

LAB 01 – ÔN TẬP JAVA CƠ BẢN



Bài 1. Viết chương trình quản lý danh sách người sử dụng. Thông tin mỗi người sử dụng gồm có: tên, mật khẩu, đường dẫn thư mục, quyền truy xuất (đọc, viết, cả hai). Chương trình có thể thực hiện các chức năng sau:

- Cho phép user đăng ký vào hệ thống.
- Cho phép user đăng nhập vào hệ thống.

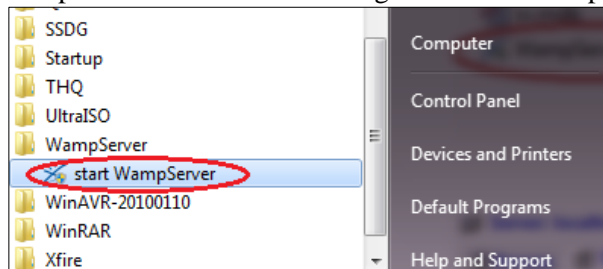
Hướng dẫn: Hiện nay, có rất nhiều hệ quản trị CSDL đang được sử dụng rộng rãi, trong phần lab này ta cùng tìm hiểu hai hệ quản trị CSDL được sử dụng phổ biến hiện nay là: mysql và SQL Server.

Phần 1: Kết nối CSDL mysql

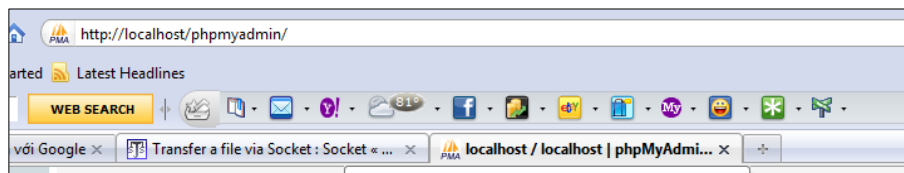
1. Cài đặt mysql thông qua gói WampServer và tạo CSDL

Bước 1: Cài đặt wampserver

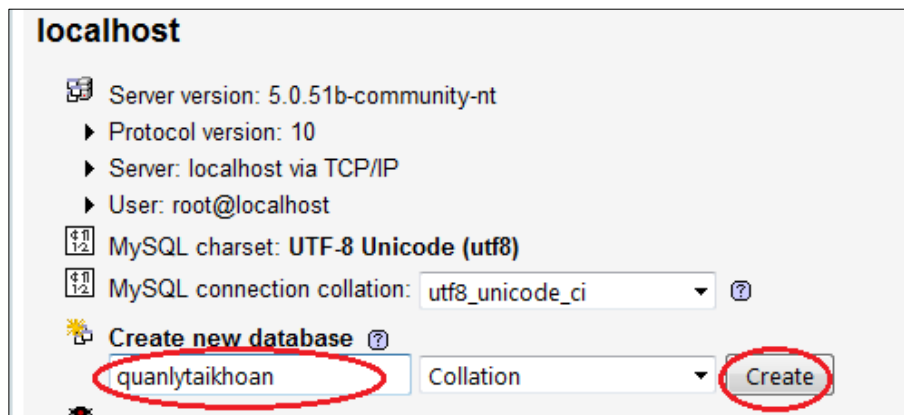
Bước 2: Sau khi cài xong WampServer ta vào Start → Program file → WampServer → start WampServer



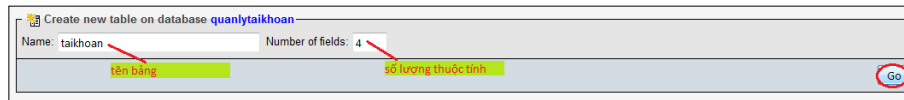
Bước 3: Vào trình duyệt Firefox gõ địa chỉ của trang phpmyadmin



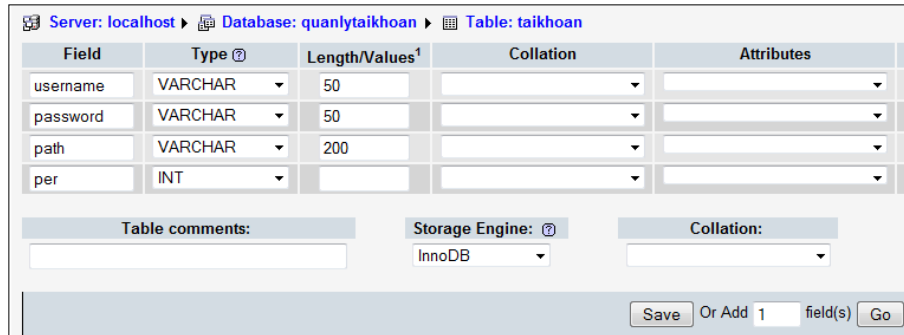
Bước 4: Tạo database quanlytaikhoan



Bước 5: Vào Database quanlytaikhoan, tạo bảng taikhoan

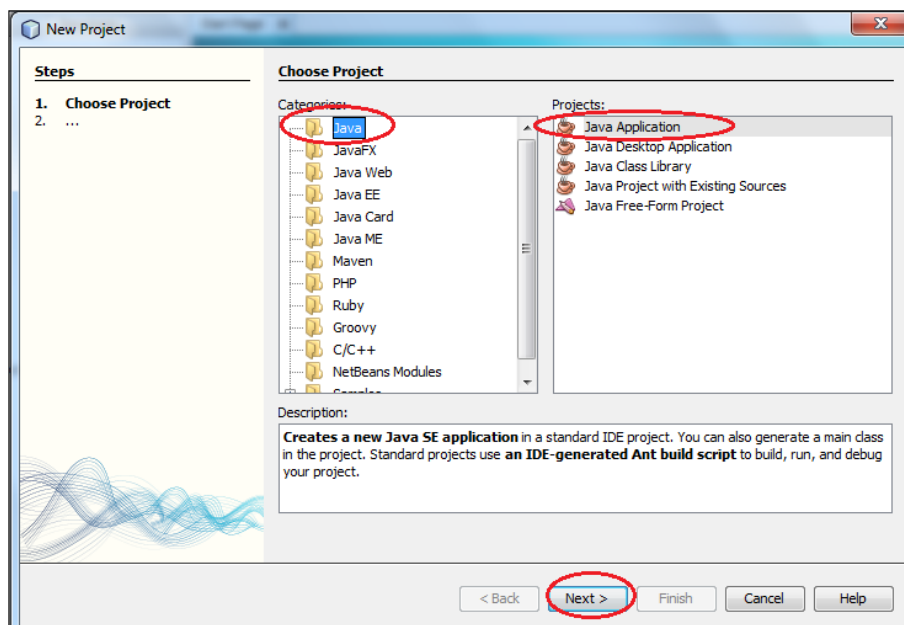
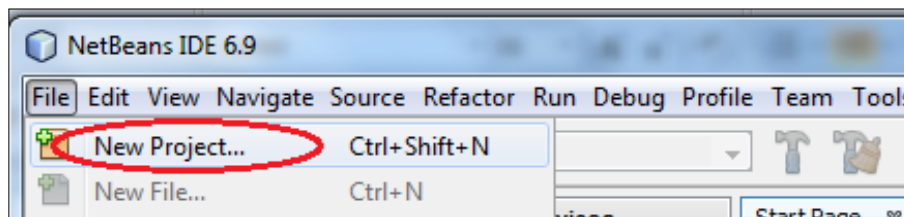


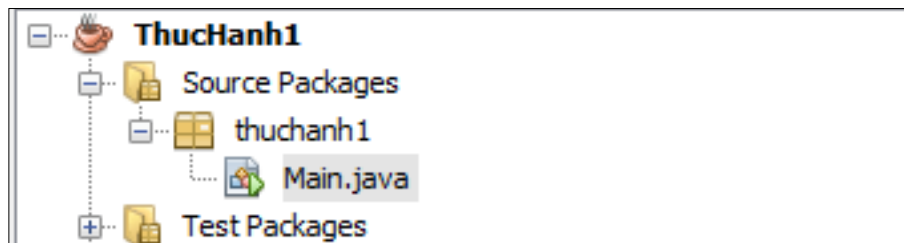
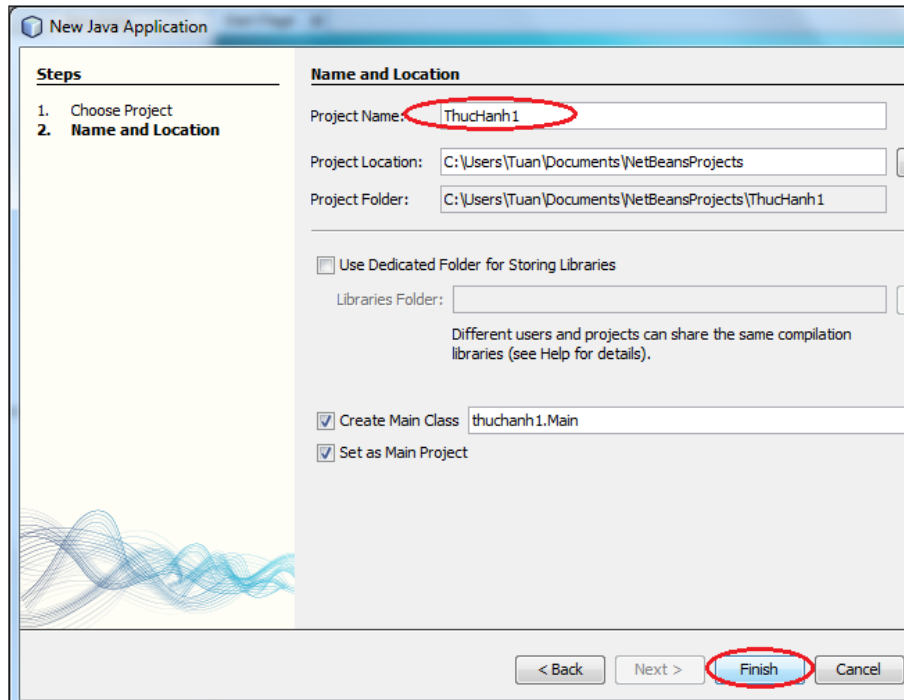
Bước 5: Tạo các thuộc tính cho bảng dữ liệu



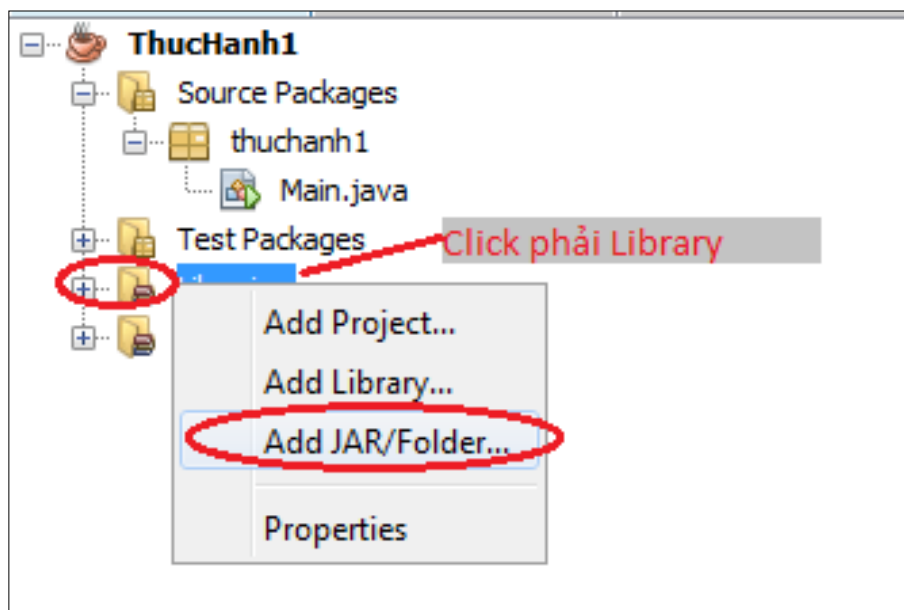
2. Tạo ra các lớp đối tượng truy xuất đến CSDL

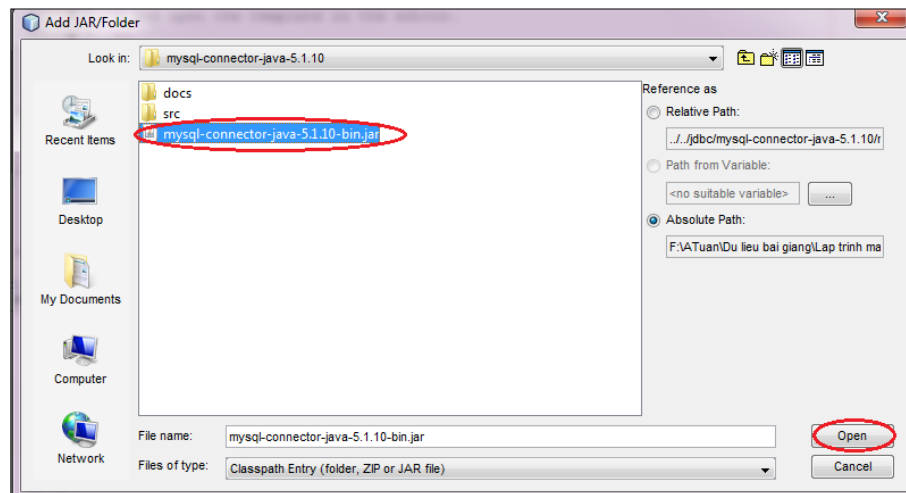
Bước 1: Tạo project mới: File → New Project



