



Bộ Công Thương
Trường Đại học Công nghiệp Thực phẩm
TP. Hồ Chí Minh

Khoa: Công nghệ Thông Tin.

Môn học: **Hệ quản trị cơ sở dữ liệu Oracle.**

ORACLE®

DATABASE

Đồ án:
ỨNG DỤNG QUẢN LÝ THƯ VIỆN.

GVHD: Nguyễn Phuong Hạc.

Nhóm thực hiện: 09.
2001190424 - Phạm Gia Bảo (Nhóm trưởng).
2001190464 - Phạm Gia Đạt.
2001190421 - Nguyễn Lê Gia Bảo.

LỜI MỞ ĐẦU

Từ trước tới nay, các thư viện dù lớn hay nhỏ đều chỉ hoạt động dựa theo cách thủ công do con người phụ trách. Nhưng với sự phát triển mạnh mẽ của ngành Công nghệ Thông tin, dẫn đến việc cần phải có các hệ thống, các phần mềm để hỗ trợ con người trong việc quản lý. Xuất phát từ nhu cầu thực tế, cần phải có một hệ thống trợ giúp cho việc xuất nhập, thao tác trong các khâu của thư viện. Chúng em đã thực hiện đồ án “*Quản lý thư viện*” mục tiêu để tìm hiểu sâu hơn về lĩnh vực pháp tích thiết kế hệ thống thông tin. Và vì là 1 đồ án đầu tay nên khó tránh khỏi nhiều sự sai sót, khuyết điểm mong cô thông cảm, hết lòng góp ý và chỉ cho chúng em các ưu, khuyết điểm để giúp chúng em có thể hoàn thiện tốt hơn sau này.

Mục lục

I. GIỚI THIỆU.....
1. Phạm vi đồ án
2. Mục tiêu
3. Lý do chọn đề tài
II. THIẾT KẾ
1. Kết nối đến Oracle
2. Giao diện
III. KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN.....
1. Kết luận
2. Định hướng phát triển
IV. TÀI LIỆU THAM KHẢO
V. BIÊN BẢN

I. GIỚI THIỆU:

1. Phạm vi đồ án:

* Xây dựng ứng dụng Quản lý thư viện, với các chức năng như:

- Đăng nhập hệ thống với quyền quản trị, người dùng.
- Thêm, xóa, sửa, tìm kiếm.
- Quản lý mượn, trả sách.
- Thống kê, in thông kê.

2. Mục tiêu:

- Tạo nên một ứng dụng hỗ trợ thủ thư cập nhật thông tin sách, xác nhận cho mượn sách và nhận lại sách khi độc giả trả sách. Hỗ trợ quản lý các thông tin về độc giả dựa trên thẻ độc giả, thông tin phiếu mượn. Hỗ trợ các chức năng quản trị chung hệ thống, trong đó người quản trị có thể thay đổi thông tin hoặc thêm bớt các thủ thư.

- Độc giả có thể tra cứu thông tin sách trên môi trường mạng nội bộ của thư viện. Tuy nhiên, việc mượn và trả sách phải thực hiện trực tiếp trên thư viện. Thủ thư sử dụng hệ thống để cập nhật và quản lý quá trình mượn trả sách. Thông tin thống kê phải đảm bảo tính chính xác, khách quan.

3. Lý do chọn đề tài:

- Nhóm chúng em chọn đề tài với mong muốn mang lại sự thống nhất và chính xác trong việc cập nhật và thao tác trên cơ sở dữ liệu. Giúp thủ thư dễ dàng thao tác, dễ dàng cập nhật và hạn chế tối đa những sai sót.

II. THIẾT KẾ:

1. Kết nối đến Oracle:

1.1. Tải gói bổ sung:

- Truy cập đường dẫn:

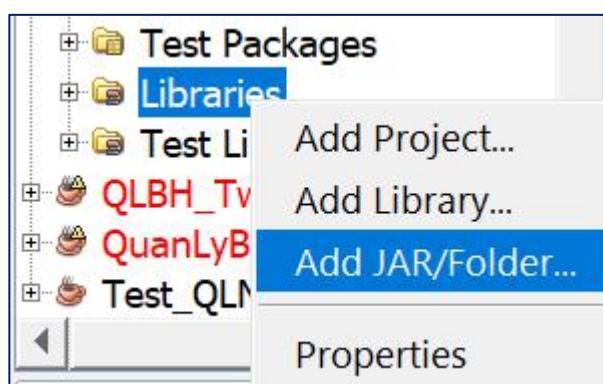
<https://www.oracle.com/database/technologies/jdbc-ucp-122-downloads.html>

- Click chọn “ojdbc8-full.tar.gz” để tiến hành download.

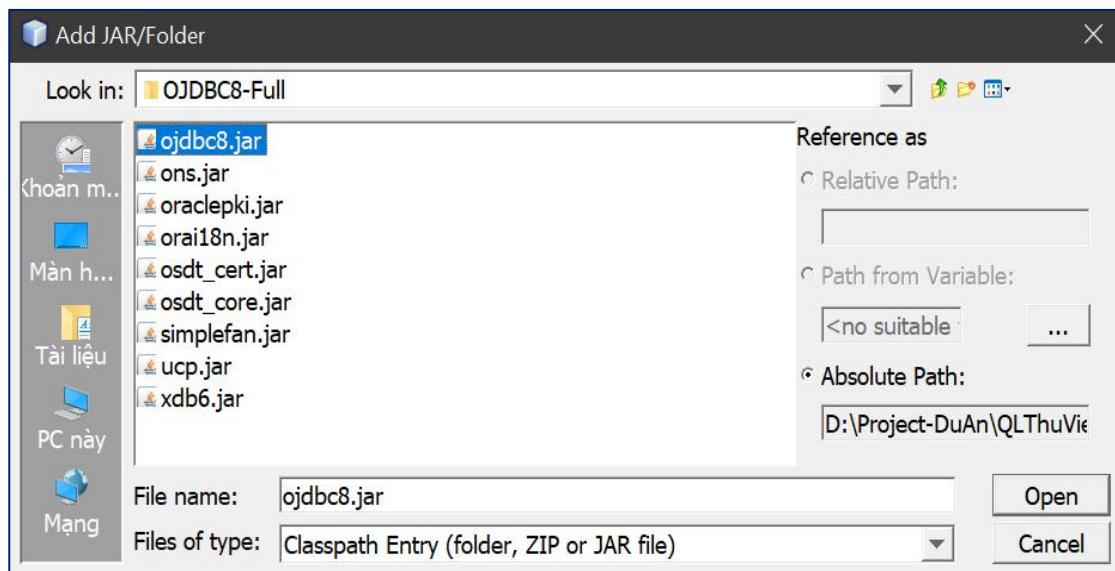
The screenshot shows the Oracle Database 12.2.0.1 JDBC Driver & UCP Downloads page. At the top, there is a navigation bar with links for Products, Industries, Resources, Support, Events, and Developer. Below the navigation bar, the URL is https://www.oracle.com/database/technologies/jdbc-ucp-122-downloads.html. The main content area is titled "Oracle Database 12.2.0.1 JDBC Driver & UCP Downloads". Underneath, it says "Zipped JDBC Driver and Companion JARs". A table lists a single item: "ojdbc8-full.tar.gz" with a download link and a "Release Notes" link. The "Release Notes" link points to a page with the following text:
(7,287,468 bytes) -
(SHA1 Checksum: 1e210e431d4a67ee13a71c5bdef8da220abe54c)
The TAR archive contains the latest 12.2.0.1 JDBC Thin driver (**ojdbc8.jar**), Universal Connection Pool (**ucp.jar**), other companion jars, and README that has more information about the contents of the tar file.

1.2. Kết nối Oracle đến Java (Netbeans):

- Import thư viện: (Libraries → Add JAR/Folder)



- Nhấn “Open” để hoàn tất:

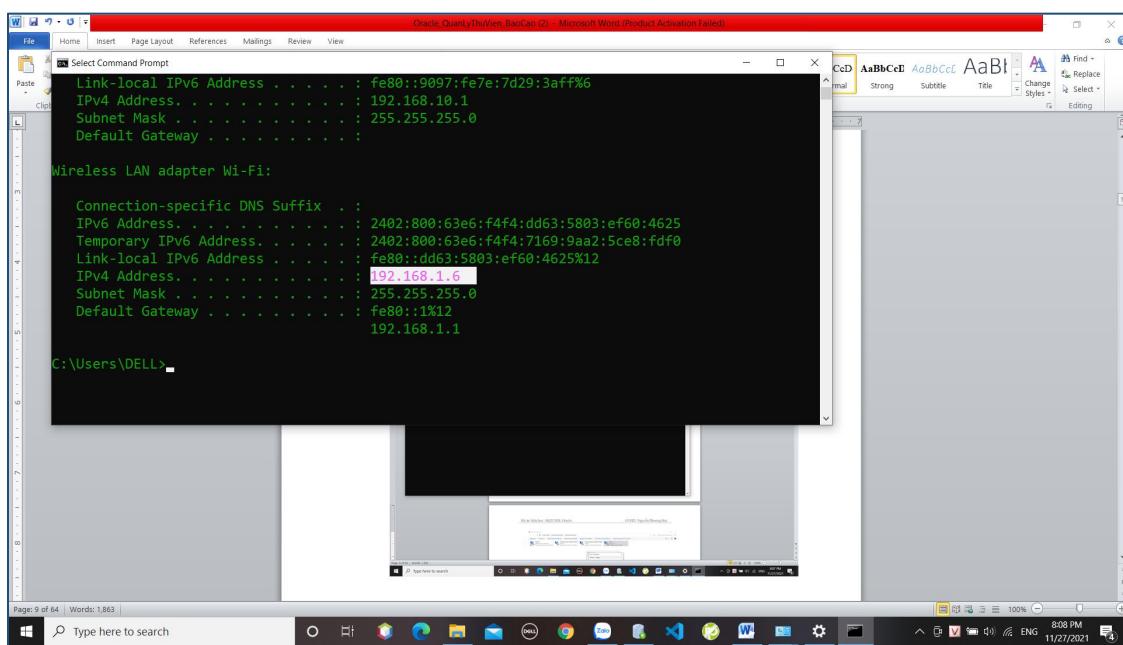
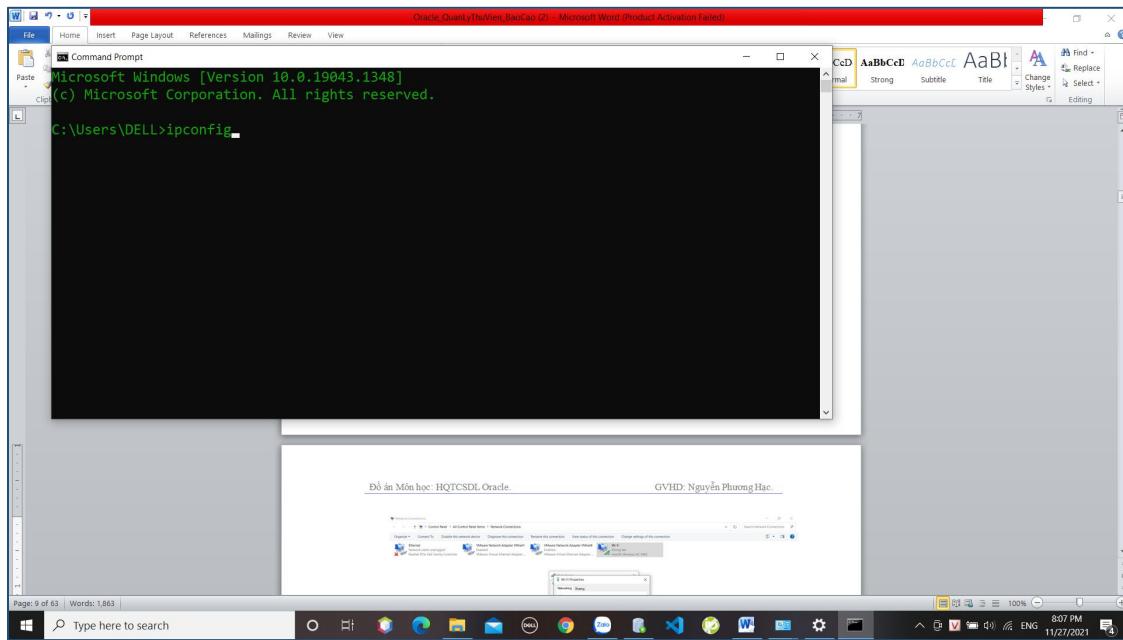


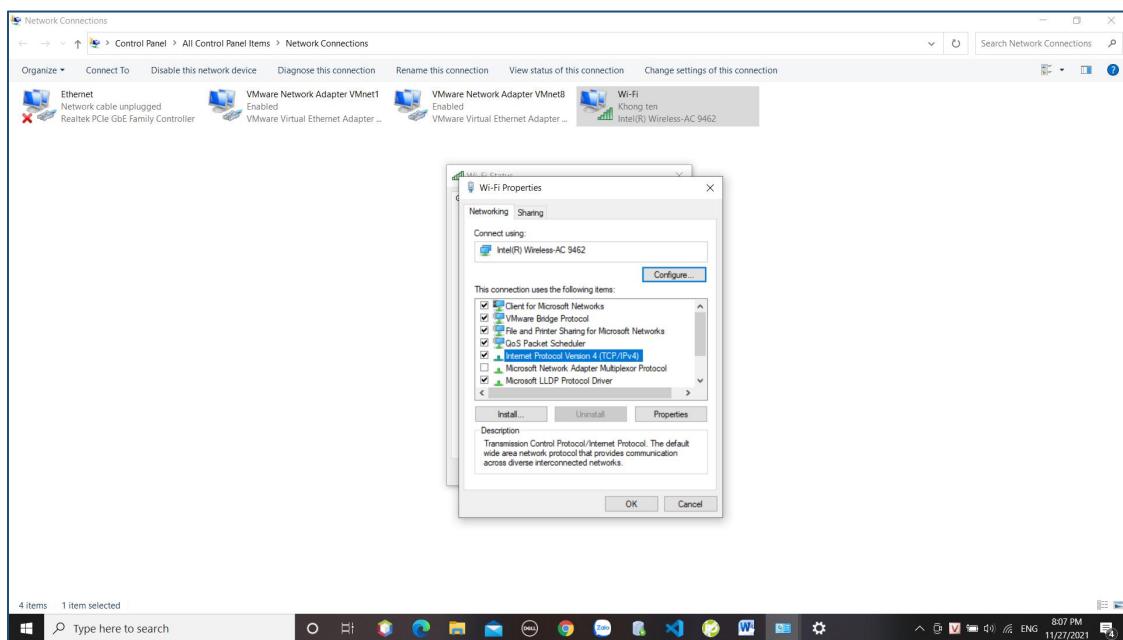
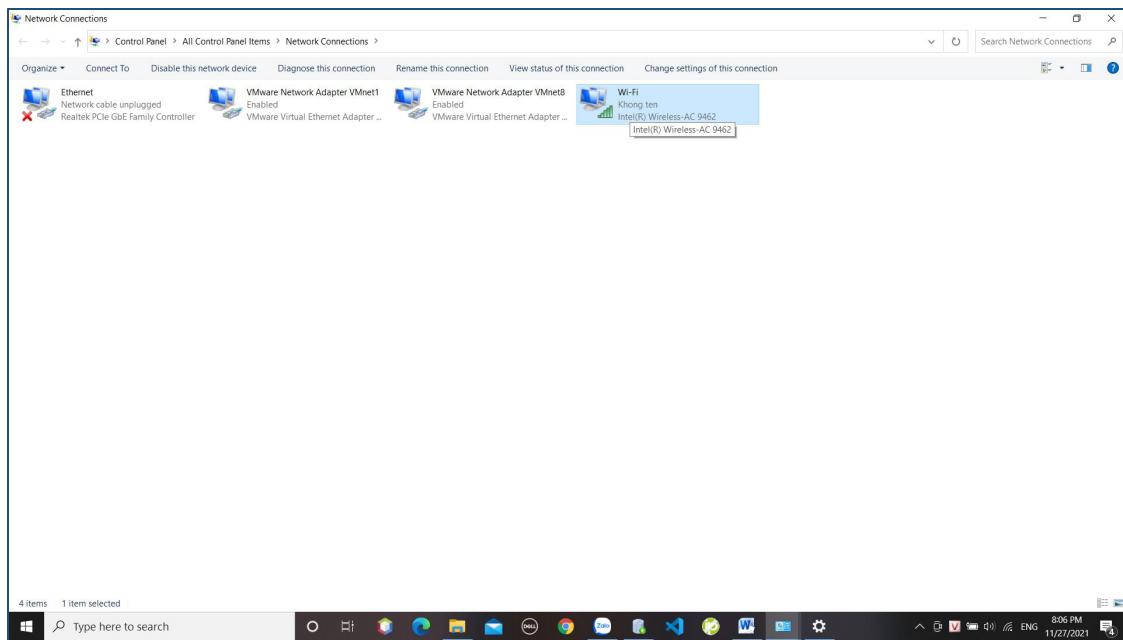
- Import thành công:

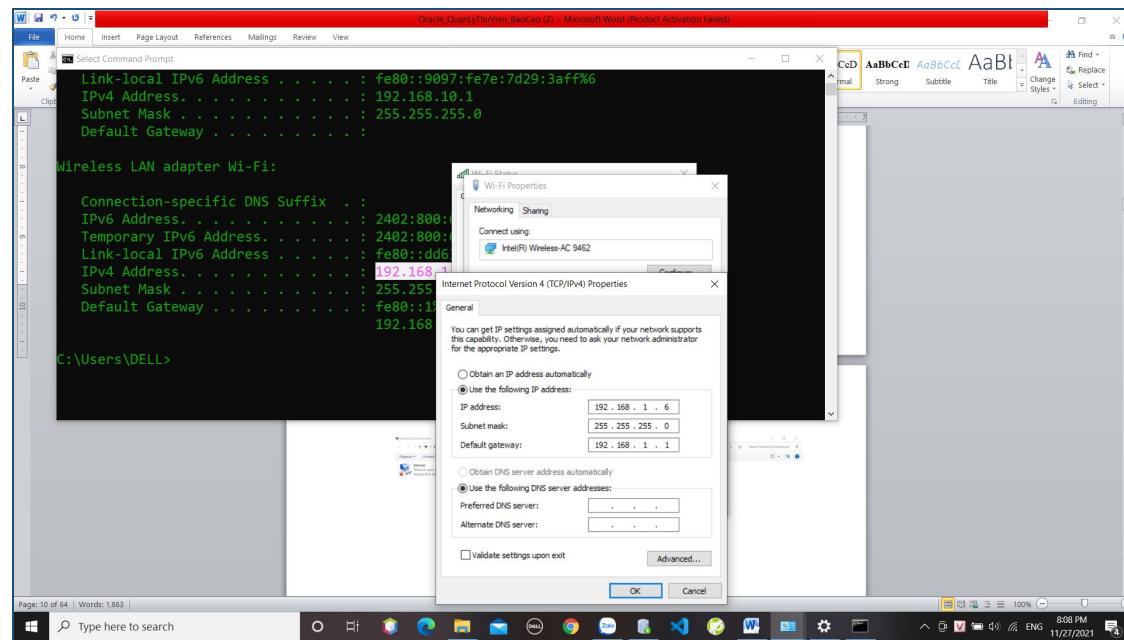


* Thực hiện kết nối từ xa:

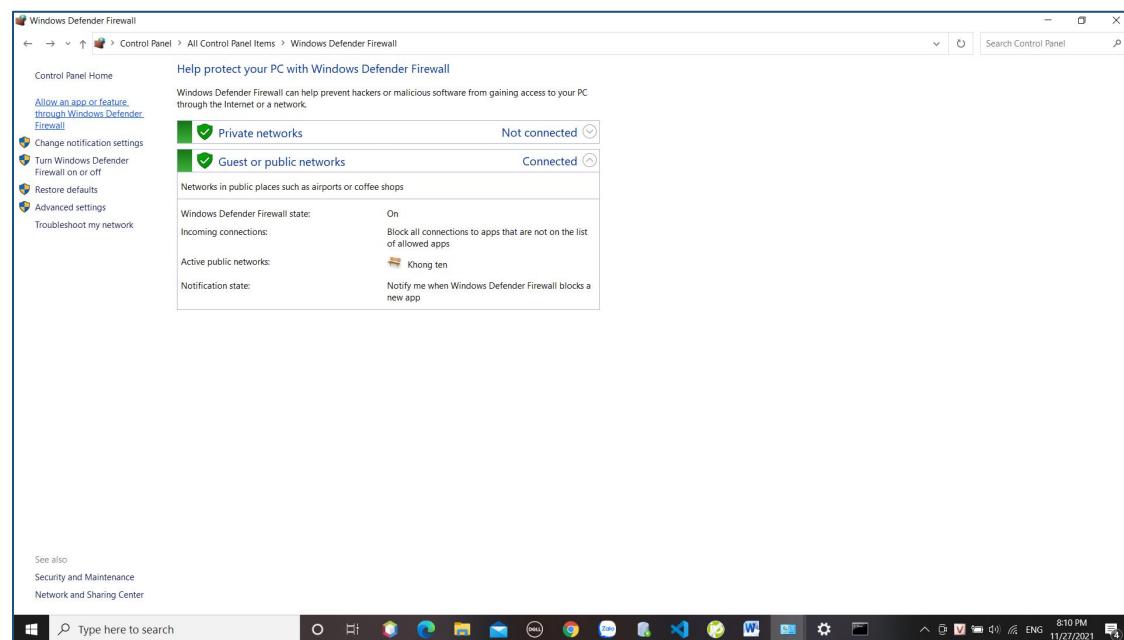
- Thiết lập địa chỉ IP tĩnh:



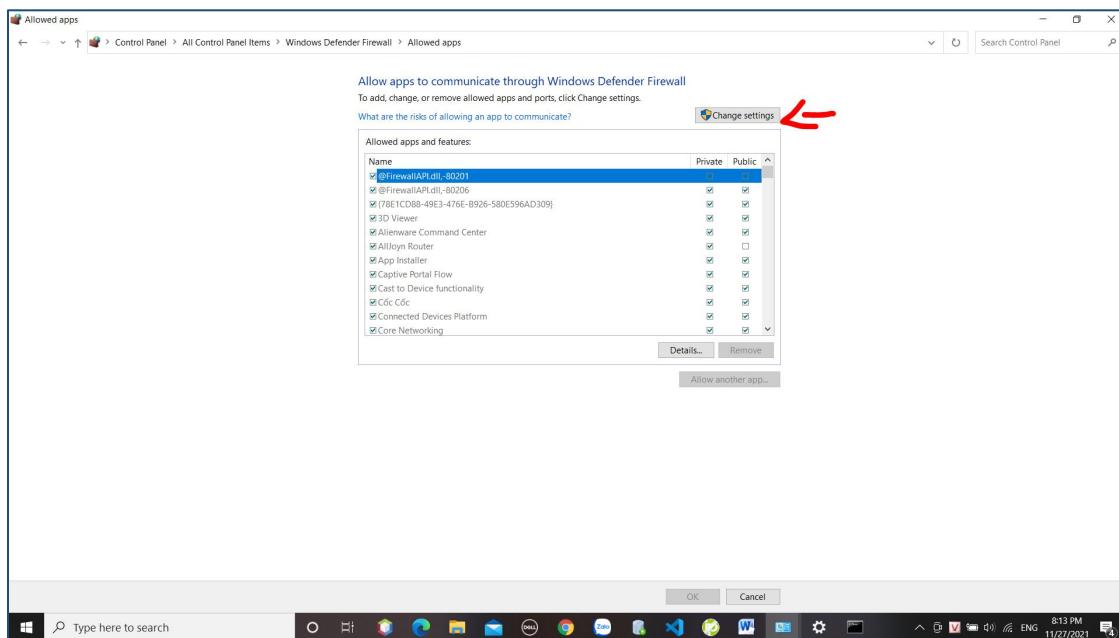
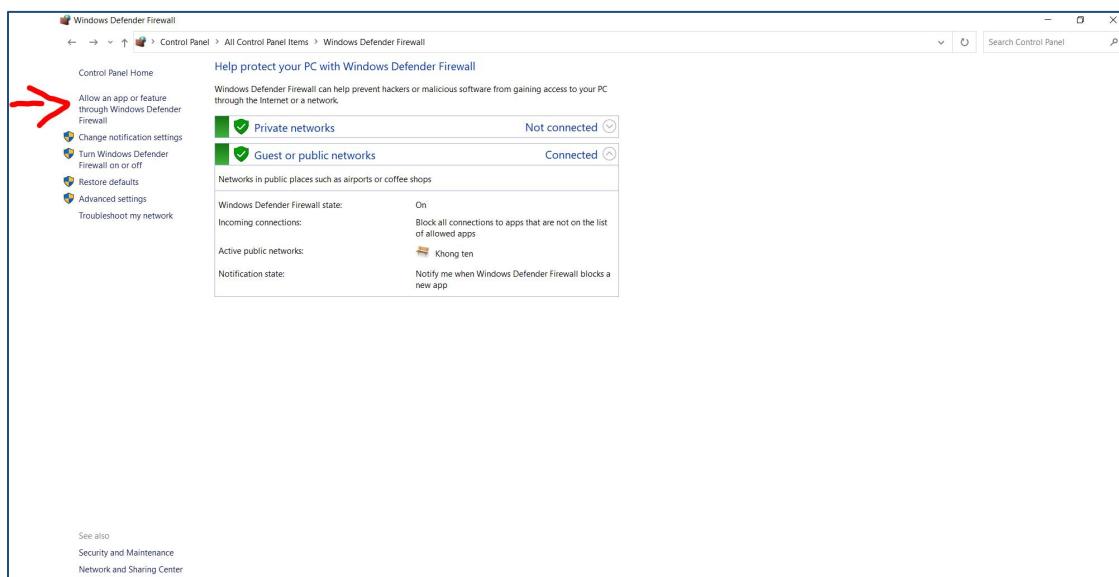


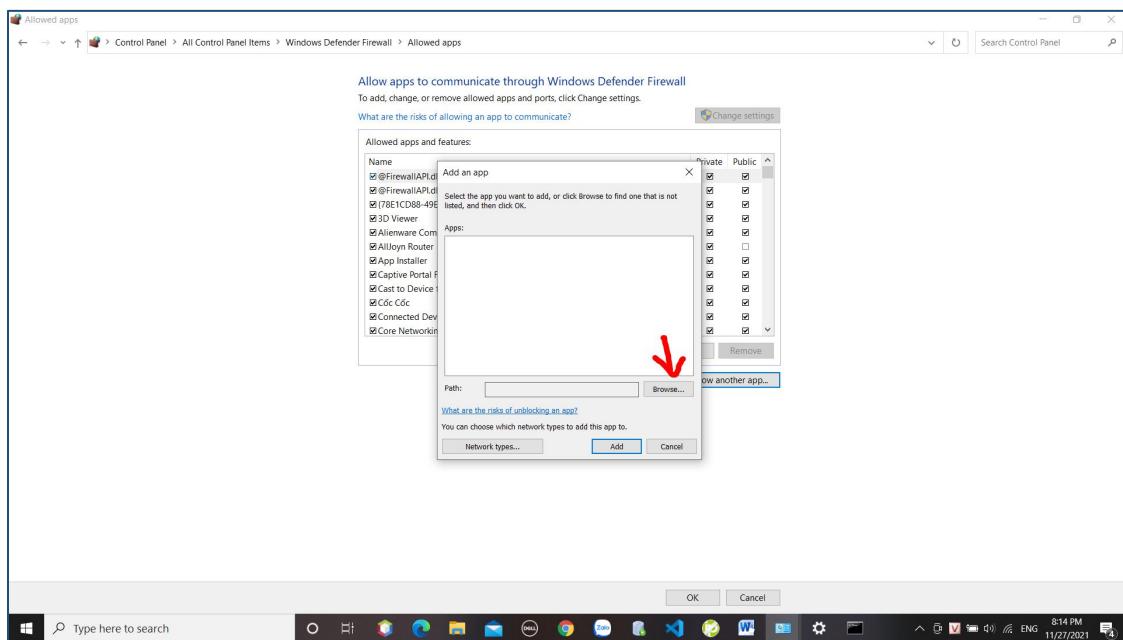
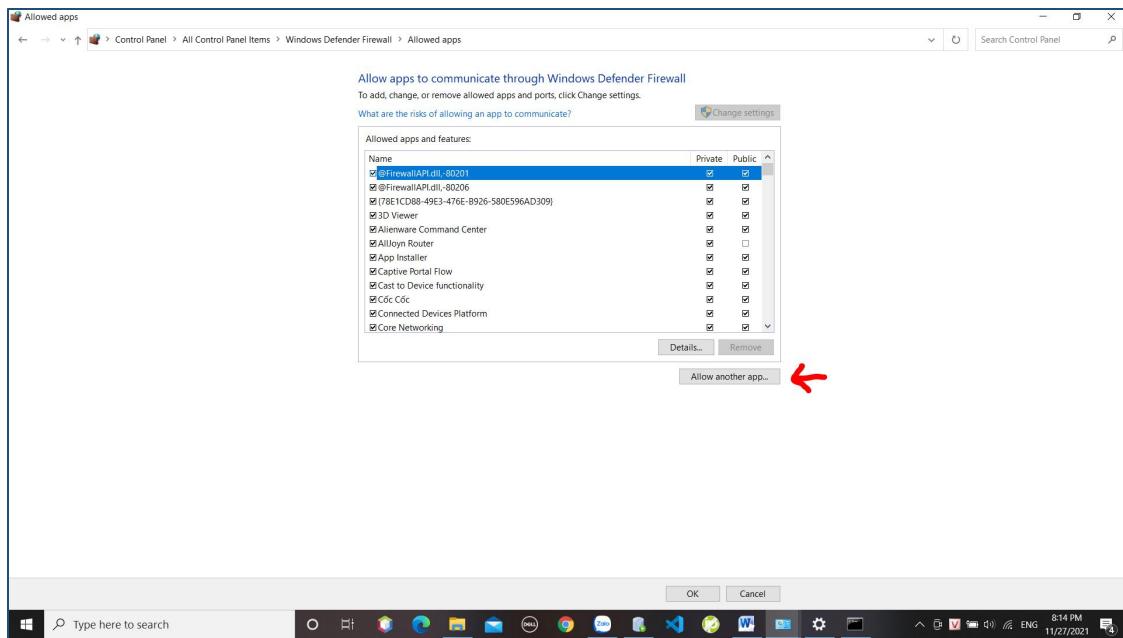


