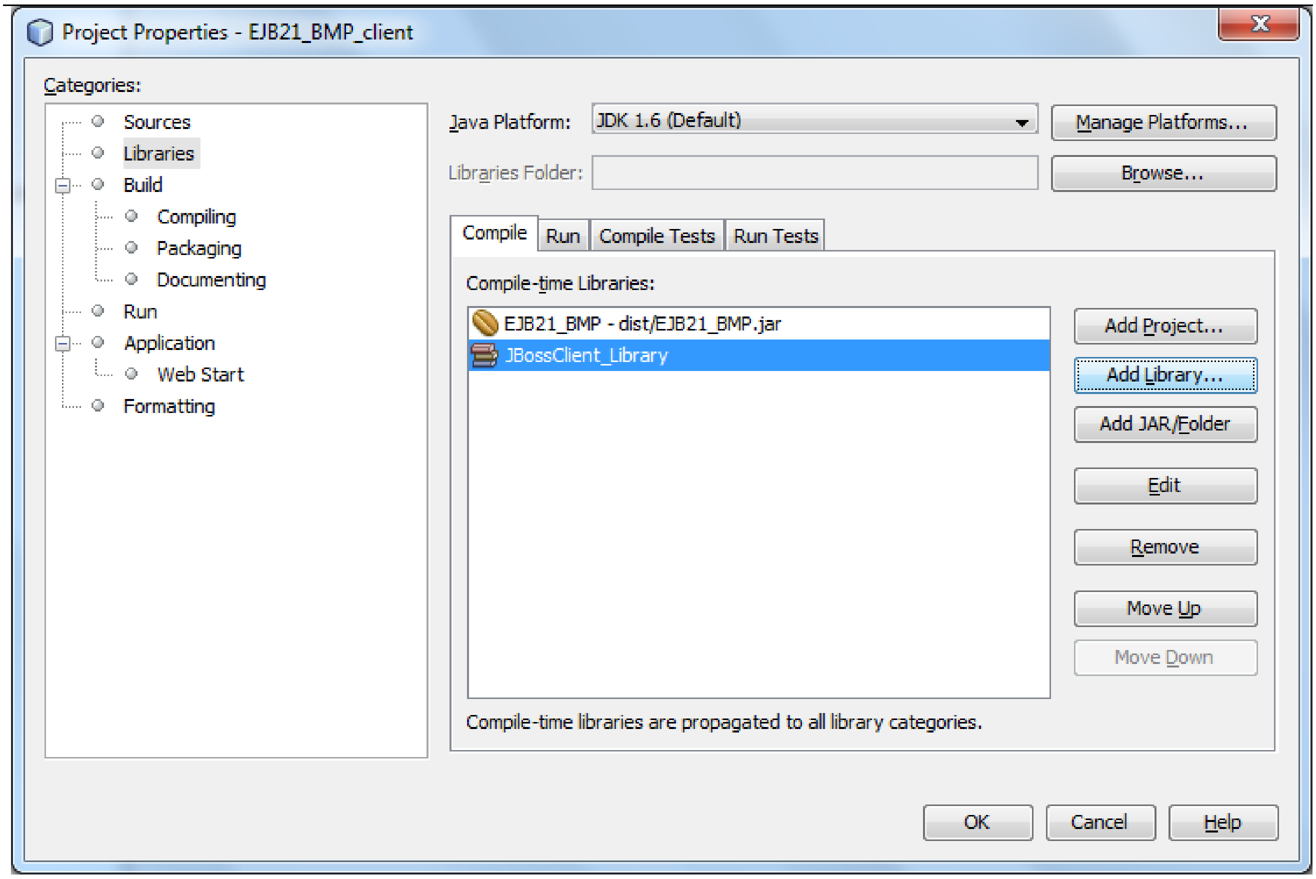
Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