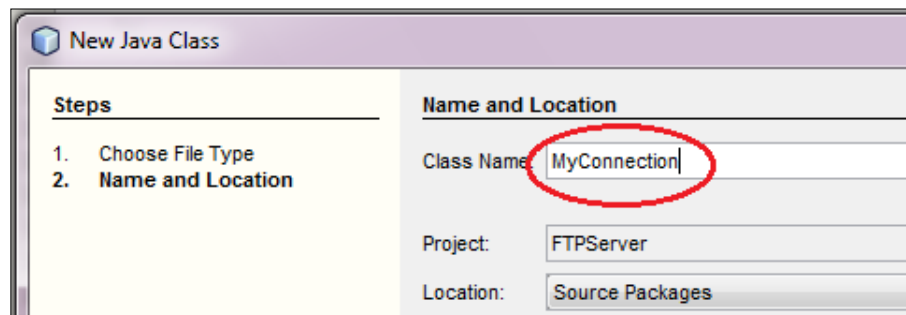
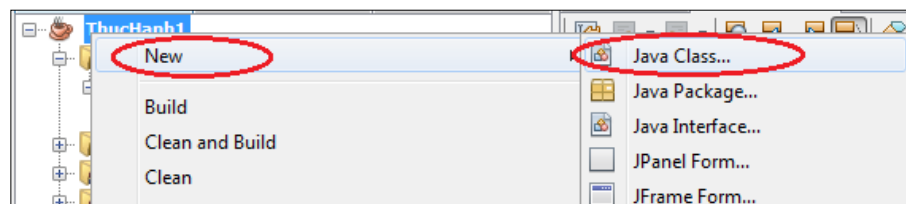


Bước 2: Thêm thư viện vào

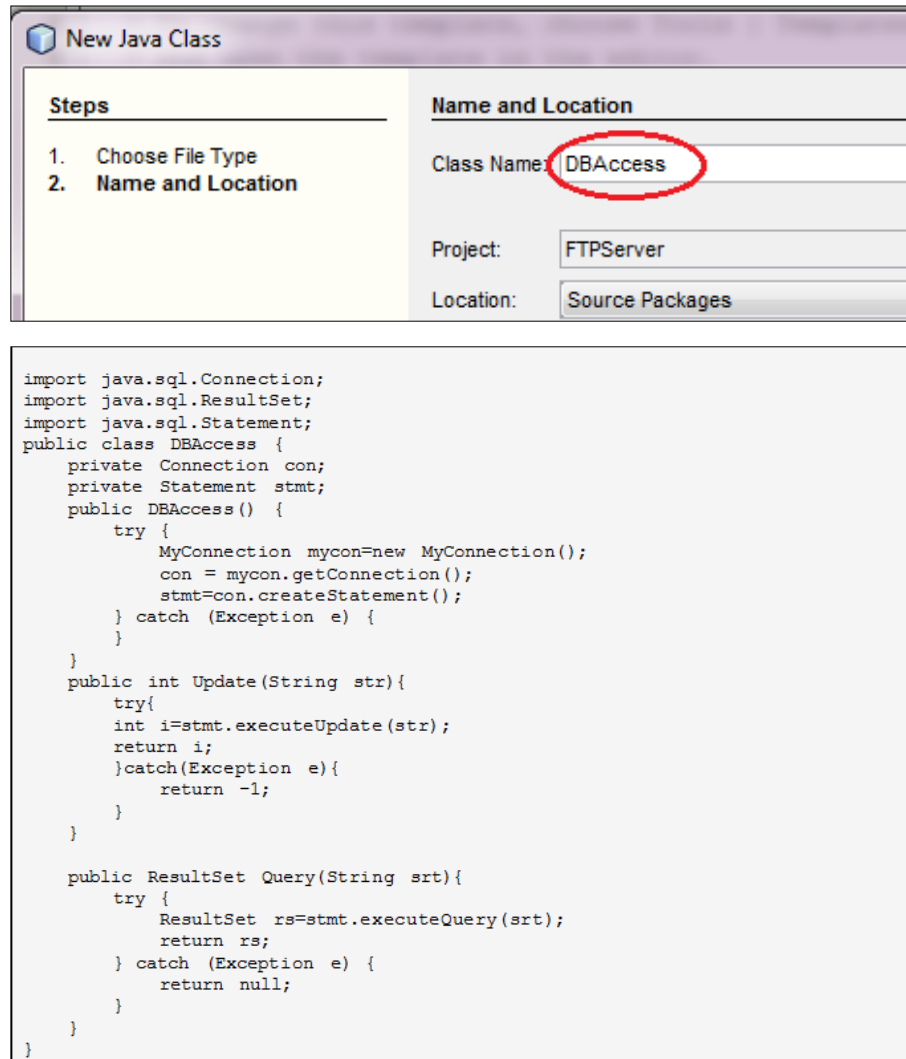




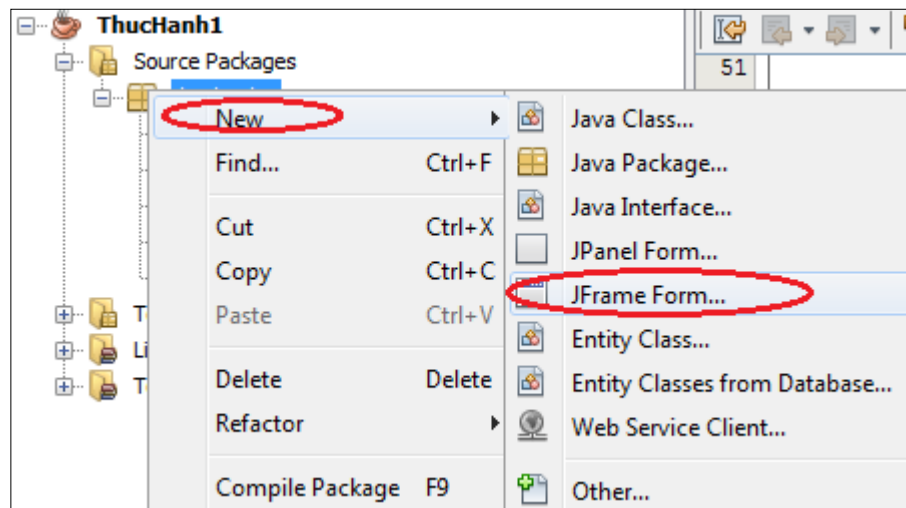
Bước 3: Tạo lớp chứa kết nối từ chương trình và mysql.

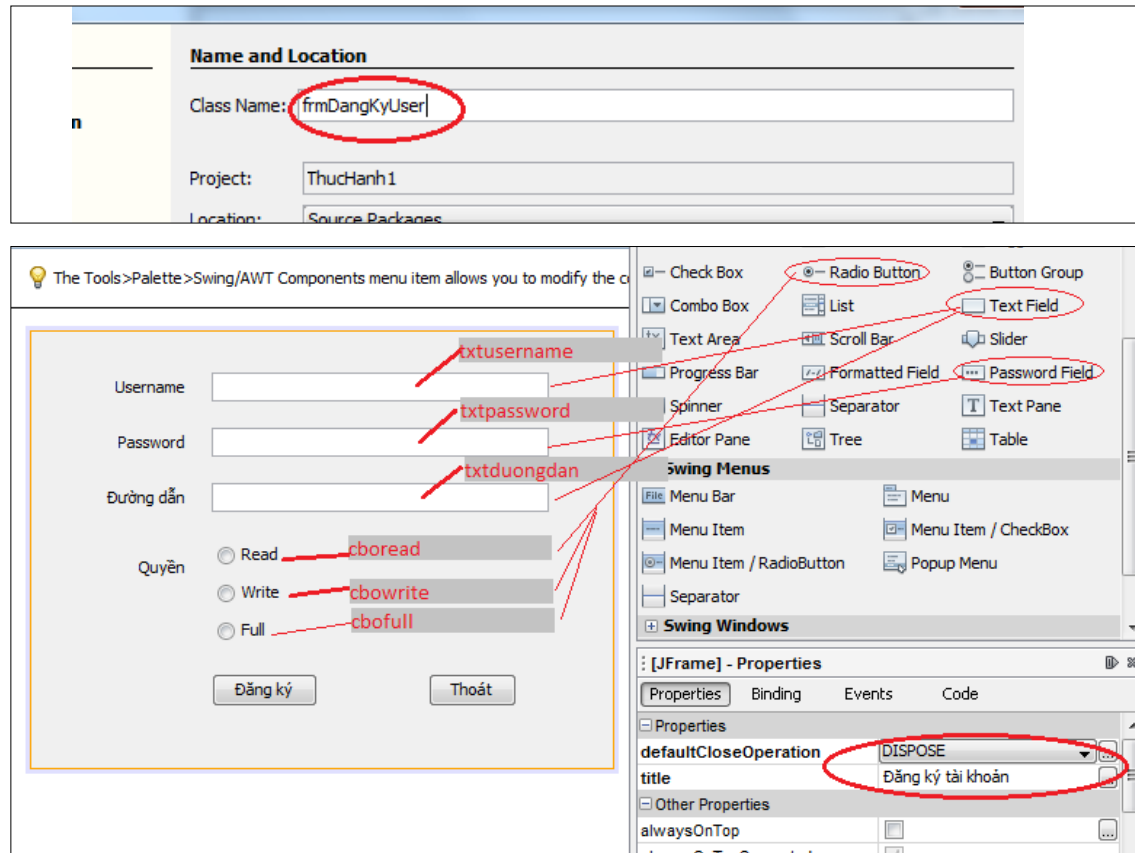


Bước 4: Tạo lớp truy xuất CSDL, và sau này khi cần truy xuất CSDL, ta chỉ cần gọi lớp này ra và thực thi các hàm bên trong nó

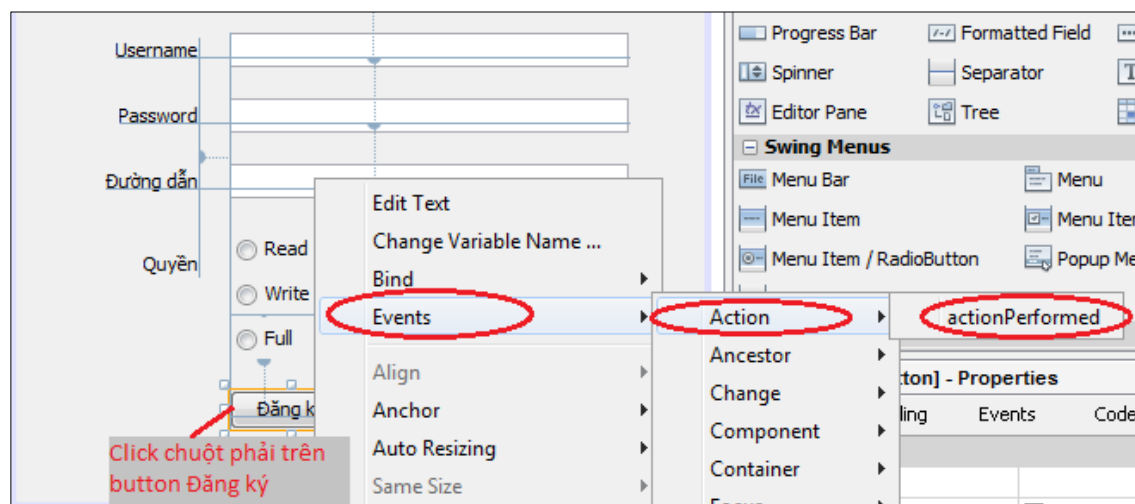


Bước 5: Tạo giao diện đăng ký user





Bước 6: Thêm sự kiện khi click chuột vào nút Đăng ký.



```

private void btndangkyActionPerformed(java.awt.event.ActionEvent evt) {
    //Lấy thông tin của User
    String username=txtusername.getText();
    String password=new String(txtpassword.getPassword());
    String duongdan=txtduongdan.getText();
    int quyen;
    if(cboread.isSelected()){
        quyen=0;
    }else if(cbowrite.isSelected()){
        quyen=1;
    }else
        quyen=2;
    //Tạo ra kết nối và thêm User vào CSDL
    DBAccess acc=new DBAccess();
    int kq= acc.Update("insert into taikhoan values('"+username+"','"+password+"','"+duongdan+"','"+quyen+"')");
    //Kiểm tra xem đã đăng ký được hay chưa
    if(kq!=0)
        JOptionPane.showMessageDialog(null, "Đăng ký User thành công.");
    else
        JOptionPane.showMessageDialog(null, "Đăng ký thất bại!");
}