- Mở tường lửa vào Windows Defender Firewall

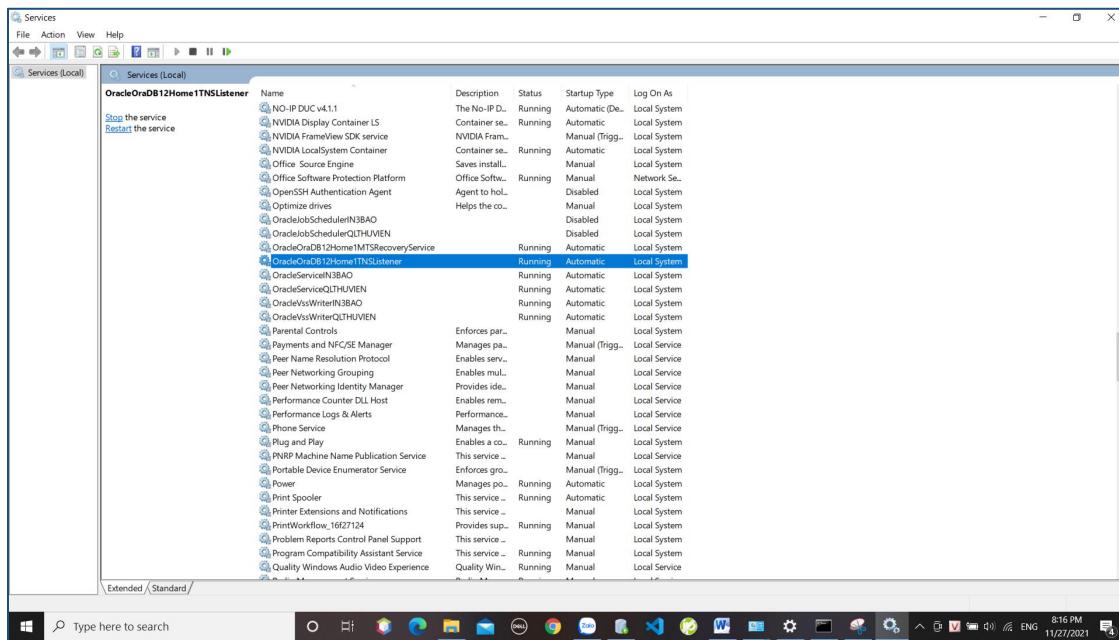
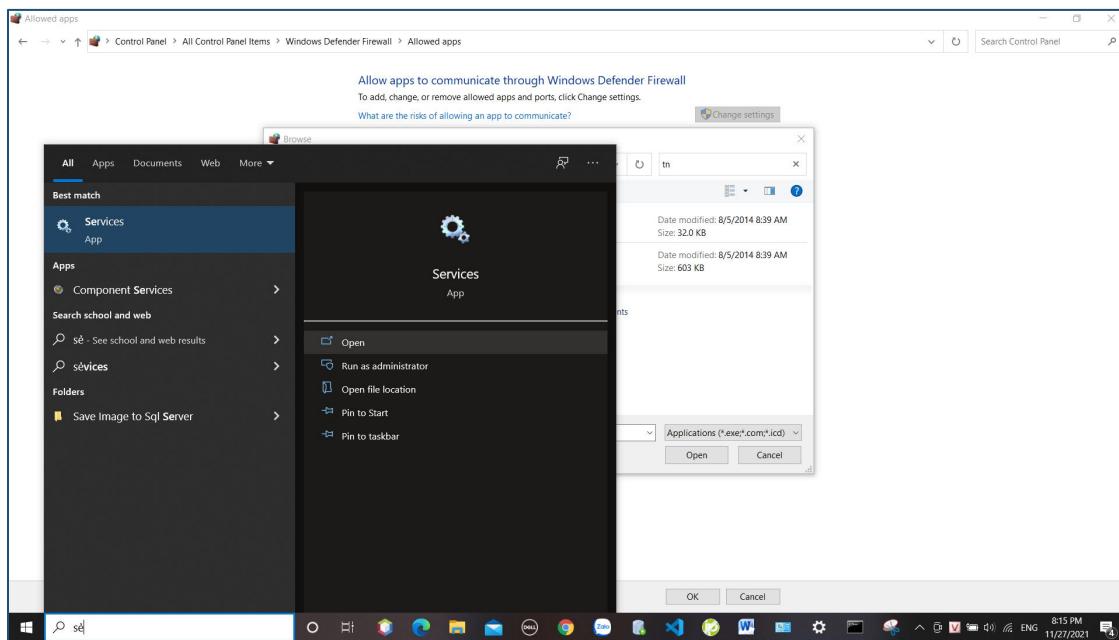


- Chọn Allowed apps.

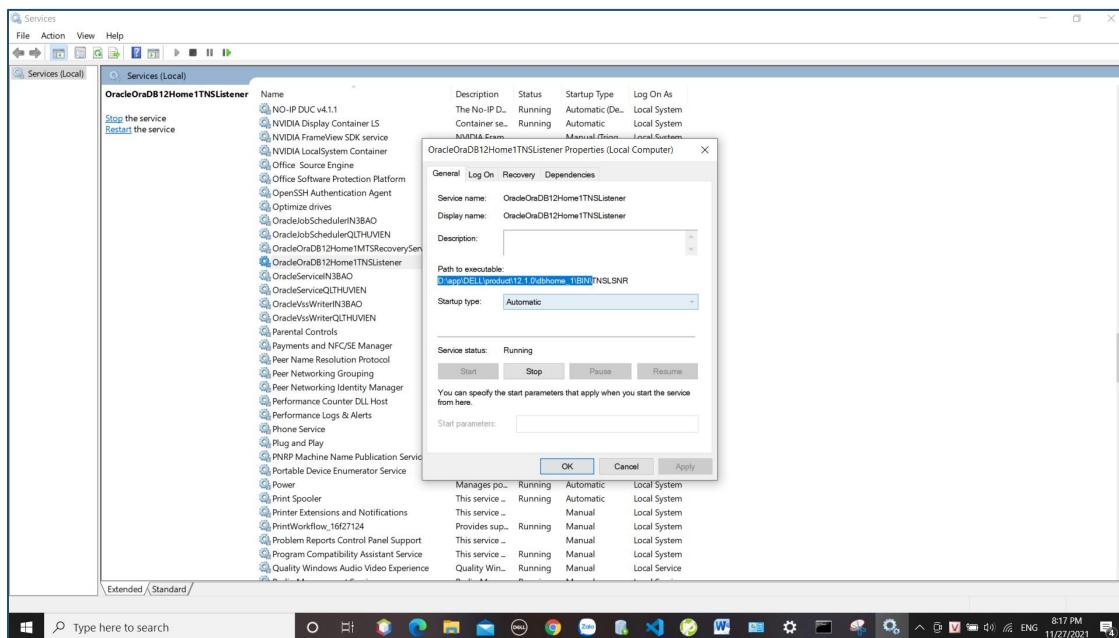




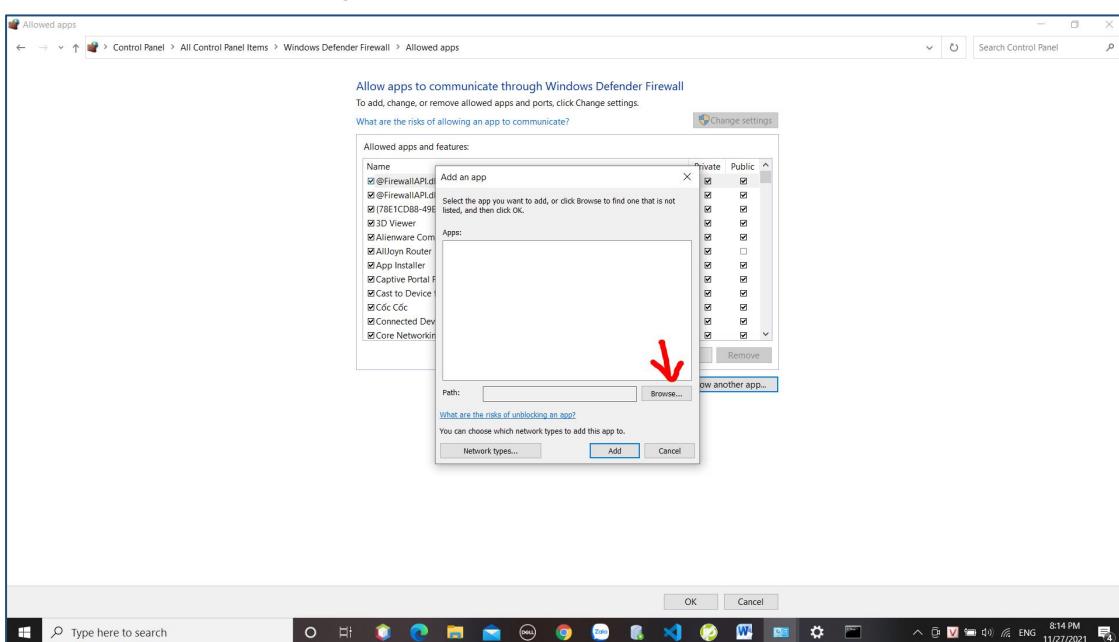
- Tìm đường dẫn của TNSListener

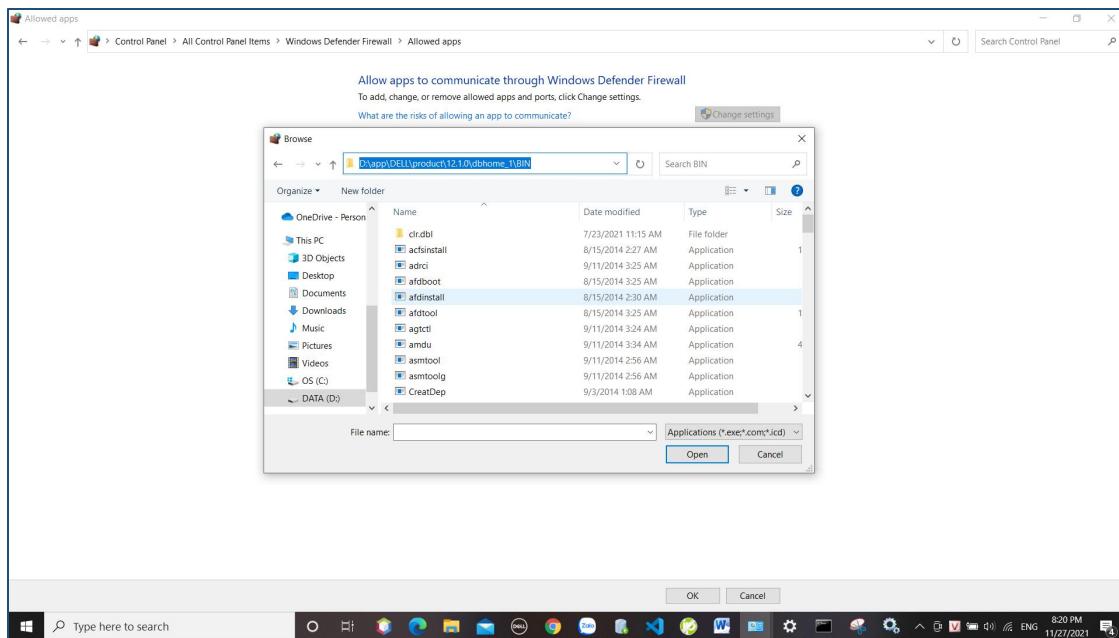


- Copy đường dẫn này

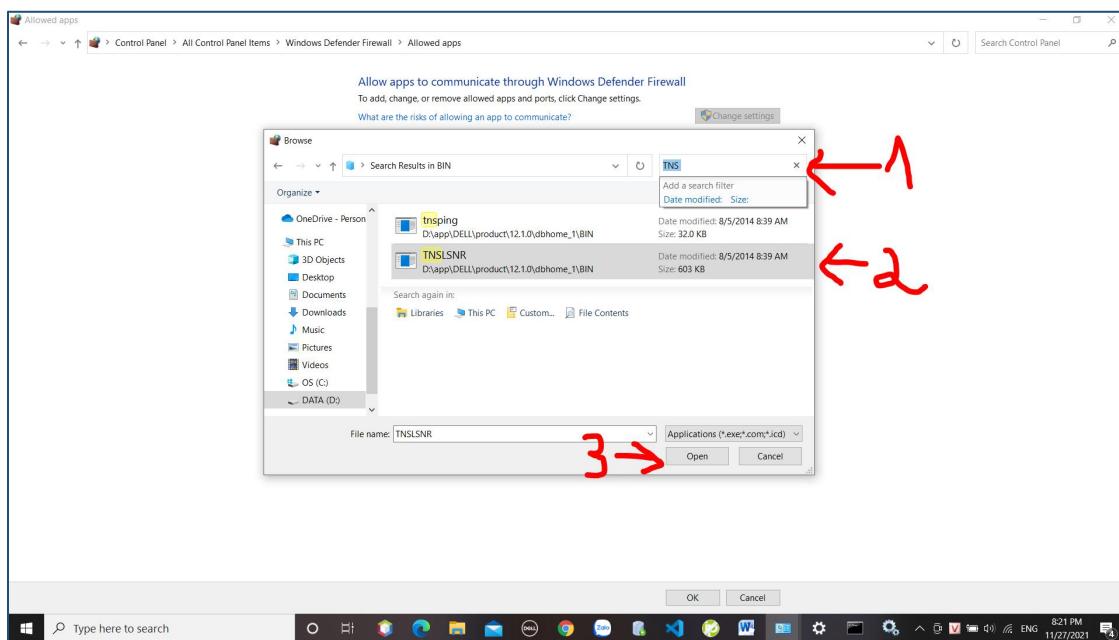


- Gán vào đường dẫn tìm kiếm

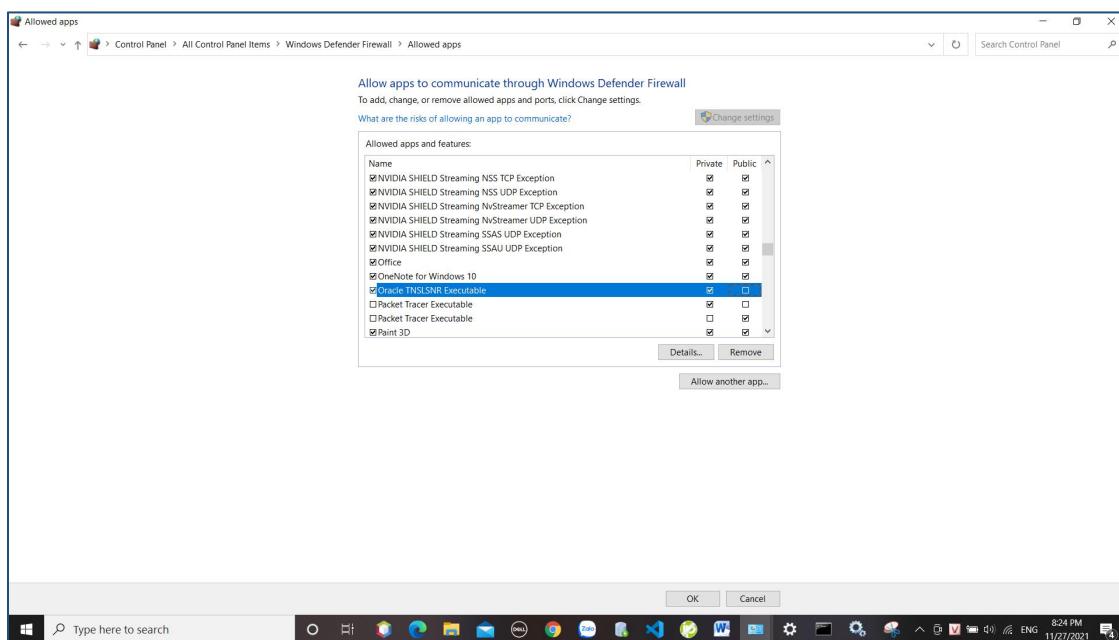
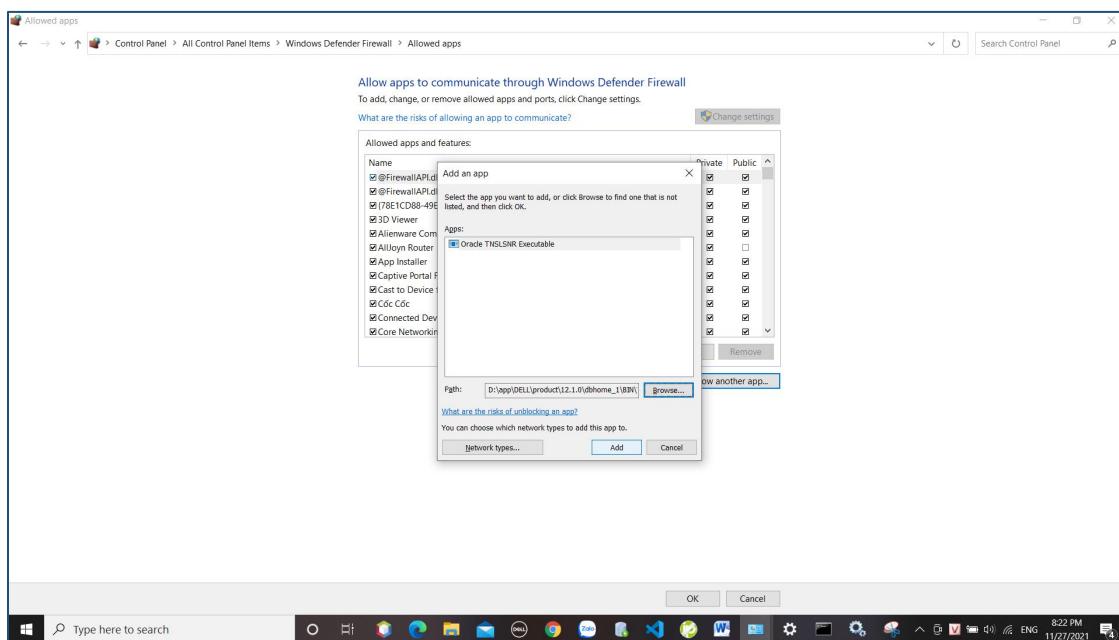




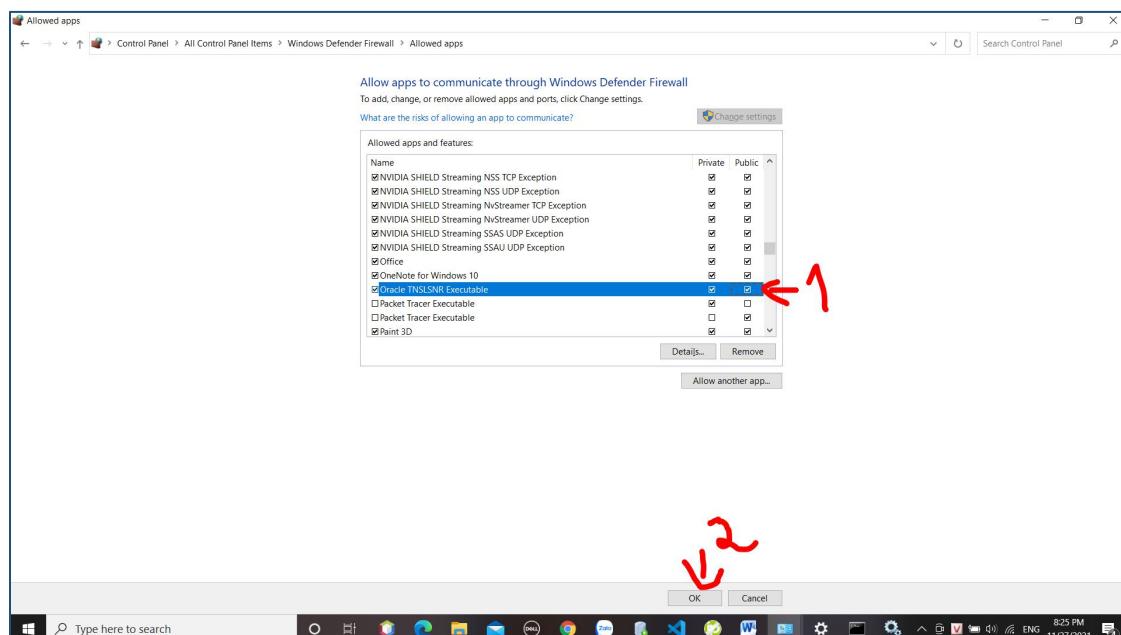
- Chọn file TNSLSNR



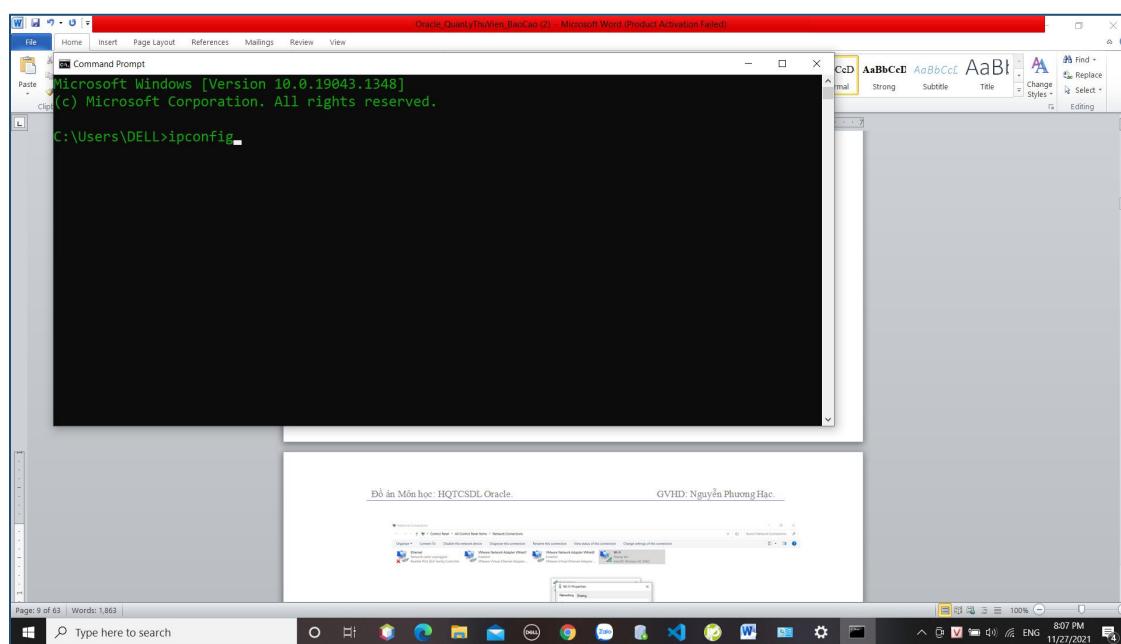
- Chọn add



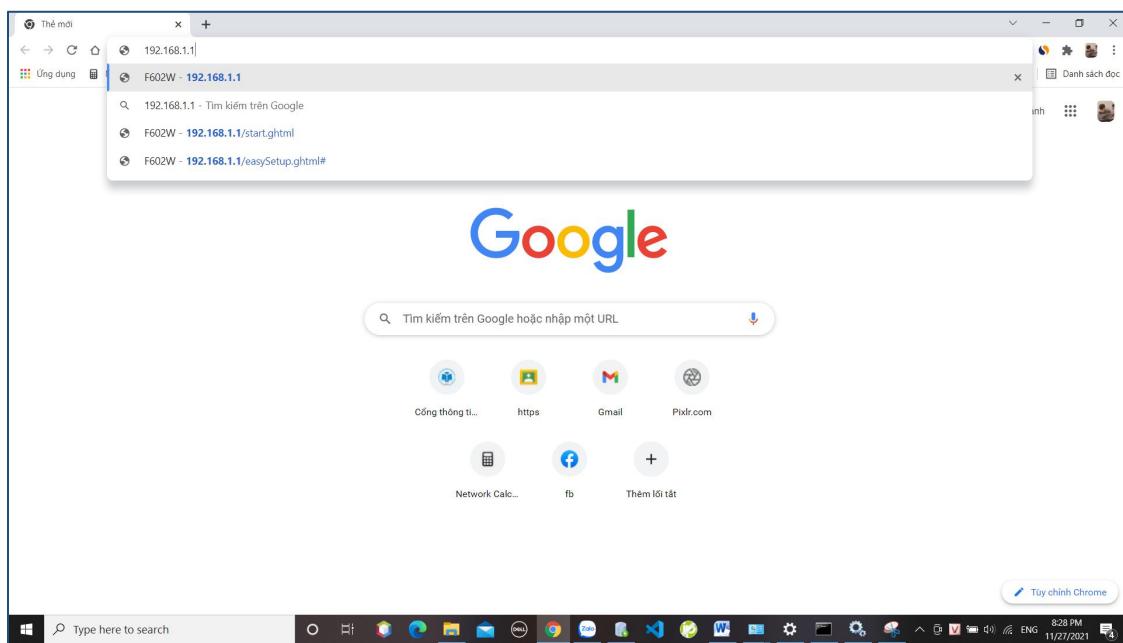
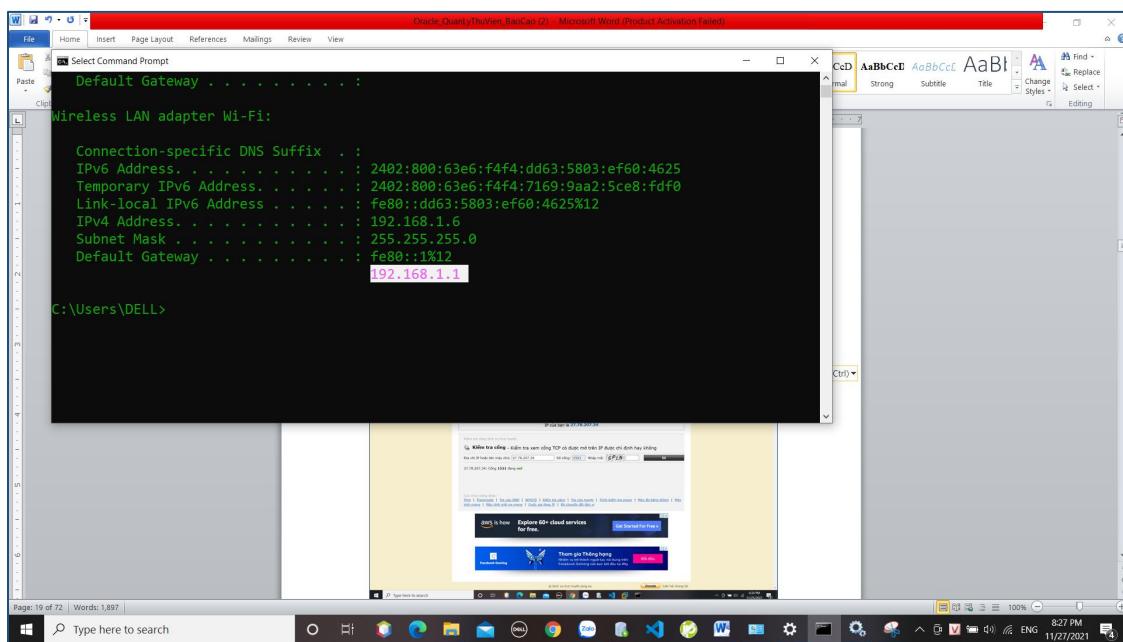
- Tích vào ô public và bấm OK.