☑ Create Main Class vovanhai.wordpress.com.Client

☑ Set as Main Project

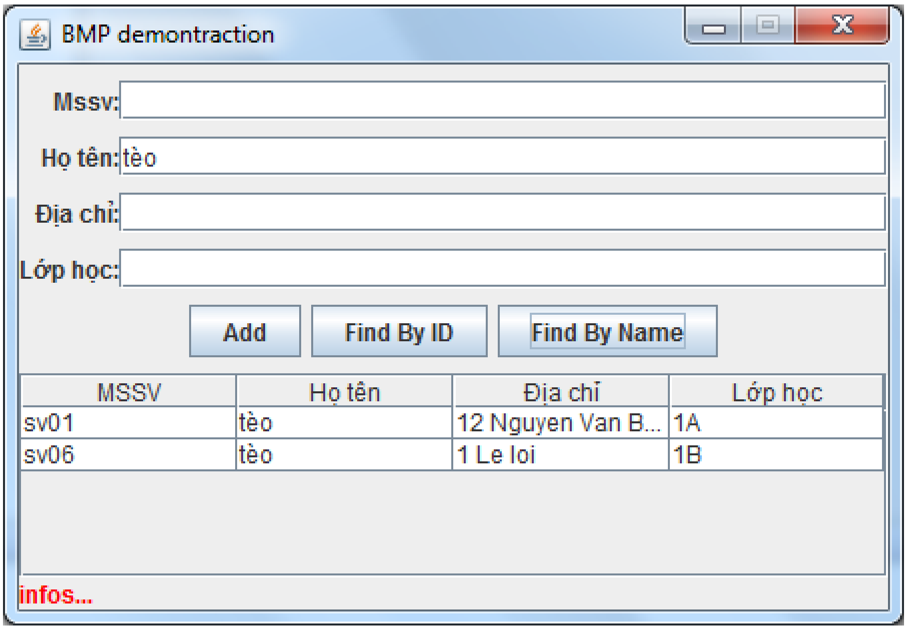
Thêm tham chiếu



Code cho lớp client

```
package vovanhai.wordpress.com;

/**
 *
 * @author VoVanHai
 */
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Enumeration;

import javax.naming.Context;
import javax.naming.InitialContext;
import javax.rmi.PortableRemoteObject;
import javax.swing.Box;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;

public class Client extends JFrame implements ActionListener {
    private JTextField tfMSSV, tfHoten, tfDiachi, tfLophoc;
    private JButton btAdd, btFindById, btFindByName;
    private DefaultTableModel dtm;
    private JTable table;
    private SinhvienRemoteHome SinhvienHome;
    private JLabel lblStatus;

    public Client() {
        super("BMP demontraction");
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setSize(440, 300);
        setResizable(false);

        Box b = Box.createVerticalBox();
        Box b1 = Box.createHorizontalBox();
        Box b2 = Box.createHorizontalBox();
        Box b3 = Box.createHorizontalBox();
        Box b4 = Box.createHorizontalBox();
        Box b5 = Box.createHorizontalBox();
        b.add(Box.createVerticalStrut(8));
        b.add(b1);
        b.add(Box.createVerticalStrut(8));
        b.add(b2);
        b.add(Box.createVerticalStrut(8));
        b.add(b3);
        b.add(Box.createVerticalStrut(8));
        b.add(b4);
        b.add(Box.createVerticalStrut(8));
        b.add(b5);
        b.add(Box.createVerticalStrut(8));

        this.add(b, BorderLayout.NORTH);
        JLabel l1, l2, l3, l4;
        b1.add(l1 = new JLabel("Mssv:", JLabel.RIGHT));
        b1.add(tfMSSV = new JTextField(20));
        b2.add(l2 = new JLabel("Họ tên:", JLabel.RIGHT));
        b2.add(tfHoten = new JTextField(20));

        b3.add(l3 = new JLabel("Địa chỉ:", JLabel.RIGHT));
        b3.add(tfDiachi = new JTextField(20));

        b4.add(l4 = new JLabel("Lớp học:", JLabel.RIGHT));
        b4.add(tfLophoc = new JTextField(20));

        l2.setPreferredSize(l4.getPreferredSize());
        l1.setPreferredSize(l4.getPreferredSize());
        l3.setPreferredSize(l4.getPreferredSize());


        b5.add(btAdd = new JButton("Add"));
        b5.add(Box.createHorizontalStrut(5));
        b5.add(btFindById = new JButton("Find By ID"));
        b5.add(Box.createHorizontalStrut(5));
        b5.add(btFindByName = new JButton("Find By Name"));

        btAdd.addActionListener(this);
        btFindById.addActionListener(this);
        btFindByName.addActionListener(this);

        lblStatus = new JLabel("infos...");
        lblStatus.setForeground(Color.red);
        this.add(lblStatus, BorderLayout.SOUTH);
        createTable();
        DoLook();//lookup
    }

    private void DoLook() {
        try {
            System.setProperty("java.naming.factory.initial",
"org.jnp.interfaces.NamingContextFactory");
            System.setProperty("java.naming.provider.url",
"localhost:1099");
            System.setProperty("java.naming.factory.url.pkgs",