```

Sự kiện cho nút Thoát

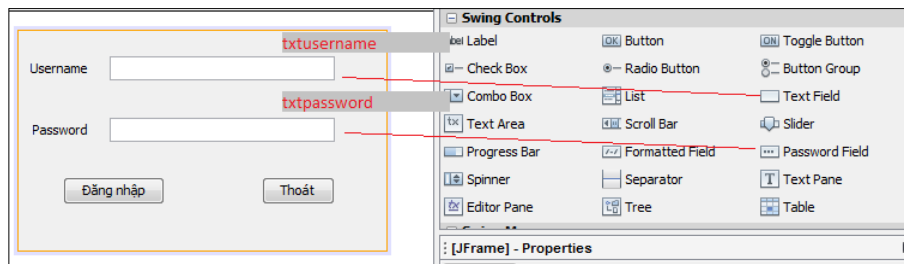
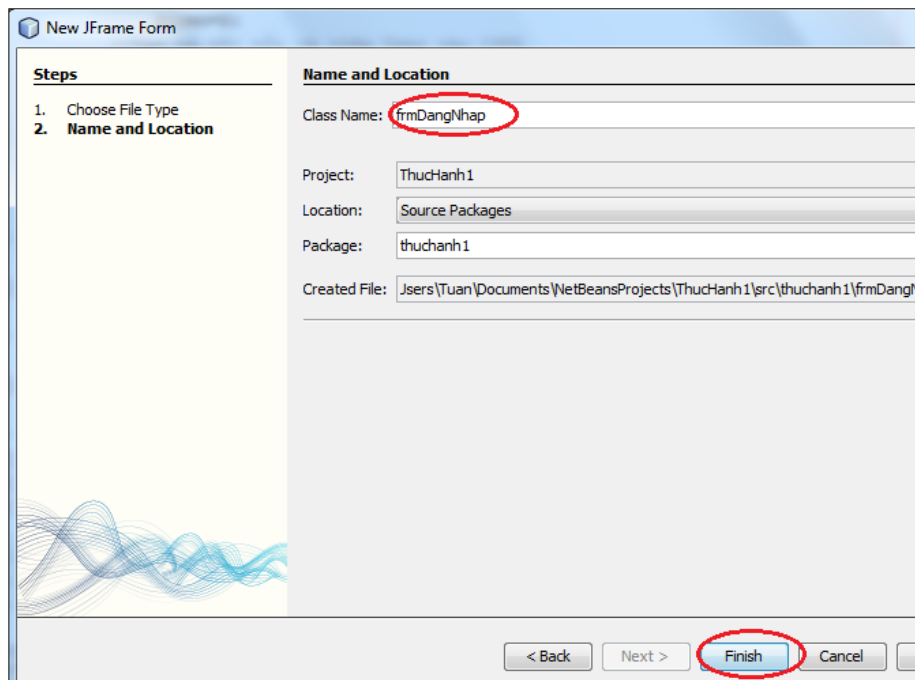
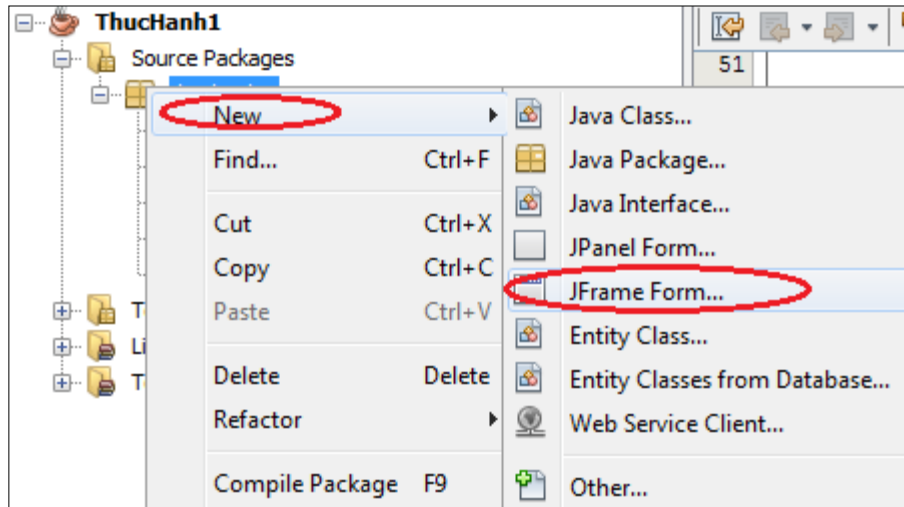
click phải trên button Thoát

```

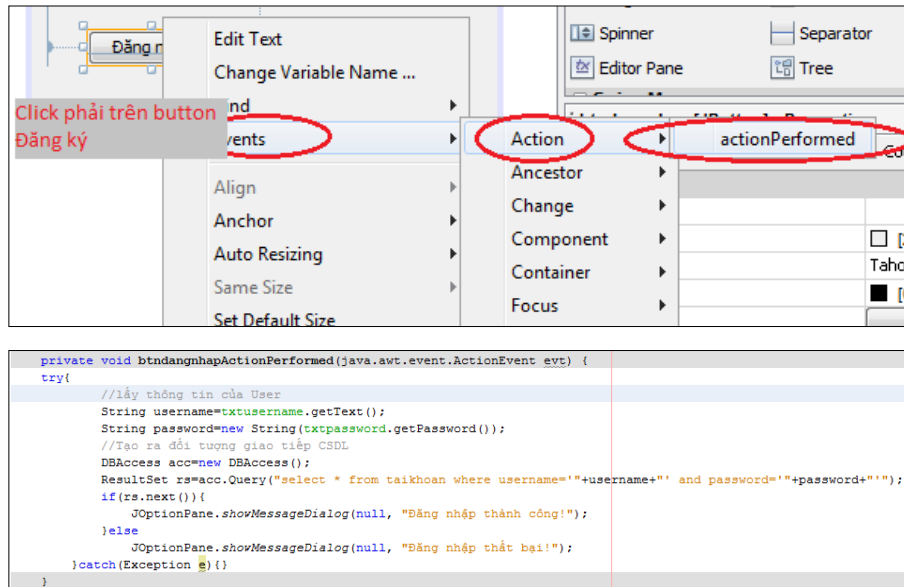
private void btnthoatActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}

```

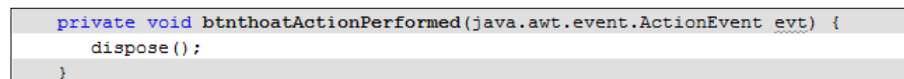
Bước 7: Tạo giao diện cho phần đăng nhập



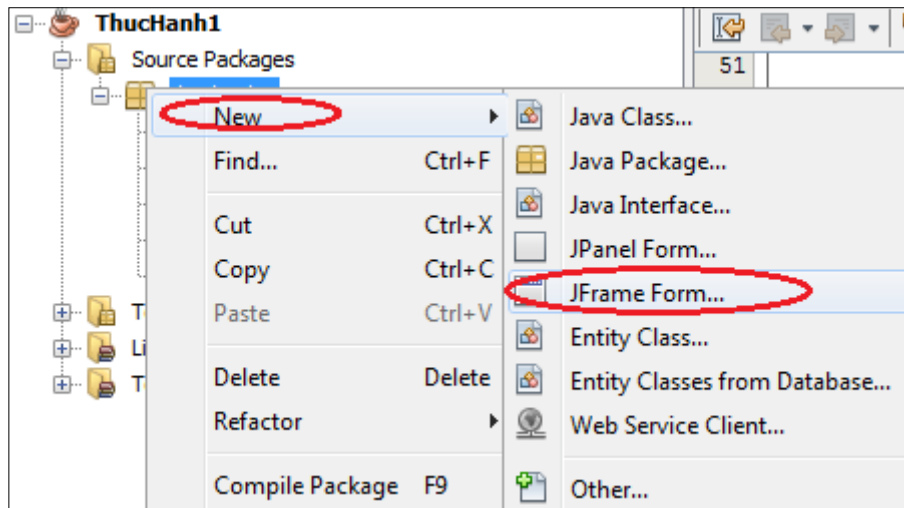
Bước 8: Thêm sự kiện cho nút đăng nhập.

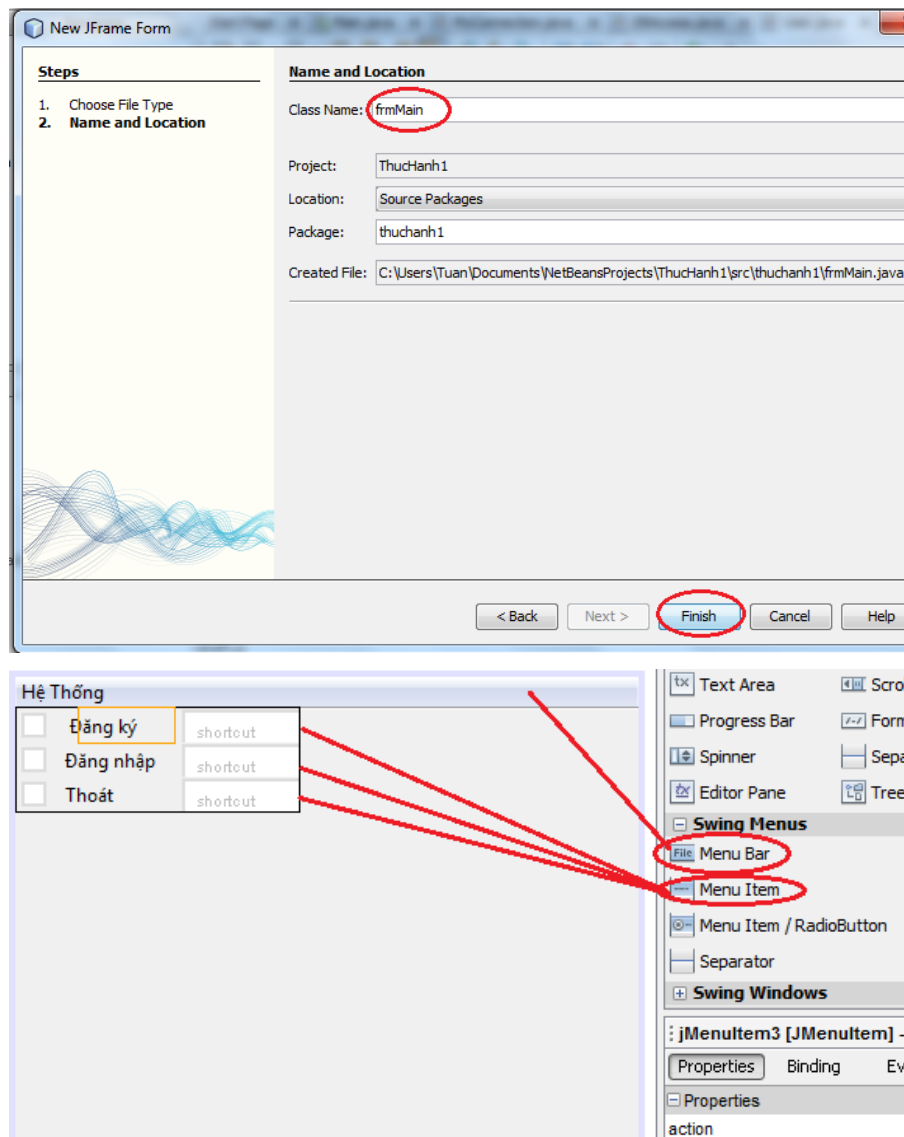


Thêm sự kiện cho button Thoát



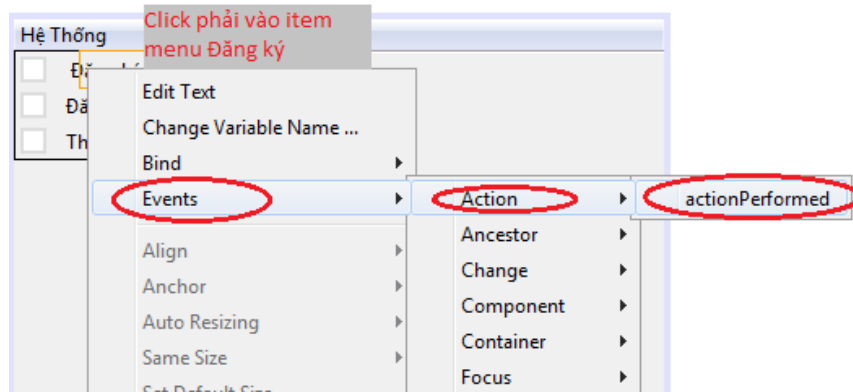
Bước 9: Tạo lớp giao diện cho chương trình





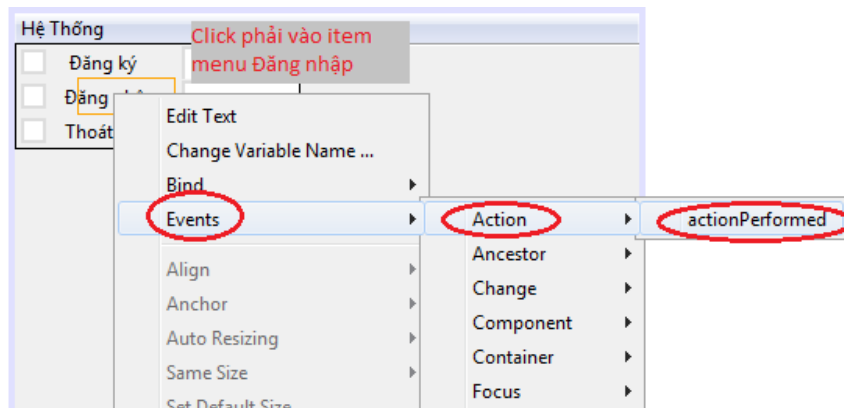
Bước 10: Thêm sự kiện cho các itemmenu

Thêm sự kiện cho itemmenu Đăng ký



```
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {  
    frmDangKyUser obj=new frmDangKyUser();//cấp phát vùng nhớ cho form Đăng ký User  
    obj.setVisible(true);//Hiện thị form Đăng ký lên màn hình  
}
```

