**TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC**

**KHOA CÔNG NGHỆ THÔNG TIN**

Icon

Description automatically generated

**LẬP TRÌNH JAVA**

**ĐỒ ÁN: QUẢN LÝ THƯ VIỆN**

**Nhóm 2**

1. **Huỳnh Thị Ngọc Thư**
2. **Nguyễn Thái Sơn**
3. **Nguyễn Thành Đức Trí**
4. **Bùi Hồng Ngọc**
5. **Bùi Duy Thanh**

**Giảng viên: Trương Bá Thái**

**Tp. Hồ Chí Minh, 20/12/2021**

**MENU**

[PHÂN CÔNG 4](#_Toc90857636)

[CHƯƠNG 1: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 6](#_Toc90857637)

[1.1 Phân tích 6](#_Toc90857638)

[1.2 Thiết kế hệ thống 6](#_Toc90857639)

[1.2.1 Chọn loại tài khoản 6](#_Toc90857640)

[1.2.2 Đăng nhập 7](#_Toc90857641)

[1.2.3 Giao diện trang chủ 7](#_Toc90857642)

[1.2.4 Nhập sách 8](#_Toc90857643)

[1.2.5 Nhân viên 9](#_Toc90857644)

[1.2.6 Danh mục 10](#_Toc90857645)

[1.2.7 Ngôn ngữ 11](#_Toc90857646)

[1.2.8 Nhà cung cấp 12](#_Toc90857647)

[1.2.9 Tài khoản 13](#_Toc90857648)

[1.2.10 Loại độc giả 14](#_Toc90857649)

[1.2.11 Thêm thẻ thư viện 15](#_Toc90857650)

[1.2.12 Gia hạn thẻ thư viện 16](#_Toc90857651)

[1.2.13 Mượn sách 16](#_Toc90857652)

[1.2.14 Trả sách 17](#_Toc90857653)

[1.2.15 Tìm sách 18](#_Toc90857654)

[CHƯƠNG 2: CÀI ĐẶT CHƯƠNG TRÌNH 19](#_Toc90857655)

[1.1 Giao diện chương trình 19](#_Toc90857656)

[1.1 Chọn loại tài khoản 19](#_Toc90857657)

[1.2 Đăng nhập 19](#_Toc90857658)

[1.3 Trang chủ độc giả 20](#_Toc90857659)

[1.4 Trang chủ hệ thống 20](#_Toc90857660)

[1.5 Nhập sách 21](#_Toc90857661)

[1.6 Cập nhập sách 21](#_Toc90857662)

[1.7 Nhân viên 22](#_Toc90857663)

[1.8 Danh mục 22](#_Toc90857664)

[1.9 Ngôn ngữ 23](#_Toc90857665)

[1.10 Nhà xuất bản 23](#_Toc90857666)

[1.11 Nhà cung cấp 24](#_Toc90857667)

[1.12 Tài khoản 24](#_Toc90857668)

[1.13 Loại độc giả 25](#_Toc90857669)

[1.14 Thẻ thư viện 25](#_Toc90857670)

[1.15 Gia hạn thẻ 26](#_Toc90857671)

[1.16 Mượn sách 26](#_Toc90857672)

[1.17 Trả sách 27](#_Toc90857673)

[1.18 Tìm kiếm sách 27](#_Toc90857674)

[1.19 Thống kê 28](#_Toc90857675)

[2. Code 28](#_Toc90857676)

[2.1 Chọn loại tài khoản 28](#_Toc90857677)

[2.2 Đăng nhập 29](#_Toc90857678)

[2.3 Giao diện trang chủ 30](#_Toc90857679)

[2.4 Nhập sách 40](#_Toc90857680)

[2.5 Nhân viên 54](#_Toc90857681)

[2.6 Danh mục 68](#_Toc90857682)

[2.7 Ngôn ngữ 79](#_Toc90857683)

[2.8 Nhà cung cấp 89](#_Toc90857684)

[2.9 Tài khoản 101](#_Toc90857685)

[2.10 Độc giả 104](#_Toc90857686)

[2.11 Thêm thẻ thư viện 112](#_Toc90857687)

[2.12 Mượn sách 122](#_Toc90857688)

[2.13 Trả sách 132](#_Toc90857689)

[2.14 Tìm sách 142](#_Toc90857690)

[2.15 Cập nhập sách 144](#_Toc90857691)

[2.16 Nhà cung cấp 150](#_Toc90857692)

[2.17 Thống kê 159](#_Toc90857693)

# PHÂN CÔNG

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Họ Tên | Công việc | Hoàn thành (%) | Đóng góp |
| 1 | Huỳnh Thị Ngọc Thư  (20211TT0665) | Tuần 7:Thiết kế cơ sở dữ liệu  Tuần 8:Thiết kế mockup  Tuần 9:   * Tạo database * Form nhân viên, * Form chọn tư cách đăng nhập * Form trang chủ   Tuần 10:   * Tạo giao diện cho bảng: * Nhà cung cấp, * Form NXB * Form Ngôn ngữ   Tuần 11: Code bảng nhân viên, tài khoản, đăng nhập  Tuần 12: Chỉnh sửa cơ sở dữ liệu, bảng đăng nhập  Tuần 13: Fix code cho các bảng, sửa cơ sở dữ liệu | 100% | 20% |
| 2 | Bùi Hồng Ngọc  (20211TT1065) | Tuần 8: Tạo giao diện cơ bản  Tuần 9: Tạo giao diện cập nhật sách, thẻ thư viện  Tuần 10 - 11 : Code bảng nhà Cung cấp  Tuần 12: Code bảng Cập nhật sách , thẻ thư viện  Tuần 13: Làm file word báo cáo | 100% | 20% |
| 3 | Nguyễn Thành Đức Trí  (20211TT0245) | Tuần 9: Tạo giao diện Đăng nhập, nhập sách  Tuần 10:Code form Nhà Xuất Bản  Tuần 11->12:Code form Nhập Sách  Tuần 13: fix code cho các bảng | 100% | 20% |
| 4 | Bùi Duy Thanh  (20211TT0660) | Tuần 8:Thiết kế mockup  Tuần 9: Tạo giao diện trả sách, thống kê  Tuần 10: Form ngôn ngữ  Tuần 11->12 Viết code trả sách  Tuần 13: làm file word báo cáo, form Thống kê | 100% | 20% |
| 5 | Nguyễn Thái Sơn  (20211TT0615) | Tuần 7: làm cơ sở dữ liệu  Tuần 9:   * Tạo database, * Form tìm sách, * Form mượn sách,   Tuần 10 :Tạo giao diện và viết code cho tìm sách  Tuần 11: Viết code Tìm Sách  Tuần 12: Viết code mượn sách  Tuần 13: Update form Tìm sách, support phần trả sách | 100% | 20% |

# CHƯƠNG 1: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Phân tích

Vì số lượng sách trong thư viện ngày càng nhiều, đồng thời các thông tin lưu trữ khác cũng tăng lên nên thư viện không thể quản lý thông tin sách, thông tin người mượn sách.. bằng cách thủ công. Do đó cần xây dựng phần mềm quản lý thư viện với giao diện thân thiện với người dùng để người quản lý dễ dàng thao tác

1. Quản lý sách
2. Quản lý thẻ thư viện
3. Quản lý thời gian mượn, trả sách
4. Quản lý độc giả
5. Tìm kiếm sách
6. Thống kê

**Hướng xây dựng ứng dụng**: Sử dụng Java JFrame xây dựng GUI, đồng thời kết nối với cơ sở dữ liệu để quản lý.

## Thiết kế hệ thống

### Chọn loại tài khoản

Graphical user interface, application

Description automatically generated

* Có 2 tư cách đăng nhập: Độc giả và nhân viên thư viện.
* Nhân viên được quản lý các danh mục, thông tin thư viện, giao diện độc giả được tìm kiếm thông tin sách

### Đăng nhập

Graphical user interface

Description automatically generated with medium confidence.

* Tài khoản: Nhập Tên tài khoản
* Mật khẩu:Nhập Mât khẩu
* Nút đăng nhập: lấy dữ liệu từ ô Đăng nhập và dữ liệu từ ô mật khẩu và kiểm tra có trùng với tài khoản và mật khẩu đã có trong sever hay không
* Nút Thoát: Thoát khỏi giao diện đăng nhập

### Giao diện trang chủ

A picture containing text

Description automatically generated

* Hiển thị các chức năng mà Thủ thư có thể thực hiện, bao gồm:
  + Quản lý: Nhập sách, Nhân viên, Danh mục, Ngôn ngữ, Nhà xuất bản, Nhà cung cấp, Tài khoản, Loại độc giả.
  + Thẻ thư viện: Thêm thẻ thư viện, Gia hạn thẻ thư viện
  + Mượn-trả sách: Mượn sách, Trả sách.
  + Tìm kiếm: Tìm sách
  + Trợ giúp
* Khi nhấn Đăng xuất -> quay trở về giao diện Chọn tư cách đăng nhập
* Khi nhấn Thoát -> thoát chương trình

### Nhập sách

Graphical user interface

Description automatically generated with medium confidence

* Chức năng: nhập sách vào của thư viện
* Thành phần
  + Mã hoá đơn: ta nhập mã hoá đơn vào
  + Mã sách: ta nhập mã sách
  + Tên sách: ta nhập tên sách vào
  + Danh mục: ta chọn một danh mục có sẵn trong combobox
  + Số trang: ta nhập số trang vào
  + Giá: ta nhập giá tiền của sách vào
  + Tác giả: ta nhập tên tác giả vào
  + Tên NXB: ta chọn một NXB có sẵn trong combobox
  + Ngôn ngữ: ta chọn một ngôn ngữ có sẵn trong combobox
  + Nhân viên: ta chọn một nhân viên có sẵn trong combobox
  + Số lượng: ta nhập vào số lượng sách vào
  + Năm xuất bản: ta nhập năm xuất bản vào
  + Nhà cung cấp: ta chọn một nhà cung cấp có sẵn trong combobox

### Nhân viên

Graphical user interface

Description automatically generated

* Chức năng: thêm, chỉnh sửa, thông tin nhân viên của thư viện
* Hành động:
  + Mỗi khi thêm nhân viên thì thông tin nhân viên được thêm vào listview
  + Khi click vào 1 hàng nhân viên trong listview thì thông tin nhân viên được hiển thị lên các trường trong giao diện.
  + Nhấn thay đổi thông tin 1 nhân viên bất kỳ, nhấn nút Sửa thì thông tin mới được cập nhật vào list view
  + Khi chọn 1 hay nhiều nhân viên và nhấn nút Xóa thì thông tin nhân viên đó bị xóa khỏi list view
  + Khi nhấn nút thoát thì quay lại giao diện chính
* Thành phần:
  + Mã nhân viên: mã số của nhân viên
  + Họ: họ nhân viên
  + Tên: tên nhân viên
  + Số điện thoại: số điện thoại nhân viên
  + Địa chỉ: địa chỉ nhân viên
  + Ngày sinh
  + CCCD: căn cước công dân nhân viên
  + Nút thêm
  + Nút sửa
  + Nút xoá
  + Nút thoát

### Danh mục

Graphical user interface, table

Description automatically generated

* Chức năng: thêm, chỉnh sửa, xoá thông tin danh mục sách
* Hành động:
  + Mỗi khi thêm danh mục thì thông tin danh mục sách sẽ được thêm vào listview
  + Khi click vào 1 hàng danh mục trong listview thì thông tin danh mục được hiển thị lên các trường trong giao diện
  + Nhấn thay đổi thông tin 1 danh mục bất kỳ, nhấn nút Sửa thì thông tin mới được cập nhật vào list view
  + Khi chọn 1 danh mục và nhấn nút Xóa thì thông tin danh mục đó bị xóa khỏi list view
  + Khi nhấn nút thoát thì quay lại giao diện chính
* Thành phần:
  + Mã danh mục: mã số danh mục
  + Tên danh mục: tên của danh mục
  + Nút mới
  + Nút thêm
  + Nút sửa
  + Nút xoá
  + Nút thoát

### Ngôn ngữ

Graphical user interface, table

Description automatically generated

* Chức năng: thêm, chỉnh sửa, xoá thông tin ngôn ngữ sách
* Hành động:
  + Mỗi khi thêm ngôn ngữ thì thông tin ngôn ngữ sách sẽ được thêm vào listview
  + Khi click vào 1 hàng ngôn ngữ trong listview thì thông tin ngôn ngữ được hiển thị lên các trường trong giao diện
  + Nhấn thay đổi thông tin 1 ngôn ngữ bất kỳ, nhấn nút Sửa thì thông tin mới được cập nhật vào list view
  + Khi chọn 1 ngôn ngữ và nhấn nút Xóa thì thông tin ngôn ngữ đó bị xóa khỏi list view
  + Khi nhấn nút thoát thì quay lại giao diện chính
* Thành phần:
  + Mã ngôn ngữ: mã số ngôn ngữ
  + Tên ngôn ngữ: tên của ngôn ngữ

### Nhà cung cấp

Graphical user interface, table

Description automatically generated

* Chức năng: thêm, chỉnh sửa, xoá thông tin nhà cung cấp
* Hành động:
  + Mỗi khi thêm nhà cung cấp thông tin nhà cung cấp sách sẽ được thêm vào listview
  + Khi click vào 1 hàng nhà cung cấp trong listview thì thông tin nhà cung cấp được hiển thị lên các trường trong giao diện
  + Nhấn thay đổi thông tin 1 nhà cung cấp bất kỳ, nhấn nút Sửa thì thông tin mới được cập nhật vào list view
  + Khi chọn 1 nhà cung cấp và nhấn nút Xóa thì thông tin nhà cung cấp đó bị xóa khỏi list view
  + Khi nhấn nút thoát thì quay lại giao diện chính
* Thành phần:
  + Mã NCC: mã của nhà cung cấp
  + Tên NCC: tên của nhà cung cấp
  + Địa chỉ: địa chỉ của nhà cung cấp
  + Số điện thoại: số điện thoại của nhà cung cấp
  + Nút mới
  + Nút thêm
  + Nút sửa
  + Nút xoá
  + Nút thoát

### Tài khoản

Graphical user interface, application

Description automatically generated

* Chức năng: thêm, chỉnh sửa, xoá thông tin tài khoản của nhân viên
* Hành động:
  + Chọn loại tài khoản cần thêm (ThuKho hoặc NhanVien)
  + Mỗi khi thêm tài khoản thông tin tài khoản sẽ được thêm vào listview
  + Khi click vào 1 hàng tài khoản trong listview thì thông tin tài khoản được hiển thị lên các trường trong giao diện
  + Nhấn thay đổi thông tin tài khoản bất kỳ, nhấn nút Sửa thì thông tin mới được cập nhật vào list view
  + Khi chọn 1 tài khoản và nhấn nút Xóa thì thông tin tài khoản đó bị xóa khỏi list view
  + Khi nhấn nút thoát thì quay lại giao diện chính
* Thành phần:
  + Mã tài khoản: nhập mã tài khoản
  + Tên đăng nhập: nhập tên đăng nhập cho tài khoản
  + Mật khẩu: nhập mật khẩu đăng nhập cho tài khoản
  + Loại tài khoản: chọn loại tài khoản trong combobox cho sẵn
  + Mã nhân viên: mã của nhân viên
  + Nút mới
  + Nút thêm
  + Nút sửa
  + Nút xoá
  + Nút thoát

### Loại độc giả

Graphical user interface, table

Description automatically generated

* Chức năng: Thêm loại độc giả
* Thành phần
  + Tên loại độc giả: nhập tên loại độc giả
  + Phí: nhập số phí theo loại độc giả
  + Số lượng tối đa: giới hạn số lượng sách mượn theo loại độc giả
* Hành động: nhập tên , phí , và số lượng sách giới hạn của loại độc giả mới,sau đó loại độc giả mới vào table . sau đó chọn vào loại độc giả trong table để xóa hoặc sửa

### Thêm thẻ thư viện

Graphical user interface

Description automatically generated

* Chức năng: Thêm thẻ thư viện
* Thành phần
  + Mã thẻ thư viện: thẻ thư viện tự động tăng
  + CCCD: ta nhập căn cước công dân
  + Ngày sinh: ta nhập ngày sinh độc giả
  + Giới tính: chọn giới tính
  + Họ độc giả: ta nhập họ độc giả
  + Tên độc giả: ta nhập Tên độc giả
  + Địa chỉ: ta nhập địa chỉ độc giả
  + Nhân viên tạo: Chọn nhân viên tạo
  + Loại độc giả:chọn loại độc giả
* Hành động: Khi nhập đủ thông tin cần thêm , nhấn button thêm thì thẻ thư viện được tạo, thêm vào listview.

