

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**HCMUTE**

**BÁO CÁO ĐỒ ÁN**  
**NGÔN NGỮ LẬP TRÌNH TIỀN TIẾN**

**Mã học phần: ADPL331379**

**Đề tài: Xây dựng ứng dụng quản lý thư viện**

**Giảng viên hướng dẫn: ThS. Trần Văn Định**

**Sinh viên thực hiện:**

**Lê Văn Cường                      19110332**

**Trần Bảo Duy                      19110339**

**TP. Hồ Chí Minh, tháng 12 năm 2021**

## NHẬN XÉT CỦA GVHD

### 1. Về nội dung đề tài và khối lượng thực hiện:

.....

.....

.....

.....

### 2. Ưu điểm:

.....

.....

.....

.....

### 3. Khuyết điểm:

.....

.....

.....

.....

### 4. Đánh giá:

.....

.....

.....

.....

.....

.....

**GIẢNG VIÊN**

Trần Văn Định

## LỜI CẢM ƠN

Sau quá trình học tập và rèn luyện tại khoa Công nghệ thông tin trường Đại học Sư Phạm Kỹ Thuật, chúng em đã được trang bị các kiến thức cơ bản, các kỹ năng thực tế để có thể hoàn thành đồ án môn học của mình.

Chúng em xin gửi lời cảm ơn chân thành đến thầy Trần Văn Định giảng viên trường Đại học Sư phạm Kỹ thuật thành phố Hồ Chí Minh đã quan tâm hướng dẫn truyền đạt những kiến thức và kinh nghiệm cho chúng em trong thời gian học tập bộ môn Ngôn ngữ lập trình tiên tiến.

Trong quá trình làm đồ án không tránh khỏi được những sai sót, chúng em mong nhận được sự góp ý của quý thầy và các bạn để được hoàn thiện hơn.

TP.Hồ Chí Minh, tháng 12 năm 2021

## MỤC LỤC

<b>PHẦN 1: TÓM TẮT CƠ SỞ LÝ THUYẾT</b> .....	1
<b>1.1. Khái niệm về lập trình hướng đối tượng trong java</b> .....	1
1.1.1. Đối tượng (Object) .....	1
1.1.2. Lớp (Class).....	1
1.1.3. Kế thừa (Inheritance) .....	1
1.1.4. Đa hình (Polymorphism) .....	2
1.1.5. Trừu tượng (Abstraction) .....	2
1.1.6. Đóng gói (Encapsulation) .....	3
<b>1.2. Java Swing</b> .....	4
1.2.1. Lớp JFrame trong Java Swing .....	5
1.2.2. Lớp JPanel trong Java Swing .....	5
1.2.3. Xử lý sự kiện trong Swing.....	6
1.2.3.1. Lớp EventObject trong Java Swing.....	8
1.2.3.2. Lớp AWTEvent .....	8
1.2.3.3. Lớp ActionEvent.....	8
1.2.3.4. Lớp InputEvent .....	9
1.2.3.5. Lớp KeyEvent .....	9
1.2.3.6. Lớp MouseEvent.....	9
<b>1.3. Lý thuyết JDBC với MySQL</b> .....	10
<b>PHẦN 2: GIỚI THIỆU HỆ THỐNG</b> .....	13
2.1. Phần mềm quản lý thư viện là gì? .....	13
2.2. Ưu điểm của hệ thống quản lý thư viện .....	13
2.3. Chức năng của phần mềm quản lý thư viện .....	13
2.4. Lời kết .....	14
<b>PHẦN 3: THIẾT KẾ HỆ THỐNG</b> .....	15
3.1. Sơ đồ chức năng .....	15
3.2. Sơ đồ lớp.....	15
3.3. Thiết kế cơ sở dữ liệu .....	16
3.4. Thiết kế giao diện .....	17
3.4.1. Tính năng đăng nhập: .....	17

<b>3.4.2. Frame của Thủ thư</b> .....	19
<b>3.4.2.1. Sách</b> .....	19
<b>3.4.2.1.1. Đầu Sách</b> .....	19
<b>3.4.2.1.2. Nhà xuất bản</b> .....	19
<b>3.4.2.1.3. Sách</b> .....	20
<b>3.4.2.2. Độc giả</b> .....	20
<b>3.4.2.3. Mượn trả</b> .....	21
<b>3.4.2.4. Hoạt động</b> .....	21
<b>3.4.2.5. Tra cứu</b> .....	22
<b>3.4.3. Frame của Quản lý:</b> .....	22
<b>3.4.3.1. Nhân viên</b> .....	22
<b>3.4.3.1.1. Danh sách nhân viên</b> .....	23
<b>3.4.3.1.2. Thông tin cá nhân</b> .....	23
<b>3.4.3.2. Thống kê</b> .....	23
<b>3.4.3.3. Hoạt động</b> .....	24
<b>PHẦN 4. KẾT QUẢ</b> .....	25
<b>4.1. Kết quả chung</b> .....	25
<b>4.2. Hạn chế</b> .....	25
<b>4.3. Hướng cải tiến</b> .....	25
<b>5. TÀI LIỆU THAM KHẢO</b> .....	27