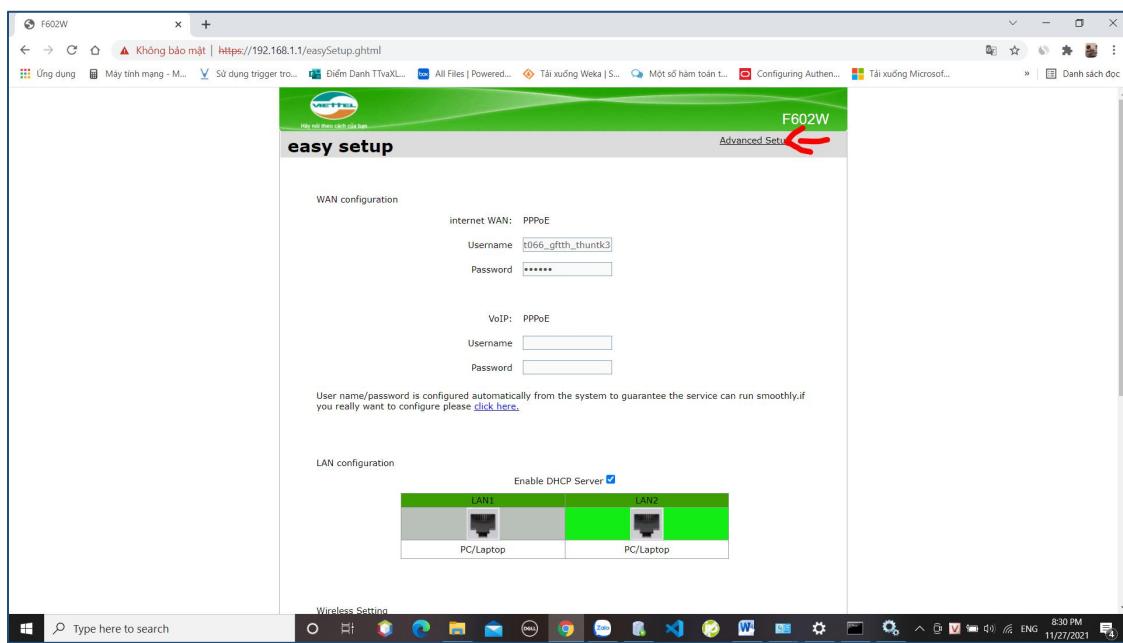
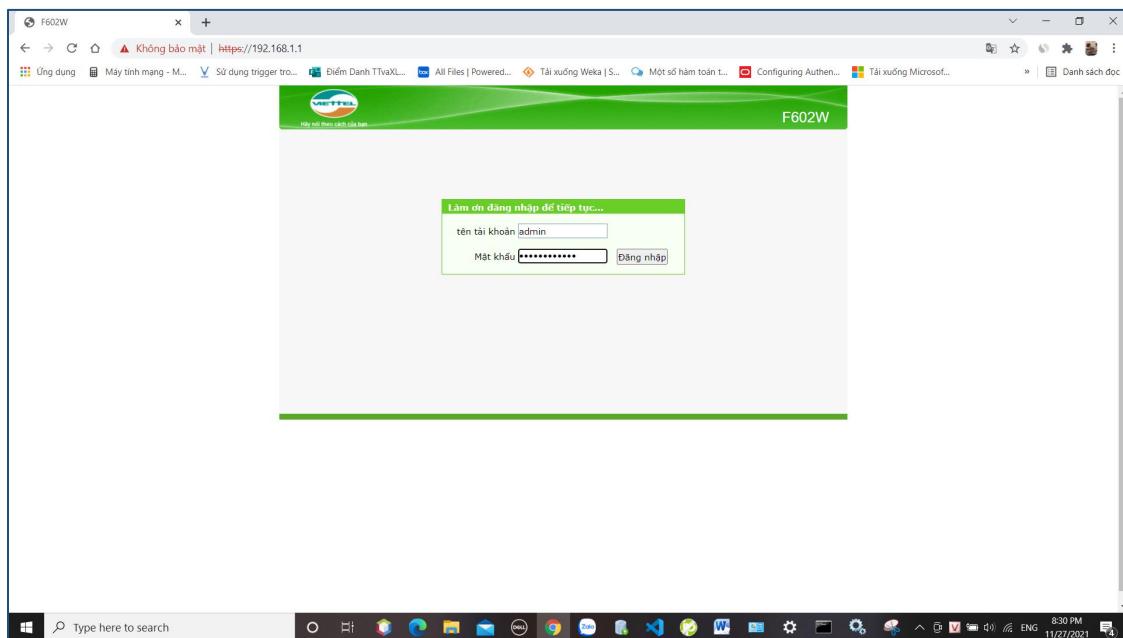
- Mở port 1521.



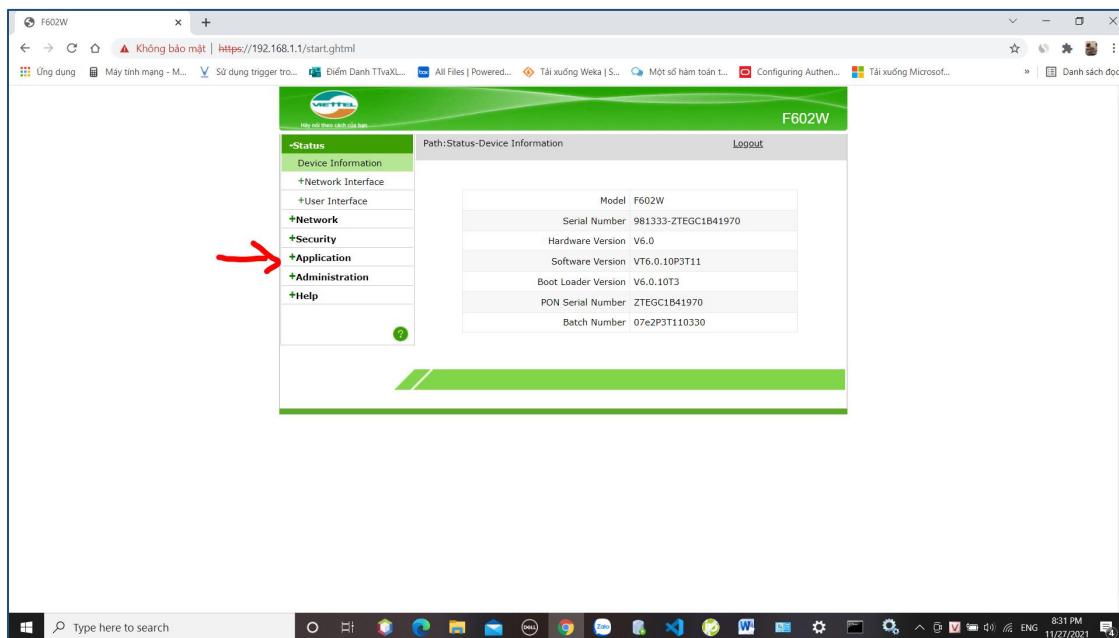
- Đăng nhập vào router thường là địa chỉ của defauft gateway



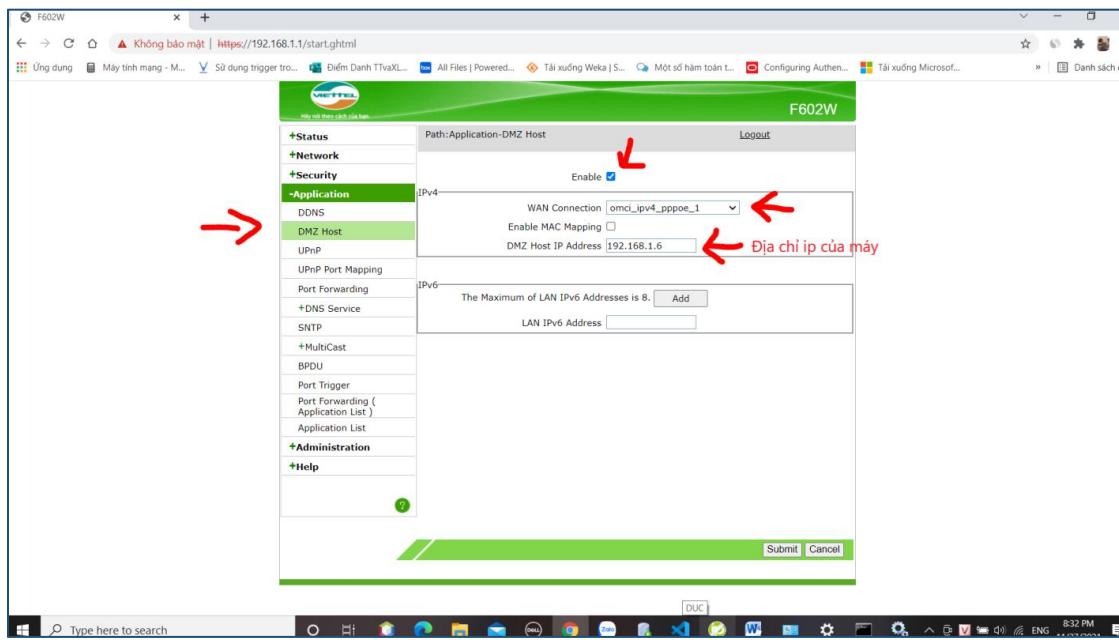
- Tài khoản là admin còn password nằm dưới router (Viettel)



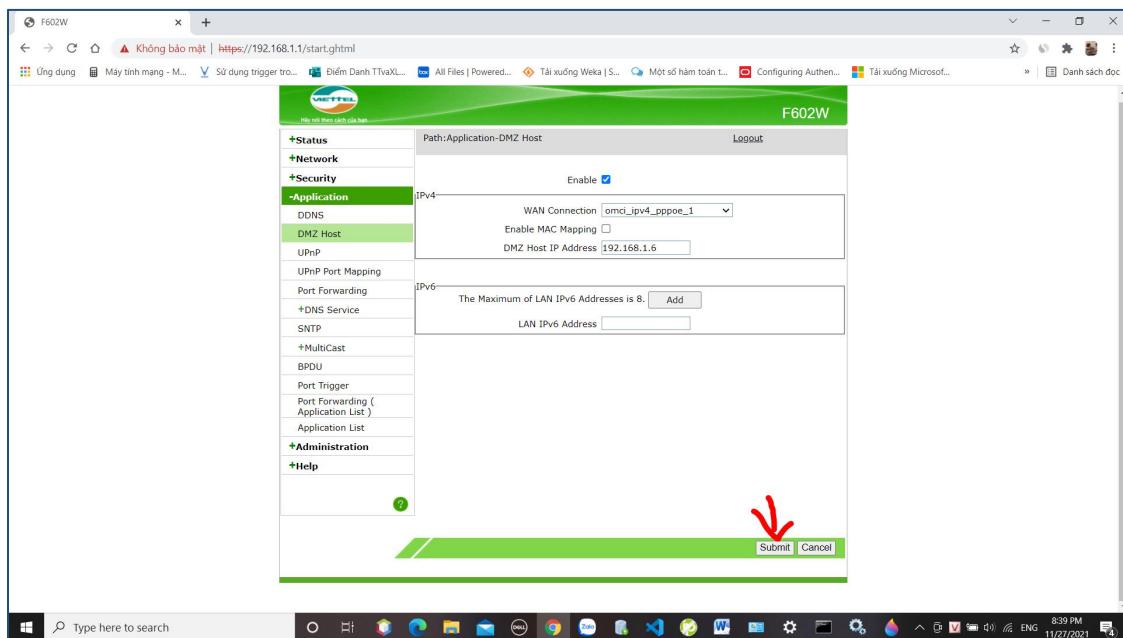
- Chọn Application



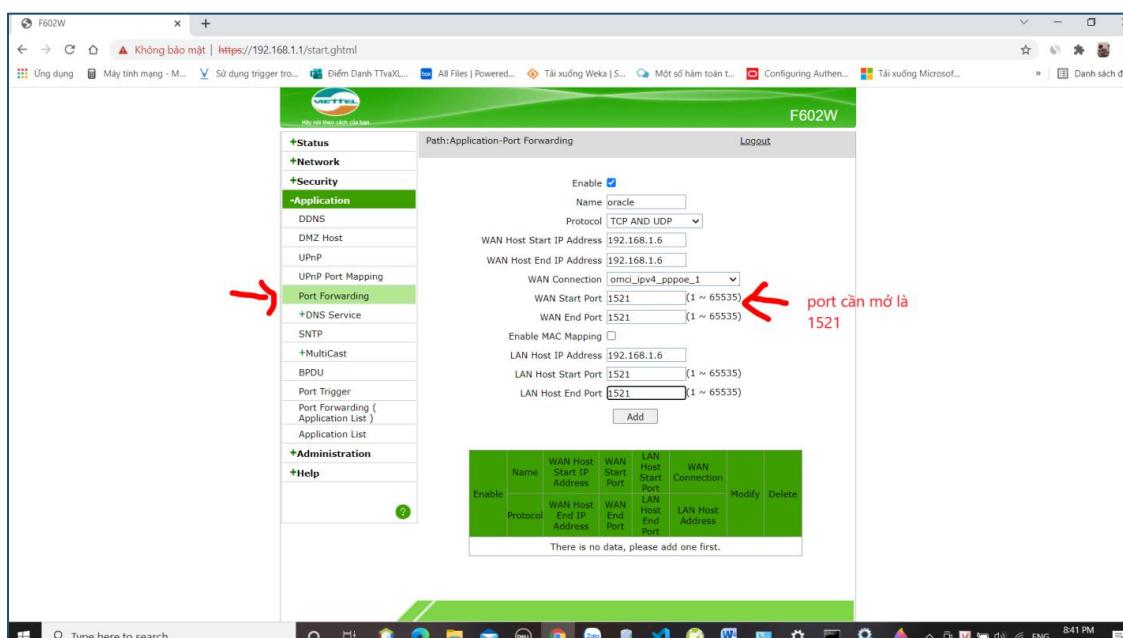
- Chọn DMZ Host



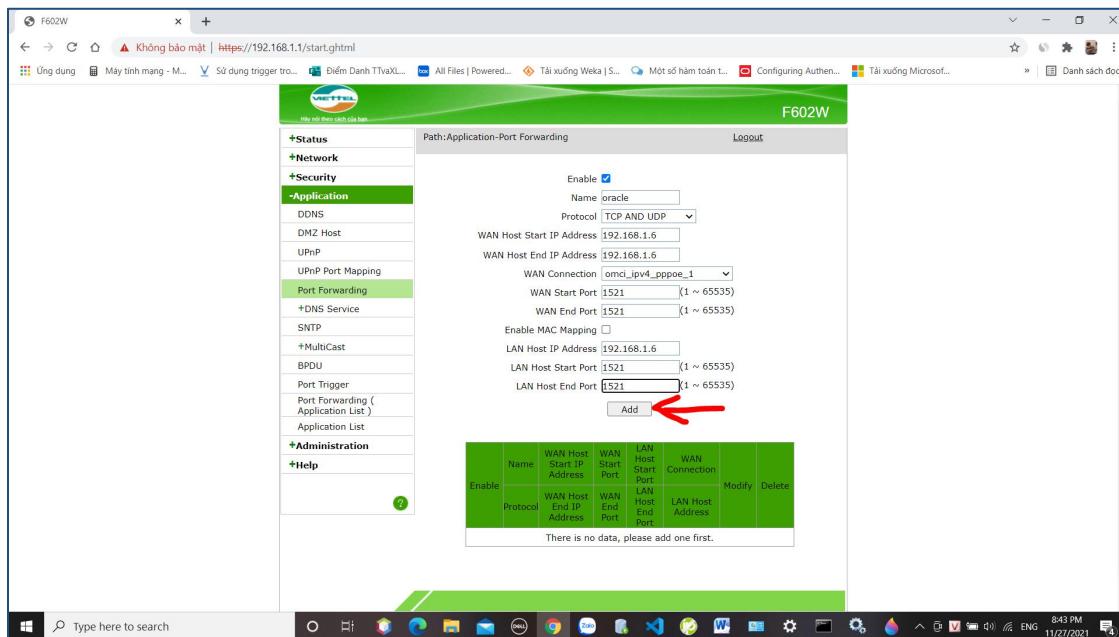
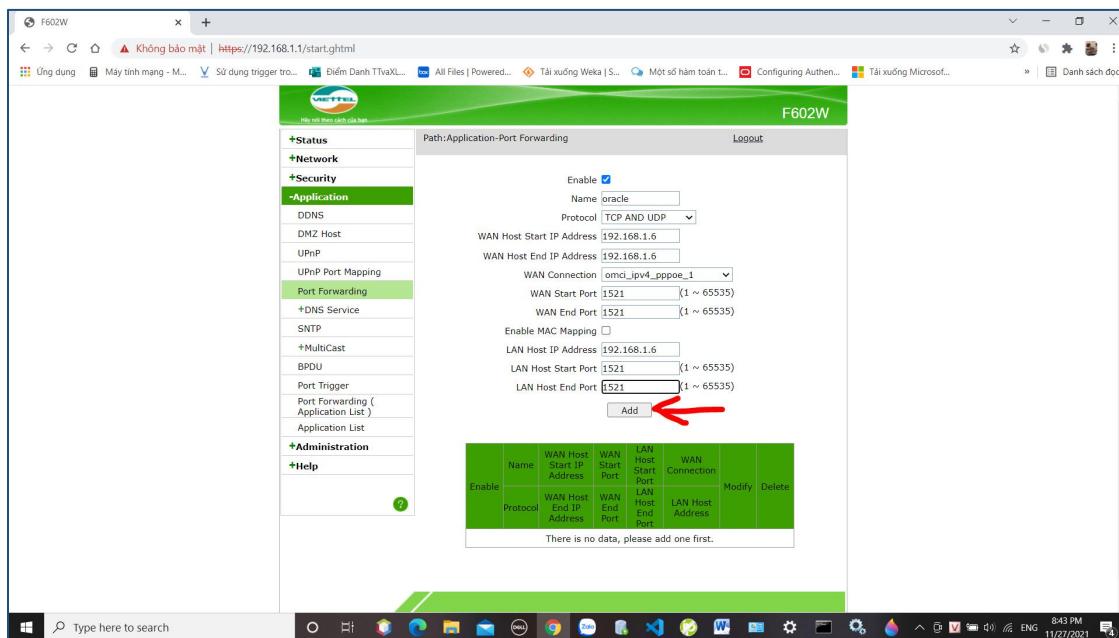
- Và click chọn Submit.



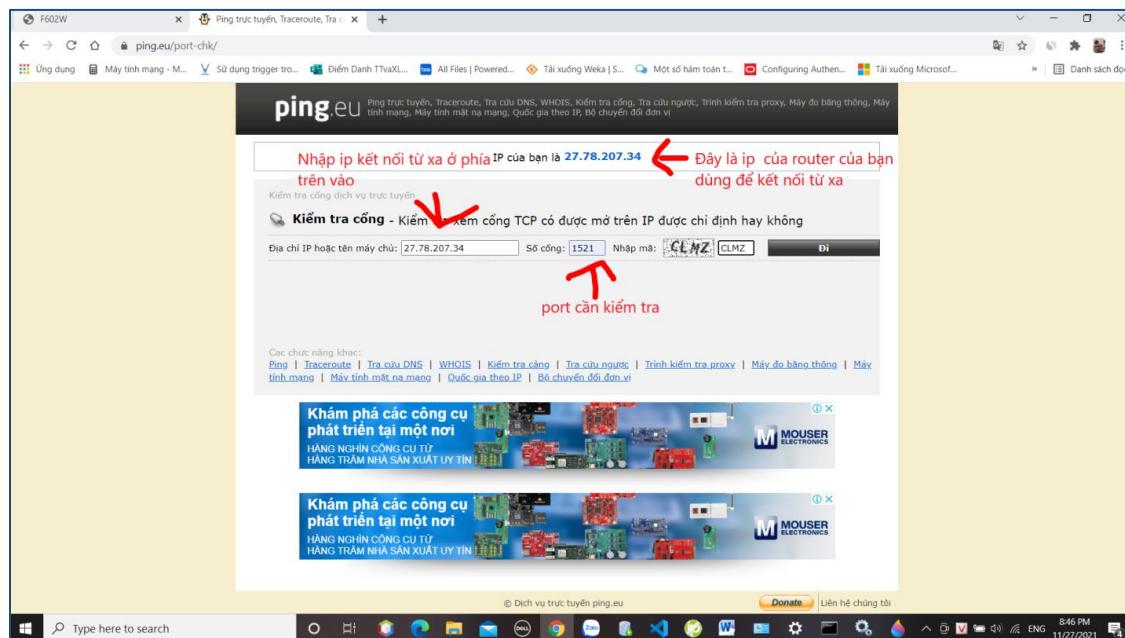
- Chọn Port Forwarding Để tiến hành mở port 1521.



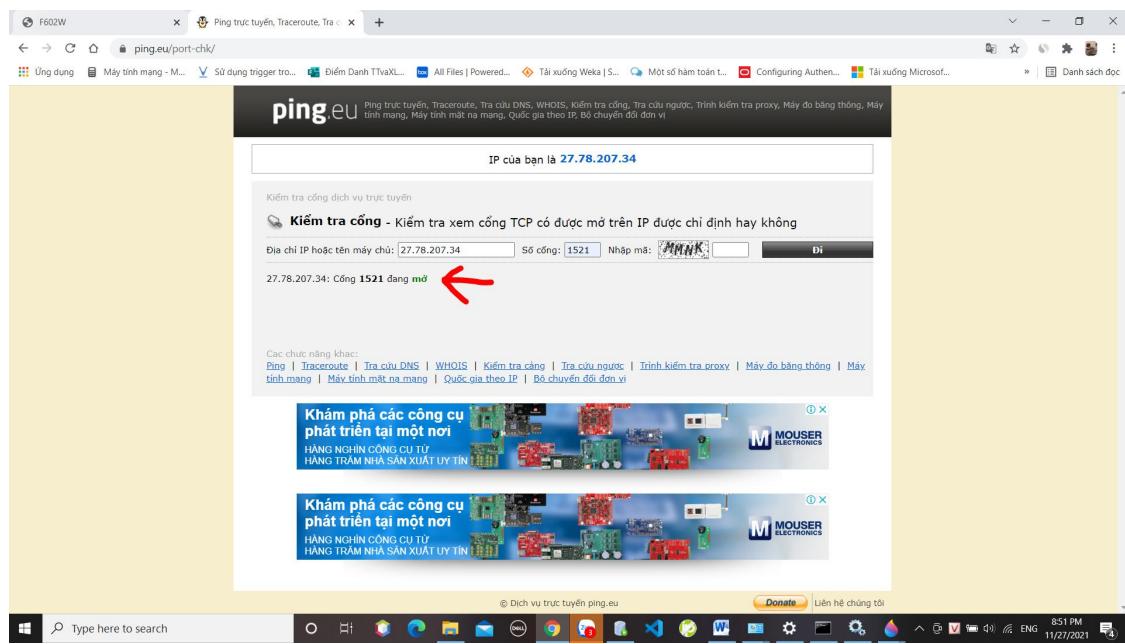
- Và chọn add



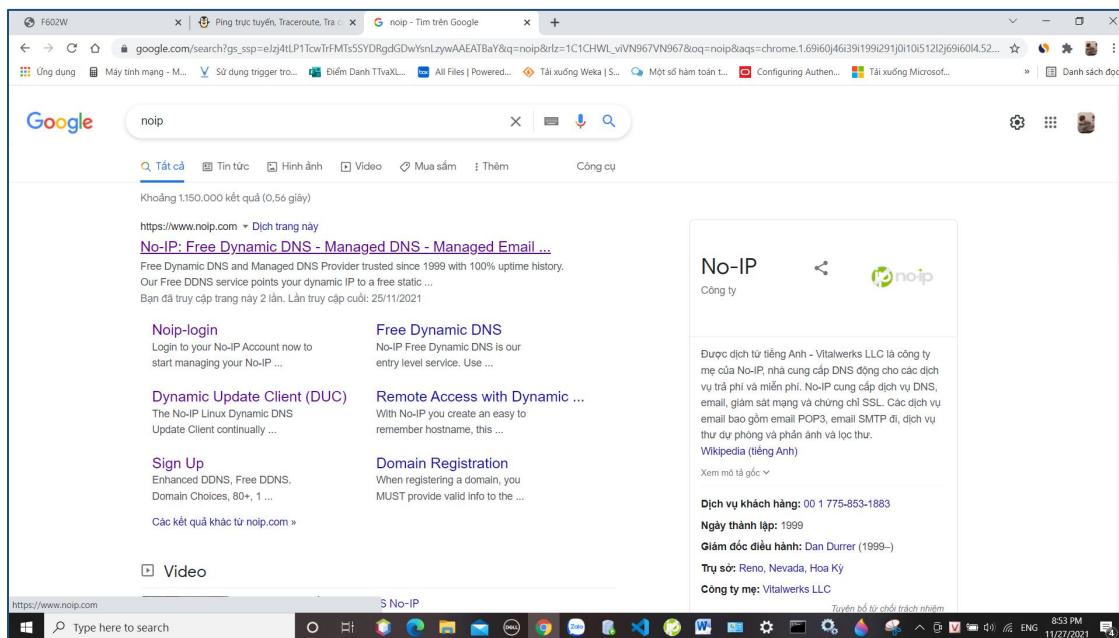
- Kiểm tra port có được mở thành công không vào ping.eu



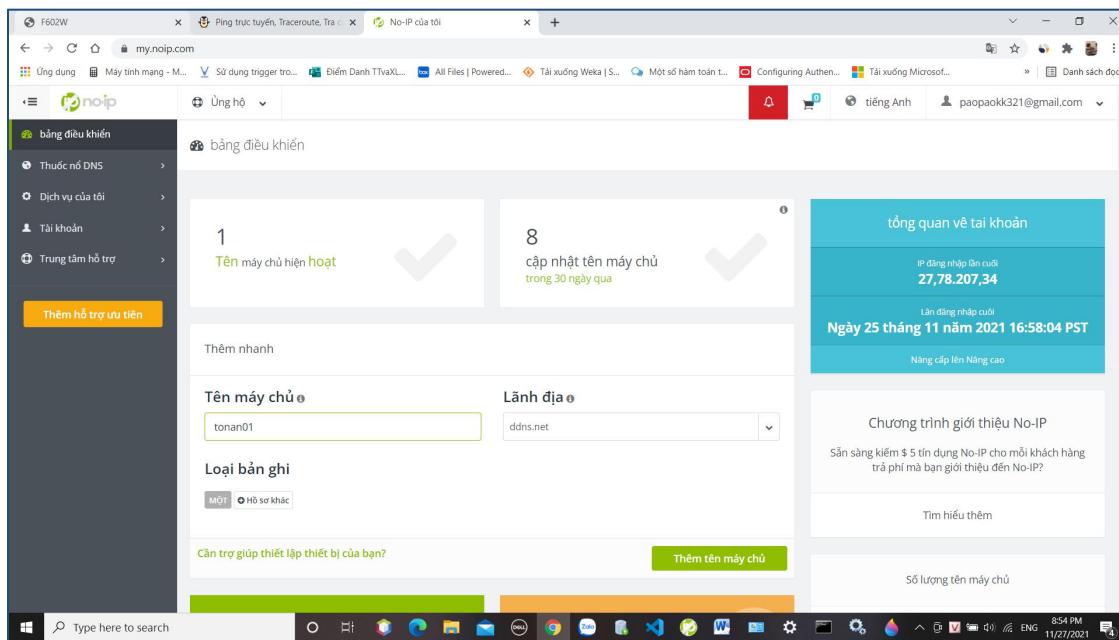
- Mở thành công



- Vào No-IP



- Tạo 1 máy chủ



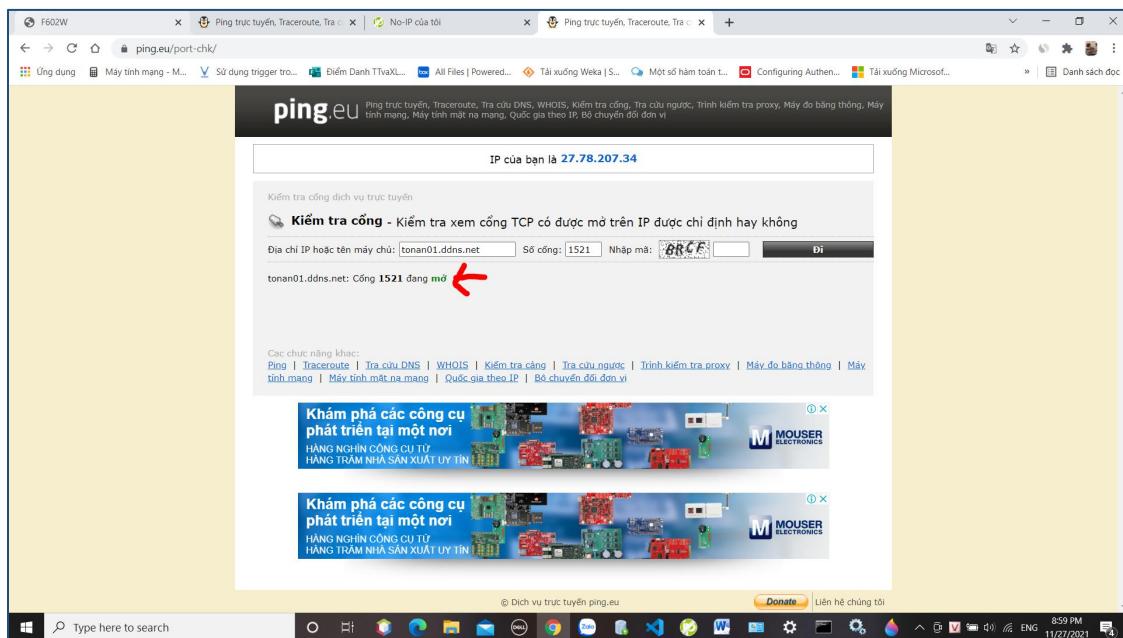
- Tên máy chủ ở đây là “tonan01.ddns.net”

The screenshot shows the No-IP control panel. On the left sidebar, under 'Thuốc nổ DNS', there are several options: 'Tên máy chủ Không có IP', 'Tên máy chủ cá nhân', 'Các nhóm', 'Ứng dụng khách cập nhật động', 'Cập nhật khách hàng', 'Hỗ trợ cấu hình thiết bị', 'Dịch vụ của tôi', 'Tài khoản', and 'Trung tâm hỗ trợ'. A yellow button at the bottom says 'Thêm hỗ trợ ưu tiên'. The main content area is titled 'Tạo tên máy chủ' and contains a table with columns: 'Tên máy chủ', 'Cập nhật cuối cùng', 'IP / Mục tiêu', and 'Kiểu'. The first row shows 'tonan01.ddns.net' with a timestamp of 'Ngày 27 tháng 11 năm 2021 05:53 PST'. A red arrow points to this row. Below the table, there's a section titled 'Trợ giúp về Tên máy chủ' with a gear icon and text about defining host names. To the right, there's a 'Chương trình giới thiệu No-IP' section with a person icon and text about saving \$5 for No-IP users.