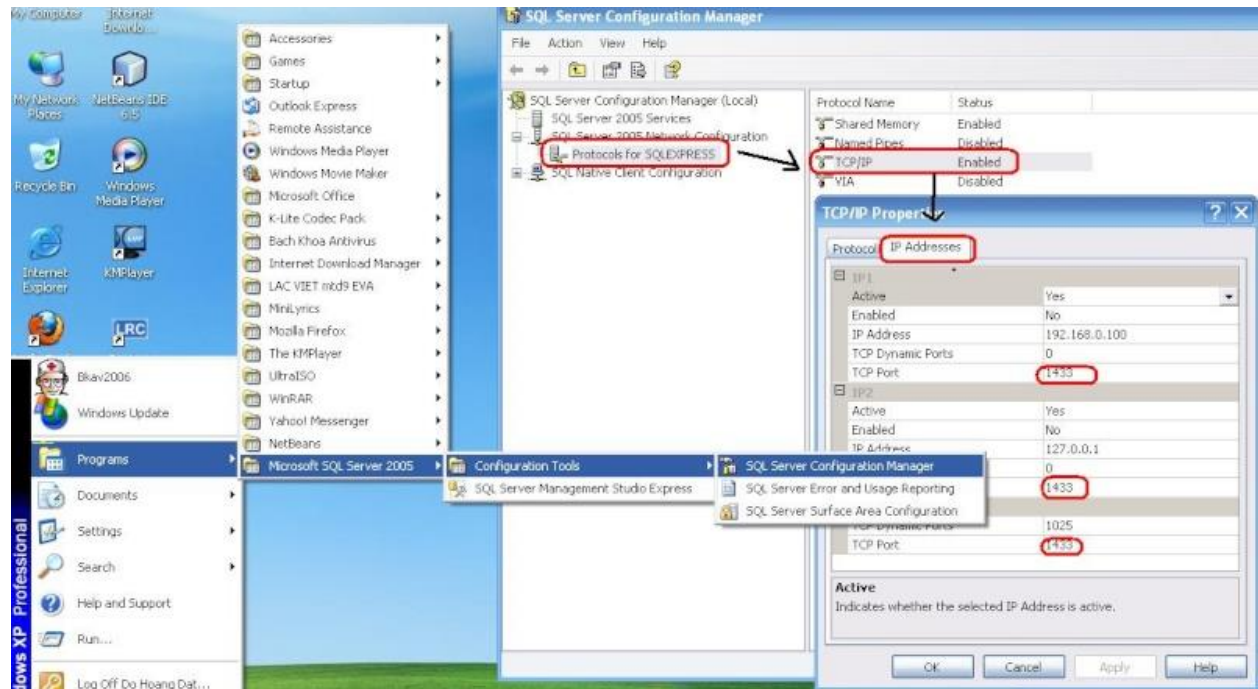
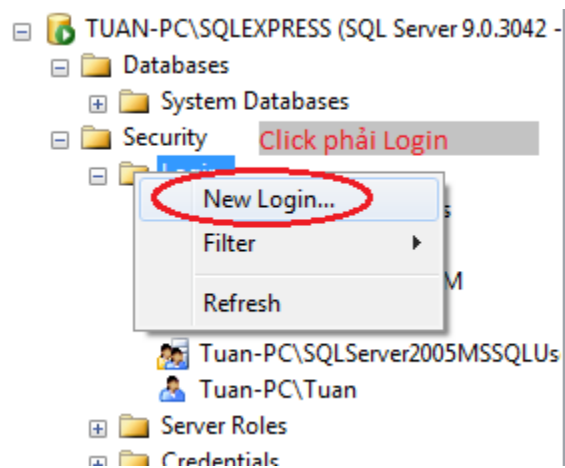
Thêm sự kiện cho itemmenu Đăng nhập

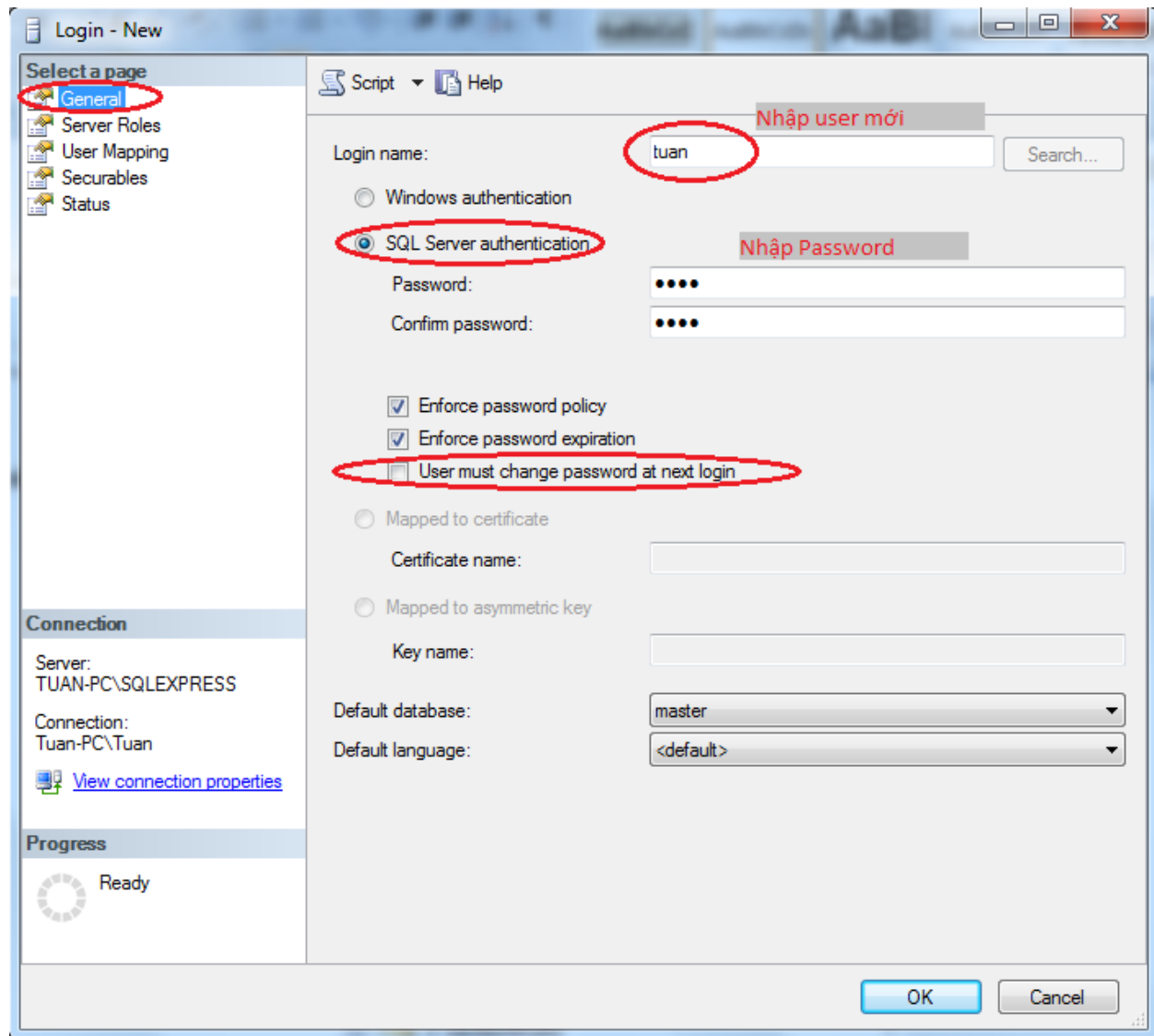


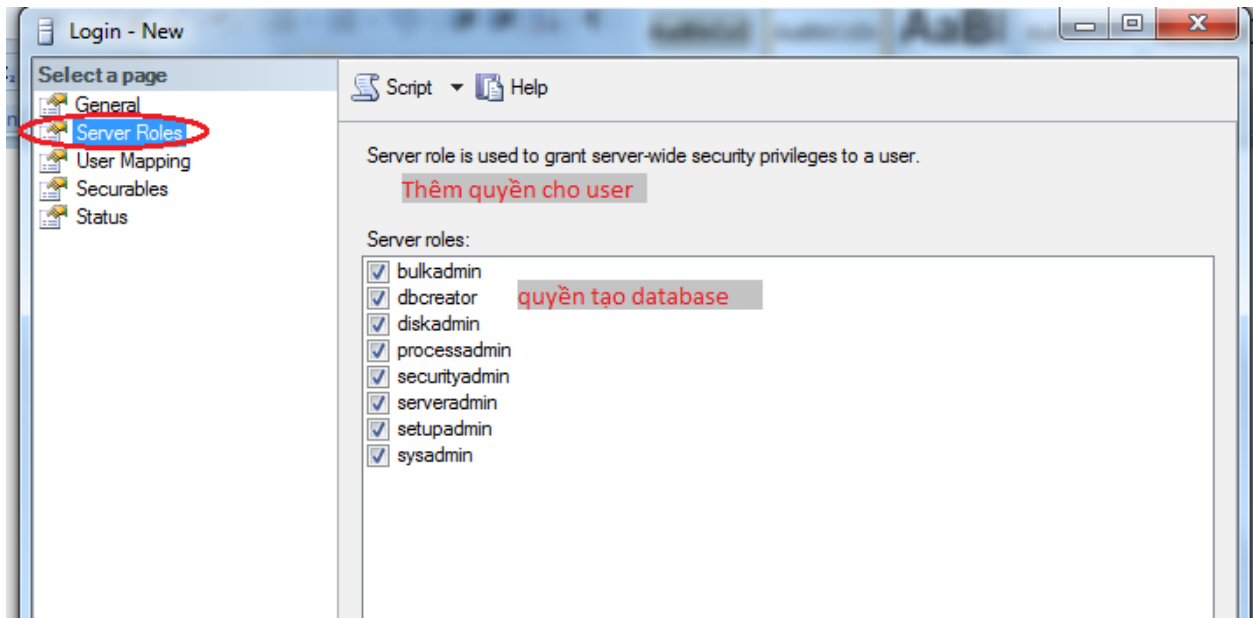
```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {  
    frmDangNhap obj=new frmDangNhap();//cấp phát vùng nhớ cho form Đăng nhập  
    obj.setVisible(true);//Hiện thị form Đăng nhập lên màn hình  
}
```

Thêm sự kiện cho itemmenu Thoát

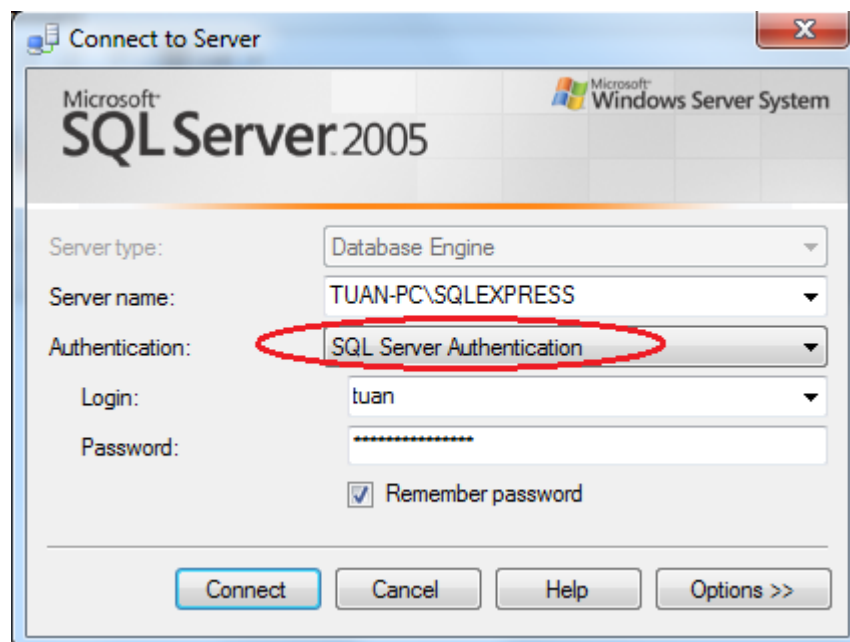
```
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0);  
}
```

Phần 2: Kết nối với cơ sở dữ liệu SQL Server.**1. Tạo CSDL SQLServer:****Bước 1:** Chỉnh lại port cho SQLEXPRESS.**Bước 2:** Tạo User mới trong sqlserver