## PHẦN 1: TÓM TẮT CƠ SỞ LÝ THUYẾT

### 1.1. Khái niệm về lập trình hướng đối tượng trong java

**Lập trình hướng đối tượng (OOP-Object-Oriented Programming)** là một phương pháp hay mô hình giúp tăng năng suất, đơn giản hóa việc bảo trì, dễ dàng mở rộng trong thiết kế phần mềm bởi việc cung cấp một vài khái niệm như:

- Đối tượng (Object)
- Lớp (Class)
- Kế thừa (Inheritance)
- Đa hình (Polymorphism)
- Trừu tượng (Abstraction)
- Đóng gói (Encapsulation)

#### 1.1.1. Đối tượng (Object)

Tất cả những thực thể có trạng thái và hành vi được biết đến như là một đối tượng. Ví dụ: bàn, ghế, bút chì, xe đạp, ô tô...

#### 1.1.2. Lớp (Class)

Tập hợp các đối tượng được gọi là lớp.

#### 1.1.3. Kế thừa (Inheritance)

Khi một đối tượng được truyền lại tất cả các thuộc tính và phương thức của đối tượng cha được gọi là kế thừa. Kế thừa giúp tái sử dụng lại mã nguồn. Nó được sử dụng cho đa hình lúc runtime.

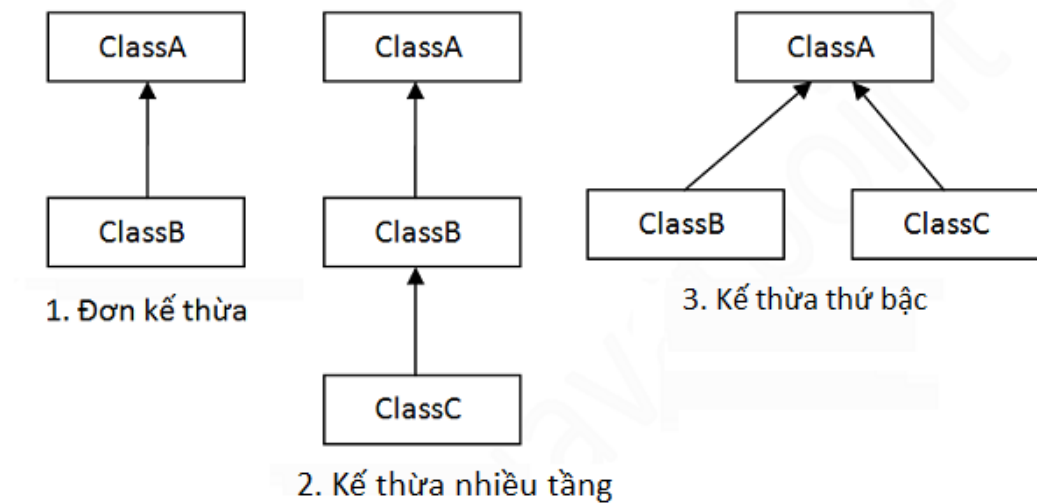
**Kế thừa trong java** là sự liên quan giữa hai class với nhau, trong đó có class cha (superclass) và class con (subclass). Khi kế thừa class con được hưởng tất cả các phương thức và thuộc tính của class cha. Tuy nhiên, nó chỉ được truy cập các thành viên public và protected của class cha. Nó không được phép truy cập đến thành viên private của class cha.

Ví dụ:

```
class Subclass-name extends Superclass-name {  
    //methods and fields
```

}

Các kiểu kế thừa trong java



#### 1.1.4. Đa hình (Polymorphism)

Khi một nhiệm vụ được thực hiện bởi nhiều cách khác nhau, tính chất này được gọi là đa hình. Ví dụ có nhiều các đề thuyết phục các khách hàng khác nhau, để vẽ một cái gì đó như hình tròn, hình chữ nhật, ...

Trong java, để áp dụng tính đa hình chúng ta sử dụng phương thức overloading hoặc overriding.