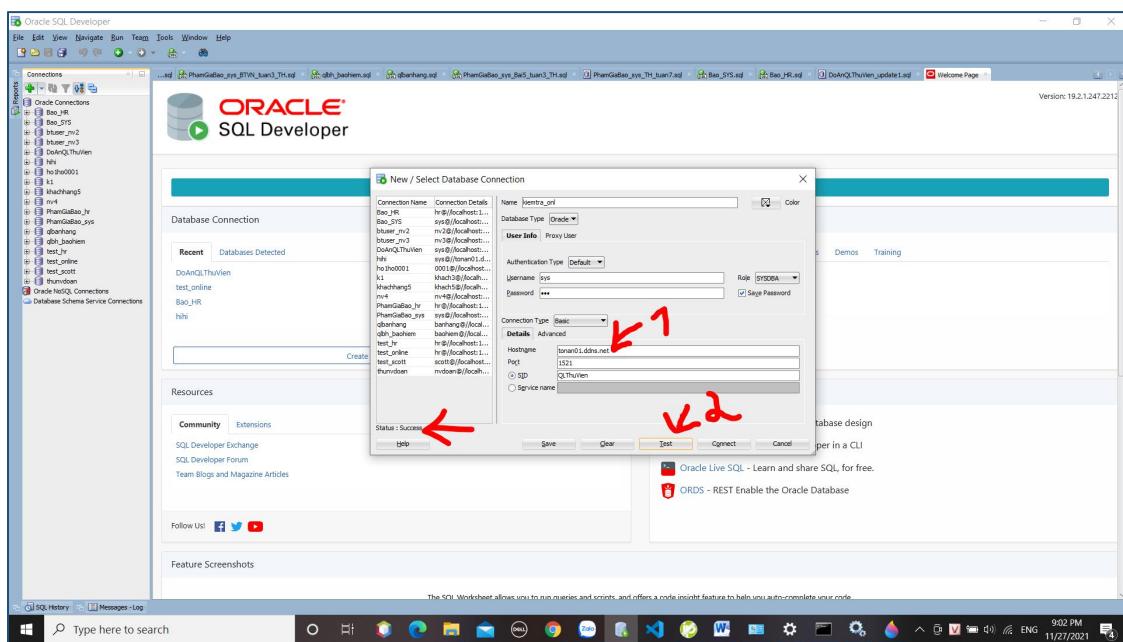
- Thực hiện kiểm tra tại ping.eu

The screenshot shows the ping.eu website. At the top, it says 'ping.eu Ping trực tuyến, Traceroute, Tra cứu DNS, WHOIS, Kiểm tra cổng, Tra cứu ngược, Trình kiểm tra proxy, Máy đo băng thông, Máy tính mạng, Máy tính mật mã mạng, Quốc gia theo IP, Bộ chuyển đổi đơn vị'. The main search bar displays 'IP của bạn là 27.78.207.34'. Below it, there's a section titled 'Kiểm tra cổng - Kiểm tra xem cổng TCP có được mở trên IP được chỉ định hay không' with an input field for 'Địa chỉ IP hoặc tên máy chủ: tonan01.ddns.net'. A red arrow points to this input field. The page also features two banners for 'Khám phá các công cụ phát triển tại một nơi' from Mouser Electronics.

- Kết nối máy chủ thành công



- Tạo connect thành công



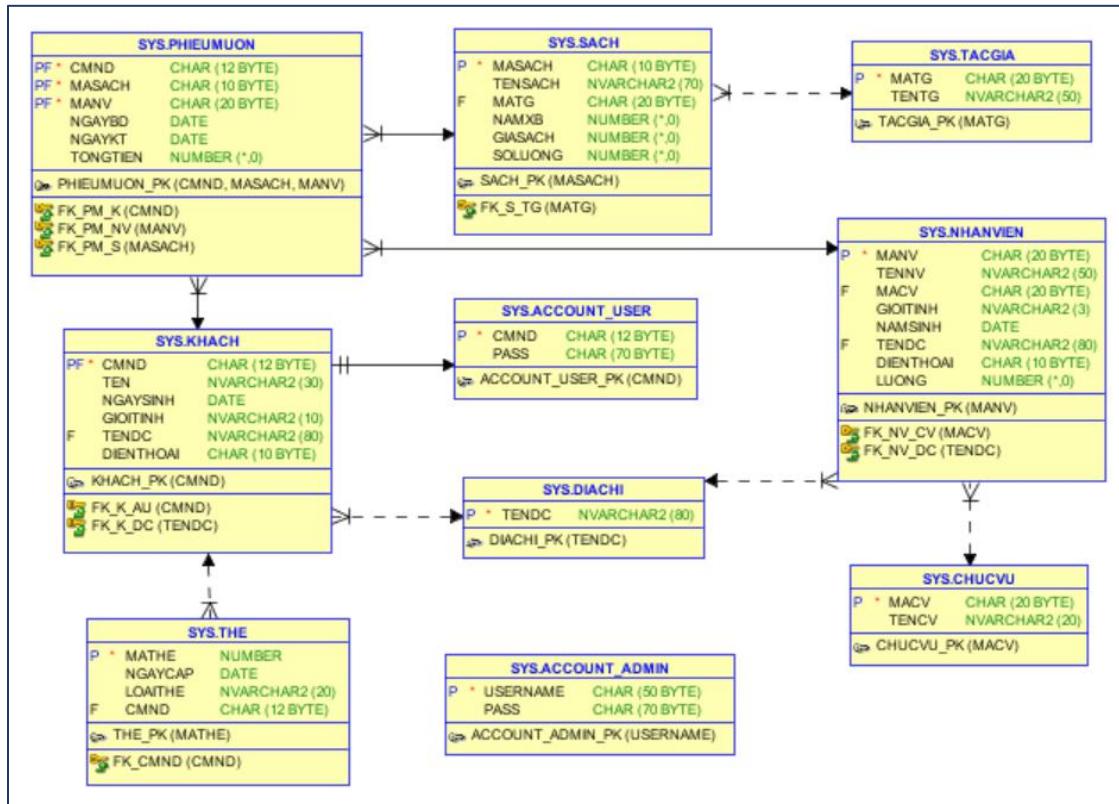
- Đã connect thành công.

```
1 select * from dba_data_files;
```

FILE_NAME	FILE_ID	TABLESPACE_NAME	BYTES	BLOCS	STATUS	RELATIVE_FNO	AUTOEXTENSIBLE	MATRIXES	MAXBLOCKS	INCREMENT_BY
D:\APP\DELL\ORADATA\OLTHUVIEN\SYSTEM01.DBF	1	SYSTEM	849346560	103680	AVAILABLE	1	YES	34359721984	4194302	1280
D:\APP\DELL\ORADATA\OLTHUVIEN\SYSaux01.DBF	3	SYSaux	1237319680	151040	AVAILABLE	3	YES	34359721984	4194302	1280
D:\APP\DELL\ORADATA\OLTHUVIEN\UNDOTBS01.DBF	5	UNDOTBS1	676331520	82560	AVAILABLE	5	YES	34359721984	4194302	640
D:\APP\DELL\ORADATA\OLTHUVIEN\USERS01.DBF	6	USERS	5242880	640	AVAILABLE	6	YES	34359721984	4194302	160
D:\APP\DELL\ORADATA\OLTHUVIEN\TEMP01.DBF	7	TEMP	1320000	20	16128	7	YES	34359721984	4194302	320
D:\APP\DELL\ORADATA\OLTHUVIEN\VIDU01.DBF	8	VIDU	104957600	12800	AVAILABLE	8	NO	0	0	0
D:\APP\DELL\ORADATA\OLTHUVIEN\TABC0L02.DBF	10	TABC0L	209715200	25600	AVAILABLE	10	NO	0	0	0
D:\APP\DELL\ORADATA\OLTHUVIEN\TABC0L01.DBF	9	TABC0L	104957600	12800	AVAILABLE	9	NO	0	0	0
D:\APP\DELL\ORADATA\OLTHUVIEN\VIDU02.DBF	11	VIDU	104857600	12800	AVAILABLE	11	NO	0	0	0
D:\APP\DELL\ORADATA\OLTHUVIEN\VIDU03.DBF	12	VIDU03	104857600	12800	AVAILABLE	12	NO	0	0	0
D:\APP\DELL\ORADATA\OLTHUVIEN\DATA02.DBF	13	DATA02	15728640	320	AVAILABLE	13	NO	0	0	0
D:\THMUCMOI\INDEX01.DBF	15	INDEX01	10485760	1280	AVAILABLE	15	YES	524288000	6400	63
D:\APP\DELL\ORADATA\OLTHUVIEN\VIDU01.DBF	13	BATTAP	104957600	12800	AVAILABLE	13	NO	0	0	0
D:\APP\DELL\ORADATA\OLTHUVIEN\INDEX01.DBF	16	HOOH	10485760	1280	AVAILABLE	16	YES	524288000	6400	63
D:\APP\DELL\ORADATA\OLTHUVIEN\INDEX045.DBF	17	HIEHE	10485760	1280	AVAILABLE	17	YES	15728640	1920	63
D:\APP\DELL\ORADATA\OLTHUVIEN\ROEO.DBF	18	TEO	10485760	1280	AVAILABLE	18	YES	15728640	1920	63

2. Thiết kế:

2.1. Cơ sở dữ liệu:



- Ràng buộc:

```

alter table KHACH
add constraint ck_SDT check (dienThoai > 0);

alter table SACH
add constraint ck_NXB check (namXB > 0);
alter table SACH
add constraint ck_Gia check (giaSach > 0);
alter table SACH
add constraint ck_SL check (soLuong > 0);

alter table NHANVIEN
add constraint uni_SDT unique (dienThoai);
alter table NHANVIEN
add constraint ck_luong check (luong > 0);

alter table PHIEUMUON
add constraint ck_TongTien check (tongTien > 0);
alter table PHIEUMUON
add constraint ck_NBD_NKT check (ngayBD < ngayKT);

```

- Nhóm quyền, gồm có 3 nhóm quyền chính:

- Nhóm quyền KHÁCH HÀNG:

```
CREATE ROLE KHACHHANG;
grant create session, connect to KHACHHANG;
grant select on KHACH to KHACHHANG;
grant select on THE to KHACHHANG;
grant select on SACH to KHACHHANG;
grant select on TACGIA to KHACHHANG;
grant select on ACCOUNT_USER to KHACHHANG;
grant select on DIACHI to KHACHHANG;
```

- Nhóm quyền NHÂN VIÊN:

```
create role NHANVIEN;
grant create session, connect to NHANVIEN;
grant select on NHANVIEN to NHANVIEN;
grant select on THE to NHANVIEN;
grant insert on THE to NHANVIEN;
grant update on THE to NHANVIEN;
grant select on SACH to NHANVIEN;
grant insert on SACH to NHANVIEN;
grant update on SACH to NHANVIEN;
grant select on TACGIA to NHANVIEN;
grant insert on TACGIA to NHANVIEN;
grant update on TACGIA to NHANVIEN;
grant select on PHIEUMUON to NHANVIEN;
grant update on PHIEUMUON to NHANVIEN;
grant insert on PHIEUMUON to NHANVIEN;
grant select on ACCOUNT_USER to NHANVIEN;
grant select on DIACHI to NHANVIEN;
```

- Nhóm quyền QUẢN TRỊ:

```
create role QUANTRI;
grant create session, connect to QUANTRI;
grant select any table to QUANTRI;
grant update any table to QUANTRI;
grant delete any table to QUANTRI;
grant insert any table to QUANTRI;
```

- Thông tin các profile:

```
❑ create profile QUANLY limit
    password_life_time 365    -- Thoi gian su dung 365 ngay
    failed_login_attempts 3   -- Nhap sai mat khau toi da 3 lan
    PASSWORD_LOCK_TIME 5     -- Khoa tai khoan trong 5 ngay
    PASSWORD_REUSE_MAX 5;    -- Doi mat khau toi da 5 ngay

❑ create profile NHANVIEN limit
    password_life_time 365    -- Thoi gian su dung 365 ngay
    failed_login_attempts 3   -- Nhap sai mat khau toi da 3 lan
    PASSWORD_LOCK_TIME 5     -- Khoa tai khoan trong 5 ngay
    PASSWORD_REUSE_MAX 5;    -- Doi mat khau toi da 5 ngay

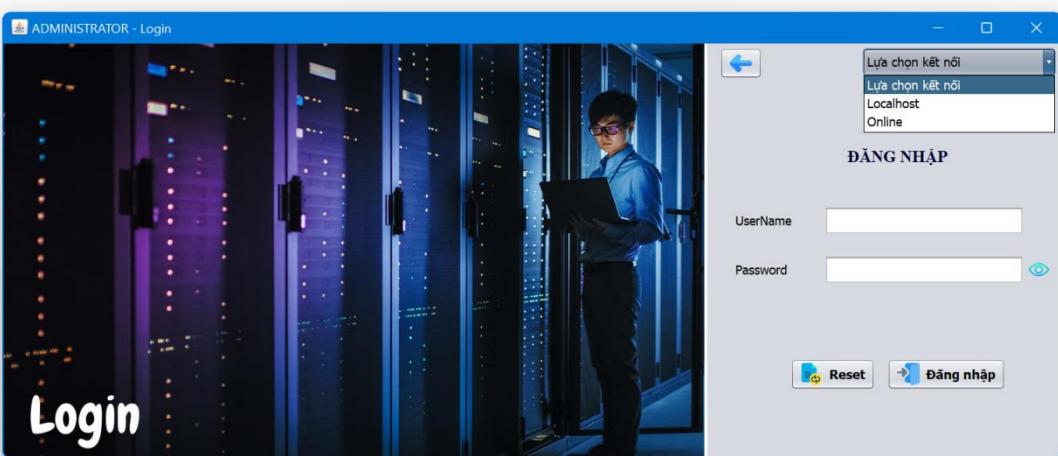
❑ create profile KHACHHANG limit
    password_life_time 90     -- Thoi gian su dung 90 ngay
    failed_login_attempts 3   -- Nhap sai mat khau toi da 3 lan
    PASSWORD_LOCK_TIME 5     -- Khoa tai khoan trong 5 ngay
    PASSWORD_REUSE_MAX 5;    -- Doi mat khau toi da 5 ngay
```

2.2. Giao diện:

- Giao diện trang chủ của chương trình, với 2 chức năng chọn quyền đăng nhập gồm quyền Administrator và quyền User.



- Giao diện đăng nhập với quyền Administrator. Có 2 lựa chọn để kết nối (localhost và online):



- Code combobox “Lựa chọn kết nối”:

```
private void cbbKetNoiActionPerformed(java.awt.event.ActionEvent evt) {
    if(cbbKetNoi.getSelectedItem() == "localhost")
    {
        BienCucBo.ketnoi = "jdbc:oracle:thin:@localhost:1521:ThuVien";
    }
    if(cbbKetNoi.getSelectedItem() == "Online")
    {
        BienCucBo.ketnoi = "jdbc:oracle:thin:@tonan01.ddns.net:1521:QLThuVien";
    }
}
```

- Code button “Đăng nhập”:

```
// Login button -
private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {
    Connection con = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

    BienCucBo.tenDN = "";
    BienCucBo.userN = "";
    BienCucBo.passW = "";

    String msg = "" + txtUserName.getText();
    msg += "\n";

    // Nếu username là 'sys'
    if(txtUserName.getText().equals("sys"))
    {
        BienCucBo.userN =txtUserName.getText() + " as sysdba";
        BienCucBo.passW =txtPassword.getText();
        BienCucBo.tenDN = "Xin chào, " +txtUserName.getText();
        try {

            Class.forName("oracle.jdbc.driver.OracleDriver");
            con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ThuVien", BienCucBo.userN,
                                            BienCucBo.passW);
            String sql = "select * from sys.ACOUNT_USER where CMND='"+txtUserName.getText()+"' "
                        + "and pass='"+txtPassword.getText()+"';

String sql = "select * from sys.ACOUNT_USER where CMND='"+txtUserName.getText()+"' "
                + "and pass='"+txtPassword.getText()+"';

pst = con.prepareStatement(sql);

rs = pst.executeQuery();

if(rs.next()){
    JOptionPane.showMessageDialog(null, "Welcome, " +msg+
                                "Bạn đã đăng nhập thành công!");
    Admin_Home ah = new Admin_Home();
    ah.setVisible(true);
    this.dispose();
}
else{
    JOptionPane.showMessageDialog(null, "Invalid UserName or Password!");
    BienCucBo.tenDN = "";
    BienCucBo.userN = "";
    BienCucBo.passW = "";
    txtUserName.requestFocus();
}
} catch (Exception e) {
    BienCucBo.tenDN = "";
    BienCucBo.userN = "";
    BienCucBo.passW = "";
    JOptionPane.showMessageDialog(null, e);
}
```

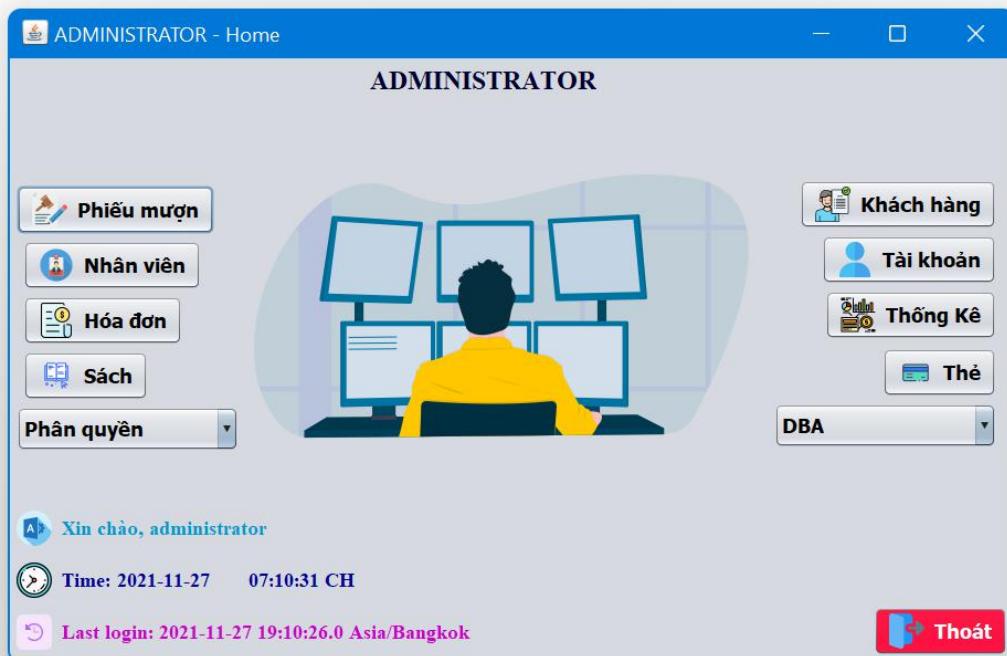
```
        JOptionPane.showMessageDialog(null, e);
    }
}
else // Không là user 'sys'
{
    BienCucBo.userN =txtUserName.getText();
    BienCucBo.passW =txtPassword.getText();
    BienCucBo.tenDN = "Xin chào, "+txtUserName.getText();
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ThuVien",
            BienCucBo.userN, BienCucBo.passW);
        String sql = "select * from sys.ACOUNT_USER where CMND='"+txtUserName.getText()+"' "
            + "and pass='"+txtPassword.getText()+"'";
        pst = con.prepareStatement(sql);

        rs = pst.executeQuery();

        if(rs.next()){
            JOptionPane.showMessageDialog(null, "Welcome, " +msg+
                "Bạn đã đăng nhập thành công!");

            Admin_Home ah = new Admin_Home();
            ah.setVisible(true);
        }
    }
    else{
        JOptionPane.showMessageDialog(null, "Invalid UserName or Password!");
        BienCucBo.tenDN = "";
        BienCucBo.userN ="";
        BienCucBo.passW = "";
        txtUserName.requestFocus();
    }
    } catch (Exception e) {
        BienCucBo.tenDN = "";
        BienCucBo.userN ="";
        BienCucBo.passW = "";
        e.printStackTrace();
    }
}
```

- Giao diện làm việc của Administrator sau khi đăng nhập thành công:



- Menu Administrator với các chức năng như:
 - Quản lý thông tin phiếu mượn.
 - Quản lý thông tin nhân viên.
 - Quản lý hóa đơn.
 - Quản lý sách.
 - Quản lý thông tin khách hàng.
 - Quản lý tài khoản.
 - Xem Thông kê.
 - Quản lý thẻ (thẻ mượn và thẻ đọc).
 - Phân quyền bao gồm: Quyền hệ thống, Quyền đối tượng, Nhóm quyền và User.
 - DBA bao gồm các thông tin: Process, Instance, Database, Datafile, Session, Parameter File, Profile, Tablespace và Audit.

- Giao diện quản lý thông tin của Nhân viên:

Mã nhân viên	Tên nhân viên	Mã chức vụ	Giới tính	Năm sinh	Tên địa chỉ	Điện thoại	Lương
NV001	Phạm Gia Bảo	TT	Nam	2001-04-12 00:00:00	Tây Ninh	0184728783	9000000
NV002	Phạm Gia Đạt	TT	Nam	2001-04-12 00:00:00	TP. Hồ Chí Minh	0184728993	9000000
NV003	Nguyễn Lê Gi...	TN	Nam	2001-05-20 00:00:00	TP. Hồ Chí Minh	0184711783	9500000
NV004	Lê Văn Minh	TN	Nam	2001-05-20 00:00:00	Bến Tre	0987728783	9500000
NV005	Mai Tuấn Thủ...	TN	Nam	2001-04-28 00:00:00	Thanh Hóa	0774728783	9500000
NV006	Lê Ngọc Ngân	TT	Nữ	2001-05-01 00:00:00	TP. Hồ Chí Minh	0180028783	9000000
NV007	Lý Uyên Nữ	TT	Nữ	2001-11-11 00:00:00	TP. Hồ Chí Minh	0184720083	9000000
NV008	Hoàng Văn TT	TN	Nam	2000-01-01 00:00:00	TP. Hồ Chí Minh	0184728783	9000000
NV009	Bùi Tân Trường	FCD	Nam	1999-12-11 00:00:00	TP. Hồ Chí Minh	0188628783	10000000
NV010	Nguyễn Thị Phan...	GĐ	Nam	1993-09-13 00:00:00	Thái Bình	0987728783	12000000
NV011	Phan Nguyễn ...	LC	Nữ	2000-10-12 00:00:00	Bến Tre	0967583859	8000000
NV012	Nguyễn Thị T...	LC	Nam	2001-09-11 00:00:00	TP. Hồ Chí Minh	0958395910	8000000

- Code của form quản lý thông tin của Nhân viên:

```

public Admin_Staff() {
    initComponents();

    this.setLocationRelativeTo(null); // Chỉnh form nằm giữa màn hình

    initComboBox();
    initComboBoxCV();

    // Load dữ liệu lên table -----
    dao=new DAO_Admin_Staff();
    vheader.add("Mã nhân viên");
    vheader.add("Tên nhân viên");
    vheader.add("Mã chức vụ");
    vheader.add("Giới tính");
    vheader.add("Năm sinh");
    vheader.add("Tên địa chỉ");
    vheader.add("Điện thoại");
    vheader.add("Lương");

    dao.LayDSAdmin_Staff_vector(vdata);
    jTablenhavien.setModel(new DefaultTableModel(vdata,vheader));

    //dao.LayDScomboboxAdmin_Staff_Dia_chi_vector(cdata);
    //cbbDiaChi.setModel(new DefaultComboBoxModel<>(cdata));
}
}

```

- Code load dữ liệu từ database lên combobox Địa chỉ:

```
// Khởi tạo phần combobox -----
private void initComboBox() {
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection(BienCucBo.ketnoi,
            BienCucBo.userN, BienCucBo.passW);
        String sql = "select * from sys.DIACHI";
        PreparedStatement pstmt = con.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery();
        cbbDiaChi.removeAllItems();
        while(rs.next()) {
            cbbDiaChi.addItem(rs.getString("tenDC"));
        }
        rs.close();
        pstmt.close();
        con.close();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
        e.printStackTrace();
    }
}
```

- Code load dữ liệu từ database lên combobox Chức vụ:

```
// Khởi tạo phần combobox -----
private void initComboBoxCV() {
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection(BienCucBo.ketnoi,
            BienCucBo.userN, BienCucBo.passW);
        String sql = "select maCV from sys.CHUCVU";
        PreparedStatement pstmt = con.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery();
        cbbChucVu.removeAllItems();
        while(rs.next()) {
            cbbChucVu.addItem(rs.getString("maCV"));
        }
        rs.close();
        pstmt.close();
        con.close();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
        e.printStackTrace();
    }
}
```