### Gia hạn thẻ thư viện

### Mượn sách

Graphical user interface

Description automatically generated

* Chức năng: Mượn sách
* Thành phần
  + Mã độc giả: ta nhập mã độc giả
  + Nhân viên: ta chọn nhân viên cho mượn
  + Mã sách: ta nhập vào mã sách muốn mượn
  + Tên độc giả: ta nhập tên độc giả
  + Ngày mượn: lấy ngày hiện tại
  + Số ngày mượn: nhập số ngày mượn
  + Mã phiếu: nhập mã phiếu mượn
  + Tình trạng sách: nhập tình trạng sách

### Trả sách

Graphical user interface

Description automatically generated

* Chức năng: lưu thông tin khi trả sách
* Thành phần
  + Mã mượn sách: ta nhập mã mượn sách
  + Ngày dự trả: ta nhập ngày dự trả
  + Ngày thực trả: ta nhập ngày thực trả
  + Mã sách: ta nhập vào mã sách đã mượn
  + Tên độc giả: ta nhập tên độc giả
  + Mã thẻ: ta nhập mã thẻ thư viện
  + Số ngày quá hạn: tự động tính dựa vào ngày dự trả và ngày thực trả
  + Tình trạng sách: nhập tình trạng sách
* Hành động : Khi chọn mã thư viện, mã sách thông tin trả sách sẽ hiển thị dưới listview kèm theo đó thông tin số ngày quá hạn cũng sẽ hiện lên, người dùng bấm nút lưu thì thông tin sẽ được lưu lại, nếu quá hạn thì thu tiền phạt

### Tìm sách

Table

Description automatically generated

* Chức năng: Tìm kiếm sách trong thư viên
* Thành phần
  + Tìm kiếm theo: chọn loại chỉ tiêu tìm
  + Nhập thông tin: nhập thông tin loại sách cần tìm
  + Tìm: Thực hiện tìm kiếm
  + Thoát: thoát khỏi chức năng tìm kiếm
  + Bảng: Hiển thị danh sách các sách
* Hành động: chọn để tìm sách theo chỉ tiêu , nhập thông tin của sách cần tìm, ấn nút tìm thì sách sẽ được hiển thị ở table

# CHƯƠNG 2: CÀI ĐẶT CHƯƠNG TRÌNH

## Giao diện chương trình

### Chọn loại tài khoản

A picture containing graphical user interface

Description automatically generated

### Đăng nhập

Graphical user interface, application

Description automatically generated

### Trang chủ độc giả

A picture containing background pattern

Description automatically generated

### Trang chủ hệ thống

A picture containing background pattern

Description automatically generated

### Nhập sách

Graphical user interface, application

Description automatically generated

### Cập nhập sách

Graphical user interface, application

Description automatically generated

### Nhân viên

Graphical user interface, table

Description automatically generated

### Danh mục

Graphical user interface, text, application, email

Description automatically generated

### Ngôn ngữ

Graphical user interface

Description automatically generated

### Nhà xuất bản

Graphical user interface

Description automatically generated

### Nhà cung cấp

Graphical user interface

Description automatically generated with medium confidence

### Tài khoản

Graphical user interface, application

Description automatically generated

### Loại độc giả

Graphical user interface

Description automatically generated

### Thẻ thư viện

Graphical user interface, application, Word

Description automatically generated

### Gia hạn thẻ

Graphical user interface, application

Description automatically generated

### Mượn sách

Graphical user interface

Description automatically generated

### Trả sách

Graphical user interface, application

Description automatically generated

### Tìm kiếm sách

Graphical user interface, timeline

Description automatically generated

### Thống kê

Table

Description automatically generated

## Code

### Chọn loại tài khoản

public frmChonTCDN() {

initComponents();

this.setLocationRelativeTo(null);

this.setTitle("Chọn tư cách đăng nhập");

}

private void loginThuThuMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

this.dispose();

frmDangNhap fr = new frmDangNhap();

fr.setVisible(true);

}

private void loginNVMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

this.dispose();

frmTrangChuDocGia frm = new frmTrangChuDocGia();

frm.setVisible(true);

}

### Đăng nhập

public frmDangNhap() {

initComponents();

this.setTitle("Đăng nhập");

this.setLocationRelativeTo(null);

}

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

String tenDN = txtTenDN.getText(); // tên đăng nhập

String mk = String.valueOf(txtMK.getPassword()); // mật khẩu

if (tenDN.length() == 0 || mk.length() == 0) {

JOptionPane.showMessageDialog(null, "Tên đăng nhập/ Mật khẩu không được để trống");

} else {

try {

ResultSet rs = dal.checkDN(tenDN, mk);

rs.last();

int row = rs.getRow();

rs.beforeFirst();

if (row > 0) {

while (rs.next()) {

if (mk.compareTo(rs.getObject(3).toString().trim()) != 0) {

JOptionPane.showMessageDialog(null, "Mật khẩu không hợp lệ");

} else {

this.dispose();

frmTrangChu frmTC = new frmTrangChu();

frmTC.setVisible(true);

}

}

} else {

JOptionPane.showMessageDialog(null, "Tên đăng nhập không hợp lệ");

}

} catch (Exception ex) {

Logger.getLogger(frmDangNhap.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

### Giao diện trang chủ

private void mnuQL\_DanhMucActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if (frmDanhMuc == null || frmDanhMuc.isClosed()) {

try {

frmDanhMuc = new frmDanhMuc();

dpMain.add(frmDanhMuc);

frmDanhMuc.setLocation(this.getWidth() / 2 - frmDanhMuc.getWidth() / 2, (this.getHeight() - 20) / 2 - frmDanhMuc.getHeight() / 2 - 20);

frmDanhMuc.setVisible(true);

} catch (Exception ex) {

Logger.getLogger(frmTrangChu.class.getName()).log(Level.SEVERE, null, ex);

}

} else {

frmDanhMuc.setLocation(this.getWidth() / 2 - frmDanhMuc.getWidth() / 2, (this.getHeight() - 20) / 2 - frmDanhMuc.getHeight() / 2 - 20);

frmDanhMuc.setVisible(true);

frmDanhMuc.toFront();

}

}

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

this.dispose();

frmChonTCDN frm = new frmChonTCDN();

frm.setVisible(true);

}

private void mnuQL\_NhapSachActionPerformed(java.awt.event.ActionEvent evt) {

if (frmNhapSach == null || frmNhapSach.isClosed()) {

try {

frmNhapSach = new frmNhapSach();

} catch (Exception ex) {

Logger.getLogger(frmTrangChu.class.getName()).log(Level.SEVERE, null, ex);

}

dpMain.add(frmNhapSach);

frmNhapSach.setLocation(this.getWidth() / 2 - frmNhapSach.getWidth() / 2, (this.getHeight() - 20) / 2 - frmNhapSach.getHeight() / 2 - 20);

frmNhapSach.setVisible(true);

} else {

frmNhapSach.setLocation(this.getWidth() / 2 - frmNhapSach.getWidth() / 2, (this.getHeight() - 20) / 2 - frmNhapSach.getHeight() / 2 - 20);

frmNhapSach.setVisible(true);

frmNhapSach.toFront();

}

}

private void mnuQL\_NgonNguActionPerformed(java.awt.event.ActionEvent evt) {

if (frmNgonNgu == null || frmNgonNgu.isClosed()) {

try {

frmNgonNgu = new frmNgonNgu();

} catch (Exception ex) {

Logger.getLogger(frmTrangChu.class.getName()).log(Level.SEVERE, null, ex);

}

dpMain.add(frmNgonNgu);

frmNgonNgu.setLocation(this.getWidth() / 2 - frmNgonNgu.getWidth() / 2, (this.getHeight() - 20) / 2 - frmNgonNgu.getHeight() / 2 - 20);

frmNgonNgu.setVisible(true);

} else {

frmNgonNgu.setLocation(this.getWidth() / 2 - frmNgonNgu.getWidth() / 2, (this.getHeight() - 20) / 2 - frmNgonNgu.getHeight() / 2 - 20);

frmNgonNgu.setVisible(true);

frmNgonNgu.toFront();

}

}

private void mnuQL\_NhanVienActionPerformed(java.awt.event.ActionEvent evt) {

if (frmNhanVien == null || frmNhanVien.isClosed()) {

try {

frmNhanVien = new frmNhanVien();

} catch (Exception ex) {

Logger.getLogger(frmTrangChu.class.getName()).log(Level.SEVERE, null, ex);

}

dpMain.add(frmNhanVien);

frmNhanVien.setLocation(this.getWidth() / 2 - frmNhanVien.getWidth() / 2, (this.getHeight() - 20) / 2 - frmNhanVien.getHeight() / 2 - 20);

frmNhanVien.setVisible(true);

} else {

frmNhanVien.setLocation(this.getWidth() / 2 - frmNhanVien.getWidth() / 2, (this.getHeight() - 20) / 2 - frmNhanVien.getHeight() / 2 - 20);

frmNhanVien.setVisible(true);

frmNhanVien.toFront();

}

}

private void mnuQL\_TheTVActionPerformed(java.awt.event.ActionEvent evt) {

if (frmTheThuVien == null || frmTheThuVien.isClosed()) {

try {

frmTheThuVien = new frmTheThuVien();

} catch (Exception ex) {

Logger.getLogger(frmTrangChu.class.getName()).log(Level.SEVERE, null, ex);

}

dpMain.add(frmTheThuVien);

frmTheThuVien.setLocation(this.getWidth() / 2 - frmTheThuVien.getWidth() / 2, (this.getHeight() - 20) / 2 - frmTheThuVien.getHeight() / 2 - 20);

frmTheThuVien.setVisible(true);

} else {

frmTheThuVien.setLocation(this.getWidth() / 2 - frmTheThuVien.getWidth() / 2, (this.getHeight() - 20) / 2 - frmTheThuVien.getHeight() / 2 - 20);

frmTheThuVien.setVisible(true);

frmTheThuVien.toFront();

}

}

private void mnuQL\_NXBActionPerformed(java.awt.event.ActionEvent evt) {

if (frmNXB == null || frmNXB.isClosed()) {

try {

frmNXB = new frmNXB();

} catch (Exception ex) {

Logger.getLogger(frmTrangChu.class.getName()).log(Level.SEVERE, null, ex);

}

dpMain.add(frmNXB);

frmNXB.setLocation(this.getWidth() / 2 - frmNXB.getWidth() / 2, (this.getHeight() - 20) / 2 - frmNXB.getHeight() / 2 - 20);

frmNXB.setVisible(true);

} else {

frmNXB.setLocation(this.getWidth() / 2 - frmNXB.getWidth() / 2, (this.getHeight() - 20) / 2 - frmNXB.getHeight() / 2 - 20);

frmNXB.setVisible(true);

frmNXB.toFront();

}

}

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

if (frmNCC == null || frmNCC.isClosed()) {

try {

frmNCC = new frmNCC();

} catch (Exception ex) {

Logger.getLogger(frmTrangChu.class.getName()).log(Level.SEVERE, null, ex);

}

dpMain.add(frmNCC);

frmNCC.setLocation(this.getWidth() / 2 - frmNCC.getWidth() / 2, (this.getHeight() - 20) / 2 - frmNCC.getHeight() / 2 - 20);

frmNCC.setVisible(true);

} else {

frmNCC.setLocation(this.getWidth() / 2 - frmNCC.getWidth() / 2, (this.getHeight() - 20) / 2 - frmNCC.getHeight() / 2 - 20);

frmNCC.setVisible(true);

frmNCC.toFront();

}

}

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

if (frmMuonSach == null || frmMuonSach.isClosed()) {

try {

frmMuonSach = new frmMuonSach();

} catch (Exception ex) {

Logger.getLogger(frmTrangChu.class.getName()).log(Level.SEVERE, null, ex);

}

dpMain.add(frmMuonSach);

frmMuonSach.setLocation(this.getWidth() / 2 - frmMuonSach.getWidth() / 2, (this.getHeight() - 20) / 2 - frmMuonSach.getHeight() / 2 - 20);

frmMuonSach.setVisible(true);

} else {

frmMuonSach.setLocation(this.getWidth() / 2 - frmMuonSach.getWidth() / 2, (this.getHeight() - 20) / 2 - frmMuonSach.getHeight() / 2 - 20);

frmMuonSach.setVisible(true);

frmMuonSach.toFront();

}

}

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

if (frmTraSach == null || frmTraSach.isClosed()) {

try {

frmTraSach = new frmTraSach();

} catch (Exception ex) {

Logger.getLogger(frmTrangChu.class.getName()).log(Level.SEVERE, null, ex);

}

dpMain.add(frmTraSach);

frmTraSach.setLocation(this.getWidth() / 2 - frmTraSach.getWidth() / 2, (this.getHeight() - 20) / 2 - frmTraSach.getHeight() / 2 - 20);

frmTraSach.setVisible(true);

} else {

frmTraSach.setLocation(this.getWidth() / 2 - frmTraSach.getWidth() / 2, (this.getHeight() - 20) / 2 - frmTraSach.getHeight() / 2 - 20);

frmTraSach.setVisible(true);

frmTraSach.toFront();

}

}

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

if (frmTimKiem == null || frmTimKiem.isClosed()) {

try {

frmTimKiem = new frmTimKiem();

} catch (Exception ex) {

Logger.getLogger(frmTrangChu.class.getName()).log(Level.SEVERE, null, ex);

}

dpMain.add(frmTimKiem);

frmTimKiem.setLocation(this.getWidth() / 2 - frmTimKiem.getWidth() / 2, (this.getHeight() - 20) / 2 - frmTimKiem.getHeight() / 2 - 20);

frmTimKiem.setVisible(true);

} else {

frmTimKiem.setLocation(this.getWidth() / 2 - frmTimKiem.getWidth() / 2, (this.getHeight() - 20) / 2 - frmTimKiem.getHeight() / 2 - 20);

frmTimKiem.setVisible(true);

frmTimKiem.toFront();

}

}

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

if (frmTaiKhoan == null || frmTaiKhoan.isClosed()) {

try {

frmTaiKhoan = new frmTaiKhoan();

} catch (Exception ex) {

Logger.getLogger(frmTrangChu.class.getName()).log(Level.SEVERE, null, ex);

}

dpMain.add(frmTaiKhoan);

frmTaiKhoan.setLocation(this.getWidth() / 2 - frmTaiKhoan.getWidth() / 2, (this.getHeight() - 20) / 2 - frmTaiKhoan.getHeight() / 2 - 20);

frmTaiKhoan.setVisible(true);

} else {

frmTaiKhoan.setLocation(this.getWidth() / 2 - frmTaiKhoan.getWidth() / 2, (this.getHeight() - 20) / 2 - frmTaiKhoan.getHeight() / 2 - 20);

frmTaiKhoan.setVisible(true);

frmTaiKhoan.toFront();

}

}

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

// TODO add your handling code here:

if (frmGiaHan == null || frmGiaHan.isClosed()) {

try {

frmGiaHan = new frmGiaHanTheTV();

} catch (Exception ex) {

Logger.getLogger(frmTrangChu.class.getName()).log(Level.SEVERE, null, ex);

}

dpMain.add(frmGiaHan);

frmGiaHan.setLocation(this.getWidth() / 2 - frmGiaHan.getWidth() / 2, (this.getHeight() - 20) / 2 - frmGiaHan.getHeight() / 2 - 20);

frmGiaHan.setVisible(true);

} else {

frmGiaHan.setLocation(this.getWidth() / 2 - frmGiaHan.getWidth() / 2, (this.getHeight() - 20) / 2 - frmGiaHan.getHeight() / 2 - 20);

frmGiaHan.setVisible(true);

frmGiaHan.toFront();

}

}

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

// TODO add your handling code here:

if (frmLoaiDocGia == null || frmLoaiDocGia.isClosed()) {

try {

frmLoaiDocGia = new frmLoaiDocGia();

} catch (Exception ex) {

Logger.getLogger(frmTrangChu.class.getName()).log(Level.SEVERE, null, ex);

}

dpMain.add(frmLoaiDocGia);

frmLoaiDocGia.setLocation(this.getWidth() / 2 - frmLoaiDocGia.getWidth() / 2, (this.getHeight() - 20) / 2 - frmLoaiDocGia.getHeight() / 2 - 20);

frmLoaiDocGia.setVisible(true);

} else {

frmLoaiDocGia.setLocation(this.getWidth() / 2 - frmLoaiDocGia.getWidth() / 2, (this.getHeight() - 20) / 2 - frmLoaiDocGia.getHeight() / 2 - 20);

frmLoaiDocGia.setVisible(true);

frmLoaiDocGia.toFront();

}

}

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

// TODO add your handling code here:

if (frmCapNhatSach == null || frmCapNhatSach.isClosed()) {

try {

frmCapNhatSach = new frmCapNhatSach();

} catch (Exception ex) {

Logger.getLogger(frmTrangChu.class.getName()).log(Level.SEVERE, null, ex);

}

dpMain.add(frmCapNhatSach);

frmCapNhatSach.setLocation(this.getWidth() / 2 - frmCapNhatSach.getWidth() / 2, (this.getHeight() - 20) / 2 - frmCapNhatSach.getHeight() / 2 - 20);

frmCapNhatSach.setVisible(true);

} else {

frmCapNhatSach.setLocation(this.getWidth() / 2 - frmCapNhatSach.getWidth() / 2, (this.getHeight() - 20) / 2 - frmCapNhatSach.getHeight() / 2 - 20);

frmCapNhatSach.setVisible(true);

frmCapNhatSach.toFront();

}

}

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

if (frmThongKe == null || frmThongKe.isClosed()) {

try {

frmThongKe = new frmThongKe();

} catch (Exception ex) {

Logger.getLogger(frmTrangChu.class.getName()).log(Level.SEVERE, null, ex);

}

dpMain.add(frmThongKe);

frmThongKe.setLocation(this.getWidth() / 2 - frmThongKe.getWidth() / 2, (this.getHeight() - 20) / 2 - frmThongKe.getHeight() / 2 - 20);

frmThongKe.setVisible(true);

} else {

frmThongKe.setLocation(this.getWidth() / 2 - frmThongKe.getWidth() / 2, (this.getHeight() - 20) / 2 - frmThongKe.getHeight() / 2 - 20);

frmThongKe.setVisible(true);

frmThongKe.toFront();

}

}

### Nhập sách

#### BLL

public class BLL\_Sach {

private DAL\_Sach dal;

public BLL\_Sach() throws Exception {

dal = new DAL\_Sach();

}

public ResultSet layDS() throws SQLException, Exception {

return dal.layDS();

}

public Boolean themSach(ET\_Sach et) throws SQLException, Exception {

return dal.themSach(et);

}

public Boolean suaSach(ET\_Sach et) throws SQLException, Exception {

return dal.suaSach(et);

}

public Boolean xoaSach(String et) throws SQLException, Exception {

return dal.xoaSach(et);

}

public boolean checkTonTai(String ma) throws SQLException{

return dal.checkTonTai(ma);

}

public int laySLSach(String ma) throws SQLException, Exception{

return Integer.parseInt(dal.laySLSach(ma));

}

public int laySLSachThucTe(String ma) throws SQLException, Exception{

return Integer.parseInt(dal.laySLSachThucTe(ma));

}

public String layTen(String ma) throws Exception{

return dal.layTen(ma);

}

public String layGia(String ma) throws Exception{

return dal.layGia(ma);

}

}

#### DAL

public class DAL\_Sach {

public DAL\_Sach() throws Exception {

}

public ResultSet layDS() throws SQLException, Exception {

Connection con = null;

ResultSet rs = null;

try {

con = DatabaseUtil.getConnection();

Statement st = con.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "select \* from books";

rs = st.executeQuery(sql);

} catch (SQLException ex) {

}

return rs;

}

public ResultSet laySach(String ten) throws SQLException, Exception {

Connection con = null;

String sql = "select \* from books where title=?";

con = DatabaseUtil.getConnection();

PreparedStatement st = con.prepareStatement(sql);

st.setString(1, ten);

return st.executeQuery();

}

public boolean themSach(ET\_Sach et) throws SQLException, Exception {

Connection con = null;

con = DatabaseUtil.getConnection();

Statement st = con.createStatement();

String sql = "insert into books(id\_book,title,quantity,price,release\_year,total\_page,id\_author,id\_publisher,id\_category,id\_language,reality\_quantity)"

+ "values('" + et.getMaSach() + "','" + et.getTenSach() + "','" + et.getSoLuong() + "','" + et.getGia() + "','" + et.getNamXB() + "','" + et.getSoTrang() + "','" + et.getMaTD() + "','" + et.getMaNXB() + "','" + et.getTheLoai() + "','" + et.getNgonNgu() + "',"+et.getsLuongThuc()+")";

if (st.executeUpdate(sql) > 0) {

return true;