**Đa hình trong java (Polymorphism)** là một khái niệm mà chúng ta có thể thực hiện một hành động bằng nhiều cách khác nhau. Polymorphism được cấu tạo từ 2 từ Hy Lạp: poly và morphs. Trong đó "poly" có nghĩa là nhiều và "morphs" có nghĩa là hình thể. Vậy polymorphism có nghĩa là nhiều hình thể.

#### 1.1.5. Trừu tượng (Abstraction)

Trừu tượng là sự ẩn đi những chi tiết bên trong và hiển thị ra các chức năng, tính chất này gọi là trừu tượng. Ví dụ: khi gọi điện thoại chúng ta không biết xử lý nội bộ thế nào, khi đi xe máy cũng vậy, mà chúng ta chỉ biết đến các chức năng thông qua giao tiếp bên ngoài.

Trong java, chúng ta áp dụng tính chất trừu tượng bằng cách sử dụng abstract class và interface.

Một lớp được khai báo với từ khóa abstract là lớp abstract trong Java. Lớp abstract có nghĩa là lớp trừu tượng, nó có thể có các phương thức abstract hoặc non-abstract.

### Phương thức trừu tượng trong Java

Một phương thức được khai báo là abstract và không có trình triển khai thì đó là phương thức trừu tượng.

Nếu bạn muốn một lớp chứa một phương thức cụ thể nhưng bạn muốn triển khai thực sự phương thức đó để được quyết định bởi các lớp con, thì bạn có thể khai báo phương thức đó trong lớp cha ở dạng abstract.

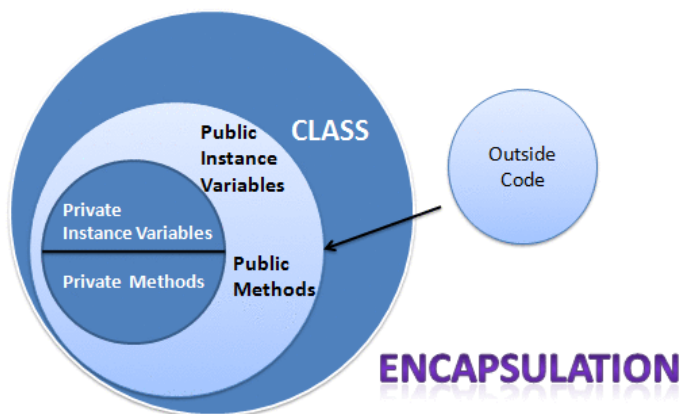
Từ khóa abstract được sử dụng để khai báo một phương thức dạng abstract. Một phương thức abstract không có thân phương thức.

### 1.1.6. Đóng gói (Encapsulation)

Việc ràng buộc giữa code và data với nhau tạo thành một khối duy nhất được biết đến là đóng gói. Ví dụ: viên thuốc con nhộng được đóng gói với nhiều loại thuốc bên trong.

Một class trong java là một ví dụ về đóng gói. Java bean là một lớp được đóng gói hoàn toàn vì tất cả các dữ liệu thành viên là private.

**Tính đóng gói trong java** là kỹ thuật ẩn giấu thông tin không liên quan và hiện thị ra thông liên quan. Mục đích chính của đóng gói trong java là giảm thiểu mức độ phức tạp phát triển phần mềm.



### Lợi ích của đóng gói trong java



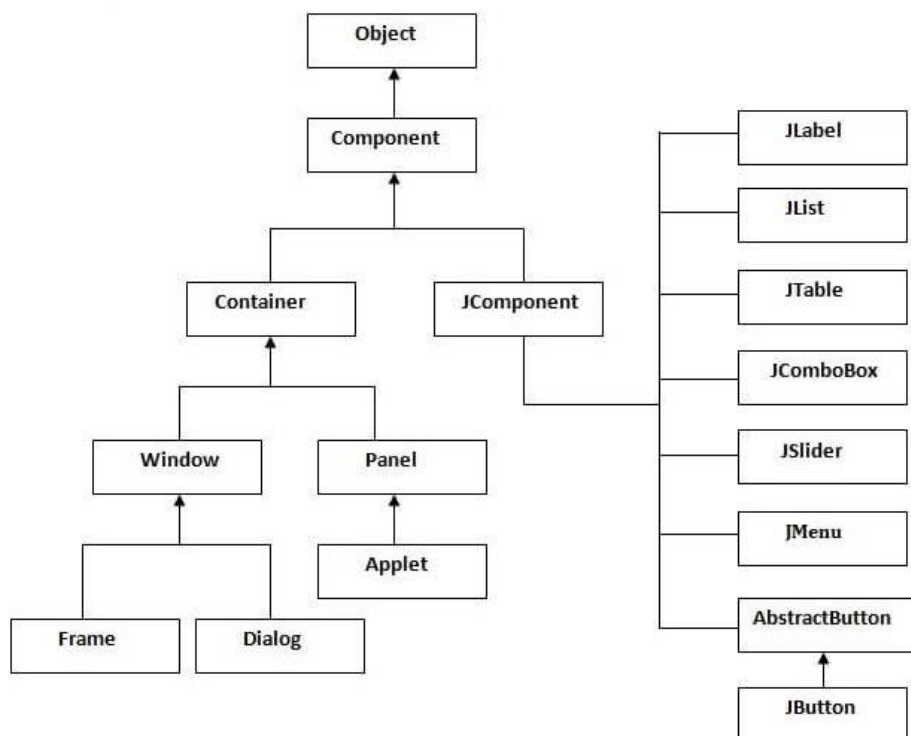
Bạn có thể tạo lớp **read-only** hoặc **write-only** bằng việc cài đặt phương thức setter hoặc getter.