- Code button Thêm mới 1 Nhân viên:

```

// Add button -----
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    // Trống -> không thực thi
    if(ktrong()==false)
    {
        return;
    }

    if(BienCucBo.ktthem()) // Đóng ý thực hiện thao tác
    {
        if(kttrung()==true) // trùng mã sách
        {
            JOptionPane.showMessageDialog(null, "Trùng Mã nhân viên!");
            return;
        }
        else // mã sách không trùng
        {
            pojo_Admin_Staff x=new pojo_Admin_Staff(txtMaNV.getText(),
                txtHoTen.getText(),cbbChucVu.getSelectedItem().toString(),
                txtGioiTinh.getText(),txtNamSinh.getText(),
                cbbDiaChi.getSelectedItem().toString(),txtSDT.getText(),txtLuong.getText());
            dao.ThemNhanVien(x);
            dao.LayDSAdmin_Staff_vector(vdata);
            jTableNhanvien.updateUI();
        }
    }
    else // mã sách không trùng
    {
        pojo_Admin_Staff x=new pojo_Admin_Staff(txtMaNV.getText(),
            txtHoTen.getText(),cbbChucVu.getSelectedItem().toString(),
            txtGioiTinh.getText(),txtNamSinh.getText(),
            cbbDiaChi.getSelectedItem().toString(),txtSDT.getText(),txtLuong.getText());
        dao.ThemNhanVien(x);
        dao.LayDSAdmin_Staff_vector(vdata);
        jTableNhanvien.updateUI();

        if(kttrung()==true)//trùng là đã thêm vào thành công
        {
            JOptionPane.showMessageDialog(null, "Thành công");
            return;
        }
        else//không trùng không có trống table
        {
            JOptionPane.showMessageDialog(null, "Thất bại");
            return;
        }
    }
}

```

- Code button Sửa thông tin của 1 Nhân viên:

```
// Edit button -----
private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {
    // Trống -> không thực thi câu lệnh
    if(ktrong()==false)//rỗng
    {
        return;
    }

    if (BienCucBo.ktsua()) // đồng ý thực thi câu lệnh
    {
        if(kttrung()==false) // không tồn tại khóa chính
        {
            JOptionPane.showMessageDialog(null, "Mã nhân viên không tồn tại ");
            return;
        }
        else//không trùng
        {
            pojo_Admin_Staff x=new pojo_Admin_Staff(txtMaNV.getText(),
            txtHoTen.getText(),cbbChucVu.getSelectedItem().toString(),
            txtGioiTinh.getText(),txtNamSinh.getText(),
            cbbDiaChi.getSelectedItem().toString(), txtSDT.getText(),
            txtLuong.getText());
            dao.SuaNhanVien(x);
            dao.LayDSAdmin_Staff_vector(vdata);
            jTablenhanvien.updateUI();
        }
    }
    else//không trùng
    {
        pojo_Admin_Staff x=new pojo_Admin_Staff(txtMaNV.getText(),
        txtHoTen.getText(),cbbChucVu.getSelectedItem().toString(),
        txtGioiTinh.getText(),txtNamSinh.getText(),
        cbbDiaChi.getSelectedItem().toString(), txtSDT.getText(),
        txtLuong.getText());
        dao.SuaNhanVien(x);
        dao.LayDSAdmin_Staff_vector(vdata);
        jTablenhanvien.updateUI();

        if(kttrung()==true)//trùng là sửa thành công
        {
            JOptionPane.showMessageDialog(null, "Thành công");
            return;
        }
        else
        {
            JOptionPane.showMessageDialog(null, "Thất bại");
            return;
        }
    }
}
```

- Code button Xóa 1 Nhân viên:

```
// Delete button -----
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // Trống -> không thực thi
    if(txtMaNV.getText().length()==0)//rỗng
    {
        JOptionPane.showMessageDialog(null, "Mã nhân viên không được để trống");
        return;
    }

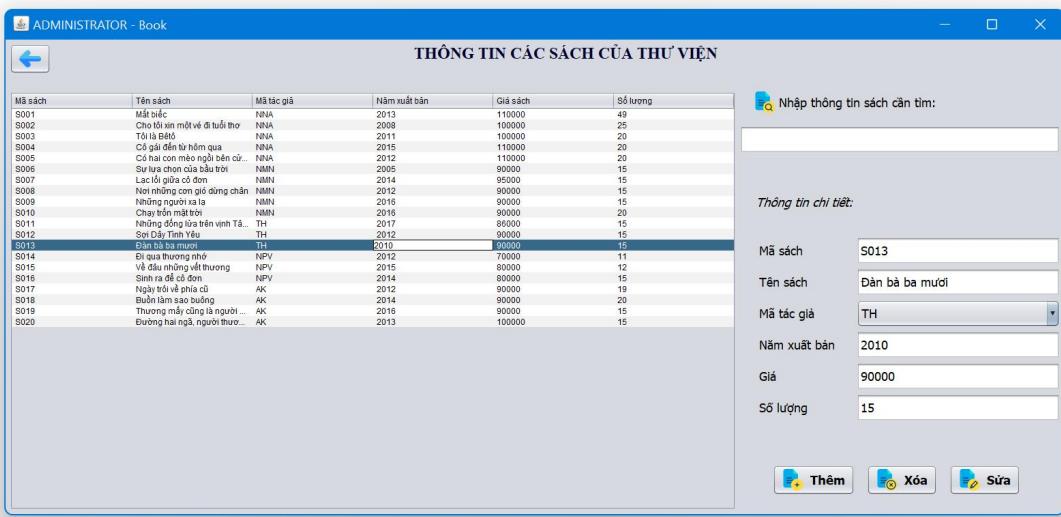
    if (BienCucBo.ktxoa())
    {
        if(kttrung()==false)//không tồn tại khóa chính
        {
            JOptionPane.showMessageDialog(null, "Mã nhân viên không tồn tại");
            return;
        }
        else // ton tai khoa chinh
        {
            pojo_Admin_Staff x=new pojo_Admin_Staff();
            x.setmNV(txtMaNV.getText());
            dao.XoaNhanVien(x);
            dao.LayDSAdmin_Staff_vector(vdata);
            jTablenhanvien.updateUI();

            if(kttrung()==false)//Không tồn tại là xóa thành công
            {
                JOptionPane.showMessageDialog(null, "Thành công");
                return;
            }
            else
            {
                JOptionPane.showMessageDialog(null, "Thất bại");
                return;
            }
        }
    }
}
```

- Code button Tìm kiếm thông tin 1 Nhân viên:

```
private void txtFindKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
        pojo_Admin_Staff x = new pojo_Admin_Staff(txtFind.getText().toUpperCase(),
            txtHoTen.getText(), cbbChucVu.getSelectedItem().toString().toUpperCase(),
            txtGioiTinh.getText(), txtNamSinh.getText(),
            cbbDiaChi.getSelectedItem().toString(), txtSDT.getText(), "");
        dao.timkiemhanhien(vdata, x);
        jTable1.updateUI();
    }
}
```

- Giao diện quản lý thông tin Sách của thư viện:



- Code form quản lý thông tin Sách của thư viện:

```
public Admin_Book() {
    initComponents();

    this.setLocationRelativeTo(null); // Chính form nằm giữa màn hình

    initComboBoxMaTG();

    // Load dữ liệu lên table -----
    dao=new DAO_Admin_Book();
    vheader.add("Mã sách");
    vheader.add("Tên sách");
    vheader.add("Mã tác giả");
    vheader.add("Năm xuất bản");
    vheader.add("Giá sách");
    vheader.add("Số lượng");

    dao.LayDSAdmin_Book_vector(vdata);
    jTable1.setModel(new DefaultTableModel(vdata,vheader));
}
```

- Code load dữ liệu từ database lên combobox Mã tác giả:

```
// Khởi tạo phần combobox mã tác giả -----
private void initComboBoxMaTG() {
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection(BienCucBo.ketnoi,
            BienCucBo.userN, BienCucBo.passW);
        String sql = "select * from sys.view_inmaTACGIA";
        PreparedStatement pstmt = con.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery();
        cbbmatagia.removeAllItems();
        while(rs.next()){
            cbbmatagia.addItem(rs.getString("maTG"));
        }
        rs.close();
        pstmt.close();
        con.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

- Code button Thêm mới 1 quyển sách:

```
private void btnThemActionPerformed(java.awt.event.ActionEvent evt) {
    // Nếu để rỗng -> không thực thi
    if(ktrong()==false)
    {
        return;
    }

    if(BienCucBo.ktthem()) // Đồng ý thực hiện thao tác
    {
        if(kttrung()==true) // trùng mã sách
        {
            JOptionPane.showMessageDialog(null, "Trùng Mã sách!");
            return;
        }
        else //mã sách không trùng
        {
            pojo_Admin_Book x = new pojo_Admin_Book(txtMaSach.getText(),
                txtTenSach.getText(),cbbmatagia.getSelectedItem().toString(),
                txtNXB.getText(),txtGia.getText(),txtSL.getText());
            dao.ThemSach(x);
            dao.LayDSAdmin_Book_vector(vdata);
            jTableSach.updateUI();

            if(kttrung()==true) // trùng là đã thêm vào thành công
            {
```

```

        else //mã sách không trùng
    {
        pojo_Admin_Book x = new pojo_Admin_Book(txtMaSach.getText(),
        txtTenSach.getText(),cbbmatagia.getSelectedItem().toString(),
        txtNXB.getText(),txtGia.getText(),txtSL.getText());
        dao.ThemSach(x);
        dao.LayDSAdmin_Book_vecter(vdata);
        jTableSach.updateUI();
    }

    if(kttrung() == true) // trùng là đã thêm vào thành công
    {
        JOptionPane.showMessageDialog(null, "Thành công");
        return;
    }
    else // không trùng không có trống table
    {
        JOptionPane.showMessageDialog(null, "Thất bại");
        return;
    }
}
}
}

```

- Code button Sửa thông tin của 1 quyển sách:

```

private void btnSuaActionPerformed(java.awt.event.ActionEvent evt) {
    // Nếu để rỗng -> không thực thi
    if(ktrong() == false)//rỗng
    {
        return;
    }

    if (BienCucBo.ktsua()) // đồng ý thực thi thao tác
    {
        if(kttrung() == false) // không tồn tại khóa chính
        {
            JOptionPane.showMessageDialog(null, "Mã Sách Không tồn tại ");
            return;
        }
        else//không trùng
        {
            pojo_Admin_Book x = new pojo_Admin_Book(txtMaSach.getText(),
            txtTenSach.getText(),cbbmatagia.getSelectedItem().toString(),
            txtNXB.getText(),txtGia.getText(),txtSL.getText());
            dao.SuaSach(x);
            dao.LayDSAdmin_Book_vecter(vdata);
            jTableSach.updateUI();
            if(kttrung() == true) // trùng là đã thêm vào thành công
            {
                JOptionPane.showMessageDialog(null, "Thành công");
                return;
            }
        }
    }
}

```

```

    if(kttrung() == true) // trùng là đã thêm vào thành công
    {
        JOptionPane.showMessageDialog(null, "Thành công");
        return;
    }
    else
    {
        JOptionPane.showMessageDialog(null, "Thất bại");
        return;
    }
}
}
}

```

- Code button Xóa 1 quyển sách:

```

private void btnXoaActionPerformed(java.awt.event.ActionEvent evt) {
    // Nếu đẻ rỗng -> không thực thi
    if(txtMaSach.getText().length()==0)//rỗng
    {
        JOptionPane.showMessageDialog(null, "CMND không được để trống");
        return;
    }

    if(BienCucBo.ktxoa()) // đồng ý thực hiện thao tác
    {
        if(kttrung()==false) // không tồn tại khóa chính
        {
            JOptionPane.showMessageDialog(null, "Mã Sách Không tồn tại");
            return;
        }
        else // khong trùng
        {
            pojo_Admin_Book x = new pojo_Admin_Book();
            x.setmS(txtMaSach.getText());
            dao.XoaSach(x);
            dao.LayDSAdmin_Book_vector(vdata);
            jTableSach.updateUI();

            if(kttrung()==false)// Không tồn tại tức là đã xóa thành công
            {
                JOptionPane.showMessageDialog(null, "Thành công");
            }
        }
    }
}

```

```

if(kttrung()==false)// Không tồn tại tức là đã xóa thành công
{
    JOptionPane.showMessageDialog(null, "Thành công");
    return;
}
else
{
    JOptionPane.showMessageDialog(null, "Thất bại");
    return;
}
}

```

- Giao diện quản lý thông tin cá nhân của khách hàng:

CMND	Tên	Tuổi	Giới tính	Địa chỉ	Điện thoại
0001	Nguyễn Công P..	1995-11-20 00:0..	Nam	TP. Hồ Chí Minh	0123456789
0002	Nguyễn Văn Toán	1995-10-20 00:0..	Nam	Thủ Thiênn Hué	0122256789
0003	Nguyễn Quang ..	1993-12-20 00:0..	Nam	TP. Hồ Chí Minh	0993456789
0004	Nguyễn Tiến Linh	1995-11-11 00:0..	Nam	TP. Hồ Chí Minh	0883456789
0005	Triệu Văn Phong	1995-09-20 00:0..	Nam	TP. Hồ Chí Minh	0123456789
0006	Nguyễn Trọng H..	1995-11-13 00:0..	Nam	TP. Hồ Chí Minh	0123096789
0007	Đặng Văn Lâm	1997-12-11 00:0..	Nam	Thủ Thiênn Hué	0123456700
0008	Võ Văn Thành	1995-11-25 00:0..	Nam	TP. Hồ Chí Minh	0123456799
0009	Trần Xuân Trường	1995-09-20 00:0..	Nam	TP. Hồ Chí Minh	0123451189
0010	Nguyễn Văn Công ..	1995-05-15 00:0..	Nam	TP. Hồ Chí Minh	0123456789
0011	Hà Đức Chính	1995-05-20 00:0..	Nam	TP. Hồ Chí Minh	0156456789
0012	Quốc Ngọc Hải	1995-02-20 00:0..	Nam	TP. Hồ Chí Minh	0123226789
0013	Bùi Tiến Dũng	1995-11-22 00:0..	Nam	TP. Hồ Chí Minh	0123456339
0014	Đỗ Duy Mạnh	1995-09-25 00:0..	Nam	TP. Hồ Chí Minh	0123456449
0015	Nguyễn Văn Anh	1995-11-25 00:0..	Nam	TP. Hồ Chí Minh	0123456789
0016	Lê Công Vinh	1995-11-21 00:0..	Nam	TP. Hồ Chí Minh	0689369172
0017	Nguyễn Hoàng ..	1995-05-21 00:0..	Nam	TP. Hồ Chí Minh	098932051
0018	Bùi Hoàng Việt A..	1995-07-20 00:0..	Nam	TP. Hồ Chí Minh	0593950192
0019	Nguyễn Thành C..	1995-03-20 00:0..	Nam	TP. Hồ Chí Minh	0192850395
0020	Đỗ Hùng Dũng	1998-08-20 00:0..	Nam	Hà Nội	0593850182

- Code form quản lý thông tin khách hàng:

```
public Admin_Customer() {
    initComponents();

    initComboBox();

    this.setLocationRelativeTo(null); // Chỉnh form nằm giữa màn hình

    // Load dữ liệu lên table -----
    dao=new DAO_Admin_Customer();
    vheader.add("CMND");
    vheader.add("Tên");
    vheader.add("Tuổi");
    vheader.add("Giới tính");
    vheader.add("Địa chỉ");
    vheader.add("Điện thoại");
    dao.LayDSAdmin_Customer_vector(vdata);
    jTableKhach.setModel(new DefaultTableModel(vdata,vheader));
}
```

- Code load dữ liệu từ database lên combobox địa chỉ:

```
// Khởi tạo phần combobox -----
private void initComboBox() {
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection(BienCucBo.ketnoi,
            BienCucBo.userN, BienCucBo.passW);
        String sql = "select * from DIACHI";
        PreparedStatement pstmt = con.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery();
        cbbDiaChi.removeAllItems();
        while(rs.next()){
            cbbDiaChi.addItem(rs.getString("tenDC"));
        }
        rs.close();
        pstmt.close();
        con.close();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
        e.printStackTrace();
    }
}
```

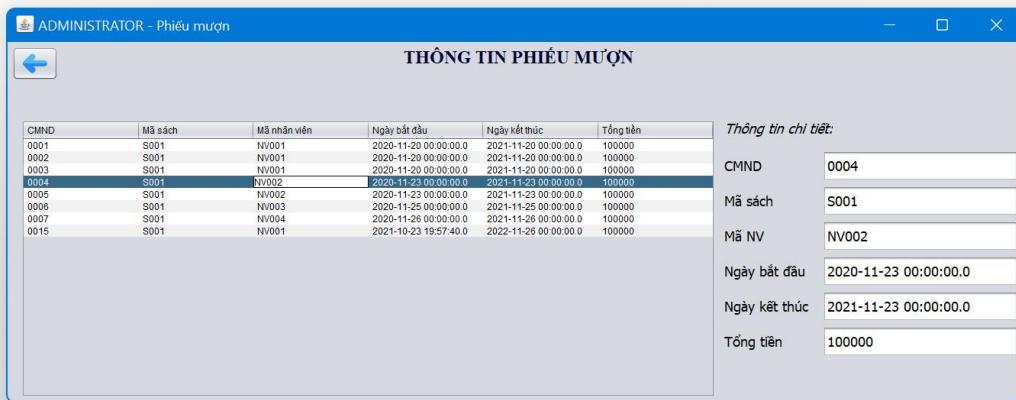
- Code Sửa thông tin của 1 khách hàng:

```
private void btnSuaActionPerformed(java.awt.event.ActionEvent evt) {
    if (JOptionPane.showConfirmDialog(this, "Bạn có muốn sửa không?")
        == JOptionPane.OK_OPTION) {
        pojo_Admin_Customer x=new pojo_Admin_Customer(txtCMND.getText(),
            txtHoTen.getText(),txtTuoi.getText(),txtGioiTinh.getText(),
            cbbDiaChi.getSelectedItem().toString(),txtSDT.getText());
        dao.SuaKhach(x);
        dao.LayDSAdmin_Customer_vector(vdata);
        jTableKhach.updateUI();
    }
}
```

- Code Xóa 1 khách hàng:

```
private void btnXoaActionPerformed(java.awt.event.ActionEvent evt) {
    if (JOptionPane.showConfirmDialog(this, "Bạn có muốn xóa không?" == JOptionPane.OK_OPTION) {
        pojo_Admin_Customer x=new pojo_Admin_Customer();
        x.setCMND(txtCMND.getText());
        dao.XoaKhach(x);
        dao.LayDSAdmin_Customer_vector(vdata);
        jTableKhach.updateUI();
    }
}
```

- Giao diện quản lý thông tin phiếu mượn:



- Code form quản lý thông tin phiếu mượn:

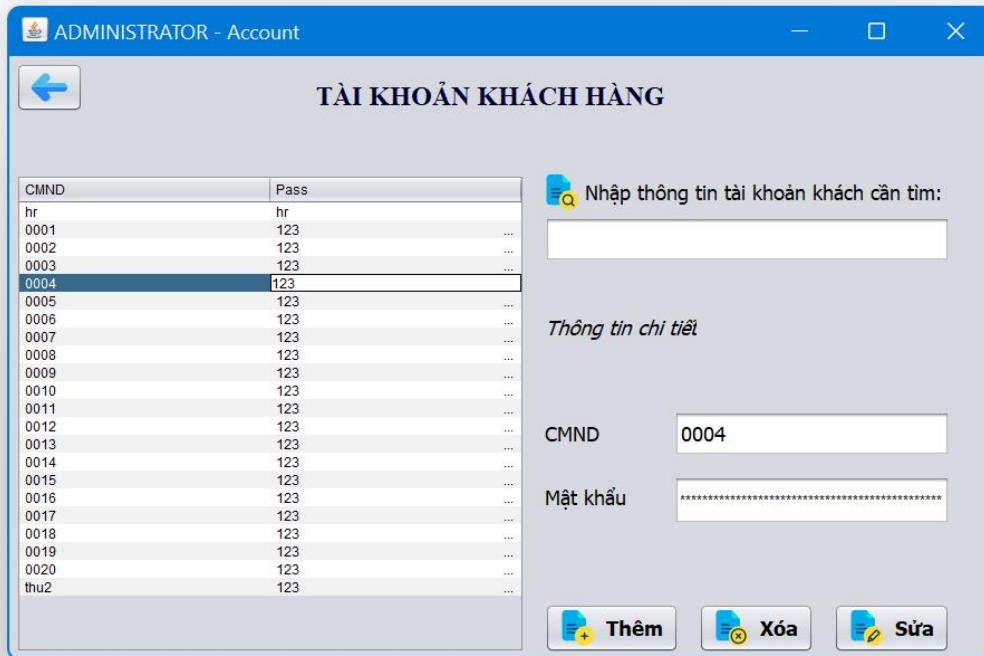
```
public Admin_PhieuMuon() {
    initComponents();

    this.setLocationRelativeTo(null); // Chính form nằm giữa màn hình

    // Load dữ liệu lên table -----
    dao=new DAO_Admin_PhieuMuon();
    vheader.add("CMND");
    vheader.add("Mã sách");
    vheader.add("Mã nhân viên");
    vheader.add("Ngày bắt đầu");
    vheader.add("Ngày kết thúc");
    vheader.add("Tổng tiền");

    dao.load(vdata);
    jTable1.setModel(new DefaultTableModel(vdata,vheader));
}
```

- Giao diện quản lý tài khoản của khách hàng:



- Code form quản lý tài khoản của khách hàng:

```
public Admin_Account() {
    initComponents();

    this.setLocationRelativeTo(null); // Chính form nằm giữa màn hình

    dao = new DAO_Admin_Account();

    // Load dữ liệu lên table -----
    vheader.add("CMND");
    vheader.add("Pass");
    dao.LayDSAdmin_Account_vector(vdata);
    jTableAccount.setModel(new DefaultTableModel(vdata,vheader));
}
```

- Code button Thêm mới 1 tài khoản của khách hàng:

```
// Add button -----
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    // Nếu đ为空 -> không thực thi
    if(ktrong()==false)
    {
        return;
    }

    if (BienCucBo.ktthem())
    {
        if(kttrung() == true) // Trùng CMND -> không thêm
        {
            JOptionPane.showMessageDialog(null, "CMND bạn nhập đã tồn tại!");
            return;
        }
        else // không trùng -> Tiến hành thêm mới 1 tài khoản
        {
            pojo_Admin_Account x=new pojo_Admin_Account(txtCMND.getText(), txtMatKhau.getText());
            dao.ThemAdmin_Account(x);
            dao.LayDSAdmin_Account_vector(vdata);
            jTableAccount.updateUI();

            if(kttrung()==true) // Kiểm tra lại nếu CMND đã được thêm
            {

```

```
                if(kttrung()==true) // Kiểm tra lại nếu CMND đã được thêm
                {
                    JOptionPane.showMessageDialog(null, "Thành công");
                    return;
                }
                else
                {
                    JOptionPane.showMessageDialog(null, "Thất bại");
                    return;
                }
            }
        }
    }
}
```

- Code button Sửa thông tin 1 tài khoản của khách hàng:

```
// Edit button -----
private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {
    // Nếu đ为空 -> không thực thi
    if(ktrong()==false) // rỗng
    {
        return;
    }

    if (BienCucBo.ktsua() == true) // đồng ý thực thi câu lệnh
    {
        if(kttrung() == false) // không tồn tại khóa chính (không trùng)
        {
            JOptionPane.showMessageDialog(null, "CMND không tồn tại");
            return;
        }
        else // trùng
        {
            pojo_Admin_Account x=new pojo_Admin_Account(txtCMND.getText(), txtMatKhau.getText());
            dao.SuaAdmin_Account(x);
            dao.LayDSAdmin_Account_vector(vdata);
            jTableAccount.updateUI();

            if(kttrung()==true)// KTra lại, nếu trùng
            {

```

```

        if(kttrung() == true) // KTra lại, nếu trùng
        {
            JOptionPane.showMessageDialog(null, "Thành công");
            return;
        }
        else
        {
            JOptionPane.showMessageDialog(null, "Thất bại");
            return;
        }
    }
}

```

- Code button Xóa 1 tài khoản của khách hàng:

```

// Delete button
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // CMND không được để trống
    if(txtCMND.getText().length() == 0)
    {
        JOptionPane.showMessageDialog(null, "CMND không được để trống");
        return;
    }

    if(BienCucBo.ktxoa()) // Đóng ý thực thi câu lệnh
    {
        if(ktkhoaNgoai() == true) // có khóa ngoại thì không xóa được
        {
            JOptionPane.showMessageDialog(null, "CMND này đang tồn tại ở Khách Hàng"
                + "\n Không thể xóa!");
            return;
        }
        if(kttrung() == false) // không tồn tại khóa chính
        {
            JOptionPane.showMessageDialog(null, "CMND không tồn tại");
            return;
        }
        else // có khóa chính
        {

```

```

            else // có khóa chính
            {
                pojo_Admin_Account x = new pojo_Admin_Account();
                x.setcM(txtCMND.getText());
                dao.XoaKAdmin_Account(x);
                dao.LayDSAdmin_Account_vector(vdata);
                jTableAccount.updateUI(); // thực thi update lên table

                if(kttrung() == false) // Không tồn tại (Đã được xóa)
                {
                    JOptionPane.showMessageDialog(null, "Thành công");
                    return;
                }
                else
                {
                    JOptionPane.showMessageDialog(null, "Thất bại");
                    return;
                }
            }
        }
    }
}

```

- Giao diện Thống kê thông tin, hoạt động của các user:

session_id	db_user	object_schema	object_name	policy_name	sql_text	statement_type
170309	HR	HR	COUNTRIES	FGA_COUNTRIES	select * from COUNTRIES	SELECT
240431	HR	HR	COUNTRIES	FGA_COUNTRIES	select * from COUNTRIES	SELECT
240431	HR	HR	JOBs	FGA_JOBs	select * from jobs	SELECT
240431	HR	HR	COUNTRIES	FGA_COUNTRIES	select * from COUNTRIES	SELECT
240438	USER1	HR	COUNTRIES	FGA_COUNTRIES	select * from hr.COUNTRIES	SELECT
240441	HR	HR	COUNTRIES	FGA_COUNTRIES	update hr.COUNTRIESset...	UPDATE
240441	HR	HR	COUNTRIES	FGA_COUNTRIES	select * from countries	SELECT
240443	HR	HR	COUNTRIES	FGA_COUNTRIES	update countniesset cou...	UPDATE
240447	HR	HR	COUNTRIES	FGA_COUNTRIES	update countniesset cou...	UPDATE
240447	HR	HR	COUNTRIES	FGA_COUNTRIES	update countniesset cou...	UPDATE
240454	HR	HR	COUNTRIES	FGA_COUNTRIES	update countniesset cou...	UPDATE
240455	HR	HR	COUNTRIES	FGA_COUNTRIES	update hr.countniesset cou...	UPDATE
240455	HR	HR	COUNTRIES	FGA_COUNTRIES	update hr.countniesset cou...	UPDATE
240458	USER1	HR	COUNTRIES	FGA_COUNTRIES	update hr.countniesset cou...	UPDATE
240458	USER1	HR	COUNTRIES	FGA_COUNTRIES	select * from hr.COUNTRIES	SELECT
240459	HR	HR	COUNTRIES	FGA_COUNTRIES	update hr.countniesset cou...	UPDATE
240466	HR	HR	COUNTRIES	FGA_COUNTRIES	select * from countries	SELECT
240466	HR	HR	COUNTRIES	FGA_COUNTRIES	update countniesset cou...	UPDATE
240475	USER1	HR	COUNTRIES	FGA_COUNTRIES	select * from hr.countries	SELECT
240475	USER1	HR	COUNTRIES	FGA_COUNTRIES	update hr.countniesset cou...	UPDATE
240475	USER1	HR	COUNTRIES	FGA_COUNTRIES	update hr.countries set Co...	UPDATE
240476	HR	HR	COUNTRIES	FGA_COUNTRIES	UPDATE COUNTRIES SET ...	UPDATE
240483	USER1	HR	COUNTRIES	FGA_COUNTRIES	Update hr.COUNTRIESSET ...	UPDATE
240483	USER1	HR	COUNTRIES	FGA_COUNTRIES	select * from hr.COUNTRIES	SELECT
240487	HR	HR	COUNTRIES	FGA_COUNTRIES	Update hr.COUNTRIESSET ...	UPDATE
240487	HR	HR	COUNTRIES	FGA_COUNTRIES	Update hr.COUNTRIESSET ...	UPDATE
240487	HR	HR	COUNTRIES	FGA_COUNTRIES	select * from countries	SELECT
240487	HR	HR	COUNTRIES	FGA_COUNTRIES	Update hr.COUNTRIESSET ...	UPDATE
240487	HR	HR	COUNTRIES	FGA_COUNTRIES	Update hr.COUNTRIESSET ...	UPDATE
240487	HR	HR	COUNTRIES	FGA_COUNTRIES	Update hr.COUNTRIESSET ...	UPDATE

- Code form Thống kê thông tin, hoạt động của các user:

```
public Admin_Statistical() {
    initComponents();

    this.setLocationRelativeTo(null); // Chỉnh form nằm giữa màn hình

    // Load dữ liệu lên table -----
    dao=new DAO_Admin_Statistical();
    vheader.add("session_id");
    vheader.add("db_user");
    vheader.add("object_schema");
    vheader.add("object_name");
    vheader.add("policy_name");
    vheader.add("sql_text");
    vheader.add("statement_type");

    dao.load(vdata);
    jTable1.setModel(new DefaultTableModel(vdata,vheader));

    // Hiển thị user bị khóa -----
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        con = DriverManager.getConnection(BienCucBo.ketnoi,
            BienCucBo.userN, BienCucBo.passW);
        String sql = "select count(*) as Khoa from sys.dba_users where account_status!='OPEN'";
        pst = con.prepareStatement(sql);
    }
}
```

```

        rs = pst.executeQuery();

        if(rs.next()){
            this.lblDangmo.setText("Số lượng các tài khoản đang bị khóa: " +rs.getString("Khoa"));
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }

    // Hiển thị user đang mở -----
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        con = DriverManager.getConnection(BienCucBo.ketnoi,
                                         BienCucBo.userN, BienCucBo.passW);
        String sql = "select count(*) as Mo from sys.dba_users where account_status='OPEN'";
        pst = con.prepareStatement(sql);
    }

```

```

    // Hiển thị user đang mở -----
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        con = DriverManager.getConnection(BienCucBo.ketnoi,
                                         BienCucBo.userN, BienCucBo.passW);
        String sql = "select count(*) as Mo from sys.dba_users where account_status='OPEN'";
        pst = con.prepareStatement(sql);

        rs = pst.executeQuery();

        if(rs.next()){
            this.lblDanglock.setText("Số lượng các tài khoản đang mở: " +rs.getString("Mo"));
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

```

- Code button In Thông kê thông tin, hoạt động của các user:

```

// Print button -
private void btn_InfongkeActionPerformed(java.awt.event.ActionEvent evt) {
    if (JOptionPane.showConfirmDialog(null, "Bạn có muốn xóa không?")
        == JOptionPane.OK_OPTION)
    {
        try {
            File f = new File("D:\\Project-DuAn\\QLThuVien_HQTCSQL-Oracle\\Print\\Admin_Statistical.txt");
            if(!f.exists()){
                f.createNewFile();
            }

            FileWriter fw = new FileWriter(f.getAbsoluteFile());
            BufferedWriter bw = new BufferedWriter(fw);

            bw.write("\t\t\tTHÔNG TIN HOẠT ĐỘNG USER\n\n");
            bw.write("SESSION_ID DB_USER OBJECT_SCHEMA OBJECT_NAME POLICY_NAME SQL_TEXT");
            bw.write("\n");
            for(int i=0; i<jTable1.getRowCount(); i++){
                for(int j=0; j<jTable1.getColumnCount(); j++){
                    bw.write((String)jTable1.getModel().getValueAt(i, j)+ " ");
                }
                bw.write("\n");
            }
        }
    }
}

```

```

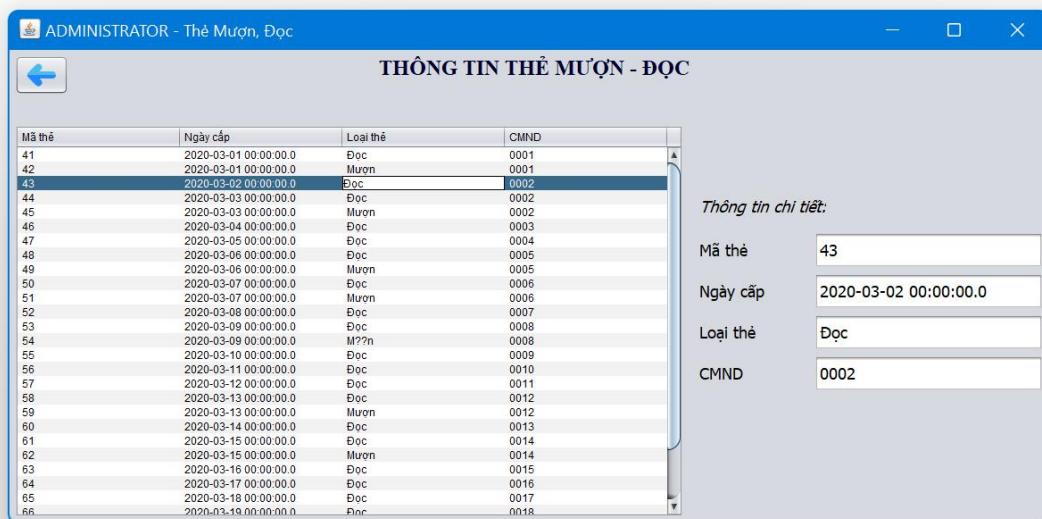
        }

        bw.close();
        fw.close();
        JOptionPane.showMessageDialog(this, "In thành công!");

    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

- Giao diện quản lý thông tin Thẻ (thẻ đọc và thẻ mượn):



- Code form quản lý thông tin Thẻ (thẻ đọc và thẻ mượn):

```

public Admin_Card() {
    initComponents();

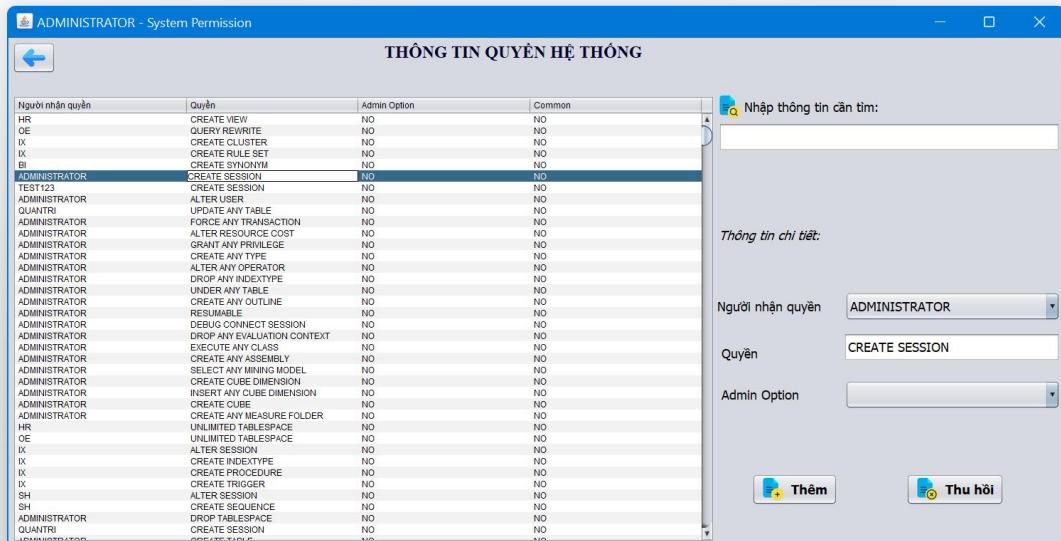
    this.setLocationRelativeTo(null); // Chỉnh form nằm giữa màn hình

    // Load dữ liệu lên table -----
    dao=new DAO_Admin_The();
    vheader.add("Mã thẻ");
    vheader.add("Ngày cấp");
    vheader.add("Loại thẻ");
    vheader.add("CMND");

    dao.load(vdata);
    jTable1.setModel(new DefaultTableModel(vdata,vheader));
}

```

- Giao diện quản lý thông tin về Quyền hệ thống:



- Code form quản lý thông tin về Quyền hệ thống:

```

public Admin_SysPermission() {
    initComponents();
    initComboBoxtenuser();
    this.setLocationRelativeTo(null); // Chỉnh form nằm giữa màn hình

    String[] s={"With admin option"};

    dao=new DAO_Admin_SysPermission();
    vheader.add("Người nhận quyền");
    vheader.add("Quyền");
    vheader.add("Admin Option");
    vheader.add("Common");
    dao.LayDSAdmin_SysPermission_vector(vdata);
    jTable1.setModel(new DefaultTableModel(vdata,vheader));

    cbbwithadmiin.removeAllItems();
    cbbwithadmiin.addItem("");
    for (String item : s) {
        cbbwithadmiin.addItem(item);
    }
}

```

- Code load dữ liệu từ database lên combobox Tên user:

```
// Khởi tạo combobox tên ser -----
private void initComboBoxtenuser() {
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection(BienCucBo.ketnoi,
            BienCucBo.userN, BienCucBo.passW);
        String sql = "select * from sys.xemtenuserhienco";
        PreparedStatement pstmt = con.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery();
        cbbtenuser.removeAllItems();
        while(rs.next()){
            cbbtenuser.addItem(rs.getString("username"));
        }
        rs.close();
        pstmt.close();
        con.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

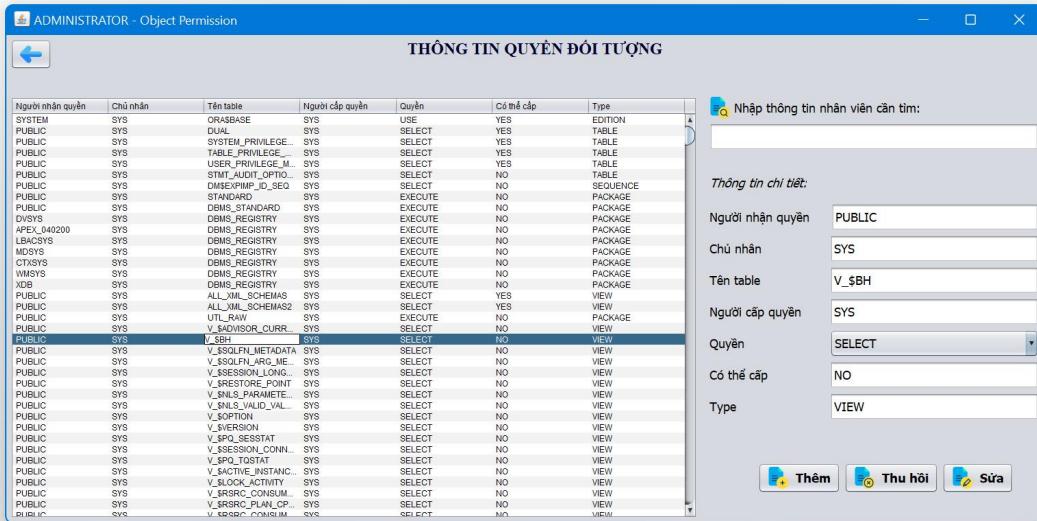
- Code Thêm quyền cho user được chọn:

```
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    if (JOptionPane.showConfirmDialog(this, "Bạn có muốn thêm không?") ==
        == JOptionPane.OK_OPTION) {
        pojo_Admin_SysPermission x = new pojo_Admin_SysPermission(txtquyen.getText(),
            cbbtenuser.getSelectedItem().toString(),cbbwithadmin.getSelectedItem().toString(),"");
        dao.Themquyen(x);
        dao.LayDSAdmin_SysPermission_vector(vdata);
        jTable1.updateUI();
        JOptionPane.showMessageDialog(null, "Thành công");
    }
}
```

- Code Thu hồi quyền của 1 user được chọn:

```
private void btnRevokeActionPerformed(java.awt.event.ActionEvent evt) {
    if (JOptionPane.showConfirmDialog(this, "Bạn có muốn Thu hồi không?") ==
        == JOptionPane.OK_OPTION) {
        pojo_Admin_SysPermission x = new pojo_Admin_SysPermission(txtquyen.getText(),
            cbbtenuser.getSelectedItem().toString(),"","");
        dao.thuhoiquyen(x);
        dao.LayDSAdmin_SysPermission_vector(vdata);
        jTable1.updateUI();
        JOptionPane.showMessageDialog(null, "Thành công");
    }
}
```

- Giao diện quản lý về thông tin của Quyền đối tượng:



- Code form quản lý về thông tin của Quyền đối tượng:

```
public Admin_ObjPermission() {
    initComponents();

    this.setLocationRelativeTo(null); // Chính form nằm giữa màn hình

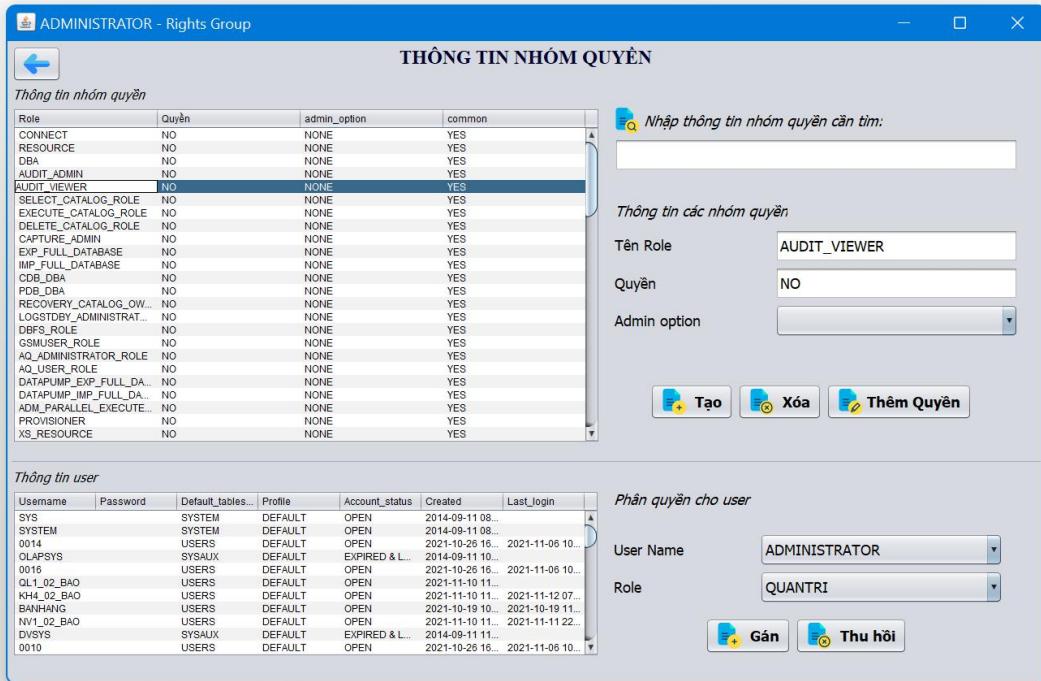
    initComboBoxPrivilege();

    dao = new DAO_Admin_ObjPermission();
    vheader.add("Người nhận quyền");
    vheader.add("Chủ nhân");
    vheader.add("Tên table");
    vheader.add("Người cấp quyền");
    vheader.add("Quyền");
    vheader.add("Có thể cấp");
    vheader.add("Type");
    dao.LayDSAdmin_ObjPermission_vecter(vdata);
    jTable1.setModel(new DefaultTableModel(vdata,vheader));
}
```

- Code load dữ liệu từ database lên combobox Privilege:

```
// Khởi tạo phần combobox -----
private void initComboBoxPrivilege(){
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection(BienCucBo.ketnoi,
                                                     BienCucBo.userN, BienCucBo.passW);
        String sql = "SELECT unique privilege FROM sys.dba_tab_privs";
        PreparedStatement pstmt = con.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery();
        cbbQuyen.removeAllItems();
        while(rs.next()){
            cbbQuyen.addItem(rs.getString("privilege"));
        }
        rs.close();
        pstmt.close();
        con.close();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
        e.printStackTrace();
    }
}
```

- Giao diện quản lý thông tin Nhóm quyền:



- Code form quản lý thông tin Nhóm quyền:

```
public Admin_RightsGroup() {
    initComponents();

    this.setLocationRelativeTo(null); // Chỉnh form nằm giữa màn hình
    initComboBoxtenuser();
    initComboBoxRole();

    String[] s={"With admin option"};
    cbbadminoption.removeAllItems();
    cbbadminoption.addItem("");
    for (String item : s) {
        cbbadminoption.addItem(item);
    }

    daoRole = new DAO_Admin_RightsGroup();
    vheader.add("Role");
    vheader.add("Quyền");
    vheader.add("admin_option");
    vheader.add("common");
    daoRole.LayDSAdmin_RightsGroup_vector(vdata);
    tblRole.setModel(new DefaultTableModel(vdata,vheader));
}
```

```
    daoUser=new DAO_Admin_TaoUser();
    vheaderU.add("Username");
    vheaderU.add("Password");
    vheaderU.add("Default_tablespace");
    vheaderU.add("Profile");
    vheaderU.add("Account_status");
    vheaderU.add("Created");
    vheaderU.add("Last_login");
    daoUser.LayDSAdmin_TaoUser_vector(vdataU);
    tblUser.setModel(new DefaultTableModel(vdataU,vheaderU));
}
```

- Code load dữ liệu lên combobox Tên user:

```
// Khởi tạo combobox tên ser -----
private void initComboBoxtenuser(){
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection(BienCucBo.ketnoi,
            BienCucBo.userN, BienCucBo.passW);
        String sql = "select * from sys.xemtenuserhienco";
        PreparedStatement pstmt = con.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery();
        cbbtenuser.removeAllItems();
        while(rs.next()){
            cbbtenuser.addItem(rs.getString("username"));
        }
        rs.close();
        pstmt.close();
        con.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

- Code load dữ liệu từ database lên combobox Role:

```
// Khởi tạo phần combobox -----
private void initComboBoxRole() {
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection(BienCucBo.ketnoi,
            BienCucBo.userN, BienCucBo.passW);
        String sql = "select * from sys.tenhomdatao";
        PreparedStatement pstmt = con.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery();
        cbbRole.removeAllItems();
        while(rs.next()){
            cbbRole.addItem(rs.getString("role"));
        }
        rs.close();
        pstmt.close();
        con.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

- Code xóa quyền:

```
// Xóa quyền -----
private void btnXoaActionPerformed(java.awt.event.ActionEvent evt) {
    // Nếu đ为空 -> không thực thi
    if(txttenRole.getText().length()==0)//rỗng
    {
        JOptionPane.showMessageDialog(null, "Tên nhóm quyền không được đ为空");
        return;
    }

    if (BienCucBo.ktxoa()) // đồng ý thực hiện thao tác
    {
        if(kttrung()==false) // không tồn tại khóa chính
        {
            JOptionPane.showMessageDialog(null, "Nhóm quyền Không tồn tại");
            return;
        }
        else // khong trùng
        {
            pojo_Admin_RightsGroup x = new pojo_Admin_RightsGroup(txttenRole.getText(),
                txtquyen.getText(), cbadminoption.getSelectedItem().toString(), "", "");
            daoRole.XoaNhomquyen(x);
            daoRole.LayDSAdmin_RightsGroup_vector(vdata);
            tblRole.updateUI();
        }
    }
}
```

```
if(kttrung()==false)// Không tồn tại tức là đã xóa thành công
{
    JOptionPane.showMessageDialog(null, "Thành công");
    return;
}
else
{
    JOptionPane.showMessageDialog(null, "Thất Bại");
    return;
}
}
```

- Code Thêm vào nhóm quyền:

```
// Thêm vào nhóm quyền -----
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    if (JOptionPane.showConfirmDialog(this, "Bạn có muốn thêm không?") == JOptionPane.OK_OPTION) {
        pojo_Admin_RightsGroup x = new pojo_Admin_RightsGroup(txttenRole.getText(),
            txtquyen.getText(), cbbadminoption.getSelectedItem().toString(), "", "");
        daoRole.ThemquyenNhomquyen(x);
        daoRole.LayDSAdmin_RightsGroup_vector(vdata);
        tblRole.updateUI();
        JOptionPane.showMessageDialog(null, "Thành công");
    }
}
```

- Code Tạo nhóm quyền:

```
// Tạo nhóm quyền -----
private void btnCreateActionPerformed(java.awt.event.ActionEvent evt) {
    // Nếu để rỗng -> không thực thi
    if(ktrong()==false)
    {
        return;
    }

    if(BienCucBo.ktthem()) // Đóng ý thực hiện thao tác
    {
        if(kttrung()==true) //trùng mã sách
        {
            JOptionPane.showMessageDialog(null, "Tên nhóm quyền đã tồn tại!");
            return;
        }
        else //mã sách không trùng
        {
            pojo_Admin_RightsGroup x = new pojo_Admin_RightsGroup(txttenRole.getText(),
                txtquyen.getText(), cbbadminoption.getSelectedItem().toString(), "", "");
            daoRole.TaoNhomquyen(x);
            daoRole.LayDSAdmin_RightsGroup_vector(vdata);
            tblRole.updateUI();
        }
    }
}
```

```
    if(kttrung()==true) // trùng là đã thêm vào thành công
    {
        JOptionPane.showMessageDialog(null, "Thành công");
        return;
    }
    else // không trùng không có trong table
    {
        JOptionPane.showMessageDialog(null, "Thất bại");
        return;
    }
}
```

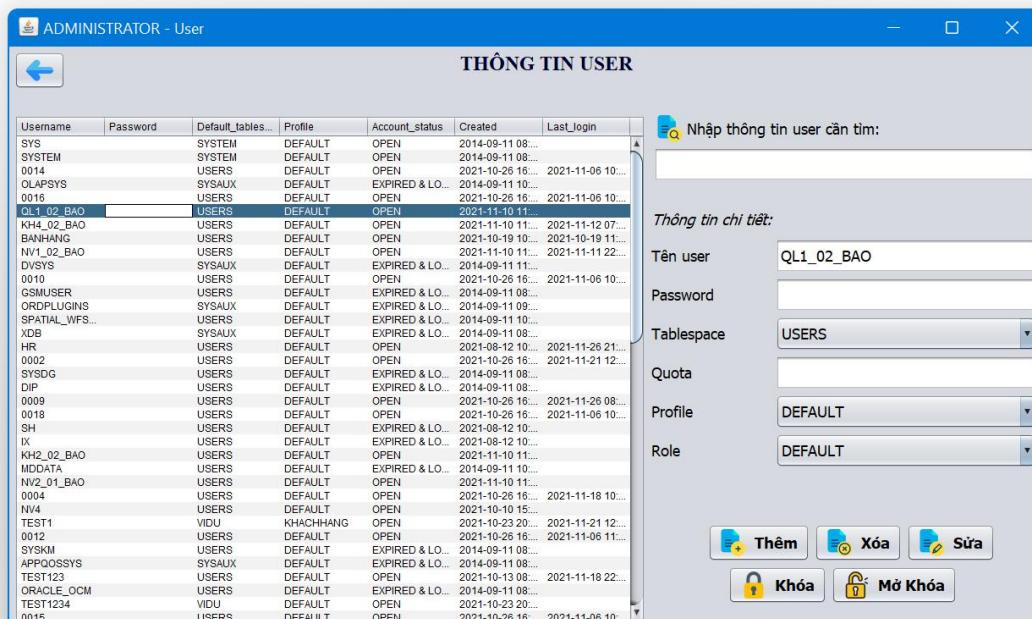
- Code Thu hồi quyền của user:

```
// Thu hồi quyền của user -----
private void btnRevokeRoleActionPerformed(java.awt.event.ActionEvent evt) {
    if (JOptionPane.showConfirmDialog(this, "Bạn có muốn thu hồi nhóm Quyền cho user này không?") == JOptionPane.OK_OPTION) {
        pojo_Admin_RightsGroup x = new pojo_Admin_RightsGroup(cbbRole.getSelectedItem().toString(),
            cbbtenuser.getSelectedItem().toString(), "", "", "");
        daoRole.ThuhoiNhomquyenchoUser(x);
        daoRole.LayDSAdmin_RightsGroup_vector(vdata);
        tblRole.updateUI();
        JOptionPane.showMessageDialog(null, "Thu hồi thành công");
    }
}
```

- Code Gán quyền cho user:

```
// Gán quyền cho user -----
private void btnPasteRoleActionPerformed(java.awt.event.ActionEvent evt) {
    if (JOptionPane.showConfirmDialog(this, "Bạn có muốn thêm nhóm quyền cho user này không?") == JOptionPane.OK_OPTION) {
        pojo_Admin_RightsGroup x = new pojo_Admin_RightsGroup(txttenRole.getText(),
            txtquyen.getText(), cbbadminoption.getSelectedItem().toString(), "", "");
        daoRole.ThemNhomquyenchoUser(x);
        daoRole.LayDSAdmin_RightsGroup_vector(vdata);
        tblRole.updateUI();
        JOptionPane.showMessageDialog(null, "Gán quyền thành công");
    }
}
```

- Giao diện quản lý thông tin 1 User:



- Code form quản lý thông tin 1 User:

```
public Admin_CreateUser() {
    initComponents();
    this.setLocationRelativeTo(null); // Chỉnh form nằm giữa màn hình

    initComboBoxTablespace();
    initComboBoxTenRole();
    initComboBoxProfile();

    dao=new DAO_Admin_TaoUser();
    vheader.add("Username");
    vheader.add("Password");
    vheader.add("Default_tablespace");
    vheader.add("Profile");
    vheader.add("Account_status");
    vheader.add("Created");
    vheader.add("Last_login");
    dao.LayDSAdmin_TaoUser_vector(vdata);
    jTable1.setModel(new DefaultTableModel(vdata,vheader));
}
```

- Code load dữ liệu từ database lên combobox Tablespace:

```

private void initComboBoxTablespace() {
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection(BienCucBo.ketnoi,
            BienCucBo.userN, BienCucBo.passW);
        String sql = "select * from sys.view_tentablespace";
        PreparedStatement pstmt = con.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery();
        cbbdefault_tablespace.removeAllItems();
        cbbdefault_tablespace.addItem("DEFAULT");
        while(rs.next()) {
            cbbdefault_tablespace.addItem(rs.getString("NAME"));
        }
        rs.close();
        pstmt.close();
        con.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

- Code load dữ liệu từ database lên combobox Profile:

```

private void initComboBoxprofile() {
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection(BienCucBo.ketnoi,
            BienCucBo.userN, BienCucBo.passW);
        String sql = "select * from sys.view_tenprofile";
        PreparedStatement pstmt = con.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery();
        cbbprofile.removeAllItems();
        cbbprofile.addItem("DEFAULT");
        while(rs.next()) {
            cbbprofile.addItem(rs.getString("profile"));
        }
        rs.close();
        pstmt.close();
        con.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

- Code load dữ liệu từ database lên combobox Role:

```

private void initComboBoxtenrole() {
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection(BienCucBo.ketnoi,
            BienCucBo.userN, BienCucBo.passW);
        String sql = "select * from sys.tennhom";
        PreparedStatement pstmt = con.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery();
        cbbrole.removeAllItems();
        cbbrole.addItem("DEFAULT");
        while(rs.next()) {
            cbbrole.addItem(rs.getString("role"));
        }
        rs.close();
        pstmt.close();
        con.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

- Code button Thêm mới 1 nhóm quyền:

```

private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    if(ktrong()==false)
    {
        return;
    }
    if(BienCucBo.ktthem())
    {
        if(kttrung()==true)//trùng mã sách
        {
            JOptionPane.showMessageDialog(null, "User đã tồn tại!");
            return;
        }
        else//mã sách không trùng
        {
            pojo_Admin_TaoUser x = new pojo_Admin_TaoUser(txtusername.getText(),
                txtpassword.getText(),cbbdefault_tablespace.getSelectedItem().toString(),
                txtquota.getText(),cbbprofile.getSelectedItem().toString(),cbbrole.getSelectedItem().toString(),cbbrol
                dao.Themuser(x);
                dao.LayDSAdmin_TaoUser_vector(vdata);
                jTable1.updateUI();
                if(kttrung()==true)//trùng là đã thêm vào thành công
                {
                    JOptionPane.showMessageDialog(null, "Thành công");
                    return;
                }
            }
        else//không trùng không có trống table
        {
            JOptionPane.showMessageDialog(null, "Thất bại");
            return;
        }
    }
}

```

- Code button Xóa 1 nhóm quyền:

```

private void btnXoaActionPerformed(java.awt.event.ActionEvent evt) {
    if(txtusername.getText().length()==0)//rỗng
    {
        JOptionPane.showMessageDialog(null, "Username không được để trống");
        return;
    }
    if (BienCucBo.ktxoa())//tồn tại
    {
        if(kttrung()==false)//không tồn tại khóa chính
        {
            JOptionPane.showMessageDialog(null, "Username Không tồn tại");
            return;
        }
        else//không trùng
        {
            pojo_Admin_TaoUser x = new pojo_Admin_TaoUser();
            x.setUsername(txtusername.getText().toUpperCase());
            dao.Xoauser(x);
            dao.LayDSAdmin_TaoUser_vector(vdata);
            jTable1.updateUI();
            if(kttrung()==false)//Không tồn tại
            {
                JOptionPane.showMessageDialog(null, "Thành công");
                return;
            }
            else
            {
                JOptionPane.showMessageDialog(null, "Thất bại");
                return;
            }
        }
    }
}