}

return false;

}

public boolean suaSach(ET\_Sach et) throws SQLException, Exception {

Connection con = null;

con = DatabaseUtil.getConnection();

Statement st = con.createStatement();

String sql = "UPDATE books"

+ " SET title ='" + et.getTenSach() + "', quantity ='" + et.getSoLuong() + "', price ='" + et.getGia() + "', release\_year ='" + et.getNamXB() + "', total\_page ='" + et.getSoTrang() + "', id\_author ='" + et.getMaTD() + "', id\_publisher ='" + et.getMaNXB() + "', id\_category ='" + et.getTheLoai() + "', id\_language ='" + et.getNgonNgu() + "', reality\_quantity="+et.getsLuongThuc()

+ " WHERE id\_book='" + et.getMaSach() + "'";

if (st.executeUpdate(sql) > 0) {

return true;

}

return false;

}

public boolean xoaSach(String et) throws SQLException, Exception {

Connection con = DatabaseUtil.getConnection();

Statement st = con.createStatement();

String sql = "DELETE "

+ "FROM books "

+ " WHERE id\_book='" + et + "'";

if (st.executeUpdate(sql) > 0) {

return true;

}

return false;

}

public Boolean checkTonTai(String ma) throws SQLException {

Connection conn = null;;

ResultSet rs = null;

int row = 0;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "Select \* from books where id\_book = '" + ma + "'";

rs = st.executeQuery(sql);

rs.last();

row = rs.getRow();

rs.beforeFirst();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

if(row >0){

return true;

}

else{

return false;

}

}

public String laySLSach(String maSach) throws Exception {

Connection conn = null;

conn = DatabaseUtil.getConnection();

ResultSet rs = null;

Statement st = conn.createStatement();

String sql = "select \* from books where id\_book = '" + maSach + "'";

rs = st.executeQuery(sql);

String sl = "";

while (rs.next()) {

sl = rs.getObject(3).toString();

}

return sl;

}

public String laySLSachThucTe(String maSach) throws Exception {

Connection conn = null;

conn = DatabaseUtil.getConnection();

ResultSet rs = null;

Statement st = conn.createStatement();

String sql = "select \* from books where id\_book = '" + maSach + "'";

rs = st.executeQuery(sql);

String sl = "";

while (rs.next()) {

sl = rs.getObject(12).toString();

}

return sl;

}

public String layGia(String maSach) throws Exception {

Connection conn = null;

conn = DatabaseUtil.getConnection();

ResultSet rs = null;

Statement st = conn.createStatement();

String sql = "select \* from books where id\_book = '" + maSach + "'";

rs = st.executeQuery(sql);

String sl = "";

while (rs.next()) {

sl = rs.getObject(4).toString();

}

return sl;

}

public String layTen(String maSach) throws Exception {

Connection conn = null;

conn = DatabaseUtil.getConnection();

ResultSet rs = null;

Statement st = conn.createStatement();

String sql = "select \* from books where id\_book = '" + maSach + "'";

rs = st.executeQuery(sql);

String ten = "";

while (rs.next()) {

ten = rs.getObject(2).toString();

}

return ten;

}

#### ET

public String getMaSach() {

return maSach;

}

public void setMaSach(String maSach) {

this.maSach = maSach;

}

public String getTenSach() {

return tenSach;

}

public void setTenSach(String tenSach) {

this.tenSach = tenSach;

}

public int getSoLuong() {

return soLuong;

}

public void setSoLuong(int soLuong) {

this.soLuong = soLuong;

}

public int getGia() {

return gia;

}

public void setGia(int gia) {

this.gia = gia;

}

public int getNamXB() {

return namXB;

}

public void setNamXB(int namXB) {

this.namXB = namXB;

}

public int getSoTrang() {

return soTrang;

}

public void setSoTrang(int soTrang) {

this.soTrang = soTrang;

}

public String getMaTD() {

return maTD;

}

public void setMaTD(String maTD) {

this.maTD = maTD;

}

public String getMaNXB() {

return maNXB;

}

public void setMaNXB(String maNXB) {

this.maNXB = maNXB;

}

public String getTheLoai() {

return theLoai;

}

public void setTheLoai(String theLoai) {

this.theLoai = theLoai;

}

public String getNgonNgu() {

return ngonNgu;

}

public void setNgonNgu(String ngonNgu) {

this.ngonNgu = ngonNgu;

}

public ET\_Sach(String maSach, String tenSach, int soLuong, int gia, int namXB, int soTrang, String maTD, String maNXB, String theLoai, String ngonNgu) {

this.maSach = maSach;

this.tenSach = tenSach;

this.soLuong = soLuong;

this.gia = gia;

this.namXB = namXB;

this.soTrang = soTrang;

this.maTD = maTD;

this.maNXB = maNXB;

this.theLoai = theLoai;

this.ngonNgu = ngonNgu;

}

public ET\_Sach(String maSach, String tenSach,String theLoai, String ngonNgu , String maTD, String maNXB,int namXB , String viTri) {

this.maSach = maSach;

this.tenSach = tenSach;

this.namXB = namXB;

this.maTD = maTD;

this.maNXB = maNXB;

this.theLoai = theLoai;

this.ngonNgu = ngonNgu;

this.viTri = viTri;

}

#### GUI

public String themTD(String Ten) throws Exception {

ResultSet rs = bllTD.layDS();

rs.last();

int row = rs.getRow();

if (row == 0) {

STT = 1;

} else {

rs.beforeFirst();

while (rs.next()) {

STT = Integer.parseInt(rs.getObject(1).toString().substring(2)) + 1;

}

}

String ma = String.format("%02d", STT);

String New = ("TG" + ma);

ET\_TacGia TD = new ET\_TacGia(New, Ten);

bllTD.themTD(TD);

return New;

}

public String themCTHH() throws Exception {

//Lấy mã Chi tiết hoá đơn

ResultSet rs = bllCTHD.layDS();

rs.last();

int row = rs.getRow();

if (row == 0) {

STT = 1;

} else {

rs.beforeFirst();

while (rs.next()) {

STT = Integer.parseInt(rs.getObject(1).toString().substring(2)) + 1;

}

}

String ma = String.format("%02d", STT);

String New = ("CT" + ma);

return New;

}

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

int gia=-1;

int sl=-1;

int st =-1;

int nxb=-1;

try{

gia=Integer.parseInt(txtGia.getText());

sl=Integer.parseInt(txtSoLuong.getText());

st=Integer.parseInt(txtSoTrang.getText());

nxb=Integer.parseInt(txtNamXB.getText());

}catch(NumberFormatException e){

}

String ten = txtTenSach.getText();

String mahd = txtMaHD.getText();

String maSach = txtMaSach.getText();

String maTD = txtTacGia.getText();

if (maTD.compareTo("") == 0 || ten.compareTo("") == 0 || mahd.compareTo("") == 0 || maSach.compareTo("") == 0) {

JOptionPane.showMessageDialog(null, "Vui lòng nhập đủ thông tin");

} else if(nxb<0||gia<0||st<0||sl<0){

JOptionPane.showMessageDialog(null, "Dữ liệu không hợp lệ");

}else{

try {

String tenDM = cboDM.getSelectedItem().toString();

String tenNXB = cboNXB.getSelectedItem().toString();

String tenNN = cboNN.getSelectedItem().toString();

String tenNCC = cboNCC.getSelectedItem().toString();

ET\_NhanVien NhanVien = (ET\_NhanVien) cboNV.getSelectedItem();

String maNV = NhanVien.getMaNhanVien();

String maDM = bllDM.layMa(tenDM);

String maNXB = bllNXB.layMa(tenNXB);

String maNN = bllNN.layMa(tenNN);

String maNCC = bllNCC.layMa(tenNCC);

if (bllHD.checkTonTai(mahd)) {

JOptionPane.showMessageDialog(null, "Mã hoá đơn đã tồn tại");

} else {

if (bllNV.checkTT(maNV)) {

if (bllTD.checkTonTai(maTD)) {

maTD = bllTD.layMa(maTD);

} else {

maTD = themTD(txtTacGia.getText());

}

if (bll.checkTonTai(maSach) == true) {

int tong = bll.laySLSach(txtMaSach.getText()) + Integer.parseInt(txtSoLuong.getText());

int tongTT = bll.laySLSachThucTe(txtMaSach.getText()) + Integer.parseInt(txtSoLuong.getText());

ET\_Sach et = new ET\_Sach(maSach, ten, tong, gia, nxb, st, maTD, maNXB, maDM, maNN, tongTT);

ET\_HoaDon HDon = new ET\_HoaDon(mahd, maNV, maNCC);

String maCT = themCTHH();

ET\_ChiTietHD CT = new ET\_ChiTietHD(maCT, st, mahd, maSach);

bllHD.themHoaDon(HDon);

bllCTHD.themChiTietHD(CT);

try {

if (bll.suaSach(et)) {

JOptionPane.showMessageDialog(null, "Thêm thành công");

} else {

JOptionPane.showMessageDialog(null, "Thêm không thành công");

}

} catch (SQLException ex) {

ex.printStackTrace();

}

resetMa();

hienHoaDon();

} else {

ET\_Sach et = new ET\_Sach(maSach, ten, sl, gia, nxb, st, maTD, maNXB, maDM, maNN, st);

try {

if (bll.themSach(et)) {

JOptionPane.showMessageDialog(rootPane, "Thêm thành công");

} else {

JOptionPane.showMessageDialog(rootPane, "Thêm không thành công");

}

hienHoaDon();

} catch (SQLException ex) {

ex.printStackTrace();

}

ET\_HoaDon HDon = new ET\_HoaDon(mahd, maNV, maNCC);

String maCT = themCTHH();

ET\_ChiTietHD CT = new ET\_ChiTietHD(maCT, st, mahd, maSach);

bllHD.themHoaDon(HDon);

bllCTHD.themChiTietHD(CT);

resetMa();

hienHoaDon();

}

} else {

JOptionPane.showMessageDialog(null, "Mã nhân viên không tồn tại");

}

}

} catch (Exception ex) {

ex.printStackTrace();

}

}

}

### Nhân viên

#### BLL

public class BLL\_NhanVien {

private DAL\_NhanVien dal;

public BLL\_NhanVien() throws Exception {

dal = new DAL\_NhanVien();

}

public Boolean themNV(ET\_NhanVien et) throws SQLException, Exception {

return dal.themNhanVien(et);

}

public Boolean xoaNV(ET\_NhanVien et) throws SQLException {

return dal.xoaNhanVien(et);

}

public Boolean suaNV(ET\_NhanVien et) throws SQLException {

return dal.suaNhanVien(et);

}

public ResultSet layNV() throws SQLException {

return dal.layNhanVien();

}

public Boolean checkTT(String ma) throws SQLException {

return dal.checkTonTai(ma);

}

public Boolean checkCCCD(String cccd) throws SQLException {

return dal.checkTonTaiCCCD(cccd);

}

public String layMa(String ten) throws Exception {

return dal.layMaNV(ten);

}

public String layTen(String ma) throws Exception {

return dal.layTen(ma);

}

#### DAL

public ResultSet layNhanVien() throws SQLException {

ResultSet rs = null;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

String sql = "select \* from staffs";

rs = st.executeQuery(sql);

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return rs;

}

public Boolean themNhanVien(ET\_NhanVien et) throws SQLException, Exception {

Boolean kq = false;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "insert into staffs (id\_staff, firstname, lastname, address, phone\_number,birthday, pid) "

+ "values ('" + et.getMaNhanVien() + "','" + et.getTen() + "','" + et.getHo() + "','" + et.getDiaChi() + "','" + et.getSoDT() + "','" + new java.sql.Date(et.getNgaySinh().getTime()) + "','" + et.getCCCD() + "')";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

//conn.close();

return kq;

}

public Boolean checkTonTai(String ma) throws SQLException {

Connection conn = null;;

ResultSet rs = null;

int row = 0;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "Select \* from staffs where id\_staff = '" + ma + "'";

rs = st.executeQuery(sql);

rs.last();

row = rs.getRow();

rs.beforeFirst();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return (row > 0);

}

public Boolean checkTonTaiCCCD(String cccd) throws SQLException {

Connection conn = null;;

ResultSet rs = null;

int row = 0;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "Select \* from staffs where pid = '" + cccd + "'";

rs = st.executeQuery(sql);

rs.last();

row = rs.getRow();

rs.beforeFirst();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return (row > 0);

}

public Boolean xoaNhanVien(ET\_NhanVien et) throws SQLException {

Boolean kq = false;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "delete from staffs where id\_staff = '" + et.getMaNhanVien() + "'";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return kq;

}

public Boolean suaNhanVien(ET\_NhanVien et) throws SQLException {

Boolean kq = false;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "update staffs "

+ "set firstname ='" + et.getTen() + "', "

+ "lastname = '" + et.getHo() + "', "

+ "address = '" + et.getDiaChi() + "',"

+ "phone\_number = '" + et.getSoDT() + "',"

+ "birthday = '" + new java.sql.Date(et.getNgaySinh().getTime()) + "',"

+ "pid = '" + et.getCCCD() + "'"

+ "where id\_staff ='" + et.getMaNhanVien() + "'";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return kq;

}

public String layMaNV(String tenNV) throws Exception {

Connection conn = null;

conn = DatabaseUtil.getConnection();

ResultSet rs = null;

Statement st = conn.createStatement();

String sql = "select \* from staffs where firstname = '"+tenNV+"' limit 1";

rs = st.executeQuery(sql);

String ma = "";

while (rs.next()) {

ma = rs.getObject(1).toString();

break;

}

return ma;

}

public String layTen(String maNV) throws Exception {

Connection conn = null;

conn = DatabaseUtil.getConnection();

ResultSet rs = null;

Statement st = conn.createStatement();

String sql = "select \* from staffs where id\_staff = '"+maNV+"' limit 1";

rs = st.executeQuery(sql);

String ten = "";

while (rs.next()) {

ten = rs.getObject(2).toString();

break;

}

return ten;

}

#### ET

public String getMaNhanVien() {

return maNhanVien;

}

public void setMaNhanVien(String maNhanVien) {

this.maNhanVien = maNhanVien;

}

public String getHo() {

return ho;

}

public void setHo(String ho) {

this.ho = ho;

}

public String getTen() {

return ten;

}

public void setTen(String ten) {

this.ten = ten;

}

public String getDiaChi() {

return diaChi;

}

public void setDiaChi(String diaChi) {

this.diaChi = diaChi;

}

public String getSoDT() {

return soDT;

}

public void setSoDT(String soDT) {

this.soDT = soDT;

}

public Date getNgaySinh() {

return ngaySinh;

}

public void setNgaySinh(Date ngaySinh) {

this.ngaySinh = ngaySinh;

}

public String getCCCD() {

return CCCD;

}

public void setCCCD(String CCCD) {

this.CCCD = CCCD;

}

public ET\_NhanVien(String maNhanVien, String ten, String ho, String diaChi, String soDT, Date ngaySinh, String CCCD) {

this.maNhanVien = maNhanVien;

this.ho = ho;

this.ten = ten;

this.diaChi = diaChi;

this.soDT = soDT;

this.ngaySinh = ngaySinh;

this.CCCD = CCCD;

}

public String toString() {

return getHo() + " " + getTen();

}

public ET\_NhanVien(String maNhanVien, String ten) {

this.maNhanVien = maNhanVien;

this.ten = ten;

}

#### GUI

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

String maNV = txtMaNV.getText(); // mã nhaanv viên

String ho = txtHoNV.getText(); // họ

String ten = txtTenNV.getText(); // tên

String diaChi = txtDiaChi.getText(); // địa chỉ

String soDT = txtSDT.getText(); // số điện thoại

Date ngaySinh = dtpNgSinh.getDate(); // ngày sinh

String cCCD = txtCCCD.getText(); // căn cước công dân

ET\_NhanVien et;

if (maNV.equals("") || ho.equals("") || ten.equals("") || diaChi.equals("") || soDT.equals("") || ngaySinh == null || cCCD.equals("")) {

JOptionPane.showMessageDialog(null, "Vui lòng điền đầy đủ thông tin");

} else {

try {

if (bll.checkTT(maNV)) {

JOptionPane.showMessageDialog(null, "Đã tồn tại NV");

} else {

if (bll.checkCCCD(cCCD)) {

JOptionPane.showMessageDialog(rootPane, maNV);

} else {

et = new ET\_NhanVien(maNV, ho, ten, diaChi, soDT, ngaySinh, cCCD);

if (bll.themNV(et)) {

JOptionPane.showMessageDialog(rootPane, "Thêm thành công");

} else {

JOptionPane.showMessageDialog(rootPane, "Thêm không thành công");

}

hienThiNV();

reset();

}

}

} catch (SQLException ex) {

ex.printStackTrace();

} catch (Exception ex) {

ex.printStackTrace();

}

}

}

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

int row = tblNhanVien.getSelectedRow();

if (row >= 0) {

String maNV = tblNhanVien.getValueAt(row, 0).toString();

String ho = txtHoNV.getText();

String ten = txtTenNV.getText();

String diaChi = txtDiaChi.getText();

String soDT = txtSDT.getText();

Date ngaySinh = dtpNgSinh.getDate();

String cCCD = txtCCCD.getText();

ET\_NhanVien et;

if (maNV.equals("") || ho.equals("") || ten.equals("") || diaChi.equals("") || soDT.equals("") || ngaySinh == null || cCCD.equals("")) {

JOptionPane.showMessageDialog(null, "Vui lòng điền đầy đủ thông tin");

} else {

try {

et = new ET\_NhanVien(maNV, ten, ho, diaChi, soDT, ngaySinh, cCCD);

if (bll.suaNV(et)) {

JOptionPane.showMessageDialog(rootPane, "Sửa thành công");

hienThiNV();

reset();

} else {

JOptionPane.showMessageDialog(rootPane, "Sửa không thành công");

}

} catch (SQLException ex) {

ex.printStackTrace();

}

}

} else {

JOptionPane.showMessageDialog(rootPane, "Vui lòng chọn nhân viên để sửa");

}

}

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

// TODO add your handling code here:

int row1 = tblNhanVien.getSelectedRow();

if (row1 >= 0) {

String maNV = tblNhanVien.getValueAt(row1, 0).toString();

String ho = txtHoNV.getText();

String ten = txtTenNV.getText();

String diaChi = txtDiaChi.getText();

String soDT = txtSDT.getText();

Date ngaySinh = dtpNgSinh.getDate();

String cCCD = txtCCCD.getText();