"org.jboss.naming");
            Context ctx = new InitialContext();
            Object ref = ctx.lookup("SinhvienBean");
            SinhvienHome = (SinhvienRemoteHome)
PortableRemoteObject.narrow(ref, SinhvienRemoteHome.class);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private void createTable() {
        String[] hd = {"MSSV", "Họ tên", "Địa chỉ", "Lớp học"};
        dtm = new DefaultTableModel(hd, 0);
        table = new JTable(dtm);
        this.add(new JScrollPane(table), BorderLayout.CENTER);
    }

    public static void main(String[] args) {
        new Client().setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        Object o = e.getSource();
        if (o.equals(btAdd)) {
            String id = tfMSSV.getText();
            String name = tfHoten.getText();
            String diachi = tfDiachi.getText();
            String lophoc = tfLophoc.getText();
            if (id.trim().equals("") || name.trim().equals("")) {
                lblStatus.setText("Fields cannot be null!\nPlease enter
it");
            } else {
                try {
                    SinhvienHome.create(id, name, diachi, lophoc);
                    lblStatus.setText("Create new item successfully!");
                } catch (Exception e1) {
                    lblStatus.setText("Can not be create new Item");
                }
            }
        } else if (o.equals(btFindById)) {
            SinhvienRemote Sinhvien = null;
            dtm.setRowCount(0);
            try {
                Sinhvien =
SinhvienHome.findByPrimaryKey(tfMSSV.getText());
                if (Sinhvien != null) {
                    String[] tmp = new String[4];
                    tmp[0] = Sinhvien.getMssv();
                    tmp[1] = Sinhvien.getHoten();
                    tmp[2] = Sinhvien.getDiachi();
                    tmp[3] = Sinhvien.getLophoc();
                    dtm.addRow(tmp);
                }
            } catch (Exception e1) {
                lblStatus.setText(e1.getMessage());
            }
        } else if (o.equals(btFindByName)) {
            String[] tmp = new String[4];
            SinhvienRemote Sinhvien = null;
            dtm.setRowCount(0);
            try {
                Enumeration en = 
SinhvienHome.findByHoten(tfHoten.getText());
                if (e != null) {
                    while (en.hasMoreElements()) {
                        Sinhvien = (SinhvienRemote) en.nextElement();
                        tmp[0] = Sinhvien.getMssv();
                        tmp[1] = Sinhvien.getHoten();
                        tmp[2] = Sinhvien.getDiachi();
                        tmp[3] = Sinhvien.getLophoc();
                        dtm.addRow(tmp);
                    }
                } else {
                    lblStatus.setText("Could not find");
                }
            } catch (Exception ex) {
                lblStatus.setText(ex.getMessage());
            }
        }
    }
}
```

Kết quả:



Mssv:

Họ tên: tèo

Địa chỉ:

Lớp học:

MSSV	Họ tên	Địa chỉ	Lớp học
sv01	tèo	12 Nguyen Van B...	1A
sv06	tèo	1 Le loi	1B

infos...