```

- Code button Sửa 1 nhóm quyền:

```

private void btnSuaActionPerformed(java.awt.event.ActionEvent evt) {
    if(ktrong()==false)//rỗng
    {
        return;
    }
    if (BienCucBo.ksua())//tồn tại
    {
        if(kttrung()==false)//không tồn tại khóa chính
        {
            JOptionPane.showMessageDialog(null, "Username Không tồn tại ");
            return;
        }
        else//không trùng
        {
            pojo_Admin_TaoUser x = new pojo_Admin_TaoUser(txtusername.getText(),
            txtpassword.getText(),cbbdefault_tablespace.getSelectedItem().toString(),
            txtquota.getText(),cbbprofile.getSelectedItem().toString(),cbbrole.getSelectedItem().toString(),cbbrol
            dao.Suauser(x);
            dao.LayDSAdmin_TaoUser_vector(vdata);
            jTable1.updateUI();
            if(kttrung()==true)//trùng
            {
                JOptionPane.showMessageDialog(null, "Thành công");
                return;
            }
            else
            {
                JOptionPane.showMessageDialog(null, "Thất bại");
                return;
            }
        }
    }
}

```

- Code button Mở khóa cho 1 User:

```

private void btnMokhoaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if(txtusername.getText().length()==0)//rỗng
    {
        JOptionPane.showMessageDialog(null, "Username không được để trống");
        return;
    }
    if(JOptionPane.showConfirmDialog(null, "Bạn có muốn Khóa user không?") == JOptionPane.OK_OPTION)
    {
        if(kttrung()==false)//không tồn tại khóa chính
        {
            JOptionPane.showMessageDialog(null, "Username Không tồn tại");
            return;
        }
        else
        {
            if(ktlockuser()==true)
            {
                JOptionPane.showMessageDialog(null, "User Đang ở trạng thái mở");
                return;
            }
            else
            {
                pojo_Admin_TaoUser x = new pojo_Admin_TaoUser();
                x.setUsername(txtusername.getText().toUpperCase());
            }
        }
    }
    else
    {
        pojo_Admin_TaoUser x = new pojo_Admin_TaoUser();
        x.setUsername(txtusername.getText().toUpperCase());
        dao.Mokhoauser(x);
        dao.LayDSAdmin_TaoUser_vector(vdata);
        jTable1.updateUI();
        if(ktlockuser()==true)
        {
            JOptionPane.showMessageDialog(null, "Thành công");
            return;
        }
        else
        {
            JOptionPane.showMessageDialog(null, "Thất bại");
            return;
        }
    }
}

```

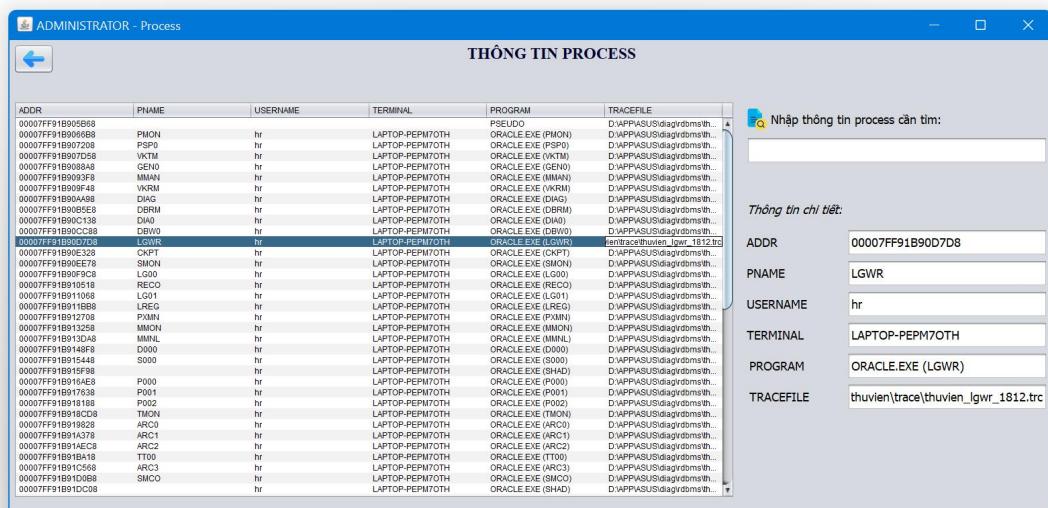
- Code button Khóa 1 User:

```

private void btkhoaActionPerformed(java.awt.event.ActionEvent evt) {
    if(txtusername.getText().length()==0)//rỗng
    {
        JOptionPane.showMessageDialog(null, "Username không được để trống");
        return;
    }
    if(JOptionPane.showConfirmDialog(null, "Bạn có muốn Khóa user không?")== JOptionPane.OK_OPTION)
    {
        if(kttrung()==false)//không tồn tại khóa chính
        {
            JOptionPane.showMessageDialog(null, "Username Không tồn tại");
            return;
        }
        else
        {
            if(ktlockuser()==false)
            {
                JOptionPane.showMessageDialog(null, "User Đang ở trạng thái khóa");
                return;
            }
            else
            {
                pojo_Admin_TaoUser x = new pojo_Admin_TaoUser();
                x.setUsername(txtusername.getText().toUpperCase());
                dao.Khoauser(x);
                dao.LayDSAAdmin_TaoUser_vecter(vdata);
                jTable1.updateUI();
                if(ktlockuser()==false)
                {
                    JOptionPane.showMessageDialog(null, "Thành công");
                    return;
                }
                else
                {
                    JOptionPane.showMessageDialog(null, "Thất bại");
                    return;
                }
            }
        }
    }
}

```

- Giao diện quản lý thông tin các Process:



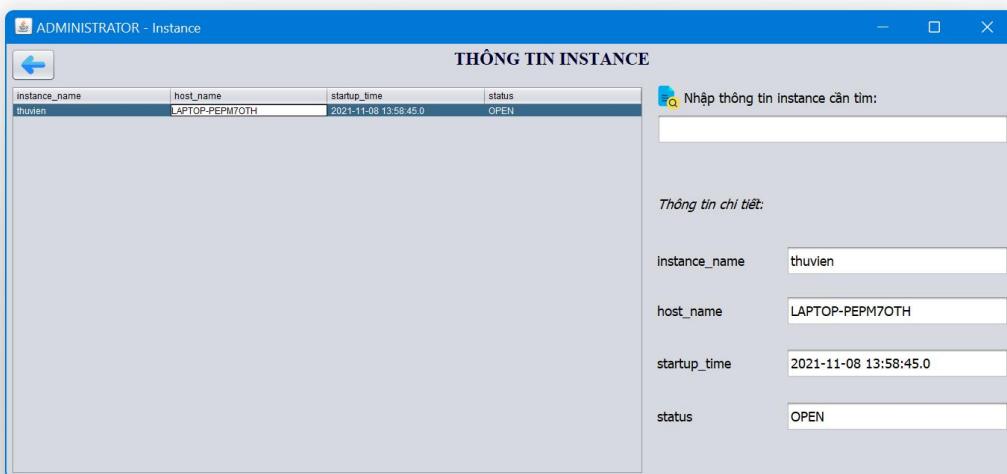
- Code form quản lý thông tin các Process:

```
public Admin_Process() {
    initComponents();

    this.setLocationRelativeTo(null); // Chỉnh form nằm giữa màn hình

    dao=new DAO_Admin_Process();
    vheader.add("ADDR");
    vheader.add("PNAME");
    vheader.add("USERNAME");
    vheader.add("TERMINAL");
    vheader.add("PROGRAM");
    vheader.add("TRACEFILE");
    dao.LayDSAdmin_Process_vector(vdata);
    jTableprocess.setModel(new DefaultTableModel(vdata,vheader));
}
```

- Giao diện quản lý Instance:



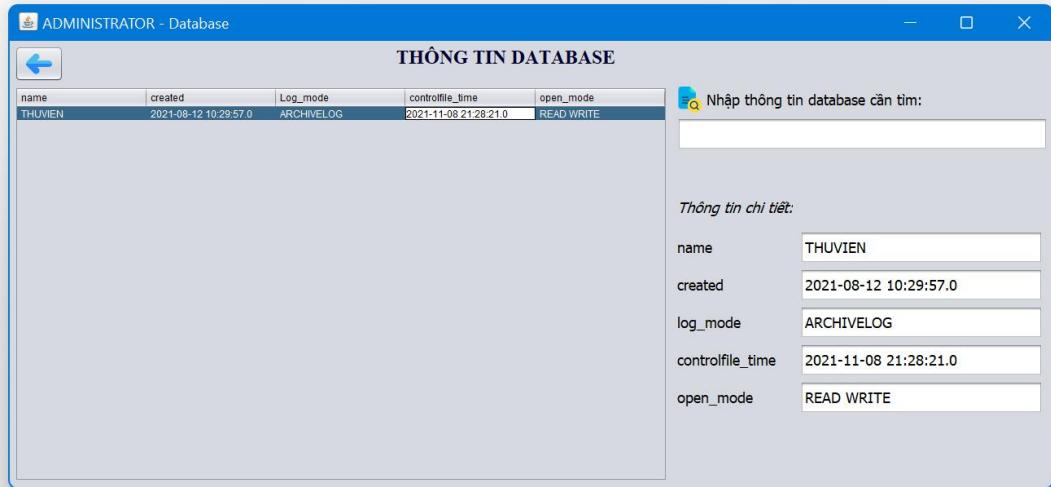
- Code form quản lý Instance:

```
public Admin_Instance() {
    initComponents();

    this.setLocationRelativeTo(null); // Chỉnh form nằm giữa màn hình

    dao=new DAO_Admin_Instance();
    vheader.add("instance_name");
    vheader.add("host_name");
    vheader.add("startup_time");
    vheader.add("status");
    dao.LayDSAdmin_Instance_vector(vdata);
    jTableinstance.setModel(new DefaultTableModel(vdata,vheader));
}
```

- Giao diện quản lý Database:



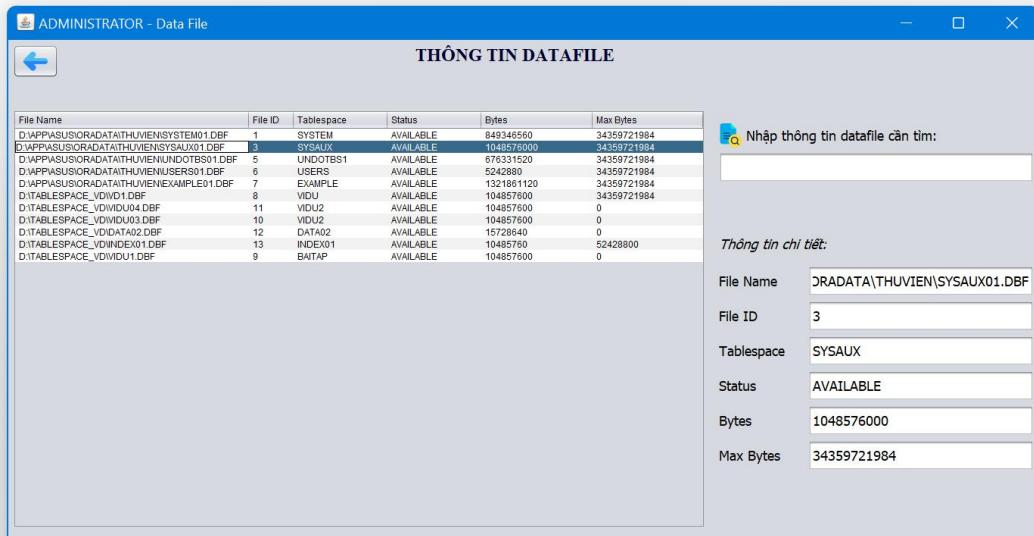
- Code form quản lý Database:

```
public Admin_Database() {
    initComponents();

    this.setLocationRelativeTo(null); // Chỉnh form nằm giữa màn hình

    dao=new DAO_Admin_Database();
    vheader.add("name");
    vheader.add("created");
    vheader.add("Log_mode");
    vheader.add("controlfile_time");
    vheader.add("open_mode");
    dao.LayDSAdmin_Database_vector(vdata);
    jTabledatabase.setModel(new DefaultTableModel(vdata,vheader));
}
```

- Giao diện quản lý thông tin Datafile:



- Code form quản lý thông tin Datafile:

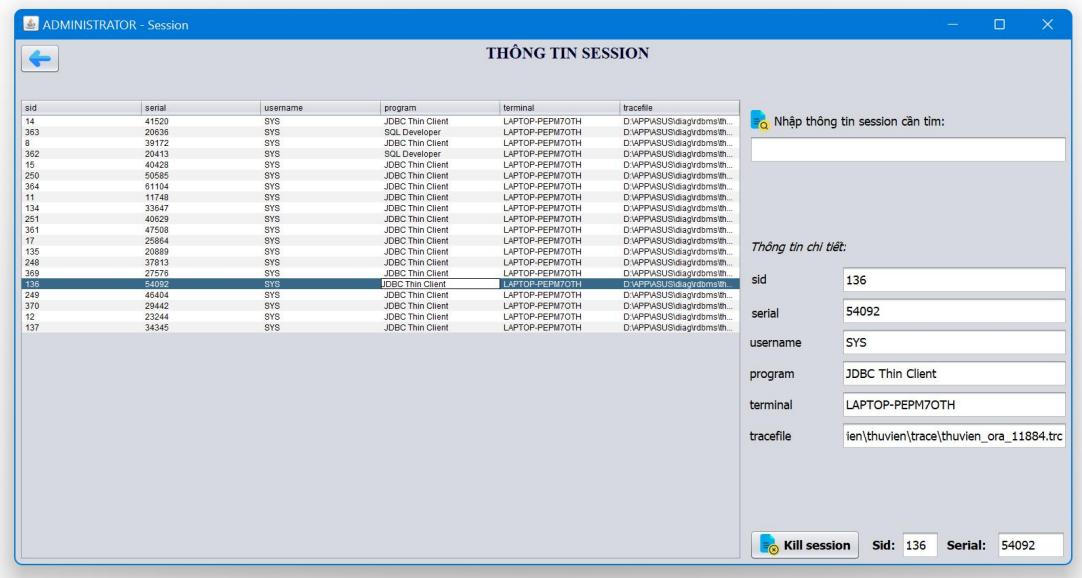
```
public Admin_Datafile() {
    initComponents();

    this.setLocationRelativeTo(null); // Chính form nằm giữa màn hình

    dao=new DAO_Admin_Datafile();
    vheader.add("File Name");
    vheader.add("File ID");
    vheader.add("Tablespace");
    vheader.add("Status");
    vheader.add("Bytes");
    vheader.add("Max Bytes");
    dao.LayDSAdmin_Datafile_vector(vdata);
    jTableDatafile.setModel(new DefaultTableModel(vdata,vheader));

}
```

- Giao diện quản lý thông tin Session:



- Code form quản lý thông tin Session:

```
public Admin_Session() {
    initComponents();

    this.setLocationRelativeTo(null); // Chính form nằm giữa màn hình

    dao=new DAO_Admin_Session();
    vheader.add("sid");
    vheader.add("serial");
    vheader.add("username");
    vheader.add("program");
    vheader.add("terminal");
    vheader.add("tracefile");
    dao.LayDSAdmin_Session_vector(vdata);
    jTableSession.setModel(new DefaultTableModel(vdata,vheader));

}
```

- Code button Kill Session của form quản lý thông tin Session:

```

private void btnKillActionPerformed(java.awt.event.ActionEvent evt) {
    if(txtkillsid.getText().equals("")) // Nếu sid rỗng
    {
        JOptionPane.showMessageDialog(null, "Vui lòng sid của session cần kill!");
        return;
    }
    if(txtkillserial.getText().equals("")) // Nếu serial rỗng
    {
        JOptionPane.showMessageDialog(null, "Vui lòng serial của session cần kill!");
        return;
    }
    else
    {
        if(JOptionPane.showConfirmDialog(null, "Bạn có muốn kill session này không?") == JOptionPane.OK_OPTION)
        {
            if(kttrung() == false) //không trùng=> không tồn tại
            {
                JOptionPane.showMessageDialog(null, "Session Không tồn tại xin vui lòng kiểm tra lại!"
                    + "sid và serial của session!");
                return;
            }
            else
            {
                pojo_Admin_Session x = new pojo_Admin_Session();

```

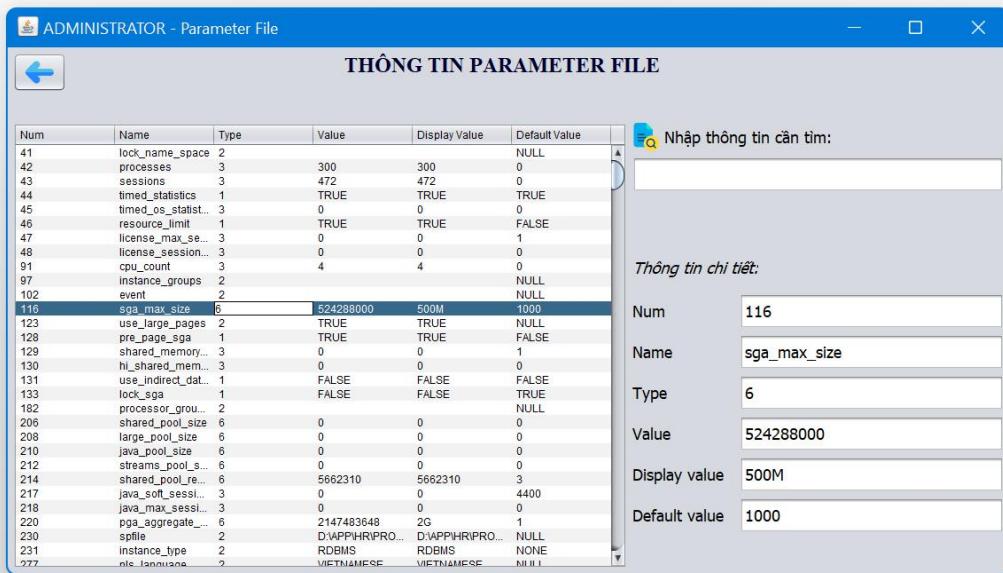


```

                else
                {
                    pojo_Admin_Session x = new pojo_Admin_Session();
                    pojo_Admin_Session y = new pojo_Admin_Session();
                    x.setSid(txtkillsid.getText());
                    y.setSerial(txtkillserial.getText());
                    dao.KillSession(x, y);
                    dao.LayDSAdmin_Session_vector(vdata);
                    jTableSession.updateUI();
                    if(kttrung() == false)
                    {
                        JOptionPane.showMessageDialog(null, "Thất bại");
                        return;
                    }
                    else
                    {
                        JOptionPane.showMessageDialog(null, "Thành công");
                        return;
                    }
                }
            }
        }
    }
}

```

- Giao diện quản lý thông tin Parameter file:



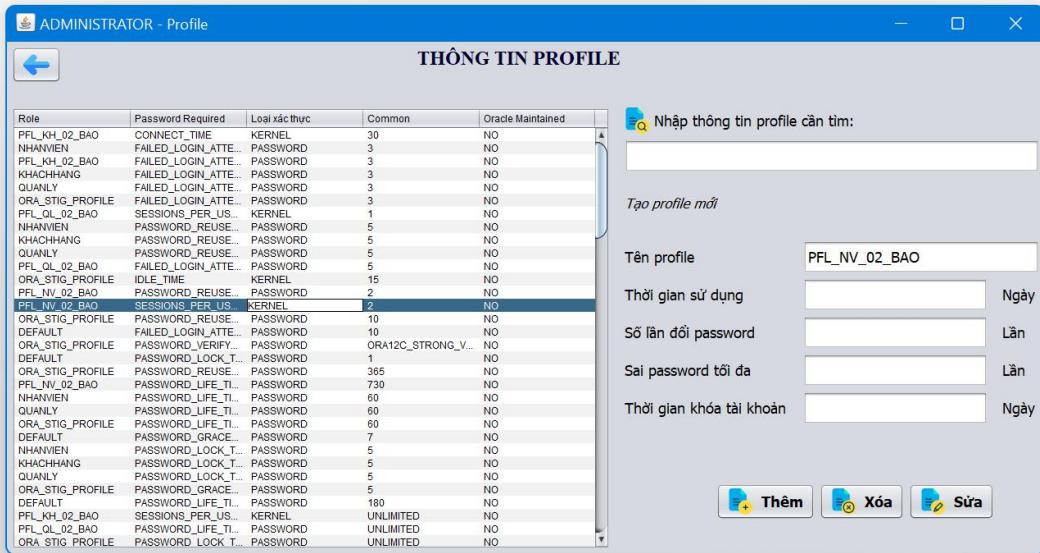
- Code form quản lý thông tin Parameter file:

```
public Admin_Parameter() {
    initComponents();

    this.setLocationRelativeTo(null); // Chính form nằm giữa màn hình

    dao = new DAO_Admin_Parameter();
    vHeader.add("Num");
    vHeader.add("Name");
    vHeader.add("Type");
    vHeader.add("Value");
    vHeader.add("Display Value");
    vHeader.add("Default Value");
    dao.LayDSAdmin_Parameter_vector(vData);
    jTable1.setModel(new DefaultTableModel(vData, vHeader));
}
```

- Giao diện quản lý thông tin của Profile:



- Code form quản lý thông tin của Profile:

```
public Admin_Profile() {
    initComponents();

    this.setLocationRelativeTo(null); // Chính form nằm giữa màn hình

    dao = new DAO_Admin_Profile();
    vheader.add("Role");
    vheader.add("Password Required");
    vheader.add("Loại xác thực");
    vheader.add("Common");
    vheader.add("Oracle Maintained");
    dao.LayDSAdmin_Profile_vector(vdata);
    jTable1.setModel(new DefaultTableModel(vdata, vheader));
}
```

- Code button Xóa Profile:

```
// Xóa profile -----
private void btnXoaActionPerformed(java.awt.event.ActionEvent evt) {
    Connection con = null;
    CallableStatement callableStmt = null;

    if (JOptionPane.showConfirmDialog(this, "Bạn có muốn xóa không?")
        == JOptionPane.OK_OPTION) {
        try {
            con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ThuVien",
                BienCucBo.userN ,BienCucBo.passW);
            String command = "{call xoa_profile(?)}";

            callableStmt = con.prepareCall(command);

            callableStmt.setString(1, txtProfileName.getText());

            // execute database Stored procedure
            callableStmt.executeUpdate();

            if(callableStmt == null){
                JOptionPane.showMessageDialog(null, "Không thể xóa Profile!");
            }
            else{
                JOptionPane.showMessageDialog(null, "Bạn đã xóa Profile thành công!");

                dao.LayDSAdmin_Profile_vector(vdata);
                jTable1.updateUI();
            }
        } catch(Exception ex) {
            JOptionPane.showMessageDialog(null, ex);
        }
    }
}
```

- Code button Sửa thông tin của Profile:

```
// Sửa profile -----
private void btnSuaActionPerformed(java.awt.event.ActionEvent evt) {
    Connection con = null;
    CallableStatement callableStmt = null;

    if (JOptionPane.showConfirmDialog(this, "Bạn có muốn sửa không?")
        == JOptionPane.OK_OPTION) {
        try {
            con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ThuVien",
                BienCucBo.userN ,BienCucBo.passW);
            String command = "{call sua_profile(?,?,?,?,?,?)}";

            callableStmt = con.prepareCall(command);

            callableStmt.setString(1, txtProfileName.getText());
            callableStmt.setString(2, txtPassword_life_time_Day.getText());
            callableStmt.setString(3, txtFailed_login_attempts_Lan.getText());
            callableStmt.setString(4, txtPASSWORD_LOCK_TIME_Day.getText());
            callableStmt.setString(5, txtPASSWORD_REUSE_MAX_Day.getText());

            // execute database Stored procedure
            callableStmt.executeUpdate();

            if(callableStmt == null){
                JOptionPane.showMessageDialog(null, "Không thể sửa Profile!");
            }
            else{
                JOptionPane.showMessageDialog(null, "Bạn đã sửa Profile thành công!");
                dao.LayDSAdmin_Profile_vector(vdata);
                jTable1.updateUI();
            }
        } catch(Exception ex) {
            JOptionPane.showMessageDialog(null, ex);
        }
    }
}
```

- Code button Thêm mới 1 Profile:

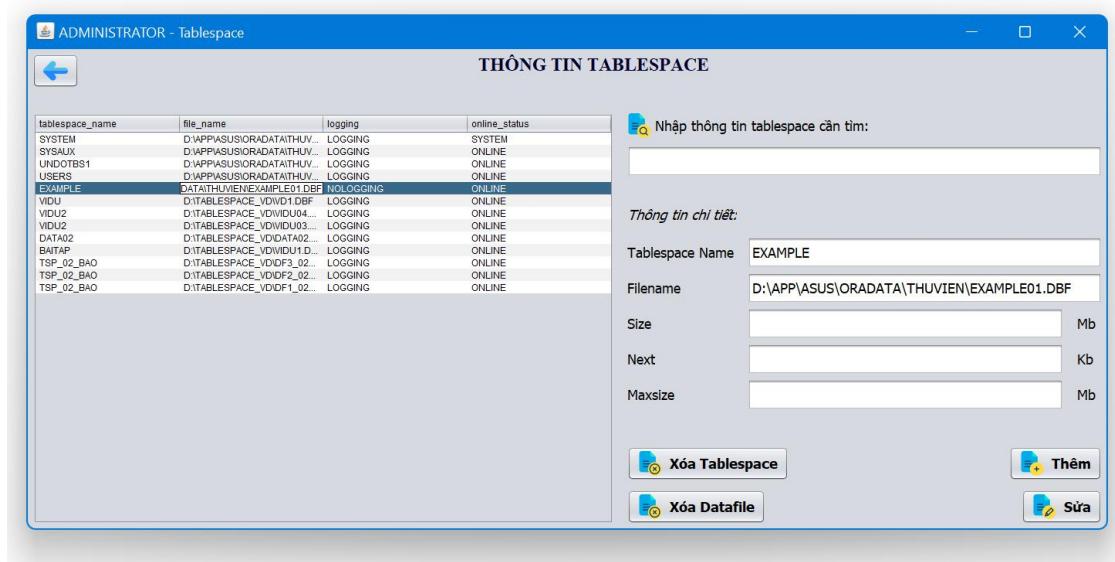
```
// Thêm mới profile -----
private void btnCreateActionPerformed(java.awt.event.ActionEvent evt) {
    Connection con = null;
    CallableStatement callableStmt = null;

    if (JOptionPane.showConfirmDialog(this, "Bạn có muốn thêm không?")
        == JOptionPane.OK_OPTION) {
        try {
            con = DriverManager.getConnection(BienCucBo.ketnoi,
                BienCucBo.userN, BienCucBo.passW);
            String command = "{call sys.tao_profile(?, ?, ?, ?, ?)}";
            callableStmt = con.prepareCall(command);

            callableStmt.setString(1, txtProfileName.getText());
            callableStmt.setString(2, txtPassword_life_time_Day.getText());
            callableStmt.setString(3, txtFailed_login_attempts_Lan.getText());
            callableStmt.setString(4, txtPASSWORD_LOCK_TIME_Day.getText());
            callableStmt.setString(5, txtPASSWORD_REUSE_MAX_Day.getText());

            // execute database Stored procedure
            callableStmt.executeUpdate();
        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null, ex);
        }
    }
}
```

- Giao diện quản lý thông tin của Tablespace:



- Code form quản lý thông tin của Tablespace:

```

public Admin_Tablespace() {
    initComponents();

    this.setLocationRelativeTo(null); // Chỉnh form nằm giữa màn hình

    dao=new DAO_Admin_Tablespace();
    vheader.add("tablespace_name");
    vheader.add("file_name");
    vheader.add("logging");
    vheader.add("online_status");
    dao.load(vdata);
    jTable1.setModel(new DefaultTableModel(vdata,vheader));
}

```

- Code button Thêm mới 1 Tablespace:

```

private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    // Nếu đ为空 -> không thực thi
    if(ktrong()==false)
    {
        return;
    }

    if(BienCucBo.ktthem()) // Đóng ý thực hiện thao tác
    {
        if(kttrung()==true) // trùng mã sách
        {
            JOptionPane.showMessageDialog(null, "Trùng Tablespace name!");
            return;
        }
        else //mã sách không trùng
        {
            pojo_Admin_Tablespace x = new pojo_Admin_Tablespace(txtTablespace.getText(),
            txtFileName.getText(),"",txtSize.getText(),
            txtnext.getText(),txtmaxsize.getText());
            dao.Themtablespace(x);
            dao.load(vdata);
            jTable1.updateUI();

            if(kttrung()==true) // trùng là đã thêm vào thành công
            {
                JOptionPane.showMessageDialog(null, "Thành công");
                return;
            }
            else // không trùng không có trống table
            {
                JOptionPane.showMessageDialog(null, "Thất bại");
                return;
            }
        }
    }
}