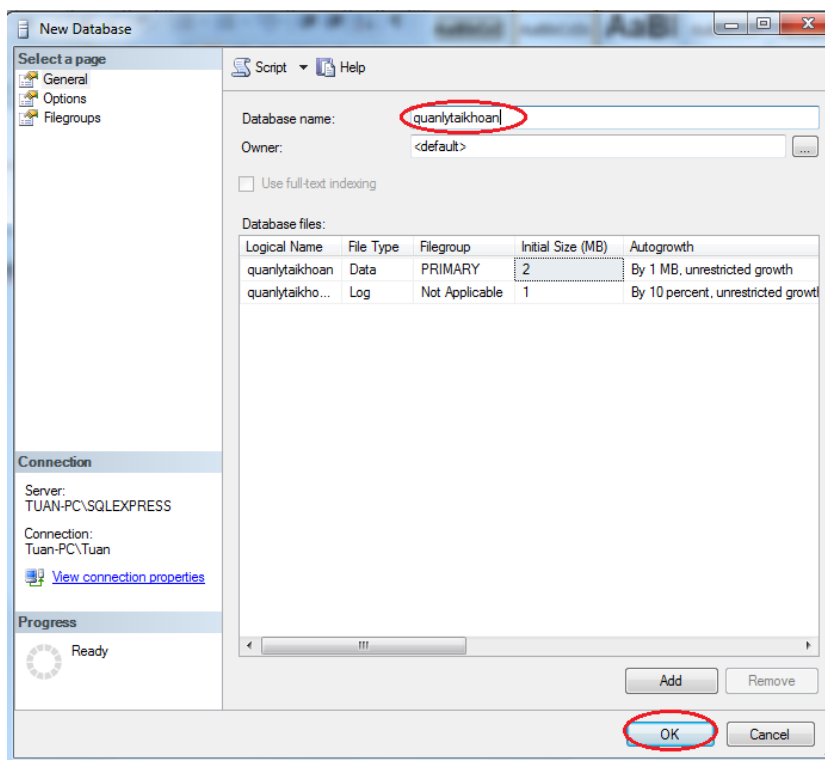
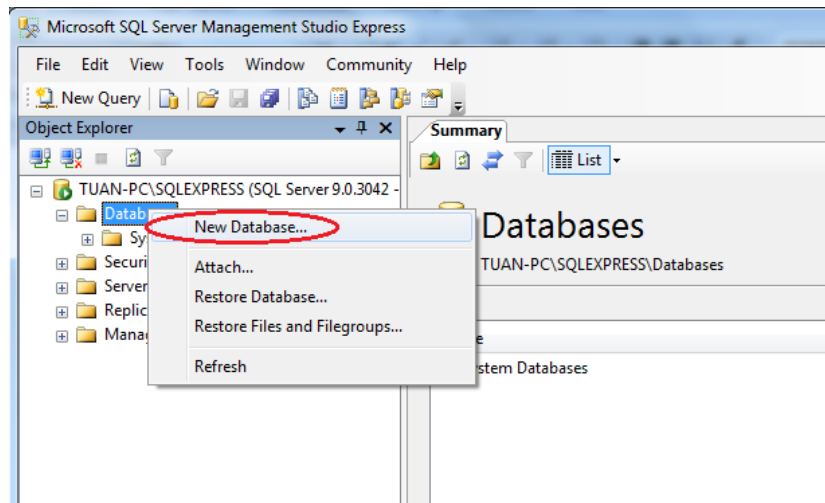




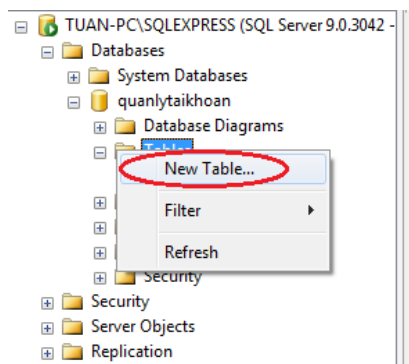
Bước 3: Đăng nhập vào Database với User mới tạo.

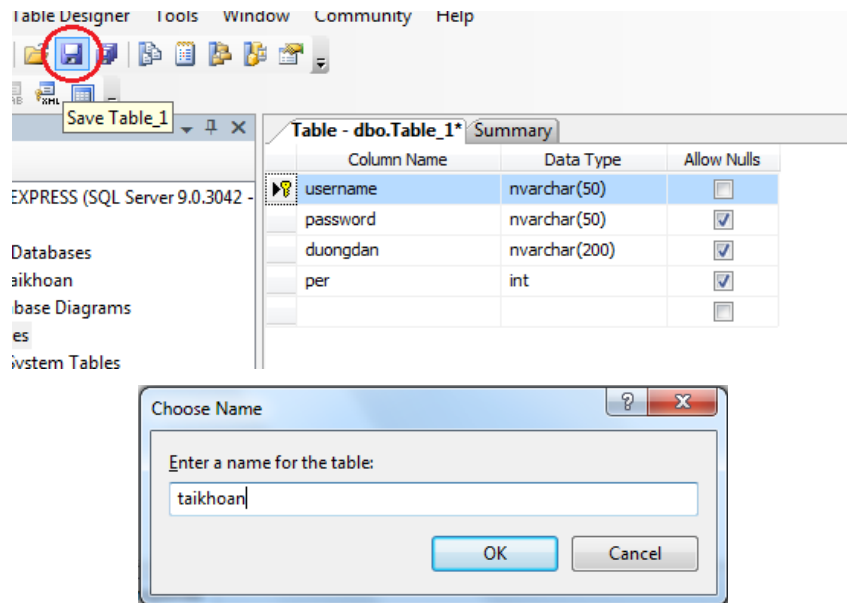


Bước 4: Tạo database mới quanlytaikhoan

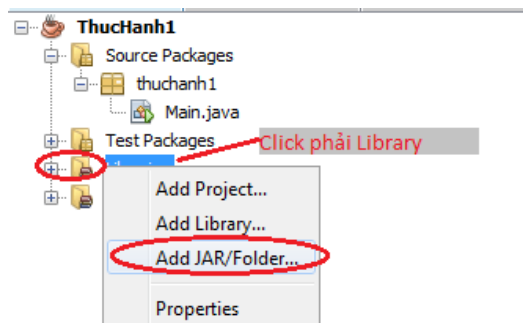


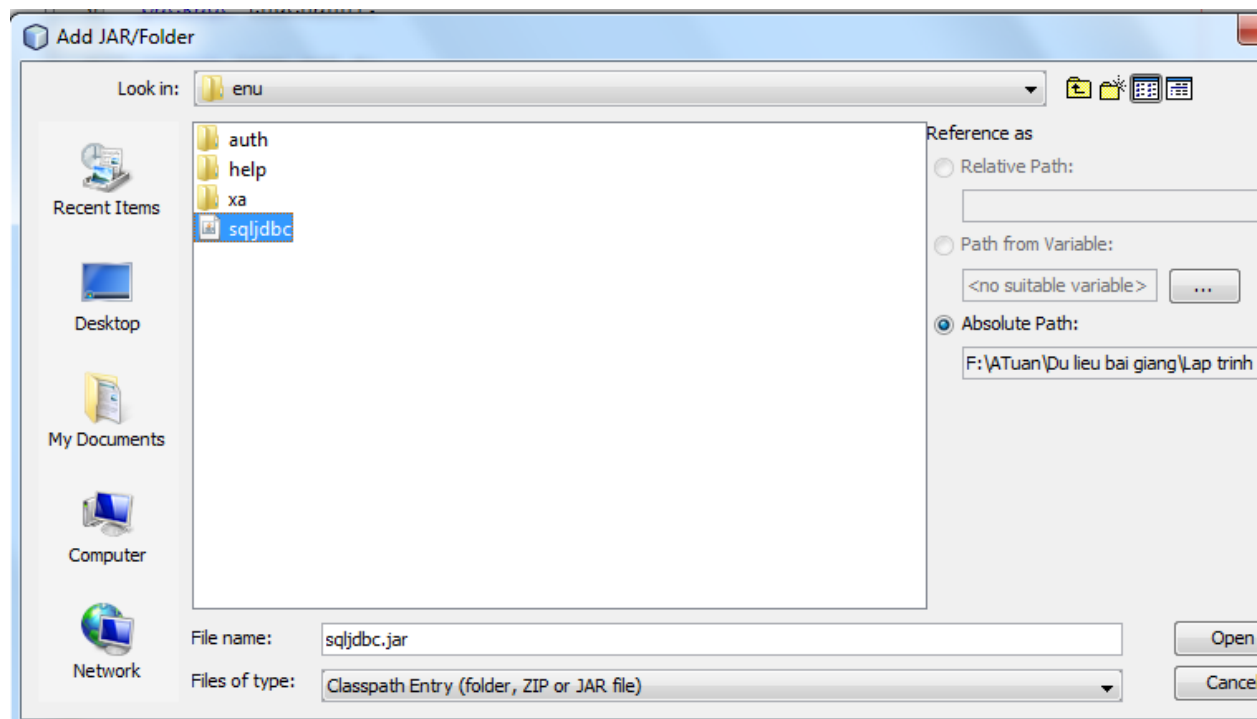
Bước 5: Tạo Table mới.





Bước 6: Tiếp theo ta thực hiện lại các bước như phần kết nối CSDL ở trên. Nhưng thay đổi phần add library như sau





Bước 7: Đồng thời thay đổi lớp MyConnection.java như sau

```
package thuchanh1;

import java.sql.*;
import javax.swing.JOptionPane;

/**
 *
 * @author Tuan
 */
public class MyConnection {
    public Connection getConnection(){
        try{
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
            String url = "jdbc:sqlserver://localhost:1433;Database=quanlytaikhoan;user=tuan;password=tuan";
            Connection con=DriverManager.getConnection(url);
            return con;
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e.toString(),"Loi",JOptionPane.ERROR_MESSAGE);
            return null;
        }
    }
}
```

Port chỉnh lại ở bước 1 tên database username password mới tạo

Bước 8: Sau khi làm xong các bước như phần kết nối CSDL mysql, ta kiểm tra chương trình đăng ký tài khoản và đăng nhập thử.

BÀI TẬP THAM KHẢO:

Bài 1. Viết chương trình quản lý danh sách người sử dụng. Thông tin mỗi người sử dụng gồm có: tên, mật khẩu, đường dẫn thư mục, quyền truy xuất (đọc, viết, cả hai). Chương trình có thể thực hiện các chức năng sau:

Tạo thêm một người sử dụng mới vào hệ thống.

Hiển thị danh sách người sử dụng.

Hiển thị danh sách người sử dụng theo quyền truy xuất.

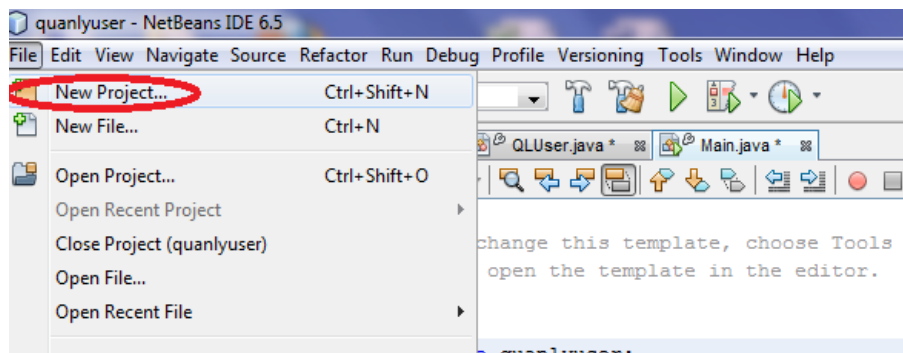
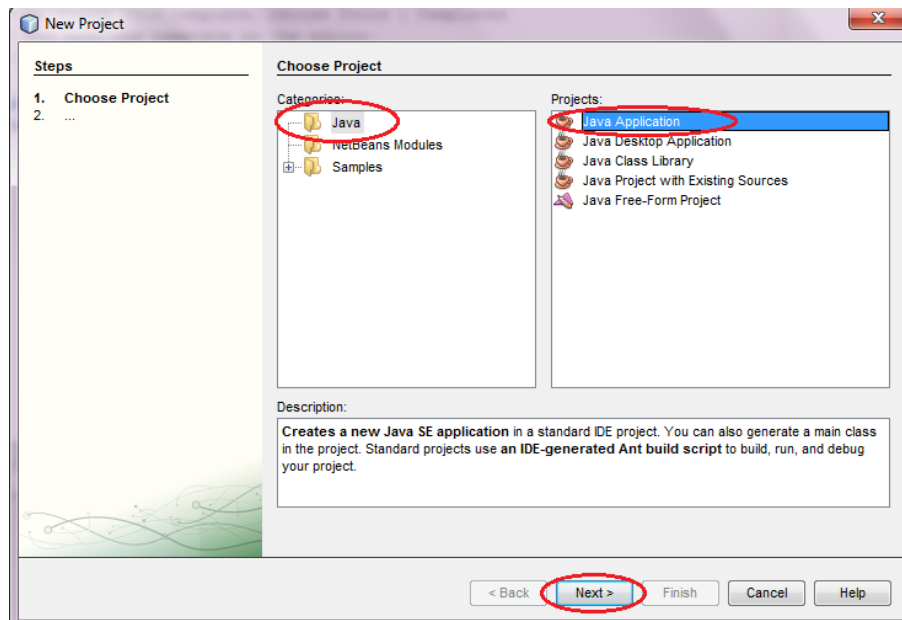
Hiển thị danh sách người sử dụng theo tên.