Bạn có thể kiểm soát đối với dữ liệu. Giả sử bạn muốn đặt giá trị của id chỉ lớn hơn 100 bạn có thể viết logic bên trong lớp setter.

## 1.2. Java Swing

Phân cấp các lớp Java Swing

Hệ thống phân cấp của API java swing được đưa ra dưới đây.



### Các phương thức thường dùng của lớp Component

Các phương thức của lớp Component được sử dụng rộng rãi trong java swing được đưa ra dưới đây.

Phương thức	Mô tả
public void add(Component c)	thêm một thành phần vào thành phần khác.

<code>public void setSize(int width, int height)</code>	thiết lập kích thước của thành phần.
<code>public void setLayout(LayoutManager m)</code>	thiết lập trình quản lý bố cục (layout) cho thành phần.
<code>public void setVisible(boolean b)</code>	thiết lập khả năng hiển thị của thành phần. Nó theo mặc định là false (ẩn)

### 1.2.1. Lớp JFrame trong Java Swing

Container là thành phần chủ chốt trong các thành phần của SWING GUI. Một Container cung cấp một không gian, là nơi đặt một thành phần. Một Container trong AWT chính là một Component và nó có thêm khả năng để thêm các thành phần khác vào chính nó.

**Lớp này kế thừa các phương thức từ các lớp sau:**

- java.awt.Frame
- java.awt.Window
- java.awt.Container
- java.awt.Component
- java.lang.Object

**Lớp JFrame này có các constructor sau:**

- **JFrame()**: Xây dựng một Frame mới, ban đầu là không nhìn thấy (invisible).
- **JFrame(GraphicsConfiguration gc)**: Tạo một Frame trong GraphicsConfiguration đã cho của một thiết bị màn hình và một title trống.
- **JFrame(String title)**: Tạo một Frame mới, ban đầu là không nhìn thấy (invisible) với title đã cho.
- **JFrame(String title, GraphicsConfiguration gc)**: Tạo một Frame với title đã cho và GraphicsConfiguration đã cho của một thiết bị màn hình.

### 1.2.2. Lớp JPanel trong Java Swing

JPanel trong Java Swing được sử dụng để tạo ra các container nơi chứa các phần tử khác. Container là thành phần chủ chốt trong các thành phần của SWING GUI. Một Container cung cấp một không gian, là nơi đặt một thành phần. Một Container trong AWT chính là một Component và nó có thêm khả năng để thêm các thành phần khác vào chính nó.

**Lớp này kế thừa các phương thức từ các lớp sau:**

- javax.swing.JComponent
- java.awt.Container
- java.awt.Component
- java.lang.Object

**Lớp này bao gồm các constructor sau:**

- **JPanel():** Tạo một JPanel mới với một double buffer và một Flow Layout.
- **JPanel(boolean isDoubleBuffered):** Tạo một JPanel mới với Flow Layout và trình đệm đã xác định.
- **JPanel(LayoutManager layout):** Tạo một JPanel mới với Layout Manager đã cho
- **JPanel(LayoutManager layout, boolean isDoubleBuffered):** Tạo một JPanel mới với Layout Manager đã cho và trình đệm đã xác định.