```

- Code button Xóa 1 Tablespace:

```

private void btnXoaActionPerformed(java.awt.event.ActionEvent evt) {

    if(txtTablespace.getText().length()==0)//rỗng
    {
        JOptionPane.showMessageDialog(null, "Tablespace name không được để trống");
        return;
    }

    if (BienCucBo.ktxoa()) // đồng ý thực hiện thao tác
    {
        if(kttrung()==false) // không tồn tại khóa chính
        {
            JOptionPane.showMessageDialog(null, "Tablespace Không tồn tại");
            return;
        }
        else // khong trung
        {
            pojo_Admin_Tablespace x = new pojo_Admin_Tablespace();
            x.setTablespace_name(txtTablespace.getText());
            dao.xoatablespace(x);
            dao.load(vdata);
            jTable1.updateUI();

            if(kttrung()==false)// Không tồn tại tức là đã xóa thành công
        }
    }
}

```

- Code button Sửa thông tin của 1 Tablespace:

```

private void btnSuaActionPerformed(java.awt.event.ActionEvent evt) {

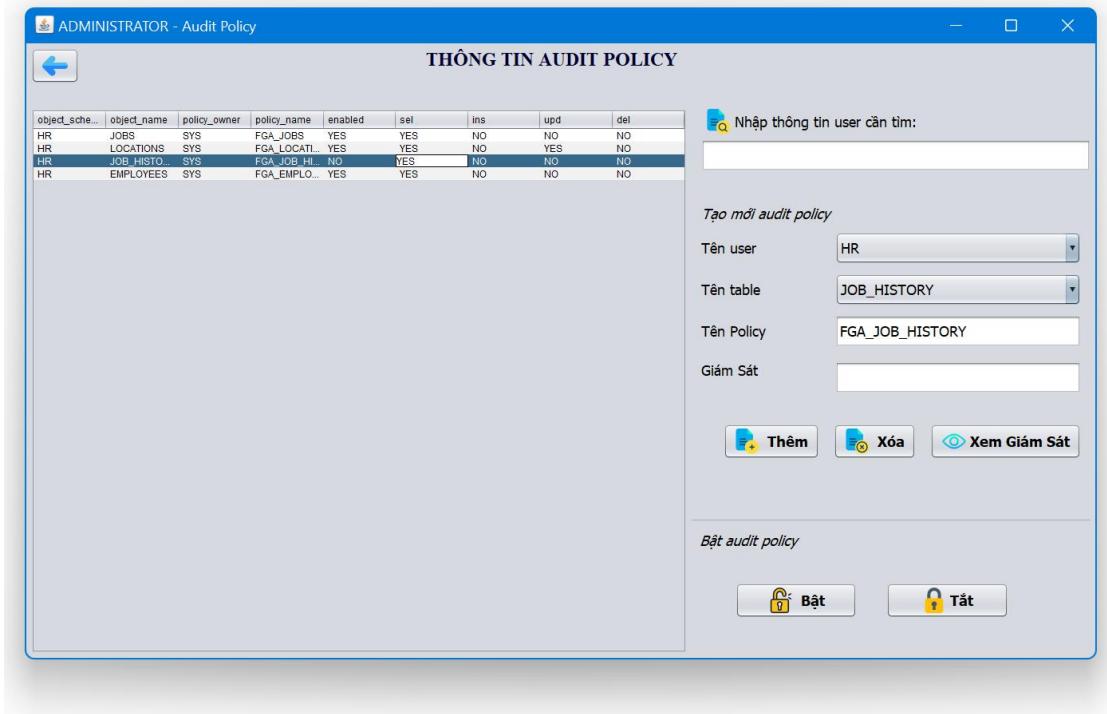
    // Nếu để rỗng -> không thực thi
    if(kttrung()==false)//rỗng
    {
        return;
    }

    if (BienCucBo.ptsua()) // đồng ý thực thi thao tác
    {
        if(kttrung()==false) // không tồn tại khóa chính
        {
            JOptionPane.showMessageDialog(null, "Tablespace Không tồn tại ");
            return;
        }
        else//khong trung
        {
            pojo_Admin_Tablespace x = new pojo_Admin_Tablespace(txtTablespace.getText(),
            txtFileName.getText(),"","","",txtSize.getText(),
            txtnext.getText(),txtmaxsize.getText());
            dao.suatablespace(x);
            dao.load(vdata);
            jTable1.updateUI();

            if(kttrung()==true) // trùng là đã thêm vào thành công
        }
    }
}

```

- Giao diện quản lý thông tin của Audit Policy:



- Code form quản lý thông tin của Audit Policy:

```

public Admin_Audit() {
    initComponents();

    this.setLocationRelativeTo(null); // Chỉnh form nằm giữa màn hình

    initComboBoxObject_schema();

    dao=new DAO_Admin_Audit();
    vheader.add("object_schema");
    vheader.add("object_name");
    vheader.add("policy_owner");
    vheader.add("policy_name");
    vheader.add("enabled");
    vheader.add("sel");
    vheader.add("ins");
    vheader.add("upd");
    vheader.add("del");
    dao.loadData(vdata);
    jTable1.setModel(new DefaultTableModel(vdata,vheader));
}

```

- Code load dữ liệu từ database lên combobox:

```
// Khởi tạo combobox ten user -----
private void initComboBoxobject_schema(){
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection(BienCucBo.ketnoi,
            BienCucBo.userN, BienCucBo.passW);
        String sql = "select * from sys.xemtenuserhienco";
        PreparedStatement pstmt = con.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery();
        cbbtenuser.removeAllItems();
        while(rs.next()){
            cbbtenuser.addItem(rs.getString("username"));
        }
        rs.close();
        pstmt.close();
        con.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

- Code load dữ liệu từ database lên combobox:

```
// Khởi tạo combobox Action -----
private void initComboBoxobject_name(){
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection(BienCucBo.ketnoi,
            BienCucBo.userN, BienCucBo.passW);
        String sql = "SELECT table_name FROM sys.dba_tables "
            + "where owner='"+cbbtenuser.getSelectedItem().toString()+"'";
        PreparedStatement pstmt = con.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery();
        cbbtentable.removeAllItems();
        while(rs.next()){
            cbbtentable.addItem(rs.getString("table_name"));
        }
        rs.close();
        pstmt.close();
        con.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

- Code button Thêm mới 1 Audit Policy:

```

private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    if(ktrong()==false)
    {
        return;
    }
    if(BienCucBo.ktthem())
    {
        if(kttrung()==true)//trùng mã sách
        {
            JOptionPane.showMessageDialog(null, "Tên policy đã tồn tại!");
            return;
        }
        else//mã sách không trùng
        {
            pojo_Admin_Audit x = new pojo_Admin_Audit(cbbtenuser.getSelectedItem().toString(),
            cbbtentable.getSelectedItem().toString(),"",
            jTFTenpolicy.getText(),"",jTextGiamsat1.getText(),"","","");
            dao.Thempolicy(x);
            dao.loadData(vdata);
            jTable1.updateUI();
            if(kttrung()==true)//trùng là đã thêm vào thành công
            {
                JOptionPane.showMessageDialog(null, "Thành công");
                return;
            }
            else//không trùng không có trong table
            {
                JOptionPane.showMessageDialog(null, "Thất bại");
                return;
            }
        }
    }
}

```

- Code button Xóa 1 Audit Policy:

```

private void btnXoaActionPerformed(java.awt.event.ActionEvent evt) {
    if(ktrong()==false)
    {
        return;
    }
    if (BienCucBo.ktxoa())//
    {
        if(kttrung()==false)//không tồn tại khóa chính
        {
            JOptionPane.showMessageDialog(null, "Username Không tồn tại");
            return;
        }
        else//không trùng
        {
            pojo_Admin_Audit x = new pojo_Admin_Audit();
            pojo_Admin_Audit y = new pojo_Admin_Audit();
            pojo_Admin_Audit z = new pojo_Admin_Audit();
            x.setObject_schema(cbbtenuser.getSelectedItem().toString());
            y.setObject_name(cbbtentable.getSelectedItem().toString());
            z.setPolicy_name(jTFTenpolicy.getText());
            dao.Xoapolicy(x,y,z);
            dao.loadData(vdata);
            jTable1.updateUI();
            if(kttrung()==false)//Không tồn tại
            {
                JOptionPane.showMessageDialog(null, "Thành công");
                return;
            }
            else
            {
                JOptionPane.showMessageDialog(null, "Thất bại");
                return;
            }
        }
    }
}

```

- Code button Bật 1 Audit Policy:

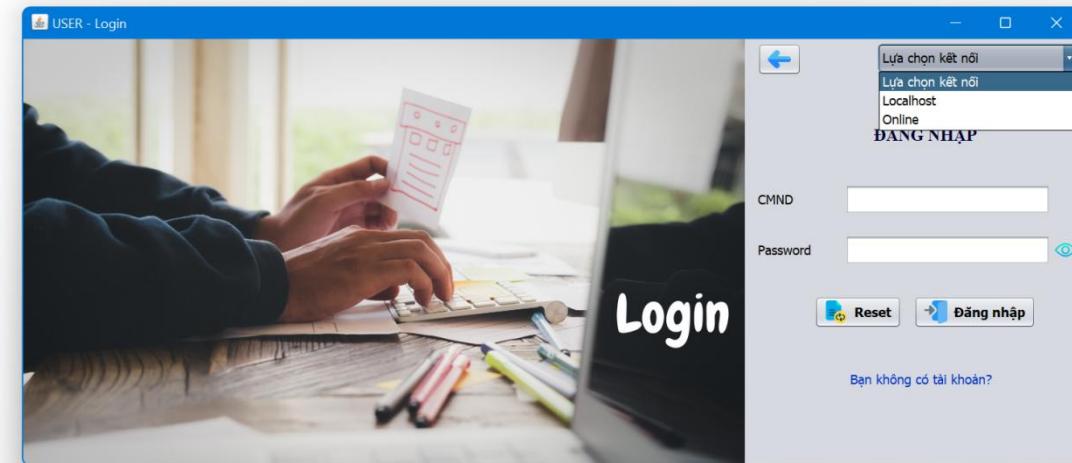
```
private void btnBatActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if(ktrong()==false)  
{  
        return;  
    }  
    if(JOptionPane.showConfirmDialog(null, "Bạn có muốn Bật Policy này không?") == JOptionPane.OK_OPTION)  
{  
        if(kttrung()==false)//không tồn tại khóa chính  
        {  
            JOptionPane.showMessageDialog(null, "Tên Policy Không tồn tại");  
            return;  
        }  
        else  
        {  
            if(ktlockuser()==true)  
            {  
                JOptionPane.showMessageDialog(null, "Policy Đang ở trạng thái Bật");  
                return;  
            }  
            else  
            {  
                pojo_Admin_Audit x = new pojo_Admin_Audit();  
                pojo_Admin_Audit y = new pojo_Admin_Audit();  
  
                else  
                {  
                    pojo_Admin_Audit x = new pojo_Admin_Audit();  
                    pojo_Admin_Audit y = new pojo_Admin_Audit();  
                    pojo_Admin_Audit z = new pojo_Admin_Audit();  
                    x.setObject_schema(cbbtenuser.getSelectedItem().toString());  
                    y.setObject_name(cbbtentable.getSelectedItem().toString());  
                    z.setPolicy_name(jTFtenpolicy.getText());  
                    dao.Batpolicy(x,y,z);  
                    dao.loadData(vdata);  
                    jTable1.updateUI();  
                    if(ktlockuser()==true)  
                    {  
                        JOptionPane.showMessageDialog(null, "Thành công");  
                        return;  
                    }  
                    else  
                    {  
                        JOptionPane.showMessageDialog(null, "Thất bại");  
                        return;  
                    }  
                }  
            }  
        }  
    }  
}
```

- Code button Tắt 1 Audit Policy:

```
private void btnTatActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if(ktrong()==false)
    {
        return;
    }
    if(JOptionPane.showConfirmDialog(null, "Bạn có muốn Bật Policy này không?") == JOptionPane.OK_OPTION)
    {
        if(kttrung()==false)//không tồn tại khóa chính
        {
            JOptionPane.showMessageDialog(null, "Tên Policy Không tồn tại");
            return;
        }
        else
        {
            if(ktlockuser()==false)
            {
                JOptionPane.showMessageDialog(null, "Policy Đang ở trạng thái Tắt");
                return;
            }
            else
            {
                pojo_Admin_Audit x = new pojo_Admin_Audit();
                pojo_Admin_Audit y = new pojo_Admin_Audit();
                pojo_Admin_Audit z = new pojo_Admin_Audit();

                x.setObject_schema(cbbtenuser.getSelectedItem().toString());
                y.setObject_name(cbbtentable.getSelectedItem().toString());
                z.setPolicy_name(jTFtenpolicy.getText());
                dao.Tatpolicy(x,y,z);
                dao.loadData(vdata);
                jTable1.updateUI();
                if(ktlockuser()==false)
                {
                    JOptionPane.showMessageDialog(null, "Thành công");
                    return;
                }
                else
                {
                    JOptionPane.showMessageDialog(null, "Thất bại");
                    return;
                }
            }
        }
    }
}
```

- Giao diện đăng nhập giành cho người dùng với 2 lựa chọn là kết nối Localhost hoặc Online.



- Code button đăng nhập:

```
// Login button -----
private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {
    Connection con = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

    BienCucBo.tenDN = "";
    BienCucBo.userN = "";
    BienCucBo.passW = "";

    UN = txtCMND.getText(); // Gán UN là CMND

    String msg = "" + txtCMND.getText();
    msg += "\n";

    // Nếu username là 'sys'
    if(txtCMND.getText().equals("sys"))
    {
        BienCucBo.userN =txtCMND.getText() + " as sysdba";
        BienCucBo.passW =txtPassword.getText();
        BienCucBo.tenDN = "Xin chào, "+txtCMND.getText();
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            con = DriverManager.getConnection(BienCucBo.ketnoi,
                BienCucBo.userN, BienCucBo.passW);
            String sql = "select * from sys.ACOUNT_USER where CMND='"+txtCMND.getText()+"' "
        }
    }
}
```

```

String sql = "select * from sys.ACCTOUNT_USER where CMND='"+txtCMND.getText()+"' "
+ "and pass='"+txtPassword.getText()+"'";
pst = con.prepareStatement(sql);
rs = pst.executeQuery();

if(rs.next()){
    JOptionPane.showMessageDialog(null, "Welcome, " +msg+
        "Bạn đã đăng nhập thành công!");
    User_Home ah = new User_Home();
    ah.setVisible(true);
    this.dispose();
}
else{
    JOptionPane.showMessageDialog(null, "Sai mật khẩu hoặc CMND!");
    BienCucBo.tenDN = "";
    BienCucBo.userN = "";
    BienCucBo.passW = "";
    txtCMND.requestFocus();
}
} catch (Exception e) {
    BienCucBo.tenDN = "";
    BienCucBo.userN = "";
    BienCucBo.passW = "";
    JOptionPane.showMessageDialog(null, e);
}
}

else // Không là user 'sys'
{
    BienCucBo.userN =txtCMND.getText();
    BienCucBo.passW =txtPassword.getText();
    BienCucBo.tenDN = "Xin chào, "+txtCMND.getText();
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        con = DriverManager.getConnection(BienCucBo.ketnoi,
            BienCucBo.userN, BienCucBo.passW);
        String sql = "select * from sys.ACCTOUNT_USER where "
            + "CMND='"+txtCMND.getText()+"' and pass='"+txtPassword.getText()+"'";
        pst = con.prepareStatement(sql);
        rs = pst.executeQuery();

        if(rs.next()){
            JOptionPane.showMessageDialog(null, "Welcome, " +msg+
                "Bạn đã đăng nhập thành công!");
            User_Home ah = new User_Home();
            ah.setVisible(true);
            this.dispose();
        }
        else{
            JOptionPane.showMessageDialog(null, "Sai mật khẩu hoặc CMND!");
            BienCucBo.tenDN = "";
            BienCucBo.userN = "";
            BienCucBo.passW = "";
            txtCMND.requestFocus();
        }
    } catch (Exception e) {
        BienCucBo.tenDN = "";
        BienCucBo.userN = "";
        BienCucBo.passW = "";
        JOptionPane.showMessageDialog(null, "Sai CMND hoặc mật khẩu!");
        e.printStackTrace();
    }
}
}

```

- Giao diện đăng ký tài khoản giành cho người dùng:



- Code button đăng ký tài khoản giành cho người dùng:

```
// Registration -
private void btnDangKyActionPerformed(java.awt.event.ActionEvent evt) {
    Connection con = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

    String name = txtCMND.getText();

    String msg = "" + name;
    msg += "\n";

    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ThuVien", "sys as sysdba", "sys");

        // Thêm tài khoản -----
        String sql = "insert into ACCOUNT_USER values ('"+txtCMND.getText()+"', '"+txtMatKhau.getText()+"')";

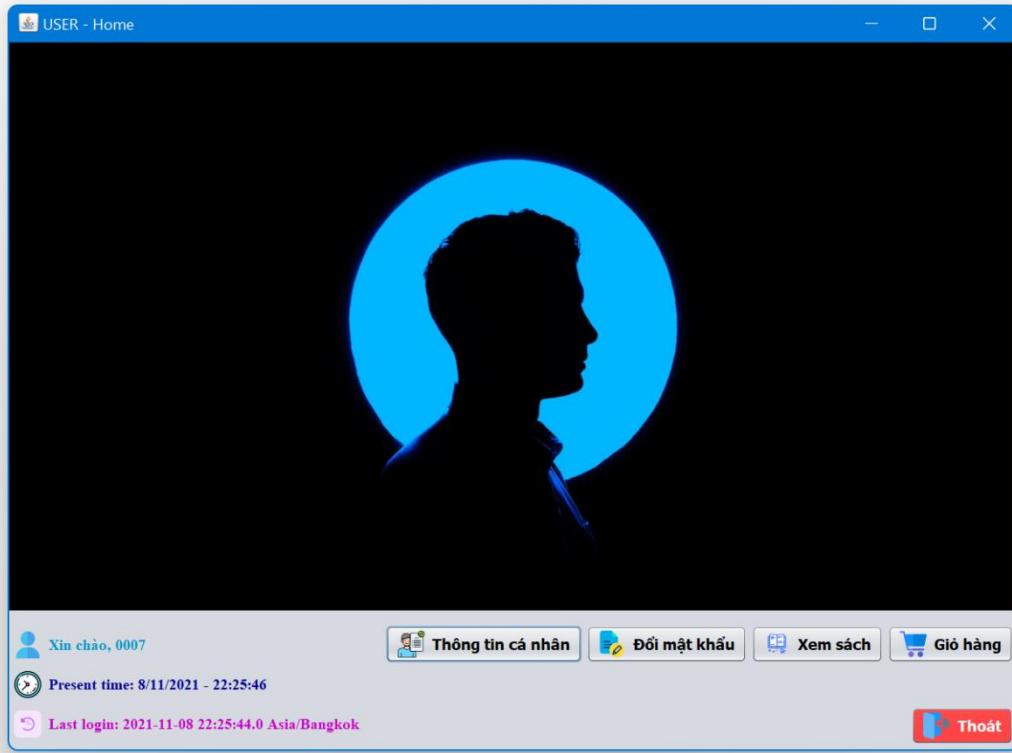
        // Thêm thông tin Khách hàng -----
        String sql2 = "insert into KHACH values ('"+txtCMND.getText()+"', "
            + "N'" +txtHoTen.getText()+"', '"+txtTuoi.getText()+"', "
            + "'"+grpRadioButton.getSelection().getActionCommand()+"', "
            + "N'" +cbbDiaChi.getSelectedItem().toString()+"', "
            + "'"+txtSDT.getText()+"')";

        Statement stmt = con.createStatement();
        int x = stmt.executeUpdate(sql);
        int y = stmt.executeUpdate(sql2);

        if(x == 0){
            JOptionPane.showMessageDialog(btnDangKy,
                "CMND đã tồn tại! Vui lòng nhập UserName khác!");
        }
        else{
            JOptionPane.showMessageDialog(btnDangKy, "Welcome, " +msg+
                "Bạn đã tạo tài khoản thành công!");
            User_Login ul = new User_Login(); // Chuyển hướng sang trang Đăng nhập
            ul.setVisible(true);
            this.dispose(); // Khi chuyển trang sẽ ẩn trang trước đó đi.
        }

        con.close();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, "Lỗi nhập thông tin!");
        e.printStackTrace();
    }
}
```

- Giao diện sau khi login thành công với tài khoản người dùng:



- Giao diện hiển thị thông tin cá nhân của người dùng đang đăng nhập:



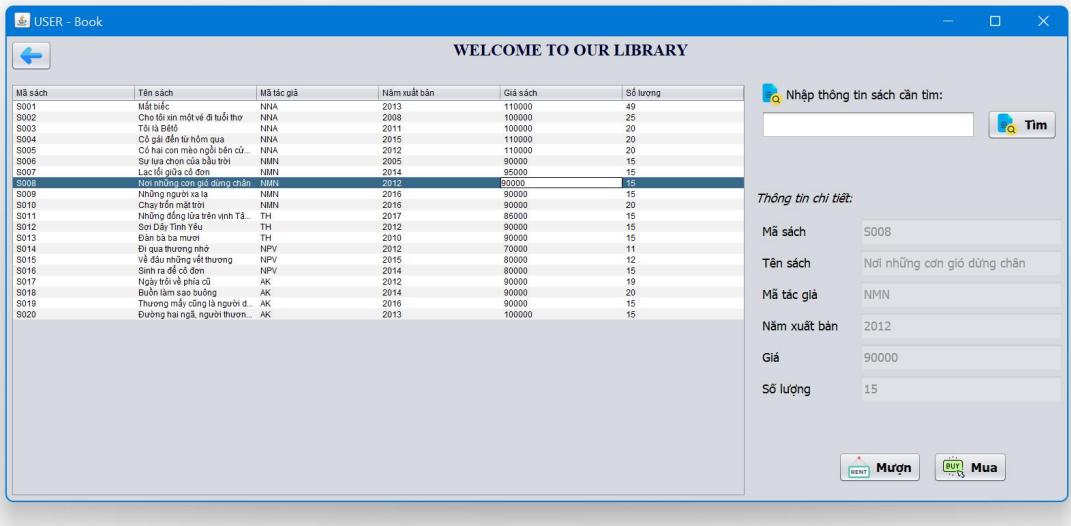
- Code button Sửa thông tin cá nhân của 1 khách hàng:

```
private void btnSuaActionPerformed(java.awt.event.ActionEvent evt) {  
    if (JOptionPane.showConfirmDialog(null, "Bạn có muốn sửa không?")  
        == JOptionPane.OK_OPTION) {  
        pojo_User_Info x = new pojo_User_Info(txtCMND.getText(),  
                                              txtHoTen.getText(), txtTuoi.getText(), txtGioiTinh.getText(),  
                                              cbbDiaChi.getSelectedItem().toString(), txtSDT.getText());  
        daoUser.SuaKhach(x);  
        daoUser.LayDSKH_vector(vData);  
        jTable1.updateUI();  
    }  
}
```

- Giao diện đổi mật khẩu của khách hàng:



- Giao diện xem sách trong thư viện của khách hàng:



- Code form xem sách của người dùng:

```

public User_Book() {
    initComponents();

    this.setLocationRelativeTo(null); // Chỉnh form nằm giữa màn hình

    dao=new DAO_User_Book();
    vheader.add("Mã sách");
    vheader.add("Tên sách");
    vheader.add("Mã tác giả");
    vheader.add("Năm xuất bản");
    vheader.add("Giá sách");
    vheader.add("Số lượng");
    dao.LayDSUser_Book_vector(vdata);
    jTableuserbook.setModel(new DefaultTableModel(vdata,vheader));
}

```

III. KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN:

1. Kết luận:

- Ứng dụng giúp người dùng dễ dàng tìm kiếm và chỉnh sửa thông tin theo ý muốn của mình.
- Hạn chế sai sót so với việc thực hiện theo cách thủ công.
- Giao diện đơn giản, dễ dàng thao tác.
- Ứng dụng vẫn còn nhiều thiếu sót và trong tương lai chúng em sẽ khắc phục, sửa đổi và hoàn thiện hơn.

2. Định hướng phát triển:

- Phát triển ứng dụng chạy trên các hệ điều hành phổ biến.
- Hoàn thiện hơn về phần cơ sở dữ liệu.
- Cải thiện giao diện đồ họa trở nên hài hòa, bắt mắt hơn và giúp người dùng dễ dàng thao tác hơn.
- Cải tiến, sáng tạo thêm nhiều chức năng mới.
- Thêm được hình ảnh, âm thanh vào ứng dụng.

IV. TÀI LIỆU THAM KHẢO:

[1]. <https://www.youtube.com/watch?v=PDw7W1TTi4c&t=961s>

[2]. <https://www.youtube.com/watch?v=WFbrRtsuQ8o>

V. BIÊN BẢN:

MSSV	Họ tên	Phân công	Tiến độ hoàn thành
2001190424	Phạm Gia Bảo	<ul style="list-style-type: none"> - Phân công nhiệm vụ cho từng thành viên. - Tạo procedure, function, view bên Oracle. - Gọi thực thi procedure, function, view bên Java. - Thực hiện kết nối, điều khiển từ xa. - Mô tả các bước kết nối, điều khiển từ xa vào file word. - Load dữ liệu từ Oracle lên table trong form Quyền hệ thống và thực hiện chức năng Tìm kiếm, Thêm, Thu hồi. - Load dữ liệu từ Oracle lên table trong form Quyền đối tượng và thực hiện các chức năng Tìm kiếm, Thêm, Xóa, Sửa. - Load dữ liệu từ Oracle lên table, combobox trong form Nhóm quyền và thực hiện các chức năng Tìm kiếm, Tạo nhóm quyền, Xóa nhóm quyền, Thêm quyền vào nhóm quyền, Gán quyền cho user, Thu hồi quyền của user. - Load dữ liệu từ Oracle lên table, combobox trong form Thông tin user và thực hiện các chức năng Thêm, Xóa, Sửa, Khóa user, Mở khóa user. - Load dữ liệu từ Oracle lên table, combobox trong 	100%

		<p>form Audit và thực hiện các chức năng Thêm, Xóa, Xem giám sát, Bật audit, Tắt audit.</p> <ul style="list-style-type: none"> - Viết code cho button Đăng nhập của form Login vào tài khoản user. - Viết code cho button Đăng nhập của form Login vào tài khoản admin. 	
2001190464	Phạm Gia Đạt	<ul style="list-style-type: none"> - Tạo bảng. - Nhập liệu nội dung cho từng bảng. - Tạo các ràng buộc. - Tạo kết nối từ Oracle đến Java. - Hiển thị thông tin của các bảng: Phiếu mượn, Nhân viên, Thẻ, Khách hàng, Sách, Thống kê và Tài khoản lên table trong form Java. - Hiển thị lần đăng nhập cuối cùng của 1 user lên label form Java. - Load dữ liệu từ Oracle lên table, combobox trong form Thông tin Khách hàng và thực hiện chức năng Sửa thông tin khách hàng. - Load dữ liệu từ Oracle lên table, combobox trong form Sách và thực hiện chức năng Tìm kiếm. 	100%
2001190421	Nguyễn Lê Gia Bảo	<ul style="list-style-type: none"> - Tạo lược đồ quan hệ Diagram. - Tạo các nhóm quyền, quyền và profile. - Thiết kế giao diện Java. - Thực hiện kết nối, điều 	100%

		<p>khiển từ xa.</p> <ul style="list-style-type: none">- Load dữ liệu từ Oracle lên table trong form Process và thực hiện chức năng Tìm kiếm.- Load dữ liệu từ Oracle lên table trong form Instance và thực hiện chức năng Tìm kiếm.- Load dữ liệu từ Oracle lên table trong form Database và thực hiện chức năng Tìm kiếm.- Load dữ liệu từ Oracle lên table trong form Datafile và thực hiện chức năng Tìm kiếm.- Load dữ liệu từ Oracle lên table trong form Session và thực hiện chức năng Kill session.- Load dữ liệu từ Oracle lên table trong form Parameter file và thực hiện chức năng Tìm kiếm.- Load dữ liệu từ Oracle lên table trong form Profile và thực hiện các chức năng Tìm kiếm, Thêm, Xóa, Sửa.- Soạn file word.	
--	--	--	--