Lưu toàn bộ thông tin người sử dụng xuống cơ sở dữ liệu

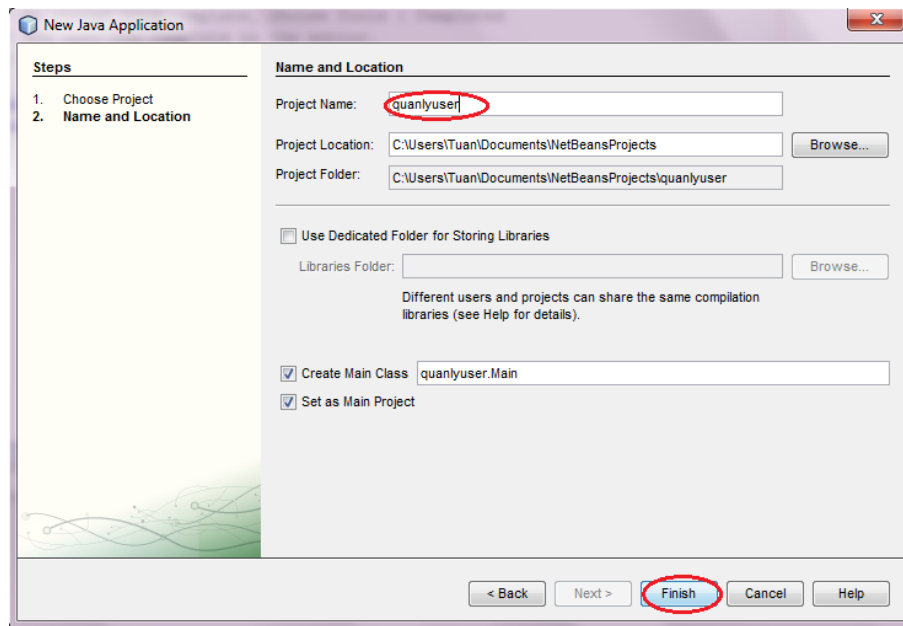
Đọc toàn bộ thông tin người sử dụng từ cơ sở dữ liệu vào chương trình.

Hướng dẫn:

Bước 1: Tạo project mới: File→New Project...

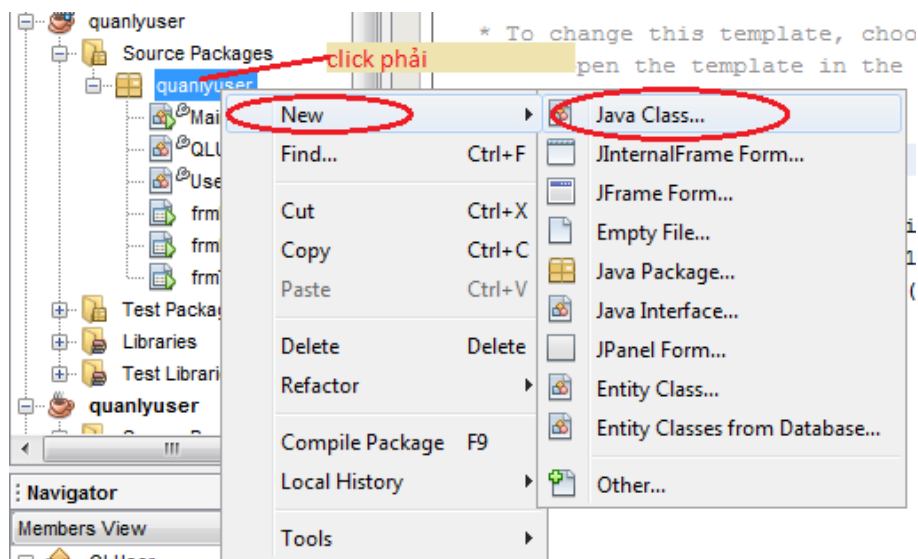
**Tạo Java Application Project**

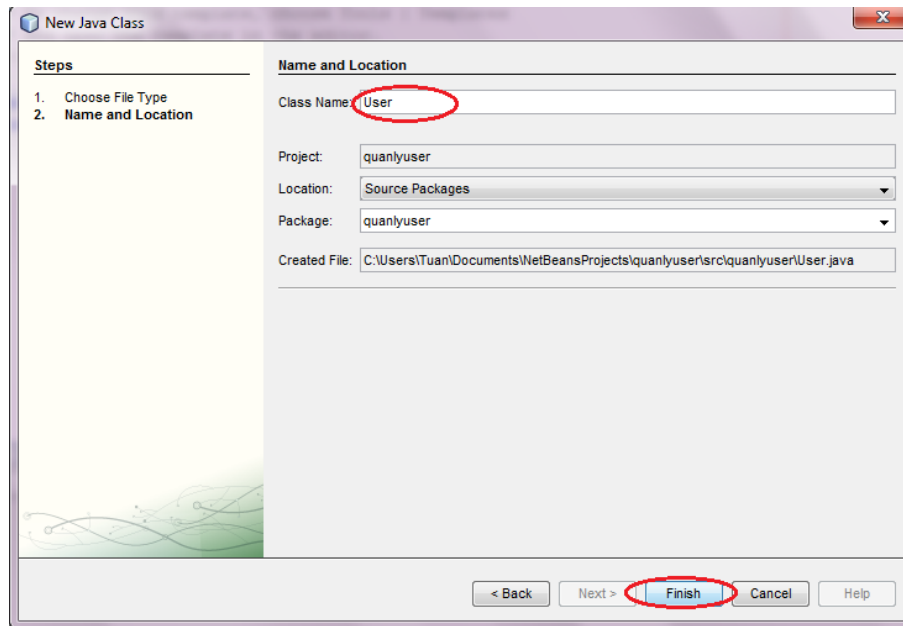
Tạo project mới tên quanlyuser →Finish



Bước 2: Tạo ra các lớp xử lý

Tạo ra lớp User





Code cho lớp User

```
package thuchanhltm;
import java.util.Vector;
public class User {
    //Danh sach cac thuoc tinh
    private String ten;
    private String matKhau;
    private String duongDan;
    private int quyen;
    public int READ =0;
    public int WRITE = 1;
    public int FULL =2;
    //ham khoi tao
    public User(String ten,String matKhau,String duongDan,int quyen){
        this.ten=ten;
        this.matKhau=matkhau;
        this.duongDan=duongdan;
        this.quyen=quyen;
    }
    //Xay dung cac getter, setter
    public String getTen(){
        return ten;
    }
    public void setTen(String ten){
        this.ten=ten;
    }
}
```

```
}  
public String getMatKhau(){  
    return matKhau;  
}  
public void setMatKhau(String matkhau){  
    this.matKhau=matkhau;  
}  
public String getDuongDan(){  
    return duongDan;  
}  
public void setDuongDan(String duongdan){  
    this.duongDan=duongdan;  
}  
  
public int getQuyen(){  
    return quyen;  
}  
public void setQuyen(int quyen){  
    this.quyen=quyen;  
}  
public boolean laUser(String ten){  
    return ten.equals(this.ten);  
}  
public Vector hienThiRow(){  
    Vector row=new Vector();  
    row.add(ten);  
    row.add(this.matKhau);  
    row.add(this.duongDan);  
    row.add(this.quyen);  
    return row;  
}  
public boolean laUser(int quyen)  
    {  
        return quyen==this.quyen;  
    }  
}
```

Tiếp tục tạo ra lớp QLUser

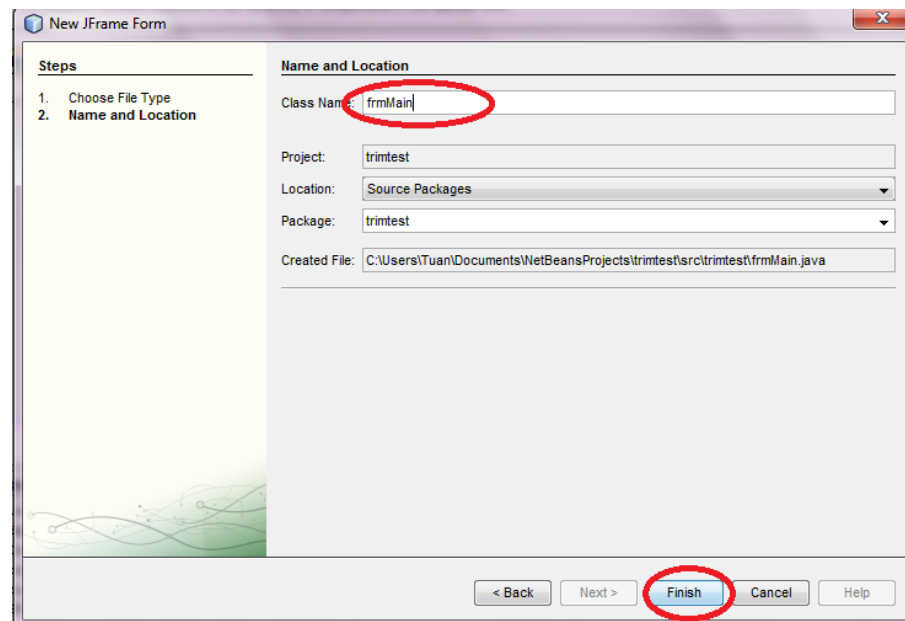
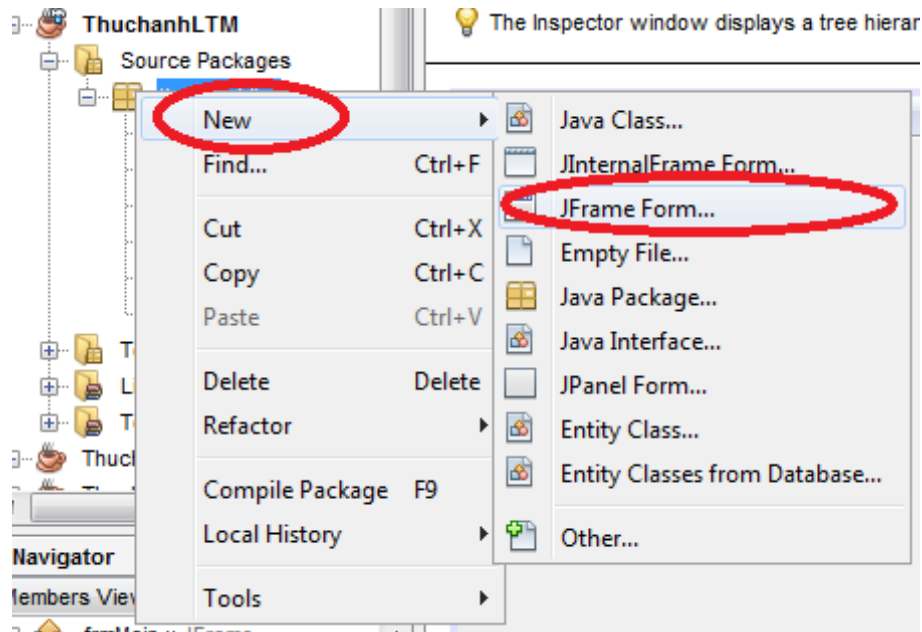
```
package thuchanhltm;  
import java.util.Vector;  
public class QLUser {  
    //Tạo mảng chứa danh sách các biến tham chiếu lưu địa chỉ user
```

```
private User user[];
//so luong user hien dang duoc luu dia chi trong danh sach
private int n;
//Mang so luong user toi da ma mang quan ly
private int MAXN;
//ham khoi tao
public QLUser(int MAXN){
    //Ham khoi tao nhan tham so so luong user toi da
    // ma chương trình có thể quản lý
    this.MAXN=MAXN;
    user=new User[MAXN];
    n=0;
}
//ham them user vao danh sach quan ly
public void themUser(User u){
    user[n]=u;
    n++;
}
//ham tao user va luu dia chi user vao danh sach de quan ly
public void themUser(String t,String mk,String dd,int q ){
    user[n]=new User(t, mk, dd, q);
    n++;
}
public Vector timKiem(String ten){
    //tao vector luu ket qua tra lai
    Vector rows=new Vector();
    for(int i=0; i<n;i++)
        if(user[i].laUser(ten))
            rows.add(user[i].hienThiRow());
    //moi dong la mot user
    return rows;
}
public Vector timKiem(int quyen){
    Vector rows=new Vector();
    for(int i=0;i<n;i++)
    {
        if(user[i].laUser(quyen))
            rows.add(user[i].hienThiRow());
    }
    return rows;
}
```

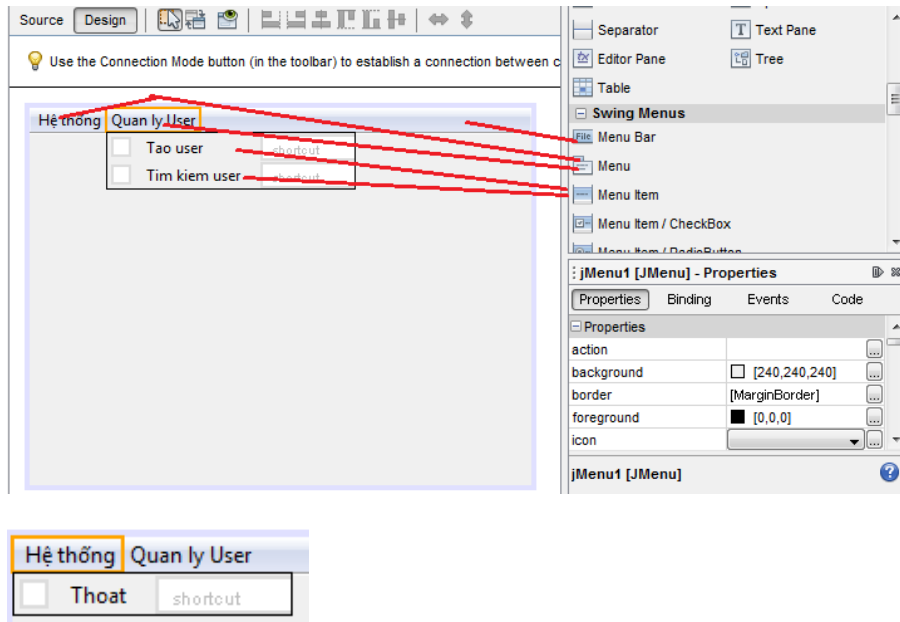
```
}
```

Bước 3: Tạo ra các lớp giao diện

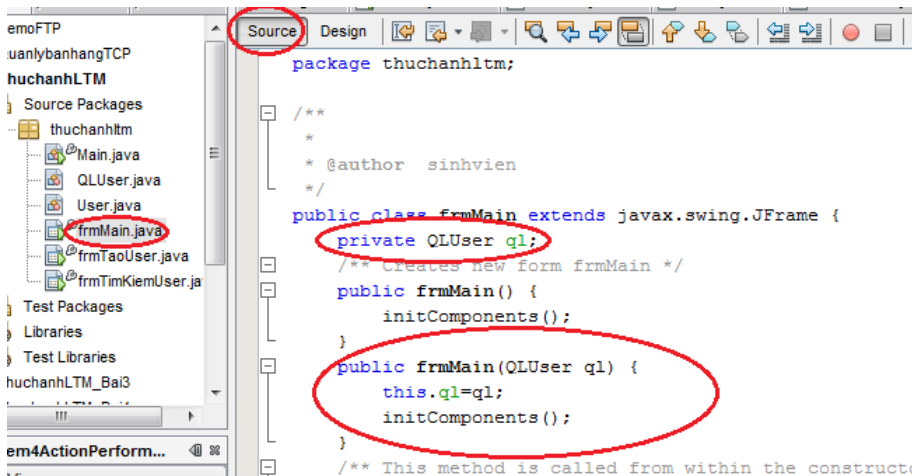
3.1 Tạo ra 1 lớp frmMain làm chương trình chính