ET\_NhanVien et;

et = new ET\_NhanVien(maNV, ho, ten, diaChi, soDT, ngaySinh, cCCD);

try {

if (bll.xoaNV(et)) {

JOptionPane.showMessageDialog(rootPane, "Xoá thành công");

} else {

JOptionPane.showMessageDialog(rootPane, "Xóa không thành công");

}

hienThiNV();

reset();

} catch (SQLException ex) {

ex.printStackTrace();

}

} else {

JOptionPane.showMessageDialog(null, "Vui lòng chọn nhân viên");

}

}

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

// TODO add your handling code here:

int kq = JOptionPane.showConfirmDialog(null, "Bạn có muốn thoát không", "Thông báo", JOptionPane.YES\_NO\_OPTION);

if (kq == 0) {

this.dispose();

}

}

private void tblNhanVienMouseClicked(java.awt.event.MouseEvent evt) {

int row = tblNhanVien.getSelectedRow();

DefaultTableModel model = (DefaultTableModel) tblNhanVien.getModel();

txtMaNV.setText(model.getValueAt(row, 0).toString());

txtTenNV.setText(model.getValueAt(row, 1).toString());

txtHoNV.setText(model.getValueAt(row, 2).toString());

txtDiaChi.setText(model.getValueAt(row, 3).toString());

txtSDT.setText(model.getValueAt(row, 4).toString());

dtpNgSinh.setDate((Date) model.getValueAt(row, 5));

txtCCCD.setText(model.getValueAt(row, 6).toString());

}

private void txtSDTKeyPressed(java.awt.event.KeyEvent evt) {

char c =evt.getKeyChar();

if(Character.isLetter(c)){

txtSDT.setEditable(false);

JOptionPane.showMessageDialog(null, "Chỉ nhập số");

}else{

txtSDT.setEditable(true);

}

}

public void hienThi() {

DefaultTableModel model = (DefaultTableModel) tblNhanVien.getModel();

model.setColumnCount(0);

model.setRowCount(0);

model.addColumn("Mã NV");

model.addColumn("Tên");

model.addColumn("Họ");

model.addColumn("Địa chỉ");

model.addColumn("Số ĐT");

model.addColumn("Ngày sinh");

model.addColumn("CCCD");

tblNhanVien.setModel(model);

}

public void hienThiNV() throws SQLException {

ResultSet rs = bll.layNV();

DefaultTableModel model = (DefaultTableModel) tblNhanVien.getModel();

model.setRowCount(0);

while (rs.next()) {

Object[] col = new Object[7];

for (int i = 1; i <= 7; i++) {

col[i - 1] = rs.getObject(i);

}

model.addRow(col);

}

}

public void reset() {

txtMaNV.setText("");

txtTenNV.setText("");

txtHoNV.setText("");

txtDiaChi.setText("");

txtSDT.setText("");

dtpNgSinh.setDate(new Date());

txtCCCD.setText("");

txtMaNV.setText("");

}

public boolean checkSo(String so) {

if (so.length() == 0) {

return false;

}

char[] c = so.toCharArray();

for (int i = 0; i < c.length; i++) {

if (!Character.isDigit(c[i])) {

return false;

}

}

return true;

}

### Danh mục

#### BLL

public class BLL\_DanhMuc {

DAL\_DanhMuc dal ;

public BLL\_DanhMuc() {

dal = new DAL\_DanhMuc();

}

public ResultSet layDM() throws Exception {

return dal.layDS();

}

public boolean themDM(ET\_DanhMuc et) throws Exception {

return dal.themDM(et);

}

public boolean xoaDM(String maDM) throws Exception {

return dal.xoaDM(maDM);

}

public boolean suaDM(ET\_DanhMuc et) throws Exception {

return dal.suaDM(et);

}

public String layMa(String ten) throws Exception {

return dal.layMaDM(ten);

}

public String layTen(String ma) throws Exception {

return dal.layTenDM(ma);

}

}

#### DAL

public ResultSet layDS() throws SQLException, Exception {

Connection conn = null;

ResultSet rs = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "select \* from categories ";

rs = st.executeQuery(sql);

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, e);

}

return rs;

}

public String layMaDM(String tenDM) throws Exception {

Connection conn = null;

conn = DatabaseUtil.getConnection();

ResultSet rs = null;

Statement st = conn.createStatement();

String sql = "select \* from categories where name = '" + tenDM + "'";

rs = st.executeQuery(sql);

String ma = "";

while (rs.next()) {

ma = (String) rs.getObject(1);

}

return ma;

}

public String layTenDM(String maDM) throws Exception {

Connection conn = null;

conn = DatabaseUtil.getConnection();

ResultSet rs = null;

Statement st = conn.createStatement();

String sql = "select \* from categories where id\_category = '" + maDM + "'";

rs = st.executeQuery(sql);

String ma = "";

while (rs.next()) {

ma = (String) rs.getObject(2);

}

return ma;

}

public Boolean checkTT(String maDM) throws SQLException, Exception {

Connection conn = null;;

ResultSet rs = null;

int row = 0;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "Select \* from categories where id\_category = '" + maDM + "'";

rs = st.executeQuery(sql);

rs.last();

row = rs.getRow();

rs.beforeFirst();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return (row > 0);

}

public Boolean themDM(ET\_DanhMuc et) throws SQLException, Exception {

Connection conn = null;

Boolean kq = false;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "insert into categories(id\_category, name)"

+ " values('" + et.getMaDanhMuc() + "','" + et.getTenDanhMuc() + "')";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, e);

}

return kq;

}

public Boolean suaDM(ET\_DanhMuc et) throws SQLException, Exception {

Connection conn = null;

Boolean kq = false;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "update categories "

+ "set name = '" + et.getTenDanhMuc() + "' "

+ "where id\_category = '" + et.getMaDanhMuc() + "'";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, e);

}

return kq;

}

public Boolean xoaDM(String maDM) throws SQLException, Exception {

Connection conn = null;

Boolean kq = false;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "delete from categories where id\_category = '" + maDM + "'";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, e);

}

return kq;

}

public ResultSet checkDM(String maDM) throws Exception {

Connection conn = null;

ResultSet rs = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

String sql = "select \* from users where username = '" + maDM + "'";

rs = st.executeQuery(sql);

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, e);

}

return rs;

}

#### ET

public class ET\_DanhMuc {

private String maDanhMuc; // mã danh mục

private String tenDanhMuc; // tên danh mục

public ET\_DanhMuc(String maDanhMuc, String tenDanhMuc) {

this.maDanhMuc = maDanhMuc;

this.tenDanhMuc = tenDanhMuc;

}

public String getMaDanhMuc() {

return maDanhMuc;

}

public void setMaDanhMuc(String maDanhMuc) {

this.maDanhMuc = maDanhMuc;

}

public String getTenDanhMuc() {

return tenDanhMuc;

}

public void setTenDanhMuc(String tenDanhMuc) {

this.tenDanhMuc = tenDanhMuc;

}

#### GUI

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

try {

layMa();

reset();

} catch (Exception ex) {

ex.printStackTrace();

}

}

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

try {

String maDM = txtMaDM.getText(); // mã danh mục

if (txtTenDM.getText().compareTo("") > 0) {

String tenDM = txtTenDM.getText(); // tên danh mục

ET\_DanhMuc et = new ET\_DanhMuc(maDM, tenDM);

if (bll.themDM(et)) {

JOptionPane.showMessageDialog(null, "Thêm thành công");

reset();

} else {

JOptionPane.showMessageDialog(rootPane, "Thêm không thành công");

}

} else {

JOptionPane.showMessageDialog(null, "Tên danh mục trống");

}

} catch (Exception ex) {

ex.printStackTrace();

}

}

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

int row = tblDanhMuc.getSelectedRow();

if (row != -1) {

try {

String maDM = txtMaDM.getText(); // mã danh mục

if (txtTenDM.getText().compareTo("") > 0) {

String tenDM = txtTenDM.getText();

ET\_DanhMuc et = new ET\_DanhMuc(maDM, tenDM);

if (bll.suaDM(et)) {

JOptionPane.showMessageDialog(rootPane, "Sửa thành công");

reset();

} else {

JOptionPane.showMessageDialog(rootPane, "Sửa không thành công");

}

}

} catch (Exception ex) {

ex.printStackTrace();

}

} else {

JOptionPane.showMessageDialog(rootPane, "Vui lòng chọn danh mục để sửa");

}

}

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

int row = tblDanhMuc.getSelectedRow(); // lấy hàng đang được chọn

if (row != -1) {

try {

String maDM = txtMaDM.getText();

String tenDM = txtTenDM.getText().compareTo("") > 0 ? txtTenDM.getText() : "";

if (maDM != "" && tenDM != "") {

ET\_DanhMuc et = new ET\_DanhMuc(maDM, tenDM);

if (bll.xoaDM(maDM)) {

JOptionPane.showMessageDialog(rootPane, "Xóa thành công");

reset();

} else {

JOptionPane.showMessageDialog(rootPane, "Xóa không thành công");

}

}

} catch (Exception ex) {

ex.printStackTrace();

}

} else {

JOptionPane.showMessageDialog(rootPane, "Vui lòng chọn danh mục để xóa");

}

}

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

int kq = JOptionPane.showConfirmDialog(null, "Bạn có muốn thoát không?", "Thông báo", JOptionPane.YES\_NO\_OPTION);

if (kq == 0) {

this.dispose();

}

}

private void tblDanhMucMouseClicked(java.awt.event.MouseEvent evt) {

int row = tblDanhMuc.getSelectedRow();

DefaultTableModel model = (DefaultTableModel) tblDanhMuc.getModel();

txtMaDM.setText(model.getValueAt(row, 0).toString());

txtTenDM.setText(model.getValueAt(row, 1).toString());

}

private void hienThi() {

DefaultTableModel model = (DefaultTableModel) tblDanhMuc.getModel();

model.setRowCount(0);

model.setColumnCount(0);

model.addColumn("Mã danh mục");

model.addColumn("Tên danh mục");

tblDanhMuc.setModel(model);

}

private void hienThiDM() throws Exception {

DefaultTableModel model = (DefaultTableModel) tblDanhMuc.getModel();

model.setRowCount(0);

ResultSet rs = bll.layDM();

while (rs.next()) {

Object[] col = new Object[2];

for (int i = 1; i <= 2; i++) {

col[i - 1] = rs.getObject(i);

}

model.addRow(col);

}

tblDanhMuc.setModel(model);

}

public void layMa() throws Exception {

STT = 0;

ResultSet rs = bll.layDM();

rs.last();

int row = rs.getRow();

if (row == 0) {

STT = 1;

} else {

rs.beforeFirst();

while (rs.next()) {

STT = Integer.parseInt(rs.getObject(1).toString().substring(2)) + 1;

}

}

String ma = String.format("%02d", STT);

txtMaDM.setText("DM" + ma);

}

private void reset() throws Exception {

layMa();

hienThi();

hienThiDM();

txtTenDM.setText("");

txtTenDM.requestFocus();

}

### Ngôn ngữ

#### BLL

private DAL\_NgonNgu dal;

public BLL\_NgonNgu() throws Exception {

dal = new DAL\_NgonNgu();

}

public ResultSet layNgonNgu() throws SQLException {

return dal.layNgonNgu();

}

public boolean themNgonNgu(ET\_NgonNgu et) throws SQLException {

return dal.themNgonNgu(et);

}

public boolean suaNgonNgu(ET\_NgonNgu et) throws SQLException {

return dal.suaNgonNgu(et);

}

public boolean xoaNgonNgu(ET\_NgonNgu et) throws SQLException {

return dal.xoaNgonNgu(et);

}

public boolean checkNN(String maNgonNgu) throws SQLException {

return dal.checkNN(maNgonNgu);

}

public boolean checkTonTai(String ma) throws SQLException{

return dal.checkTonTai(ma);

}

public String layMa(String ten) throws Exception {

return dal.layMaNN(ten);

}

public String layTen(String ma) throws Exception {

return dal.layTen(ma);

**}**

#### DAL

private Connection conn;

public DAL\_NgonNgu() throws Exception {

conn = DatabaseUtil.getConnection();

}

public ResultSet layNgonNgu() throws SQLException {

Connection conn = null;

ResultSet rs = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

String sql = "select \* from languages";

rs = st.executeQuery(sql);

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return rs;

}

public Boolean themNgonNgu(ET\_NgonNgu et) throws SQLException {

Statement st = conn.createStatement();

String sql = "INSERT INTO languages(id\_language, name)"

+ "VALUES(?, ?)";

try {

PreparedStatement pst = conn.prepareStatement(sql);

pst.setString(1, et.getMaNgonNgu());

pst.setString(2, et.getTenNgonNgu());

return pst.executeUpdate() > 1;

} catch (Exception e) {

e.printStackTrace();

}

return false;

}

public Boolean suaNgonNgu(ET\_NgonNgu et) throws SQLException {

Statement st = conn.createStatement();

String sql = "UPDATE languages SET name = ? WHERE id\_language = ?";

try {

PreparedStatement pst = conn.prepareStatement(sql);

pst.setString(2, et.getMaNgonNgu());

pst.setString(1, et.getTenNgonNgu());

return pst.executeUpdate() > 1;

} catch (Exception e) {

e.printStackTrace();

}

return false;

}

public Boolean xoaNgonNgu(ET\_NgonNgu et) throws SQLException {

Statement st = conn.createStatement();

String sql = "DELETE FROM languages WHERE id\_language = ?";

try {

PreparedStatement pst = conn.prepareStatement(sql);

pst.setString(1, et.getMaNgonNgu());

return pst.executeUpdate() > 1;

} catch (Exception e) {

e.printStackTrace();

}

return false;

}

public Boolean checkNN(String maNgonNgu) throws SQLException {

Connection conn = null;;

ResultSet rs = null;

int row = 0;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "Select \* from languages where id\_language = '" + maNgonNgu + "'";

rs = st.executeQuery(sql);

rs.last();

row = rs.getRow();

rs.beforeFirst();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return (row > 0);

}

public Boolean checkTonTai(String ma) throws SQLException {

Connection conn = null;;

ResultSet rs = null;

int row = 0;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "Select \* from languages where id\_language = '" + ma + "'";

rs = st.executeQuery(sql);

rs.last();

row = rs.getRow();

rs.beforeFirst();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return (row > 0);

}

public String layMaNN(String tenNN) throws Exception {

Connection conn = null;

conn = DatabaseUtil.getConnection();

ResultSet rs = null;

Statement st = conn.createStatement();

String sql = "select \* from languages where name = '"+tenNN+"' limit 1";

rs = st.executeQuery(sql);

String ma = "";

while (rs.next()) {

ma = rs.getObject(1).toString();

break;

}

return ma;

}

public String layTen(String maNN) throws Exception {

Connection conn = null;

conn = DatabaseUtil.getConnection();

ResultSet rs = null;

Statement st = conn.createStatement();

String sql = "select \* from languages where id\_language = '"+maNN+"' limit 1";

rs = st.executeQuery(sql);

String ma = "";

while (rs.next()) {

ma = rs.getObject(2).toString();

break;

}

return ma;

}

#### ET

public ET\_NgonNgu(String maNgonNgu, String tenNgonNgu) {

this.maNgonNgu = maNgonNgu;

this.tenNgonNgu = tenNgonNgu;

}

public String getMaNgonNgu() {

return maNgonNgu;

}

public void setMaNgonNgu(String maNgonNgu) {

this.maNgonNgu = maNgonNgu;

}

public String getTenNgonNgu() {

return tenNgonNgu;

}

public void setTenNgonNgu(String tenNgonNgu) {

this.tenNgonNgu = tenNgonNgu;

}

#### GUI

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

String maNgonNgu = txtMaNgonNgu.getText();

String tenNgonNgu = txtTenNgonNgu.getText();

ET\_NgonNgu et = new ET\_NgonNgu(maNgonNgu, tenNgonNgu);

if (maNgonNgu == "" || tenNgonNgu == "") {

JOptionPane.showMessageDialog(null, "Vui lòng điền đầy đủ thông tin");

} else {

try {

if (bll.checkNN(maNgonNgu)) {

JOptionPane.showMessageDialog(null, "Đã tồn tại ngôn ngữ");

} else {

et = new ET\_NgonNgu(maNgonNgu, tenNgonNgu);

if (bll.themNgonNgu(et)) {

JOptionPane.showMessageDialog(rootPane, "Thêm không thành công");

} else {

JOptionPane.showMessageDialog(rootPane, "Thêm thành công");

reset();

}

}

} catch (SQLException ex) {

ex.printStackTrace();

} catch (Exception ex) {

ex.printStackTrace();

}

}

}

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

String maNgonNgu = txtMaNgonNgu.getText();

String tenNgonNgu = txtTenNgonNgu.getText();

ET\_NgonNgu et = new ET\_NgonNgu(maNgonNgu, tenNgonNgu);

if (maNgonNgu == "" || tenNgonNgu == "") {

JOptionPane.showMessageDialog(null, "Vui lòng điền đầy đủ thông tin");

} else {

try {

bll.suaNgonNgu(et);

JOptionPane.showMessageDialog(rootPane, "Sửa thành công");

reset();

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(rootPane, "Sửa không thành công");

}

}

}

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

String maNgonNgu = txtMaNgonNgu.getText();

String tenNgonNgu = txtTenNgonNgu.getText();

ET\_NgonNgu et;

int row = tblNgonNgu.getSelectedRow();

if (row >= 0) {

et = new ET\_NgonNgu(maNgonNgu, tenNgonNgu);

try {

if (bll.xoaNgonNgu(et)) {

JOptionPane.showMessageDialog(rootPane, "Xoá không thành công");

} else {

JOptionPane.showMessageDialog(rootPane, "Xóa thành công");

reset();

}

} catch (SQLException ex) {

ex.printStackTrace();

}

} else {

JOptionPane.showMessageDialog(null, "Vui lòng chọn ngôn ngữ");

}

}

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

int kq = JOptionPane.showConfirmDialog(null, "Bạn có muốn thoát không", "Thông báo", JOptionPane.YES\_NO\_OPTION);

if (kq == 0) {

this.dispose();

}

}

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

try {

reset();

} catch (SQLException ex) {

ex.printStackTrace();

}

}

private void hienThi() {

DefaultTableModel model = (DefaultTableModel) tblNgonNgu.getModel();

model.setColumnCount(0);

model.setRowCount(0);

model.addColumn("Mã ngôn ngữ");

model.addColumn("Tên ngôn ngữ");

tblNgonNgu.setModel(model);

}

public void hienThiNN() throws SQLException {

ResultSet rs = bll.layNgonNgu();