3.2 Tạo giao diện gồm Menu Bar, Menu, Menu Item.



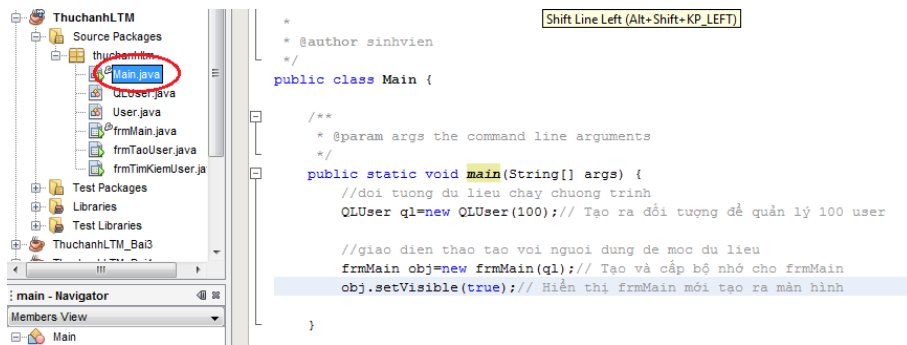
Thêm các thuộc tính và constructor cho frmMain



3.3 Sau khi tạo xong giao diện, ta bấm Shift + F6 để chạy thử.

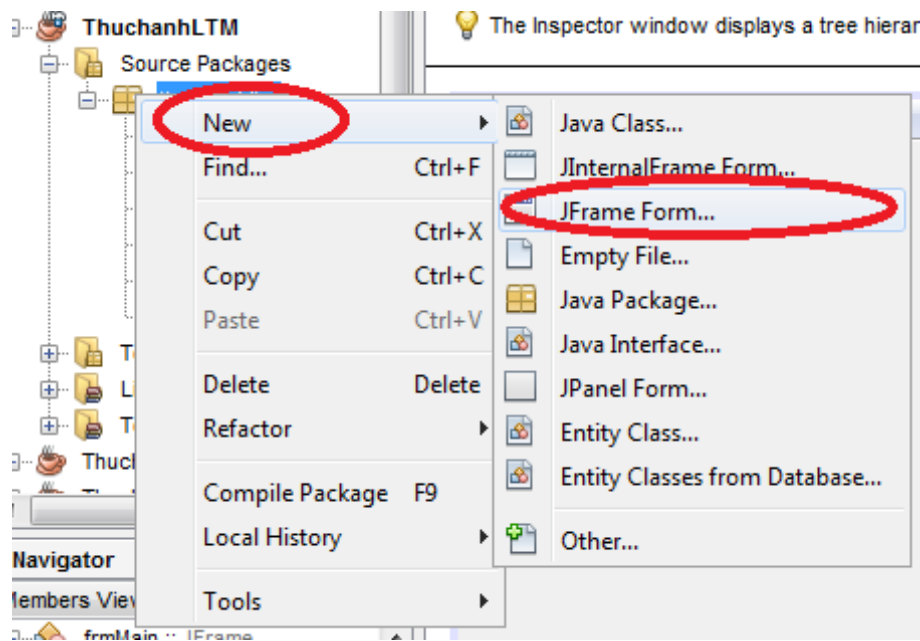
3.4 Khi tạo project thì netbean sẽ tự tạo ra một lớp Main chứa hàm Main và khi ta biên dịch và chạy project thì hàm main này sẽ thực hiện.

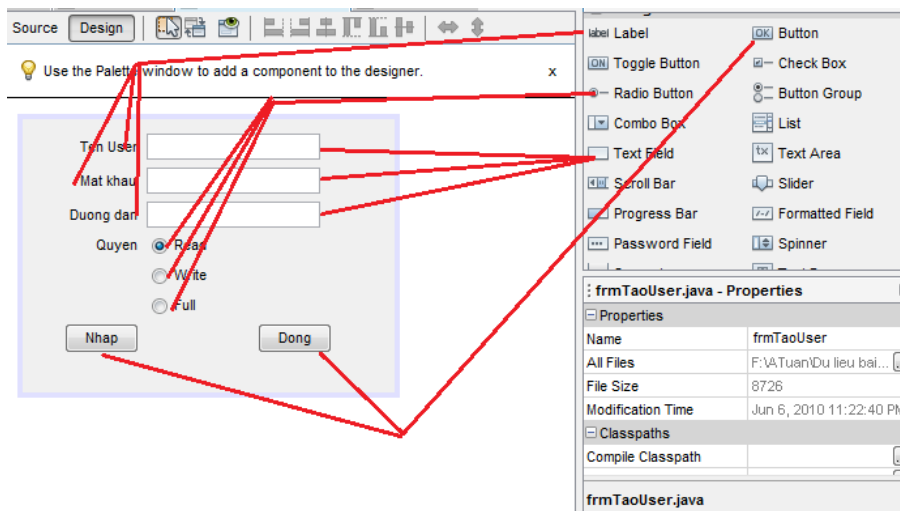
Trong hàm main này, ta tạo ra 2 đối tượng thuộc lớp QLUser và frmMain.



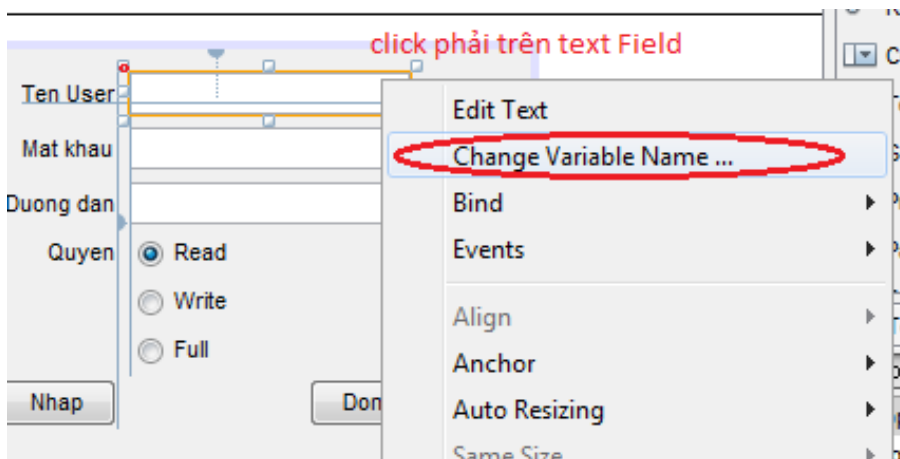
3.5 Bấm F5 để kiểm tra xem có hiện frmMain chưa.

3.6 Tạo JFrame Form mới tên là frmTaoUser có giao diện như sau

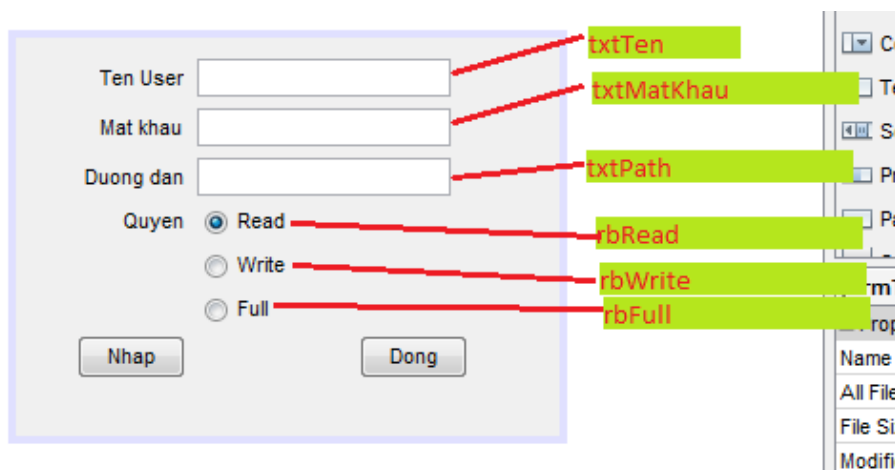




3.7 Tiến hành đặt tên cho các đối tượng: Click phải lên đối tượng → Change Variable Name → Nhập tên biến đối tượng

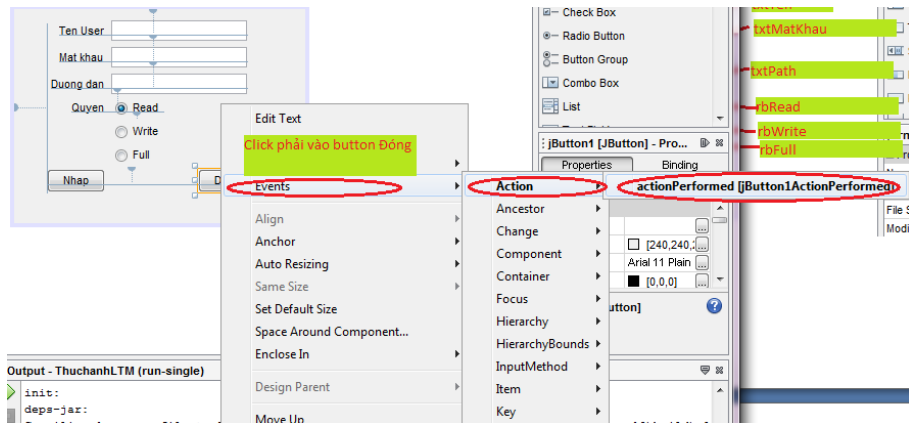


Đặt tên các biến đối tượng còn lại:



3.8 Sau khi hoàn tất các bước trên, ta tiến hành thêm sự kiện cho các button Dong và Nhap.

Đối với button Đóng ta muốn khi click vào button thì form đóng lại, ta làm như sau.

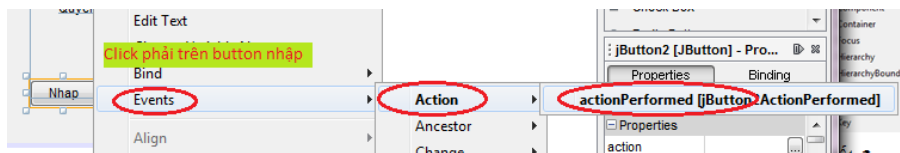


Màn hình code xuất hiện và ta chỉ cần viết đoạn code bên dưới

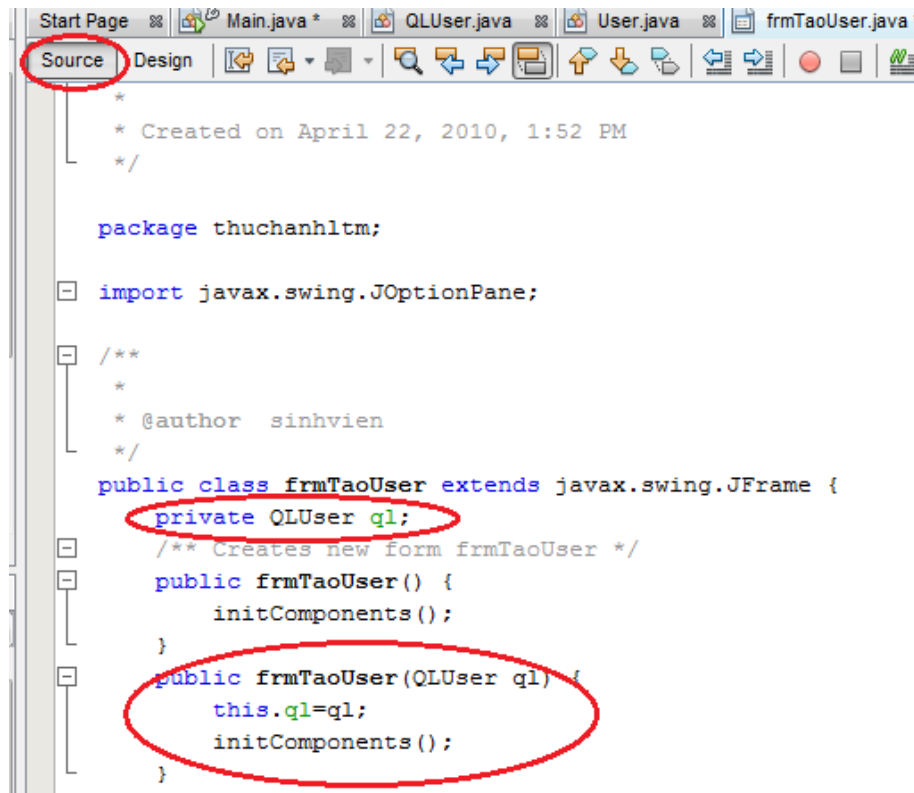
```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
```

Bây giờ khi frmTaoUser đang chạy mà ta click vào button đóng thì form sẽ đóng lại.