DefaultTableModel model = (DefaultTableModel) tblNgonNgu.getModel();

model.setNumRows(0);

while (rs.next()) {

Object[] col = new Object[2];

for (int i = 1; i <= 2; i++) {

col[i - 1] = rs.getObject(i);

}

model.addRow(col);

}

tblNgonNgu.getSelectionModel().addListSelectionListener(new ListSelectionListener() {

public void valueChanged(ListSelectionEvent e) {

if (tblNgonNgu.getSelectedRow() >= 0) {

txtMaNgonNgu.setText(tblNgonNgu.getValueAt(tblNgonNgu.getSelectedRow(), 0) + "");

txtTenNgonNgu.setText(tblNgonNgu.getValueAt(tblNgonNgu.getSelectedRow(), 1) + "");

}

}

});

}

public void reset() throws SQLException {

ResultSet rs = bll.layNgonNgu();

rs.last();

int row = rs.getRow();

if (row == 0) {

STT = 1;

} else {

rs.beforeFirst();

while (rs.next()) {

STT = Integer.parseInt(rs.getObject(1).toString().substring(2)) + 1;

}

}

String ma = String.format("%02d", STT);

txtMaNgonNgu.setText("NN" + ma);

hienThi();

hienThiNN();

txtTenNgonNgu.setText("");

}

### Nhà cung cấp

#### BLL

DAL\_NhaCungCap dal;

public BLL\_NhaCungCap() {

dal = new DAL\_NhaCungCap();

}

public String layMa(String ten) throws Exception {

return dal.layMaNCC(ten);

}

public Boolean themNCC(ET\_NhaCungCap et) throws SQLException, Exception {

return dal.themNCC(et);

}

public Boolean xoaNCC(String maNCC) throws SQLException {

return dal.xoaNCC(maNCC);

}

public Boolean suaNCC(ET\_NhaCungCap et) throws SQLException {

return dal.suaNCC(et);

}

public Boolean checkTT(String ma) throws SQLException {

return dal.checkTonTai(ma);

}

public ResultSet layNCC() throws SQLException {

return dal.layNCC();

}

public String layTenNCC(String maNCC) throws SQLException, Exception {

return dal.layTenNCC(maNCC);

}

#### DAL

public ResultSet layNCC() throws SQLException {

ResultSet rs = null;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "select \* from suppliers";

rs = st.executeQuery(sql);

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return rs;

}

public Boolean themNCC(ET\_NhaCungCap et) throws SQLException, Exception {

Boolean kq = false;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "INSERT INTO suppliers(id\_supplier, supplier\_name,address,phone\_number)"

+ "values ('" + et.getMaNCC()+ "','"+ et.getTenNCC()+ "','"+ et.getDiaChiNCC()+ "','"+ et.getSdtNCC()+ "')";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return kq;

}

public Boolean checkTonTai(String ma) throws SQLException {

Connection conn = null;;

ResultSet rs = null;

int row = 0;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "Select \* from suppliers where id\_supplier = '" + ma + "'";

rs = st.executeQuery(sql);

rs.last();

row = rs.getRow();

rs.beforeFirst();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return (row > 0);

}

public Boolean xoaNCC(String maNCC) throws SQLException {

Boolean kq = false;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "delete from suppliers where id\_supplier = '" + maNCC+"'";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return kq;

}

public Boolean suaNCC(ET\_NhaCungCap et) throws SQLException {

Boolean kq = false;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "update suppliers "

+ "set supplier\_name = '" + et.getTenNCC()+ "', "

+ "address = '" + et.getDiaChiNCC()+ "',"

+ "phone\_number = '" + et.getSdtNCC() + "' "

+ "where id\_supplier = '" + et.getMaNCC()+ "'";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return kq;

}

public String layMaNCC(String tenNCC) throws Exception {

Connection conn = null;

conn = DatabaseUtil.getConnection();

ResultSet rs = null;

Statement st = conn.createStatement();

String sql = "select \* from suppliers where supplier\_name ='" + tenNCC + "' limit 1";

rs = st.executeQuery(sql);

String ma = "";

while (rs.next()) {

ma = (String) rs.getObject(1);

break;

}

return ma;

}

public String layTenNCC(String maNCC) throws Exception {

Connection conn = null;

conn = DatabaseUtil.getConnection();

ResultSet rs = null;

Statement st = conn.createStatement();

String sql = "select \* from suppliers where id\_supplier ='" + maNCC + "' limit 1";

rs = st.executeQuery(sql);

String ma = "";

while (rs.next()) {

ma = (String) rs.getObject(2);

break;

}

return ma;

}

#### ET

private String maNCC; // mã nhà cung cấp

private String tenNCC; // tên nhà cung cấp

private String diaChiNCC; // địa chỉ

private String sdtNCC; // số điện thoại

public ET\_NhaCungCap() {

this.maNCC = "";

this.tenNCC = "";

this.diaChiNCC = "";

this.sdtNCC = "";

}

public ET\_NhaCungCap(String maNCC, String tenNCC, String diaChiNCC, String sdtNCC) {

this.maNCC = maNCC;

this.tenNCC = tenNCC;

this.diaChiNCC = diaChiNCC;

this.sdtNCC = sdtNCC;

}

public String getMaNCC() {

return maNCC;

}

public void setMaNCC(String maNCC) {

this.maNCC = maNCC;

}

public String getTenNCC() {

return tenNCC;

}

public void setTenNCC(String tenNCC) {

this.tenNCC = tenNCC;

}

public String getDiaChiNCC() {

return diaChiNCC;

}

public void setDiaChiNCC(String diaChiNCC) {

this.diaChiNCC = diaChiNCC;

}

public String getSdtNCC() {

return sdtNCC;

}

public void setSdtNCC(String sdtNCC) {

this.sdtNCC = sdtNCC;

}

#### GUI

public void hienThi() {

DefaultTableModel model = (DefaultTableModel) tbNCC.getModel();

model.setColumnCount(0);

model.setRowCount(0);

model.addColumn("Mã NCC");

model.addColumn("Tên NCC");

model.addColumn("Địa chỉ");

model.addColumn("Số ĐT");

tbNCC.setModel(model);

}

public void hienThiNCC() throws SQLException {

ResultSet rs = bll.layNCC();

DefaultTableModel model = (DefaultTableModel) tbNCC.getModel();

model.setNumRows(0);

while (rs.next()) {

Object[] col = new Object[4];

for (int i = 1; i <= 4; i++) {

col[i - 1] = rs.getObject(i);

}

model.addRow(col);

}

}

public void layMa() throws Exception {

ResultSet rs = bll.layNCC();

rs.last();

int row = rs.getRow();

if (row == 0) {

STT = 1;

} else {

rs.beforeFirst();

while (rs.next()) {

STT = Integer.parseInt(rs.getObject(1).toString().substring(3)) + 1;

}

}

String ma = String.format("%02d", STT);

txtMaNCC.setText("NCC" + ma);

txtTenNCC.setText("");

txtDiaChi.setText("");

txtSDT.setText("");

txtTenNCC.requestFocus();

}

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

String maNCC = txtMaNCC.getText();

String tenNCC = txtTenNCC.getText();

String diaChi = txtDiaChi.getText();

String soDT = txtSDT.getText();

if (maNCC.compareTo("") == 0

|| tenNCC.compareTo("") == 0

|| diaChi.compareTo("") == 0

|| soDT.compareTo("") == 0) {

JOptionPane.showMessageDialog(null, "Vui lòng điền đầy đủ thông tin");

} else {

ET\_NhaCungCap et = new ET\_NhaCungCap(maNCC, tenNCC, diaChi, soDT);

try {

if (bll.checkTT(maNCC)) {

JOptionPane.showMessageDialog(null, "Đã tồn tại Nha Cung Cap");

} else {

if (bll.themNCC(et)) {

JOptionPane.showMessageDialog(rootPane, "Thành công");

STT++;

layMa();

hienThiNCC();

} else {

JOptionPane.showMessageDialog(rootPane, "Không thành công");

}

}

} catch (SQLException ex) {

ex.printStackTrace();

} catch (Exception ex) {

ex.printStackTrace();

}

}

}

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

int dong = tbNCC.getSelectedRow();

String maNCC = tbNCC.getValueAt(dong, 0).toString();

String tenNCC = txtTenNCC.getText();

String diaChi = txtDiaChi.getText();

String soDT = txtSDT.getText();

ET\_NhaCungCap et;

if (dong != -1) {

if (maNCC.compareTo("") == 0

|| tenNCC.compareTo("") == 0

|| diaChi.compareTo("") == 0

|| soDT.compareTo("") == 0) {

JOptionPane.showMessageDialog(null, "Vui lòng điền đầy đủ thông tin");

} else {

try {

et = new ET\_NhaCungCap(maNCC, tenNCC, diaChi, soDT);

if (bll.suaNCC(et)) {

JOptionPane.showMessageDialog(rootPane, "Thành công");

layMa();

hienThiNCC();

} else {

JOptionPane.showMessageDialog(rootPane, "Không thành công");

}

} catch (Exception ex) {

ex.printStackTrace();

}

}

} else {

JOptionPane.showMessageDialog(null, "Vui lòng chọn nhà cung cấp");

}

}

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

int row = tbNCC.getSelectedRow();

if (row != -1) {

try {

String maNCC = txtMaNCC.getText();

String tenNCC = txtTenNCC.getText().compareTo("") > 0 ? txtTenNCC.getText() : "";

String diaChi = txtDiaChi.getText().compareTo("") > 0 ? txtDiaChi.getText() : "";

String soDT = txtSDT.getText().compareTo("") > 0 ? txtSDT.getText() : "";

if (maNCC != "" && tenNCC != "" && diaChi != "" && soDT != "") {

ET\_NhaCungCap et = new ET\_NhaCungCap(maNCC, tenNCC, diaChi, soDT);

if (bll.xoaNCC(maNCC)) {

JOptionPane.showMessageDialog(rootPane, "Thành công");

layMa();

hienThiNCC();

} else {

JOptionPane.showMessageDialog(rootPane, "Không thành công");

}

}

} catch (Exception ex) {

ex.printStackTrace();

}

} else {

JOptionPane.showMessageDialog(null, "Vui lòng chọn nhà cung cấp");

}

}

private void tbNCCMouseClicked(java.awt.event.MouseEvent evt) {

int row = tbNCC.getSelectedRow();

DefaultTableModel model = (DefaultTableModel) tbNCC.getModel();

txtMaNCC.setText(model.getValueAt(row, 0).toString());

txtTenNCC.setText(model.getValueAt(row, 1).toString());

txtDiaChi.setText(model.getValueAt(row, 2).toString());

txtSDT.setText(model.getValueAt(row, 3).toString());

}

private void txtSDTKeyPressed(java.awt.event.KeyEvent evt) {

char c =evt.getKeyChar();

if(Character.isLetter(c)){

txtSDT.setEditable(false);

JOptionPane.showMessageDialog(null, "Chỉ nhập số");

}else{

txtSDT.setEditable(true);

}

}

### Tài khoản

#### BLL

public BLL\_TaiKhoan() throws Exception {

dal = new DAL\_TaiKhoan();

}

public ResultSet layDS() throws SQLException, Exception {

return dal.layDS();

}

public Boolean checkTT(String maTK) throws SQLException, Exception {

return dal.checkTK(maTK);

}

public Boolean themTK(ET\_TaiKhoan et) throws SQLException, Exception {

return dal.themTK(et);

}

public Boolean xoaTK(ET\_TaiKhoan et) throws SQLException, Exception {

return dal.xoaTK(et);

}

public Boolean suaTK(ET\_TaiKhoan et) throws SQLException, Exception {

return dal.suaTK(et);

}

#### DAL

public ResultSet layDS() throws SQLException, Exception {

Connection conn = null;

ResultSet rs = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "select \* from users";

rs = st.executeQuery(sql);

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, e);

}

return rs;

}

public ResultSet checkDN(String tenDN, String mk) throws Exception {

Connection conn = null;

ResultSet rs = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

String sql = "select \* from users where username = '"+tenDN+"' and password = '" + mk+"'";

rs = st.executeQuery(sql);

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, e);

}

return rs;

}

public Boolean checkTK(String maTK) throws SQLException, Exception {

Connection conn = null;;

ResultSet rs = null;

int row = 0;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "Select \* from users where id\_user = '" + maTK + "'" ;

rs = st.executeQuery(sql);

rs.last();

row = rs.getRow();

rs.beforeFirst();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return (row > 0);

}

public Boolean themTK(ET\_TaiKhoan et) throws SQLException, Exception {

Connection conn = null;

Boolean kq = false;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "insert into users(id\_user, username, password, role, id\_staff)"

+ " values('"+et.getMaTK()+"','"+et.getTenDN()+"','"+et.getMatKhau()+"','"+et.getLoaiTK()+"','"+et.getMaNV()+"')";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, e);

}

return kq;

}

public Boolean suaTK(ET\_TaiKhoan et) throws SQLException, Exception {

Connection conn = null;

Boolean kq = false;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "update users "

+ "set username = '"+et.getTenDN()+"',"

+ " password = '"+et.getMatKhau()+"', "

+ "role = '"+et.getLoaiTK()+"', "

+ "id\_staff = '"+et.getMaNV()+"'"

+ "where id\_user = '"+ et.getMaTK()+"'";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, e);

}

return kq;

}

public Boolean xoaTK(ET\_TaiKhoan et) throws SQLException, Exception {

Connection conn = null;

Boolean kq = false;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "delete from users where id\_user = '"+et.getMaTK()+"'";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, e);

}

return kq;

}

#### ET

private String maTK; // mã tài khoản

private String tenDN; // tên đăng nhập

private String matKhau; // mật khẩu

private String loaiTK; // loại tài khoản

private String maNV; // mã nhân viên

public String getMaTK() {

return maTK;

}

public void setMaTK(String maTK) {

this.maTK = maTK;

}

public String getTenDN() {

return tenDN;

}

public void setTenDN(String tenDN) {

this.tenDN = tenDN;

}

public String getMatKhau() {

return matKhau;

}

public void setMatKhau(String matKhau) {

this.matKhau = matKhau;

}

public String getLoaiTK() {

return loaiTK;

}

public void setLoaiTK(String loaiTK) {

this.loaiTK = loaiTK;

}

public String getMaNV() {

return maNV;

}

public void setMaNV(String maNV) {

this.maNV = maNV;

}

public ET\_TaiKhoan(String maTK, String tenDN, String matKhau, String loaiTK, String maNV) {

this.maTK = maTK;

this.tenDN = tenDN;

this.matKhau = matKhau;

this.loaiTK = loaiTK;

this.maNV = maNV;

}

#### GUI

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

String maTK = txtMaTK.getText();

String tenDN = txtTenDN.getText();

String mk = txtMK.getText();

String loaiTK = cboLoaiTK.getSelectedItem().toString();

String maNV = txtMaNV.getText();

ET\_TaiKhoan et;

BLL\_NhanVien bllNV = null;

if (tenDN.compareTo("") == 0 || mk.compareTo("") == 0 || loaiTK.compareTo("") == 0 || maNV.compareTo("") == 0) {

JOptionPane.showMessageDialog(null, "Vui lòng điền đầy đủ thông tin");

} else {

try {

if (bll.checkTT(maTK)) {

JOptionPane.showMessageDialog(null, "Đã tồn tại tài khoản này");

} else {

bllNV = new BLL\_NhanVien();

if (!bllNV.checkTT(maNV)) {

JOptionPane.showMessageDialog(null, "Không tồn tại nhân viên này");

} else {

et = new ET\_TaiKhoan(maTK, tenDN, mk, loaiTK, maNV);

bll.themTK(et);

layMa();

hienThiTK();

}

}

} catch (Exception ex) {

ex.printStackTrace();

}

}

}

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

int row = tblTaiKhoan.getSelectedRow();

String maTK = txtMaTK.getText();

String tenDN = txtTenDN.getText();

String mk = txtMK.getText();

String loaiTK = cboLoaiTK.getSelectedItem().toString();

String maNV = txtMaNV.getText();

ET\_TaiKhoan et;

BLL\_NhanVien bllNV = null;

if (tenDN.compareTo("") == 0 || mk.compareTo("") == 0 || loaiTK.compareTo("") == 0 || maNV.compareTo("") == 0) {

JOptionPane.showMessageDialog(null, "Vui lòng điền đầy đủ thông tin");

} else {

try {

bllNV = new BLL\_NhanVien();

if (!bllNV.checkTT(maNV)) {

JOptionPane.showMessageDialog(null, "Không tồn tại nhân viên này");

} else {

et = new ET\_TaiKhoan(maTK, tenDN, mk, loaiTK, maNV);

if (bll.suaTK(et)) {

JOptionPane.showMessageDialog(null, "Sửa thành công");

layMa();

hienThiTK();;

} else {

JOptionPane.showMessageDialog(null, "Sửa không thành công");

}

}

} catch (Exception ex) {

ex.printStackTrace();

}

}

}

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

ET\_TaiKhoan et;

int row = tblTaiKhoan.getSelectedRow();

if (row >= 0) {

int rep = JOptionPane.showConfirmDialog(rootPane, "Bạn có muốn xóa không?", "Thông báo", JOptionPane.YES\_NO\_OPTION);

if (rep == 0) {

String maTK = txtMaTK.getText();

String tenDN = txtTenDN.getText();

String mk = txtMK.getText();

String loaiTK = cboLoaiTK.getSelectedItem().toString();

String maNV = txtMaNV.getText();

et = new ET\_TaiKhoan(maTK, tenDN, mk, loaiTK, maNV);

try {

if (bll.xoaTK(et)) {

JOptionPane.showMessageDialog(rootPane, "Xóa thành công");

layMa();

hienThiTK();

} else {

JOptionPane.showMessageDialog(rootPane, "Xóa không thành công");

}

} catch (Exception ex) {

ex.printStackTrace();

public Boolean suaLoai(ET\_LoaiDocGia et) throws Exception {

return dal.suaLoai(et);

### 2.10 Độc giả

#### DAL

public Boolean themDocGia(ET\_DocGia et) throws SQLException, Exception {

Boolean kq = false;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "insert into readers (id\_reader, firstname,lastname, birthday,gender,pid,address,reader\_type) "

+ "values ('" + et.getMaDocGia() + "','" + et.getTen()+ "','" + et.getHo()+ "','" + new java.sql.Date(et.getNgaySinh().getTime()) + "','" + et.getGioiTinh() + "','" + et.getCccd() + "','" + et.getDiaChi() + "'," + et.getloaiDG() + ")";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

//conn.close();

return kq;

}

// check tồn tại độc giả

public Boolean checkTT(String cccd) throws SQLException, Exception {

Connection conn = null;;