3.9 Thêm sự kiện cho button Nhập



Màn hình code xuất hiện và ta tiếp tục thêm phần code xử lý thêm User. Nhưng trước khi viết đoạn code nhập thêm User, trong phần Source, ta phải thêm một thuộc tính cho Form và thêm 1 constructor cho Form.

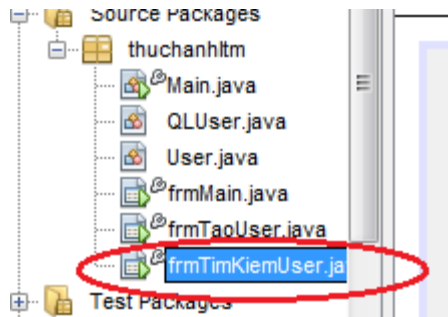


Sau đó viết thêm phần code cho button nhập User

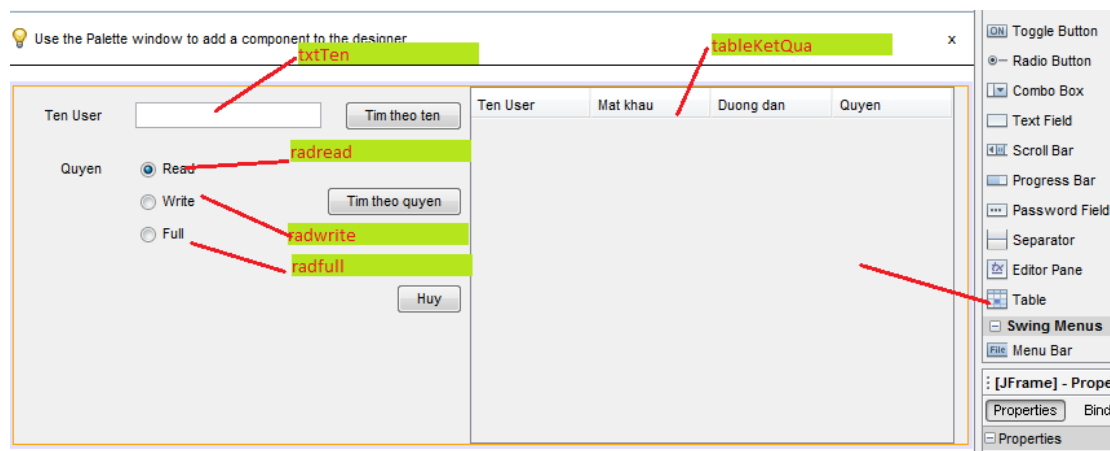
```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    //tao mot user
    String ten=txtTen.getText();
    String matkhau = txtMatKhau.getText();
    String path=txtPath.getText();
    int acc=0;
    if(this.rbFull.isSelected())
        acc=2;
    if(this.rbRead.isSelected())
        acc=0;
    if(this.rbWrite.isSelected())
        acc=1;
    User u=new User(ten,matkhau,path,acc);
    //them vao
    this.ql.themUser(u);
    JOptionPane.showMessageDialog(null, "Nhập User Thành công!");
}
```

3.10 Bấm Shift + F6 để test form, ở bước này vẫn chưa thể nhập User một cách hoàn chỉnh được.

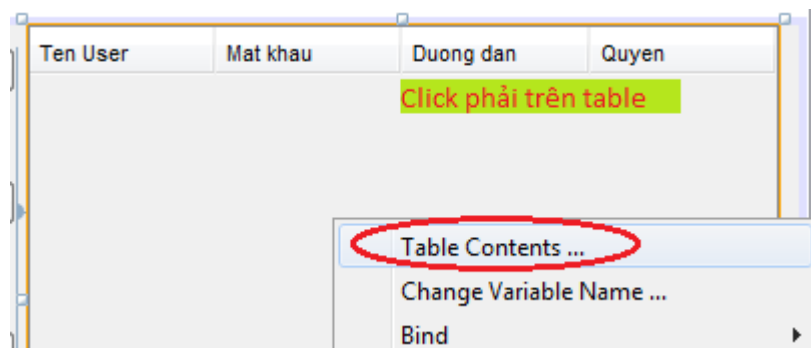
3.11 Ta tiếp tục tạo thêm 1 JFrame Form nữa

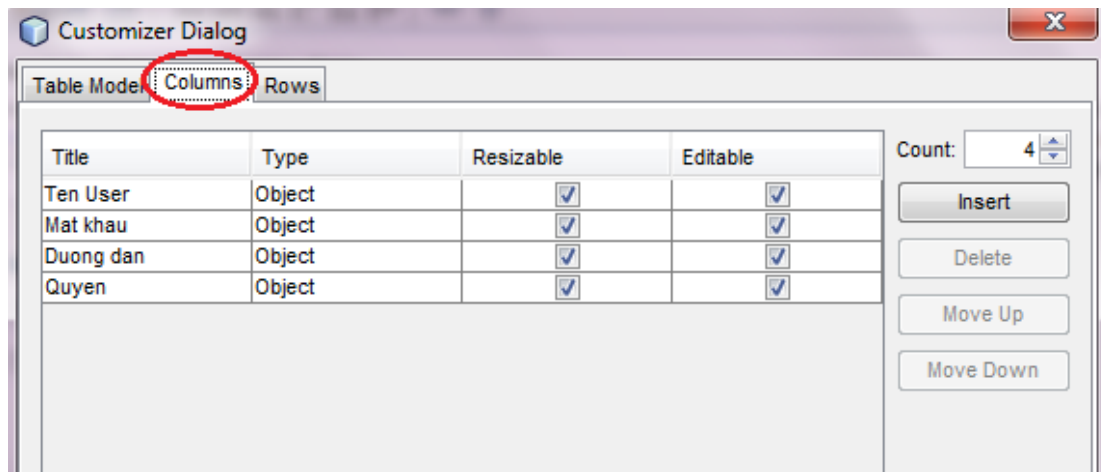


Tiến hành đặt tên cho các radio button, JTextField, Table



Thay đổi nội dung các cột của table bằng cách click phải lên table → Table Contents...





3.12 Sau đó, ta thêm vào phần Source các khai báo thư viện, thuộc tính và constructor.

```
package thuchanhltm;

import java.util.Vector;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author sinhvien
 */
public class frmTimKiemUser extends javax.swing.JFrame {
    private QLUser ql;

    /** Creates new form frmTimKiemUser */
    public frmTimKiemUser() {
        initComponents();
    }

    public frmTimKiemUser(QLUser ql) {
        this.ql=ql;
        initComponents();
    }

    /** This method is called from within the constructor t
```

3.13 Tiếp theo, ta Viết sự kiện click cho các button

Sự kiện click cho button Huy

Click phải lên button Hủy → Events → Action → Action Performed

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
```

Sự kiện cho button Tìm theo tên

Click phải lên button Tìm theo tên → Events → Action → Action Performed

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

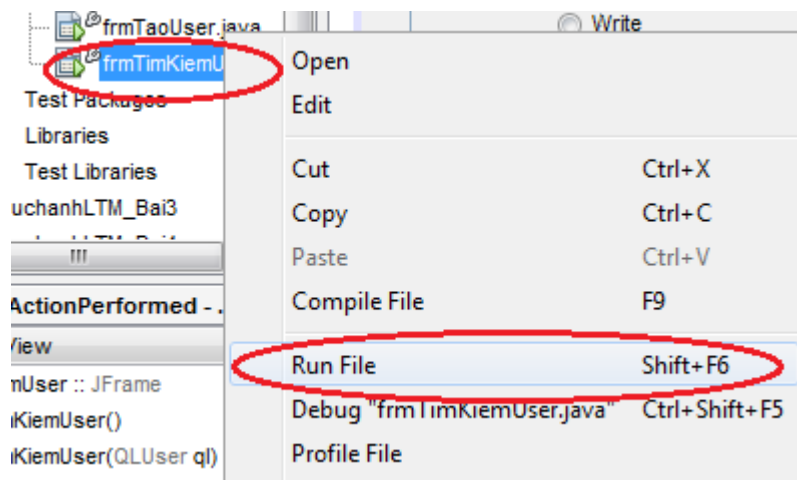
    String ten=txtTen.getText();
    Vector rows=ql.timKiem(ten);
    DefaultTableModel dm=(DefaultTableModel) this.tableKetQua.getModel();
    for(int i=0;i<rows.size();i++)
    {
        dm.addRow( (Vector) rows.get(i) );
    }
    this.validate();
}
```

Sự kiện cho button Tìm theo quyền

Click phải lên button Tìm theo quyền → Events → Action → Action Performed

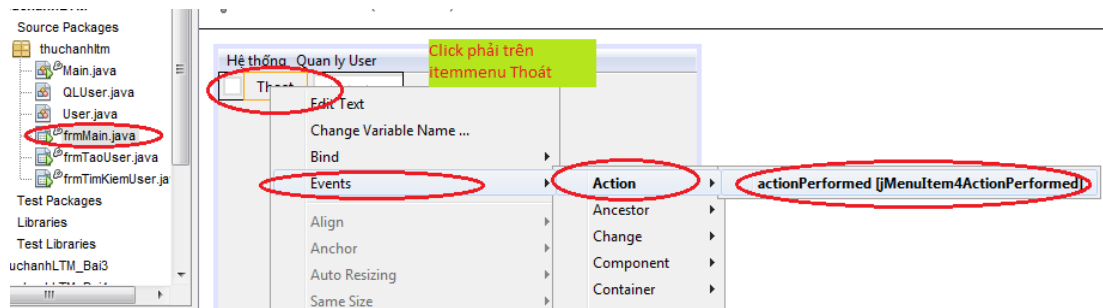
```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    int quyen=2;
    if (radread.isSelected())
        quyen=0;
    else if(radwrite.isSelected())
        quyen=1;
    Vector rows=ql.timKiem(quyen);
    DefaultTableModel dm=(DefaultTableModel) this.tableKetQua.getModel();
    for(int i=0;i<rows.size();i++)
    {
        dm.addRow( (Vector) rows.get(i) );
    }
    this.validate();
}
```

3.14 Chạy file vừa thiết kế :Shift + F6



3.15 Tiếp theo, ta liên kết các Form vừa thiết kế để tạo ra ứng dụng hoàn chỉnh.

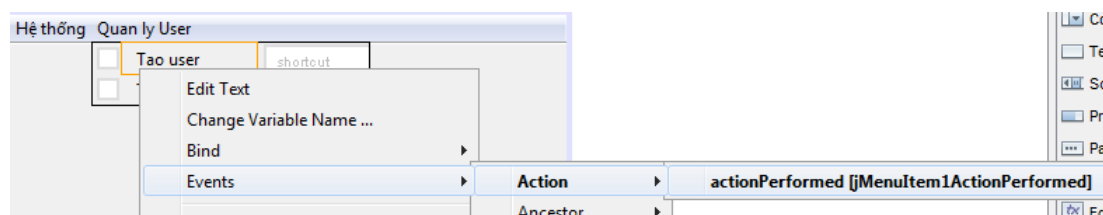
Thêm sự kiện cho item menu Thoát



```
private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0);  
}
```

ở đây, Ta không dùng hàm `this.dispose()`; mà dùng hàm `System.exit(0)`; . Có sự khác biệt giữa hai hàm này: Hàm `dispose()` chỉ đóng Form thôi chứ chương trình chính vẫn chạy bình thường còn hàm `exit(0)`; của đối tượng `System` sẽ thoát luôn ứng dụng và đóng tất cả các Form của ứng dụng.

3.16 Thêm sự kiện cho item menu Tạo User



```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {  
    frmTaoUser obj=new frmTaoUser(q1);  
    obj.setVisible(true);  
}
```

3.17 Thêm sự kiện cho item menu Tìm kiếm User

```
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {  
    frmTimKiemUser obj=new frmTimKiemUser(q1);  
    obj.setVisible(true);  
}
```

3.18 Bấm F6 để chạy thử chương trình.

Bài 2. Viết chương trình quản lý nhân viên. Có 2 loại: Nhân viên biên chế và nhân viên hợp đồng. Các thông tin chung của nhân viên là : họ tên, phòng. Thông tin riêng là:

- NVBC: hệ số lương, số năm CT.
- NVHD: lương hợp đồng, Loại HĐ(NH,DH).

Chương trình thực hiện được các chức năng:

- Tạo lập và lưu các nhân viên
- Liệt kê danh sách nhân viên.
- Liệt kê danh sách nhân viên theo loại: HĐ, BC
- Tính tổng lương toàn bộ nhân viên.
- Liệt kê danh sách nhân viên hợp đồng dài hạn.