ResultSet rs = null;

int row = 0;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "Select \* from readers where pid ='" + cccd+"'";

rs = st.executeQuery(sql);

rs.last();

row = rs.getRow();

rs.beforeFirst();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return (row > 0);

}

// lấy danh sách độc giả

public ResultSet layDS() throws SQLException, Exception {

Connection conn = null;;

ResultSet rs = null;

int row = 0;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "Select \* from readers";

rs = st.executeQuery(sql);

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return rs;

}

public Boolean suaDocGia(ET\_DocGia et) throws SQLException {

Boolean kq = false;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "update readers "

+ "set firstname = '" + et.getTen() + "',"

+ "lastname = '" + et.getHo() + "',"

+ "birthday = '" + et.getNgaySinh() + "',"

+ "gender = '" + et.getGioiTinh() + "',"

+ "pid = '" + et.getCccd() + "',"

+ "address = '" + et.getDiaChi() + " "

+ "where id\_reader ='" + et.getMaDocGia() + "'";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return kq;

}

#### ET

private String maDocGia; // mã độc giả

private String ho; // họ độc giả

private String ten; // tên độc giả

private Date ngaySinh; // ngày sinh

private String gioiTinh; // giới tính

private String cccd;// căn cước công dân

private String diaChi; // địa chỉ

private int loaiDG; // loại độc giả

public String getMaDocGia() {

return maDocGia;

}

public void setMaDocGia(String maDocGia) {

this.maDocGia = maDocGia;

}

public String getHo() {

return ho;

}

public void setHo(String ho) {

this.ho = ho;

}

public String getTen() {

return ten;

}

public void setTen(String ten) {

this.ten = ten;

}

public Date getNgaySinh() {

return ngaySinh;

}

public void setNgaySinh(Date ngaySinh) {

this.ngaySinh = ngaySinh;

}

public String getGioiTinh() {

return gioiTinh;

}

public void setGioiTinh(String gioiTinh) {

this.gioiTinh = gioiTinh;

}

public String getCccd() {

return cccd;

}

public void setCccd(String cccd) {

this.cccd = cccd;

}

public String getDiaChi() {

return diaChi;

}

public void setDiaChi(String diaChi) {

this.diaChi = diaChi;

}

public int getloaiDG() {

return loaiDG;

}

public void setTrangThai(int trangThai) {

this.loaiDG = trangThai;

}

public ET\_DocGia(String maDocGia, String ho, String ten, Date ngaySinh, String gioiTinh, String cccd, String diaChi, int loaiDG) {

this.maDocGia = maDocGia;

this.ho = ho;

this.ten = ten;

this.ngaySinh = ngaySinh;

this.gioiTinh = gioiTinh;

this.cccd = cccd;

this.diaChi = diaChi;

this.loaiDG = loaiDG;

}

#### GUI

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

if (txtTenLoai.getText().compareTo("") != 0) {

if (checkSo(txtSL.getText())) {

if (checkSo(txtPhi.getText())) {

String tenLoai = txtTenLoai.getText();

int phi = Integer.parseInt(txtPhi.getText());

int sl = Integer.parseInt(txtSL.getText());

ET\_LoaiDocGia et = new ET\_LoaiDocGia(0, tenLoai, sl, phi);

try {

if (bll.themLoai(et)) {

JOptionPane.showMessageDialog(rootPane, "Thành công");

reset();

} else {

JOptionPane.showMessageDialog(rootPane, "Không thành công");

}

} catch (Exception ex) {

Logger.getLogger(frmLoaiDocGia.class.getName()).log(Level.SEVERE, null, ex);

}

} else {

JOptionPane.showMessageDialog(null, "Phí không hợp lệ");

}

} else {

JOptionPane.showMessageDialog(null, "Số lượng không hợp lệ");

}

} else {

JOptionPane.showMessageDialog(null, "Vui lòng điền đầy đủ thông tin");

}

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

if (txtTenLoai.getText().compareTo("") != 0) {

if (checkSo(txtSL.getText())) {

if (checkSo(txtPhi.getText())) {

int row = tblLoaiDocGia.getSelectedRow();

int maLoai = Integer.parseInt(tblLoaiDocGia.getValueAt(row, 0).toString());

String tenLoai = txtTenLoai.getText();

int phi = Integer.parseInt(txtPhi.getText());

int sl = Integer.parseInt(txtSL.getText());

ET\_LoaiDocGia et = new ET\_LoaiDocGia(maLoai, tenLoai, sl, phi);

try {

if (bll.suaLoai(et)) {

JOptionPane.showMessageDialog(rootPane, "Sửa Thành công");

reset();

} else {

JOptionPane.showMessageDialog(rootPane, "Sửa không thành công");

}

} catch (Exception ex) {

Logger.getLogger(frmLoaiDocGia.class.getName()).log(Level.SEVERE, null, ex);

}

} else {

JOptionPane.showMessageDialog(null, "Phí không hợp lệ");

}

} else {

JOptionPane.showMessageDialog(null, "Số lượng không hợp lệ");

}

} else {

JOptionPane.showMessageDialog(null, "Vui lòng điền đầy đủ thông tin");

}

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

int row = tblLoaiDocGia.getSelectedRow();

if (row != -1) {

try {

int maLoai = Integer.parseInt(tblLoaiDocGia.getValueAt(row, 0).toString());

if (bll.xoaLoai(maLoai)) {

JOptionPane.showMessageDialog(rootPane, "Xóa thành công");

reset();

} else {

JOptionPane.showMessageDialog(rootPane, "Xóa không thành công");

}

} catch (Exception ex) {

ex.printStackTrace();

}

} else {

JOptionPane.showMessageDialog(null, "Vui lòng chọn nhà cung cấp");

}

}

### Thêm thẻ thư viện

#### BLL

private DAL\_TheThuVien dal;

public BLL\_TheThuVien() throws Exception {

dal = new DAL\_TheThuVien();

}

public ResultSet layDSThe() throws SQLException {

return dal.layDSThe();

}

public ResultSet layDSTheTV() throws SQLException {

return dal.layDSThe();

}

public Boolean themThe(ET\_TheThuVien et) throws SQLException, Exception {

return dal.themTheThuVien(et);

}

public Boolean checkTT(String ma) throws SQLException {

return dal.checkTonTai(ma);

}

public ResultSet layTheTheoMaThe(String maThe) throws SQLException {

return dal.layTheTheoMaThe(maThe);

}

public Boolean giaHan(String ma, Date ngHH) throws SQLException, Exception {

return dal.giaHanTheThuVien(ma, ngHH);

}

#### DAL

public ResultSet layDSThe() throws SQLException {

ResultSet rs = null;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "SELECT id\_lib\_card, readers.lastname ,readers.firstname,readers.pid,readers.gender,readers.birthday,readers.address, created\_at,id\_staff, expiration\_date FROM member\_card join readers on member\_card.id\_reader = readers.id\_reader";

rs = st.executeQuery(sql);

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

} return rs;

}

public ResultSet layTheTheoMaThe(String maTheTV) throws SQLException {

ResultSet rs = null;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "SELECT \* from member\_card WHERE id\_lib\_card = '"+maTheTV+"'";

rs = st.executeQuery(sql);

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return rs;

}

public ResultSet layDSTheTV() throws SQLException {

ResultSet rs = null;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "SELECT \* from member\_card";

rs = st.executeQuery(sql);

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return rs;

}

public Boolean themTheThuVien(ET\_TheThuVien et) throws SQLException, Exception {

Boolean kq = false;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "insert into member\_card (id\_lib\_card,id\_staff,id\_reader,expiration\_date) "

+ "values ('" + et.getMaThe()+ "','" + et.getMaNhanVien()+ "','" + et.getMaDocGia()+ "','"+new java.sql.Date(et.getNgayHetHan().getTime())+"')";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return kq;

}

public Boolean checkTonTai(String ma) throws SQLException {

Connection conn = null;;

ResultSet rs = null;

int row = 0;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "Select \* from member\_card where id\_lib\_card = '" + ma + "'";

rs = st.executeQuery(sql);

rs.last();

row = rs.getRow();

rs.beforeFirst();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return (row > 0);

}

public Boolean giaHanTheThuVien(String maThe, java.util.Date ngHH) throws SQLException, Exception {

Boolean kq = false;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

long millis = System.currentTimeMillis();

String sql = "UPDATE member\_card SET expiration\_date = '"+new java.sql.Date(ngHH.getTime())+"' WHERE id\_lib\_card = '"+maThe+"'";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return kq;

}

#### ET

private String maThe; // mã thẻ thư viện

private String maDocGia; // mã độc giả

private String maNhanVien; // mã nhân viên

private Date ngayTao; // ngày tạo

private Date ngayHetHan; // ngày hết hạn

public ET\_TheThuVien(String maThe, String maDocGia, String maNhanVien, Date ngayTao, Date ngayHetHan) {

this.maThe = maThe;

this.maDocGia = maDocGia;

this.maNhanVien = maNhanVien;

this.ngayTao = ngayTao;

this.ngayHetHan = ngayHetHan;

}

public String getMaThe() {

return maThe;

}

public void setMaThe(String maThe) {

this.maThe = maThe;

}

public String getMaDocGia() {

return maDocGia;

}

public void setMaDocGia(String maDocGia) {

this.maDocGia = maDocGia;

}

public String getMaNhanVien() {

return maNhanVien;

}

public void setMaNhanVien(String maNhanVien) {

this.maNhanVien = maNhanVien;

}

public Date getNgayTao() {

return ngayTao;

}

public void setNgayTao(Date ngayTao) {

this.ngayTao = ngayTao;

}

public Date getNgayHetHan() {

return ngayHetHan;

}

public void setNgayHetHan(Date ngayHetHan) {

this.ngayHetHan = ngayHetHan;

}

#### GUI

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

String maThe = txtMaThe.getText();

String maDG = "";

try {

maDG = layMaDG();

} catch (Exception ex) {

Logger.getLogger(frmTheThuVien.class.getName()).log(Level.SEVERE, null, ex);

}

Date ngaySinh = dtNgaySinh.getDate();

String nhanVien = null;

try {

ET\_NhanVien nv = (ET\_NhanVien) cbMaNV.getSelectedItem();

nhanVien = nv.getMaNhanVien();

} catch (Exception ex) {

Logger.getLogger(frmTheThuVien.class.getName()).log(Level.SEVERE, null, ex);

}

String ho = txtHo.getText();

String ten = txtTen.getText();

String diaChi = txtDiaChi.getText();

String gioiTinh = "Nam";

if (rdNam.isSelected()) {

gioiTinh = rdNam.getText();

} else {

gioiTinh = rdNu.getText();

}

String CCCD = txtCCCD.getText();

ET\_TheThuVien et;

ET\_DocGia etdg;

if (maThe.compareTo("") == 0 || ten.compareTo("") == 0 || ho.compareTo("") == 0 || diaChi.compareTo("") == 0 || CCCD.compareTo("") == 0) {

JOptionPane.showMessageDialog(null, "Vui lòng điền đầy đủ thông tin");

} else {

try {

if (bllThe.checkTT(maThe) == true) {

JOptionPane.showMessageDialog(null, "Thẻ đã tồn tại");

} else {

if (bllDG.checkTT(CCCD) == true) {

JOptionPane.showMessageDialog(null, "Độc giả đã tồn tại, vui lòng gia gia hạn thẻ?");

} else {

ET\_LoaiDocGia etLoaiDG = (ET\_LoaiDocGia) cboLoaiDocGia.getSelectedItem();

int loaiDocGia = etLoaiDG.getMaLoai();

etdg = new ET\_DocGia(maDG, ho, ten, ngaySinh, gioiTinh, CCCD, diaChi, loaiDocGia);

Calendar cal = Calendar.getInstance();

Date ngTao = cal.getTime();

cal.add(Calendar.YEAR, 1);

Date ngayHetHan = cal.getTime();

et = new ET\_TheThuVien(maThe, maDG, nhanVien, ngTao, ngayHetHan);

boolean kq = true;

kq = bllDG.themDG(etdg);

kq = false;

kq = bllThe.themThe(et);

if (kq) {

JOptionPane.showMessageDialog(rootPane, "Thêm thành công");

reset();

hienThiThe();

} else {

JOptionPane.showMessageDialog(rootPane, "Thêm không thành công");

}

}

}

} catch (Exception ex) {

ex.printStackTrace();

}

}

}

private void txtCCCDFocusLost(java.awt.event.FocusEvent evt) {

String cccd = txtCCCD.getText();

try {

if (bllDG.checkTT(cccd)) {

JOptionPane.showMessageDialog(null, "Độc giả đã tồn tại, vui lòng gia gia hạn thẻ?");

}

} catch (Exception ex) {

Logger.getLogger(frmTheThuVien.class.getName()).log(Level.SEVERE, null, ex);

}

public String layMaDG() throws Exception {

STT = 0;

ResultSet rs = bllDG.layDS();

rs.last();

int row = rs.getRow();

if (row == 0) {

STT = 1;

} else {

rs.beforeFirst();

while (rs.next()) {

STT = Integer.parseInt(rs.getObject(1).toString().substring(2)) + 1;

}

}

String ma = String.format("%03d", STT);

return "DG" + ma;

}

public String layMaTheTV() throws Exception {

STT = 0;

ResultSet rs = bllDG.layDS();

rs.last();

int row = rs.getRow();

if (row == 0) {

STT = 1;

} else {

rs.beforeFirst();

while (rs.next()) {

STT = Integer.parseInt(rs.getObject(1).toString().substring(2)) + 1;

}

}

String ma = String.format("%03d", STT);

return "TTV" + ma;

}

### Mượn sách

#### BLL

public class BLL\_MuonSach {

DAL\_MuonSach dal\_muonsach;

public BLL\_MuonSach() throws Exception {

this.dal\_muonsach = new DAL\_MuonSach();

}

public ResultSet LayDSNhanVien() throws Exception {

return dal\_muonsach.layDSNhanVien();

}

public ResultSet LayDSPM() throws Exception {

return dal\_muonsach.LayDSPhieu();

}

public boolean DuocMuon(String mathe) throws SQLException {

ResultSet rs = dal\_muonsach.GetStatus(mathe);

int dangMuon = 0, duocMuon = 0;

while (rs.next()) {

dangMuon = Integer.parseInt(rs.getString("sum"));

duocMuon = Integer.parseInt(rs.getString("quantity"));

}

if (dangMuon >= duocMuon) {

return false;

}

return true;

}

public boolean HetHan(String mathe) throws SQLException {

ResultSet rs = dal\_muonsach.GetStatus(mathe);

Date ngayhethan = new Date();

while (rs.next()) {

ngayhethan = rs.getDate("hethan");

}

Date now = new Date();

return ngayhethan.compareTo(now) > 0;

}

public boolean GetQtyBook(String masach) throws SQLException {

ResultSet rs = dal\_muonsach.GetQtyBook(masach);

int book = 0;

while (rs.next()) {

book = rs.getInt("reality\_quantity");

}

Date now = new Date();

return book > 0;

}

public void SetTable(JTable tb) throws Exception {

ResultSet rs = LayDSPM();

while (rs.next()) {

String maphieu = rs.getString("id\_rental\_detail");

String mathe = rs.getString("id\_lib\_card");

String manv = rs.getString("id\_staff");

String masach = rs.getString("id\_book");

String ngaymuon = rs.getString("borrowed\_date");

String ngaytra = rs.getString("return\_date");

String tinhtrang = rs.getString("status");

String trangthai = rs.getString("status\_borow");

String[] row = {maphieu, mathe, manv, masach, ngaymuon, ngaytra, tinhtrang, trangthai};

DefaultTableModel tableModel = (DefaultTableModel) tb.getModel();

tableModel.addRow(row);

}

}

public boolean ChoMuon(ET\_MuonSach phieu) throws SQLException {

return dal\_muonsach.ChoMuonSach(phieu);

}

public String GetTenDG(String madg) throws SQLException, Exception {

ResultSet rs = dal\_muonsach.LayDG(madg);

String ten = "";

while (rs.next()) {

ten = rs.getString("TenDG");

}

return ten;

}

public HashMap hm\_NV() throws Exception {

ResultSet rs = LayDSNhanVien();

HashMap<String, String> hm\_nv = new HashMap<String, String>();

try {

while (rs.next()) {

hm\_nv.put(rs.getString("MaNV"), rs.getString("TenNV"));

}

} catch (Exception e) {

}

return hm\_nv;

}

#### DAL

public boolean ChoMuonSach(ET\_MuonSach phieu) throws SQLException {

String sql = "INSERT INTO `qltv`.`rental\_detail` (`borrowed\_date`, `return\_date`, `id\_book`, `id\_lib\_card`, `id\_staff`, `status`) VALUES ( ? , ? , ? , ? , ? , ?);";

try {

PreparedStatement pst = conn.prepareStatement(sql);

pst.setDate(1, phieu.getNgaymuon());

pst.setDate(2, phieu.getNgaytra());

pst.setString(3, phieu.getMasach());

pst.setString(4, phieu.getMathe());

pst.setString(5, phieu.getManv());

pst.setString(6, phieu.getTinhTrang());

if (pst.executeUpdate() > 0) {

String sql2 = "update books set reality\_quantity = reality\_quantity - 1 where id\_book = ? ;";

try

{

PreparedStatement pst1 = conn.prepareStatement(sql2);

pst1.setString(1, phieu.getMasach());

pst1.executeUpdate();

}

catch (SQLException ex)

{

JOptionPane.showMessageDialog(null, ex);

} return true;

}

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, e);

}

return false;

} public ResultSet GetStatus(String mathe) {

ResultSet rs = null;

PreparedStatement pst = null;

String sql = "select distinct count(dt.status\_borow) as 'sum', c.expiration\_date 'hethan', rt.quantity from member\_card c \n"

+ "join readers rd on c.id\_reader = rd.id\_reader \n"

+ "join reader\_type rt on rd.reader\_type = rt.type\_id \n"

+ "join rental\_detail dt on dt.id\_lib\_card = c.id\_lib\_card\n"

+ "where c.id\_lib\_card = ? and dt.status\_borow like 'Đang mượn'";

try {

pst = conn.prepareStatement(sql);

pst.setString(1, mathe);

rs = pst.executeQuery();

} catch (SQLException e) {

}

return rs;

}

public ResultSet GetQtyBook(String masach) {

ResultSet rs = null;

PreparedStatement pst = null;

String sql = "select reality\_quantity from books where id\_book = ? ;";

try {

pst = conn.prepareStatement(sql);

pst.setString(1, masach);

rs = pst.executeQuery();

} catch (SQLException e) {

}

return rs;

}

public ResultSet LayDG(String madg) throws SQLException, Exception {

ResultSet rs = null;

PreparedStatement pst = null;

String sql = "select concat(firstname,' ',lastname) as 'TenDG' from readers rd join member\_card c on rd.id\_reader = c.id\_reader where c.id\_lib\_card = ?";

try {

pst = conn.prepareStatement(sql);

pst.setString(1, madg);

rs = pst.executeQuery();

} catch (SQLException e) {

}

return rs;

}

public ResultSet LayDSPhieu() throws SQLException, Exception {

Connection conn = null;

ResultSet rs = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "SELECT \* FROM qltv.rental\_detail order by id\_rental\_detail desc";

rs = st.executeQuery(sql);

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, e);

}

return rs;

}

public ResultSet layDSNhanVien() throws SQLException, Exception {

Connection conn = null;

ResultSet rs = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "Select id\_staff as `MaNV`, concat(lastname,' ',firstname,'-',id\_staff) as 'TenNV' from staffs order by id\_staff asc";

rs = st.executeQuery(sql);

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, e);

}

return rs;

}

#### ET

String masach; // mã sách

String mathe; // mã thẻ

String manv; // mã nhân viên

java.sql.Date ngaymuon; // ngày mượn

java.sql.Date ngaytra; // ngày trả

String tinhTrang; // tình trạng

public String getMasach() {

return masach;

}

public String getMathe() {

return mathe;

}

public String getManv() {

return manv;

}

public java.sql.Date getNgaymuon() {

return ngaymuon;

}

public java.sql.Date getNgaytra() {

return ngaytra;

}

public String getTinhTrang() {

return tinhTrang;

}

public ET\_MuonSach(String masach, String mathe, String manv, Date ngaymuon, int soNgayMuon, String tinhTrang) {

this.masach = masach;

this.mathe = mathe;

this.manv = manv;

this.ngaymuon = new java.sql.Date(ngaymuon.getTime());

Calendar c = Calendar.getInstance();

c.setTime(ngaymuon);

c.add(Calendar.DATE, soNgayMuon);

java.util.Date utilDate = c.getTime();

this.ngaytra = new java.sql.Date(utilDate.getTime());

this.tinhTrang = tinhTrang;

#### GUI

public void loadCBNV() {

cbNhanVien.removeAllItems();

Map<String, String> map = new TreeMap<>(hm\_nv);

for (Map.Entry<String, String> entry : map.entrySet()) {

Object value = entry.getValue();

cbNhanVien.addItem(value.toString());

}

}

public String getMaNV() {

for (Entry<String, String> entry : hm\_nv.entrySet()) {

if (entry.getValue().equals(cbNhanVien.getSelectedItem())) {

return entry.getKey();

}

}

return "";

}

public void LoadDSPM() throws Exception {

DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

dtm.setRowCount(0);

bll\_muonsach.SetTable(jTable1);

}

private void txtMaDocGiaFocusLost(java.awt.event.FocusEvent evt) {

try {

String text = bll\_muonsach.GetTenDG(txtMaDocGia.getText());

txtTenDocGia.setText(text);

if (text.equals("") && !txtMaDocGia.getText().equals("")) {

JOptionPane.showMessageDialog(null, "Mã thẻ không tồn tại", "Thông báo", JOptionPane.INFORMATION\_MESSAGE);

}

LoadDSPM();

} catch (Exception ex) {

}

}

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

}

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

try {

if (!bll\_muonsach.HetHan(txtMaDocGia.getText())) {

JOptionPane.showMessageDialog(null, "Thẻ đã hết hạn, vui lòng gian hạn thêm", "Thông báo", JOptionPane.INFORMATION\_MESSAGE);

return;

}

if (!bll\_muonsach.DuocMuon(txtMaDocGia.getText())) {

JOptionPane.showMessageDialog(null, "Mượn vượt quá mức !!!", "Thông báo", JOptionPane.INFORMATION\_MESSAGE);

return;

}

if (!bll\_muonsach.GetQtyBook(txtMasach.getText())) {

JOptionPane.showMessageDialog(null, "Sách đã hết !!!", "Thông báo", JOptionPane.INFORMATION\_MESSAGE);

return;

}

} catch (SQLException ex) {

Logger.getLogger(frmMuonSach.class.getName()).log(Level.SEVERE, null, ex);

}

ET\_MuonSach muonsach = new ET\_MuonSach(txtMasach.getText(), txtMaDocGia.getText(), getMaNV(), ngayMuonJDate.getDate(), Integer.parseInt((String) cbNgay.getSelectedItem()), txtTinhTrang.getText());

try {

bll\_muonsach.ChoMuon(muonsach);

clearTextFields(jPanel1);

JOptionPane.showMessageDialog(null, "Mượn thành công", "Thông báo", JOptionPane.INFORMATION\_MESSAGE);

LoadDSPM();

} catch (SQLException ex) {

JOptionPane.showConfirmDialog(null, jPanel1, "Mượn thất bại lỗi" + ex, JOptionPane.OK\_OPTION, JOptionPane.PLAIN\_MESSAGE);

try {

} catch (Exception ex1) {

Logger.getLogger(frmMuonSach.class.getName()).log(Level.SEVERE, null, ex1);

}

try {

LoadDSPM();

} catch (Exception ex1) {

Logger.getLogger(frmMuonSach.class.getName()).log(Level.SEVERE, null, ex1);

}

} catch (Exception ex) {

Logger.getLogger(frmMuonSach.class.getName()).log(Level.SEVERE, null, ex);

}

}

### Trả sách

#### BLL

private DAL\_TraSachh dal;

public BLL\_TraSachh() throws Exception {

dal = new DAL\_TraSachh();

}

public ResultSet layTraSach() throws SQLException {

return dal.layTraSach();

}

public boolean themTraSach(ET\_TraSachh et) throws SQLException {

return dal.themTraSach(et);

}

public ResultSet timTraSach(String ma) throws SQLException {

return dal.timTraSach(ma);

}

public ResultSet layDSNhanVien() throws Exception {

return dal.layDSNhanVien();

}

public String layTenDocGia(String madg) throws SQLException, Exception {

ResultSet rs = dal.layTenDocGia(madg);

String ten = "";

while (rs.next()) {

ten = rs.getString("TenDG");

}

return ten;

}

public HashMap hm\_NV() throws Exception {

ResultSet rs = layDSNhanVien();

HashMap<String, String> hm\_nv = new HashMap<String, String>();

try {

while (rs.next()) {

hm\_nv.put(rs.getString("MaNV"), rs.getString("TenNV"));

}

} catch (Exception e) {

}

return hm\_nv;

}

#### DAL

public ResultSet layTraSach() throws SQLException {

Statement st = conn.createStatement();

String sql = "select \* from return\_detail";

return st.executeQuery(sql);

}

public Boolean themTraSach(ET\_TraSachh et) throws SQLException {

Statement st = conn.createStatement();

String sql = "insert into return\_detail(id\_rental\_detail, expected\_return\_date, actually\_return\_date, id\_book, reader, id\_lib\_card, days\_overdue, id\_staff, status)"

+ "values(?,?,?,?,?,?,?,?,?)";

try {

PreparedStatement pst = conn.prepareStatement(sql);

pst.setInt(1, et.getMaMuonSach());

pst.setDate(2, et.getNgayDuTra());

pst.setDate(3, et.getNgayThucTra());

pst.setString(4, et.getMaSach());

pst.setString(5, et.getTenDocGia());

pst.setString(6, et.getMaThuVien());

pst.setInt(7, et.getSoNgayQuaHan());

//pst.setInt(8, et.getTienPhat());

pst.setString(8, et.getMaNhanVien());

pst.setString(9, et.getTinhTrangSach());

if (pst.executeUpdate() > 0) {

updateQuty(et.getMaSach());

updatePhieu(et.getMaMuonSach());

}

} catch (SQLException ex) {

}

return false;

}

void updateQuty(String masach) {

String sql = "update books set reality\_quantity = reality\_quantity + 1 where id\_book = ? ;";

try {

PreparedStatement pst1 = conn.prepareStatement(sql);

pst1.setString(1, masach);

pst1.executeUpdate();

} catch (Exception e) {

e.printStackTrace();

}

}

void updatePhieu(int maphieu) {

String sql = "update rental\_detail set status\_borow = 'Đã trả' where id\_rental\_detail = ? ;";

try {

PreparedStatement pst1 = conn.prepareStatement(sql);

pst1.setInt(1, maphieu);

pst1.executeUpdate();

} catch (Exception e) {

e.printStackTrace();

}

}

public ResultSet timTraSach(String ma) throws SQLException {

Statement st = conn.createStatement();

String sql = "select \* from rental\_detail where id\_rental\_detail = '" + ma + "' and status\_borow = 'Đang mượn'";

ResultSet rs = null;

try {

PreparedStatement pst = conn.prepareStatement(sql);

rs = st.executeQuery(sql);

} catch (Exception e) {

e.printStackTrace();

}

return rs;

}

public ResultSet layDSNhanVien() throws SQLException, Exception {

Connection conn = null;

ResultSet rs = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "Select id\_staff as `MaNV`, concat(lastname,' ',firstname,'-',id\_staff) as 'TenNV' from staffs order by id\_staff asc";

rs = st.executeQuery(sql);

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, e);

}

return rs;

}

public ResultSet layTenDocGia(String madg) throws SQLException, Exception {

ResultSet rs = null;

PreparedStatement pst = null;

String sql = "select concat(firstname,' ',lastname) as 'TenDG' from readers join member\_card on readers.id\_reader = member\_card.id\_reader where member\_card.id\_lib\_card = ?";

try {

pst = conn.prepareStatement(sql);

pst.setString(1, madg);

rs = pst.executeQuery();

} catch (SQLException e) {

}

return rs;

}

#### ET

private int maMuonSach; // mã sách mượn

private java.sql.Date ngayDuTra; // ngày dự trả

private java.sql.Date ngayThucTra; // ngày thực trả

private String maSach; // mã sách

private String tenDocGia; // tên độc giả

private String maThuVien; // mã thư viện

private int soNgayQuaHan; // số ngày quá hạn

private String maNhanVien; // mã nhân viên

private String tinhTrangSach; // tình trạng sách

public ET\_TraSachh(int maMuonSach, Date ngayDuTra, Date ngayThucTra, String maSach, String tenDocGia, String maThuVien, int soNgayQuaHan, String maNhanVien, String tinhTrangSach) {

this.maMuonSach = maMuonSach;

this.ngayDuTra = ngayDuTra;

this.ngayThucTra = ngayThucTra;

this.maSach = maSach;

this.tenDocGia = tenDocGia;

this.maThuVien = maThuVien;

this.soNgayQuaHan = soNgayQuaHan;

this.maNhanVien = maNhanVien;

this.tinhTrangSach = tinhTrangSach;

}

#### GUI

public void hienThi() {

DefaultTableModel model = (DefaultTableModel) tblTraSach.getModel();

model.setColumnCount(0);

model.setRowCount(0);

model.addColumn("Mã trả sách");

model.addColumn("Mã mượn sách");

model.addColumn("Ngày dự trả");

model.addColumn("Ngày thực trả");

model.addColumn("Mã sách");

model.addColumn("Tên đọc giả");

model.addColumn("Mã thẻ thư viện");

model.addColumn("Số ngày quá hạn");

model.addColumn("Tiền phạt");

model.addColumn("Mã nhân viên");

model.addColumn("Tình trạng sách");

tblTraSach.setModel(model);

}

public void hienThiTraSach() throws SQLException {

DefaultTableModel model = (DefaultTableModel) tblTraSach.getModel();

model.setRowCount(0);

ResultSet rs = bll.layTraSach();

while (rs.next()) {

Object[] col = new Object[10];

for (int i = 1; i <= 10; i++) {

col[i - 1] = rs.getObject(i);

}

model.addRow(col);

}

}

public void soNgayQuaHan() {

SimpleDateFormat fr = new SimpleDateFormat("yyyy-MM-dd");

Date ngayDuKienTra = null;

Date ngayTra = null;

try {

if (kq) {

ngayTra = fr.parse(fr.format(dtpNgayThucTra.getDate()));

ngayDuKienTra = fr.parse(fr.format(dtpNgayDuKienTra.getDate()));

int datediff = (int) (ngayTra.getTime() - ngayDuKienTra.getTime());

int days = datediff / (60 \* 60 \* 1000) / 24;

if (days < 0) {

txtSoNgayQuaHan.setText("0");

} else {

txtSoNgayQuaHan.setText(Long.toString(days));

}

} else {

txtSoNgayQuaHan.setText("0");

}

} catch (Exception e) {

e.printStackTrace();

}

}

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

String maMuon = txtMaMuonSach.getText();

java.sql.Date ngayDuTra = new java.sql.Date(dtpNgayDuKienTra.getDate().getTime());

java.sql.Date ngayThucTra = new java.sql.Date(dtpNgayThucTra.getDate().getTime());

String maSach = txtMaSach.getText();

String tenDocGia = txtDocGia.getText();

String maThuVien = txtMaTheThuVien.getText();

String soNgayQuaHan = txtSoNgayQuaHan.getText();

String maNhanVien = getMaNV();

String tinhTrangSach = txtaTinhTrangSach.getText();

ET\_TraSachh et = new ET\_TraSachh(Integer.parseInt(maMuon), ngayDuTra, ngayThucTra, maSach, tenDocGia, maThuVien, Integer.parseInt(soNgayQuaHan), maNhanVien, tinhTrangSach);

try {

if (bll.themTraSach(et)) {

JOptionPane.showMessageDialog(rootPane, "Thêm thành công");

}

} catch (SQLException ex) {

JOptionPane.showMessageDialog(rootPane, "Thêm không thành công");

}

try {

hienThiTraSach();

} catch (SQLException ex) {

ex.printStackTrace();

}

}

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

try {

ResultSet rs = bll.timTraSach(txtMaMuonSach.getText());

if (rs.next() == false) {

JOptionPane.showMessageDialog(this, "Không tìm thấy mã mượn sách");

txtMaMuonSach.setText("");

txtMaSach.setText("");

txtDocGia.setText("");

txtaTinhTrangSach.setText("");

txtMaTheThuVien.setText("");

txtSoNgayQuaHan.setText("");

kq = false;

dtpNgayDuKienTra.setDate(null);

} else {

txtMaMuonSach.setText(rs.getString("id\_rental\_detail"));

dtpNgayDuKienTra.setDate(rs.getDate("return\_date"));

txtMaSach.setText(rs.getString("id\_book"));

txtMaTheThuVien.setText(rs.getString("id\_lib\_card"));

kq = true;

try {

String rs1 = bll.layTenDocGia(txtMaTheThuVien.getText());

txtDocGia.setText(rs1);

} catch (Exception ex) {

ex.printStackTrace();

}

txtaTinhTrangSach.setText(rs.getString("status"));

}

} catch (Exception e) {

}

soNgayQuaHan();

### Tìm sách

#### BLL

public BLL\_TimSach() throws Exception {

this.dal\_timsach = new DAL\_TimSach();

}

public void setCB(JComboBox cbb, HashMap<String, String> hm) {

cbb.removeAllItems();

for (Map.Entry<String, String> entry : hm.entrySet()) {

Object key = entry.getKey();

cbb.addItem(key.toString());

}

}

public void TimSach(String searchBy, String kw, JTable tb) throws Exception {

ResultSet rs = dal\_timsach.layDSSach(searchBy, kw);

while (rs.next()) {

String id = rs.getString("id\_book");

String tt = rs.getString("title");

String ctg = rs.getString("ctg");

String au = rs.getString("author\_name");

String p = rs.getString("p\_name");

String lan = rs.getString("lan\_name");

String total = rs.getString("total\_page");

String price = rs.getString("price");

String location = rs.getString("location");

String[] row = {id, tt, ctg, au, p, lan, total, price, location};

DefaultTableModel tableModel = (DefaultTableModel) tb.getModel();

tableModel.addRow(row);

}

}

#### DAL

public ResultSet layDSSach(String searchBy, String kw) throws SQLException, Exception {

Connection conn = null;

ResultSet rs = null;

PreparedStatement pst = null;

String sql = "select b.id\_book, b.title,ctg.name 'ctg',a.author\_name, p.name 'p\_name', lan.name 'lan\_name', total\_page, price,location from books b\n"

+ "join authors a on b.id\_author = a.id\_author\n"

+ "join categories ctg on ctg.id\_category = b.id\_category\n"

+ "join languages lan on lan.id\_language = b.id\_language\n"

+ "join publishers p on p.id\_publisher = b.id\_publisher "

+ searchBy;

try {

conn = DatabaseUtil.getConnection();

pst = conn.prepareStatement(sql);

pst.setString(1, kw);

rs = pst.executeQuery();

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, e);

}

return rs;

}

#### GUI

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

DefaultTableModel dtm = (DefaultTableModel) TbSach.getModel();

dtm.setRowCount(0);

try {

bll\_tim.TimSach(hm.get(cbtimkiem.getSelectedItem()), txtNhapTT.getText(), TbSach);

} catch (Exception ex) {

Logger.getLogger(frmTimKiem.class.getName()).log(Level.SEVERE, null, ex);

}

}

void LoadCB() {

hm.put("Mã sách", "where b.id\_book = ? ");

hm.put("Tên sách", "where b.title like concat('%', ? , '%')");

hm.put("Thể loại", "where ctg.name Like concat('%', ? , '%')");

hm.put("Tác giả", "where a.author\_name like concat('%', ? , '%')");

hm.put("Nhà XB", "where p.name like concat('%', ? , '%')");

hm.put("Ngôn ngữ", "where lan.name like concat('%', ? , '%')");

bll\_tim.setCB(cbtimkiem, hm);

}

### Cập nhập sách

#### BLL

DAL\_CapNhatSach dal;

public Bll\_CapNhatSach() {

dal = new DAL\_CapNhatSach();

}

public ResultSet layDSSach() throws SQLException {

return dal.layDSSach();

}

public Boolean suaSach(ET\_Sach et) throws SQLException {

return dal.suaSach(et);

}

#### DAL

public ResultSet layDSSach() throws SQLException {

ResultSet rs = null;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "SELECT `id\_book`, `title` ,`id\_category`, `id\_language`, `id\_author`, `id\_publisher`,`release\_year` , `location` FROM `books`";

rs = st.executeQuery(sql);

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return rs;

}

public Boolean suaSach(ET\_Sach et) throws SQLException {

Boolean kq = false;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "update books "

+ "set title = '" + et.getTenSach()+ "', "

+ "id\_category = '" + et.getTheLoai()+ "',"

+ "id\_language = '" + et.getNgonNgu()+ "',"

+ "id\_author = '" + et.getMaTD()+ "',"

+ "id\_publisher = '" + et.getMaNXB()+ "',"

+ "release\_year = " + et.getNamXB()+ ","

+ "location = '" + et.getViTri()+ "'"

+ "where id\_book ='" + et.getMaSach()+ "'";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return kq;

}

#### GUI

public void hienThiSach() throws SQLException, Exception {

ResultSet rs = bll.layDSSach();

DefaultTableModel model = (DefaultTableModel) tbSach.getModel();

model.setRowCount(0);

while (rs.next()) {

Object[] col = new Object[8];

for (int i = 1; i <= 8; i++) {

if(i==3){

col[i - 1] =bllTL.layTen(rs.getObject(i).toString());

}else if(i==4){

col[i - 1] =bllNN.layTen(rs.getObject(i).toString());

}else if(i==5){

col[i - 1] =bllTG.layTen(rs.getObject(i).toString());

}else if(i==6){

col[i - 1] =bllNXB.layTen(rs.getObject(i).toString());

}else

col[i - 1] = rs.getObject(i);

}

model.addRow(col);

}

private void tbSachMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

int row = tbSach.getSelectedRow();

DefaultTableModel model = (DefaultTableModel) tbSach.getModel();

txtMasach.setText(model.getValueAt(row, 0).toString());

txtTensach.setText(model.getValueAt(row, 1).toString());

try {

cbTheLoai.setSelectedItem(model.getValueAt(row, 2).toString());

} catch (Exception ex) {

ex.printStackTrace();

}

try {

cbNgonNgu.setSelectedItem(model.getValueAt(row, 3).toString());

} catch (Exception ex) {

ex.printStackTrace();

}

try {

cbTacGia.setSelectedItem(model.getValueAt(row, 4).toString());

} catch (Exception ex) {

ex.printStackTrace();

}

try {

cbNXB.setSelectedItem(model.getValueAt(row, 5).toString());

} catch (Exception ex) {

ex.printStackTrace();

}

txtNamXB.setText(model.getValueAt(row, 6).toString());

txtViTri.setText(model.getValueAt(row, 7).toString());

}

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

// TODO add your handling code here:

int row = tbSach.getSelectedRow();

if (row >= 0) {

String maSach = tbSach.getValueAt(row, 0).toString();

String tenSach = txtTensach.getText();

String theLoai = null;

try {

theLoai = bllTL.layMa(cbTheLoai.getSelectedItem().toString());

} catch (Exception ex) {

ex.printStackTrace();

}

String ngonNgu = null;

try {

ngonNgu = bllNN.layMa(cbNgonNgu.getSelectedItem().toString());

} catch (Exception ex) {

ex.printStackTrace();

}

String tacGia = null;

try {

tacGia = bllTG.layMa(cbTacGia.getSelectedItem().toString());

} catch (Exception ex) {

ex.printStackTrace();

}

String NXB = null;

try {

NXB = bllNXB.layMa(cbNXB.getSelectedItem().toString());;

} catch (Exception ex) {

ex.printStackTrace();

}

int namXB = Integer.parseInt(txtNamXB.getText());

String viTri = txtViTri.getText();

ET\_Sach et;

if (maSach.compareTo("") == 0

|| tenSach.compareTo("") == 0

|| viTri.compareTo("") == 0

|| (txtNamXB.getText()).compareTo("") == 0) {

JOptionPane.showMessageDialog(null, "Vui lòng điền đầy đủ thông tin");

} else {

try {

et = new ET\_Sach(maSach, tenSach, theLoai, ngonNgu, tacGia, NXB, namXB , viTri);

if (bll.suaSach(et)) {

JOptionPane.showMessageDialog(rootPane, "Thành công");

} else {

JOptionPane.showMessageDialog(rootPane, "Không thành công");

}

hienThiSach();

} catch (SQLException ex) {

Logger.getLogger(frmNhanVien.class.getName()).log(Level.SEVERE, null, ex);

} catch (Exception ex) {

Logger.getLogger(frmCapNhatSach.class.getName()).log(Level.SEVERE, null, ex);

}

}

} else {

JOptionPane.showMessageDialog(null, "Vui lòng chọn sách");

}

}

### Nhà cung cấp

#### BLL

public BLL\_NhaCungCap() {

dal = new DAL\_NhaCungCap();

}

public String layMa(String ten) throws Exception {

return dal.layMaNCC(ten);

}

public Boolean themNCC(ET\_NhaCungCap et) throws SQLException, Exception {

return dal.themNCC(et);

}

public Boolean xoaNCC(String maNCC) throws SQLException {

return dal.xoaNCC(maNCC);

}

public Boolean suaNCC(ET\_NhaCungCap et) throws SQLException {

return dal.suaNCC(et);

}

public Boolean checkTT(String ma) throws SQLException {

return dal.checkTonTai(ma);

}

public ResultSet layNCC() throws SQLException {

return dal.layNCC();

}

public String layTenNCC(String maNCC) throws SQLException, Exception {

return dal.layTenNCC(maNCC);

}

#### DAL

public ResultSet layNCC() throws SQLException {

ResultSet rs = null;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "select \* from suppliers";

rs = st.executeQuery(sql);

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return rs;

}

public Boolean themNCC(ET\_NhaCungCap et) throws SQLException, Exception {

Boolean kq = false;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "INSERT INTO suppliers(id\_supplier, supplier\_name,address,phone\_number)"

+ "values ('" + et.getMaNCC()+ "','"+ et.getTenNCC()+ "','"+ et.getDiaChiNCC()+ "','"+ et.getSdtNCC()+ "')";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return kq;

}

public Boolean checkTonTai(String ma) throws SQLException {

Connection conn = null;;

ResultSet rs = null;

int row = 0;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

String sql = "Select \* from suppliers where id\_supplier = '" + ma + "'";

rs = st.executeQuery(sql);

rs.last();

row = rs.getRow();

rs.beforeFirst();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return (row > 0);

public Boolean xoaNCC(String maNCC) throws SQLException {

Boolean kq = false;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "delete from suppliers where id\_supplier = '" + maNCC+"'";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return kq;

}

public Boolean suaNCC(ET\_NhaCungCap et) throws SQLException {

Boolean kq = false;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement();

String sql = "update suppliers "

+ "set supplier\_name = '" + et.getTenNCC()+ "', "

+ "address = '" + et.getDiaChiNCC()+ "',"

+ "phone\_number = '" + et.getSdtNCC() + "' "

+ "where id\_supplier = '" + et.getMaNCC()+ "'";

if (st.executeUpdate(sql) > 0) {

kq = true;

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

return kq;

}

public String layMaNCC(String tenNCC) throws Exception {

Connection conn = null;

conn = DatabaseUtil.getConnection();

ResultSet rs = null;

Statement st = conn.createStatement();

String sql = "select \* from suppliers where supplier\_name ='" + tenNCC + "' limit 1";

rs = st.executeQuery(sql);

String ma = "";

while (rs.next()) {

ma = (String) rs.getObject(1);

break;

}

return ma;

}

public String layTenNCC(String maNCC) throws Exception {

Connection conn = null;

conn = DatabaseUtil.getConnection();

ResultSet rs = null;

Statement st = conn.createStatement();

String sql = "select \* from suppliers where id\_supplier ='" + maNCC + "' limit 1";

rs = st.executeQuery(sql);

String ma = "";

while (rs.next()) {

ma = (String) rs.getObject(2);

break;

}

return ma;

}

#### ET

public ET\_NhaCungCap(String maNCC, String tenNCC, String diaChiNCC, String sdtNCC) {

this.maNCC = maNCC;

this.tenNCC = tenNCC;

this.diaChiNCC = diaChiNCC;

this.sdtNCC = sdtNCC;

}

public String getMaNCC() {

return maNCC;

}

public void setMaNCC(String maNCC) {

this.maNCC = maNCC;

}

public String getTenNCC() {

return tenNCC;

}

public void setTenNCC(String tenNCC) {

this.tenNCC = tenNCC;

}

public String getDiaChiNCC() {

return diaChiNCC;

}

public void setDiaChiNCC(String diaChiNCC) {

this.diaChiNCC = diaChiNCC;

}

public String getSdtNCC() {

return sdtNCC;

}

public void setSdtNCC(String sdtNCC) {

this.sdtNCC = sdtNCC;

}

#### GUI

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

String maNCC = txtMaNCC.getText();

String tenNCC = txtTenNCC.getText();

String diaChi = txtDiaChi.getText();

String soDT = txtSDT.getText();

if (maNCC.compareTo("") == 0

|| tenNCC.compareTo("") == 0

|| diaChi.compareTo("") == 0

|| soDT.compareTo("") == 0) {

JOptionPane.showMessageDialog(null, "Vui lòng điền đầy đủ thông tin");

} else {

ET\_NhaCungCap et = new ET\_NhaCungCap(maNCC, tenNCC, diaChi, soDT);

try {

if (bll.checkTT(maNCC)) {

JOptionPane.showMessageDialog(null, "Đã tồn tại Nha Cung Cap");

} else {

if (bll.themNCC(et)) {

JOptionPane.showMessageDialog(rootPane, "Thành công");

STT++;

layMa();

hienThiNCC();

} else {

JOptionPane.showMessageDialog(rootPane, "Không thành công");

}

}

} catch (SQLException ex) {

ex.printStackTrace();

} catch (Exception ex) {

ex.printStackTrace();

}

}

}

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

int dong = tbNCC.getSelectedRow();

String maNCC = tbNCC.getValueAt(dong, 0).toString();

String tenNCC = txtTenNCC.getText();

String diaChi = txtDiaChi.getText();

String soDT = txtSDT.getText();

ET\_NhaCungCap et;

if (dong != -1) {

if (maNCC.compareTo("") == 0

|| tenNCC.compareTo("") == 0

|| diaChi.compareTo("") == 0

|| soDT.compareTo("") == 0) {

JOptionPane.showMessageDialog(null, "Vui lòng điền đầy đủ thông tin");

} else {

try {

et = new ET\_NhaCungCap(maNCC, tenNCC, diaChi, soDT);

if (bll.suaNCC(et)) {

JOptionPane.showMessageDialog(rootPane, "Thành công");

layMa();

hienThiNCC();

} else {

JOptionPane.showMessageDialog(rootPane, "Không thành công");

}

} catch (Exception ex) {

ex.printStackTrace();

}

}

} else {

JOptionPane.showMessageDialog(null, "Vui lòng chọn nhà cung cấp");

}

}

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

int row = tbNCC.getSelectedRow();

if (row != -1) {

try {

String maNCC = txtMaNCC.getText();

String tenNCC = txtTenNCC.getText().compareTo("") > 0 ? txtTenNCC.getText() : "";

String diaChi = txtDiaChi.getText().compareTo("") > 0 ? txtDiaChi.getText() : "";

String soDT = txtSDT.getText().compareTo("") > 0 ? txtSDT.getText() : "";

if (maNCC != "" && tenNCC != "" && diaChi != "" && soDT != "") {

ET\_NhaCungCap et = new ET\_NhaCungCap(maNCC, tenNCC, diaChi, soDT);

if (bll.xoaNCC(maNCC)) {

JOptionPane.showMessageDialog(rootPane, "Thành công");

layMa();

hienThiNCC();

} else {

JOptionPane.showMessageDialog(rootPane, "Không thành công");

}

}

} catch (Exception ex) {

ex.printStackTrace();

}

} else {

JOptionPane.showMessageDialog(null, "Vui lòng chọn nhà cung cấp");

}

private void tbNCCMouseClicked(java.awt.event.MouseEvent evt) {

int row = tbNCC.getSelectedRow();

DefaultTableModel model = (DefaultTableModel) tbNCC.getModel();

txtMaNCC.setText(model.getValueAt(row, 0).toString());

txtTenNCC.setText(model.getValueAt(row, 1).toString());

txtDiaChi.setText(model.getValueAt(row, 2).toString());

txtSDT.setText(model.getValueAt(row, 3).toString());

}

### Thống kê

#### BLL

private DAL\_ThongKe dal;

public BLL\_ThongKe() throws Exception {

dal = new DAL\_ThongKe();

}

public ResultSet muonNhieu() throws SQLException {

return dal.muonNhieu();

}

public ResultSet muonNhieuNhatThang() throws SQLException {

return dal.muonNhieuNhatThang();

}

public ResultSet docGiaChuaTraSach() throws SQLException {

return dal.docGiaChuaTraSach();

}

public ResultSet docGiaQuaHanTra() throws SQLException {

return dal.docGiaQuaHanTra();

}

#### DAL

public ResultSet muonNhieu() throws SQLException {

ResultSet rs = null;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

String sql = "SELECT COUNT(rental\_detail.id\_book) as 'SoLanMuon', books.id\_book , books.title, categories.name, languages.name "

+ "FROM books, categories,languages, rental\_detail "

+ "WHERE books.id\_category = categories.id\_category AND books.id\_language = languages.id\_language AND books.id\_book = rental\_detail.id\_book "

+ "GROUP BY rental\_detail.id\_book "

+ "ORDER BY COUNT(rental\_detail.id\_book) DESC LIMIT 5";

rs = st.executeQuery(sql);

} catch (Exception ex) {

Logger.getLogger(DAL\_ThongKe.class.getName()).log(Level.SEVERE, null, ex);

}

return rs;

}

public ResultSet muonNhieuNhatThang() throws SQLException {

ResultSet rs = null;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

String sql = "SELECT MONTH(rental\_detail.borrowed\_date) AS 'Thang', COUNT(rental\_detail.id\_book) AS 'SoLanMuon' FROM books, categories,languages, rental\_detail "

+ "WHERE books.id\_category = categories.id\_category AND books.id\_language = languages.id\_language "

+ "AND books.id\_book = rental\_detail.id\_book "

+ "GROUP BY MONTH(rental\_detail.borrowed\_date) ";

rs = st.executeQuery(sql);

} catch (Exception ex) {

Logger.getLogger(DAL\_ThongKe.class.getName()).log(Level.SEVERE, null, ex);

}

return rs;

}

public ResultSet docGiaChuaTraSach() throws SQLException {

ResultSet rs = null;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

String sql = "SELECT readers.id\_reader,concat(lastname,' ',firstname) as 'TenDG',books.title, rental\_detail.borrowed\_date "

+ "FROM member\_card , readers , rental\_detail, books "

+ "WHERE member\_card.id\_reader = readers.id\_reader AND member\_card.id\_lib\_card = rental\_detail.id\_lib\_card "

+ "AND books.id\_book = rental\_detail.id\_book AND rental\_detail.status\_borow = 'Đang mượn'";

rs = st.executeQuery(sql);

} catch (Exception ex) {

Logger.getLogger(DAL\_ThongKe.class.getName()).log(Level.SEVERE, null, ex);

}

return rs;

}

public ResultSet docGiaQuaHanTra() throws SQLException {

ResultSet rs = null;

Connection conn = null;

try {

conn = DatabaseUtil.getConnection();

Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

String sql = "SELECT readers.id\_reader,concat(lastname,' ',firstname) as 'TenDG',books.title, rental\_detail.borrowed\_date, rental\_detail.return\_date, "

+ "DAY(rental\_detail.return\_date) - DAY(rental\_detail.borrowed\_date) AS 'SoNgayMuon', "

+ "DATEDIFF(NOW(),rental\_detail.return\_date) AS 'SoNgayQuaHan' FROM member\_card , readers , rental\_detail, books "

+ "WHERE member\_card.id\_reader = readers.id\_reader AND member\_card.id\_lib\_card = rental\_detail.id\_lib\_card "

+ "AND books.id\_book = rental\_detail.id\_book AND DATEDIFF(NOW(),rental\_detail.return\_date) > 0";

rs = st.executeQuery(sql);

} catch (Exception ex) {

Logger.getLogger(DAL\_ThongKe.class.getName()).log(Level.SEVERE, null, ex);

}

return rs;

}

#### GUI

private void cbbthongkeItemStateChanged(java.awt.event.ItemEvent evt) {

int index = cbbthongke.getSelectedIndex();

if (index == 0) {

try {

ResultSet rs = dal.muonNhieu();

DefaultTableModel model = (DefaultTableModel) tblThongKe.getModel();

model.setColumnCount(0);

model.addColumn("Mã sách");

model.addColumn("Tên sách");

model.addColumn("Thể loại");

model.addColumn("Ngôn ngữ");

model.addColumn("Số lượng");

model.setRowCount(0);

tblThongKe.setModel(model);

while (rs.next()) {

model.addRow(new Object[]{rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5), rs.getString(1)});

}

} catch (SQLException ex) {

Logger.getLogger(frmThongKe.class.getName()).log(Level.SEVERE, null, ex);

}

}

if (index == 1) {

try {

ResultSet rs = dal.muonNhieuNhatThang();

DefaultTableModel model = (DefaultTableModel) tblThongKe.getModel();

model.setColumnCount(0);

model.addColumn("Tháng");

model.addColumn("Số lượng sách");

model.setRowCount(0);

tblThongKe.setModel(model);

while (rs.next()) {

model.addRow(new Object[]{rs.getString(1), rs.getString(2)});

}

} catch (SQLException ex) {

Logger.getLogger(frmThongKe.class.getName()).log(Level.SEVERE, null, ex);

}

}

if (index == 2) {

try {

ResultSet rs = dal.docGiaChuaTraSach();

DefaultTableModel model = (DefaultTableModel) tblThongKe.getModel();

model.setColumnCount(0);

model.addColumn("Mã độc giả");

model.addColumn("Tên độc giả");

model.addColumn("Tên sách");

model.addColumn("Ngày mượn");

model.setRowCount(0);

tblThongKe.setModel(model);

while (rs.next()) {

model.addRow(new Object[]{rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4)});

}

} catch (SQLException ex) {

Logger.getLogger(frmThongKe.class.getName()).log(Level.SEVERE, null, ex);

}

}

if (index == 3) {

try {

ResultSet rs = dal.docGiaQuaHanTra();

DefaultTableModel model = (DefaultTableModel) tblThongKe.getModel();

model.setColumnCount(0);

model.addColumn("Mã độc giả");

model.addColumn("Tên độc giả");

model.addColumn("Tên sách");

model.addColumn("Ngày mượn");

model.addColumn("Ngày trả");

model.addColumn("Số ngày mượn");

model.addColumn("Số ngày quá hạn");

model.setRowCount(0);

tblThongKe.setModel(model);

while (rs.next()) {

model.addRow(new Object[]{rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5), rs.getString(6), rs.getString(7)});

}

} catch (SQLException ex) {

Logger.getLogger(frmThongKe.class.getName()).log(Level.SEVERE, null, ex);

}

}

}