**Các phương thức của lớp JPanel:**

- **AccessibleContext getAccessibleContext():** Lấy AccessibleContext được liên kết với JPanel này.
- **PanelUI getUI():** Trả về đối tượng L&F mà truyền thành phần này
- **String getUIClassID():** Trả về một chuỗi xác định tên của lớp L&F mà truyền thành phần này
- **protected String paramString():** Trả về một biểu diễn chuỗi của JPanel
- **void setUI(PanelUI ui):** Thiết lập đối tượng L&F mà truyền thành phần
- **void updateUI():** Phục hồi thuộc tính UI về một giá trị Look và Feel hiện tại.

### **1.2.3. Xử lý sự kiện trong Swing**

**Event** là một sự thay đổi trong trạng thái của đối tượng, chẳng hạn như sự kiện mô tả sự thay đổi trong trạng thái của source. Các sự kiện được tạo ra là do tương tác của người dùng với các thành phần UI. Ví dụ như việc nhấn vào một nút button, di chuyển chuột, nhập ký tự thông qua bàn phím, ... Các sự kiện có thể được phân chia thành hai loại sau:

- Foreground Event: Những sự kiện này yêu cầu tương tác trực tiếp của người dùng. Chúng được tạo ra do tương tác của người dùng với các thành phần đồ họa trong Graphical User Interface. Ví dụ như nhấn vào một nút, di chuyển chuột, chọn một item từ list, ...

- Background Event: Các sự kiện này yêu cầu tương tác của người dùng cuối cùng, ví dụ như tín hiệu ngắt hệ điều hành, hardware hoặc software failure

**Xử lý sự kiện (Event Handling)** là một kỹ thuật kiểm soát sự kiện và quyết định những gì cần thực hiện nếu một sự kiện xảy ra. Kỹ thuật này có code, mà được biết như là Event Handler, được thực thi khi một sự kiện xảy ra. Java sử dụng Delegation Event Model để xử lý các sự kiện. Model này định nghĩa kỹ thuật chuẩn để tạo và xử lý các sự kiện. Model này bao gồm hai thành phần quan trọng sau:

- Source: đây là một đối tượng mà trên đó sự kiện xuất hiện. Source chịu trách nhiệm cung cấp thông tin về sự kiện đã xảy ra tới bộ xử lý Handler của nó.

- Listener: Còn được biết như là Event Handler. Listener chịu trách nhiệm tạo phản hồi tới một sự kiện. Theo quan điểm của Java, Listener cũng là một đối tượng. Listener đợi tới khi nó nhận một sự kiện. Khi một sự kiện đã được nhận, Listener xử lý sự kiện đó và sau đó trả về kết quả.

Khai báo thư viện xử lý sự kiện trong java

```
import java.awt.event.ActionListener;  
import java.awt.event.ActionEvent;
```

Ví dụ về thêm sự kiện cho object trong java

```

        JButton btnTabSach = new JButton("S\u00E1ch");
        btnTabSach.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                changePanel(panel_Sach);
            }
        });
    }
}

```

### 1.2.3.1. Lớp EventObject trong Java Swing

Đây là lớp gốc (root class) từ đó tất cả đối tượng về trạng thái sự kiện sẽ được kế thừa. Tất cả sự kiện được xây dựng với một tham chiếu tới đối tượng đó, là source. Lớp này được định nghĩa trong java.util package. Cú pháp khai báo của lớp java.util.EventObject như sau:

```

public class EventObject
    extends Object
        implements Serializable

```

Lớp EventObject có một trường là protected Object source. Đây là đối tượng mà trên đó sự kiện được xảy ra. Lớp EventObject có một constructor là **EventObject(Object source)** dùng để xây dựng một nguyên mẫu sự kiện.

#### Các phương thức của lớp EventObject:

- Object getSource(): Đối tượng mà trên đó sự kiện xảy ra.
- String toString(): Trả về một biểu diễn chuỗi của EventObject này.

### 1.2.3.2. Lớp AWTEvent

Đây là lớp sự kiện gốc (root class) cho tất cả sự kiện AWTEvent. Lớp này và các lớp con của nó thay thế lớp ban đầu java.awt.Event. Lớp này được định nghĩa trong java.awt package. Lớp AWTEvent có phương thức getID() được sử dụng để xác định kiểu của sự kiện. Cú pháp khai báo của lớp java.awt.AWTEvent là:

```

public class AWTEvent
    extends EventObject

```

### 1.2.3.3. Lớp ActionEvent

Lớp `ActionEvent` được định nghĩa trong `java.awt.event` package. `ActionEvent` được tạo ra khi một nút được nhấn hoặc một item của một danh sách được nhấn đúp. Cú pháp khai báo cho lớp `ActionEvent` như sau:

```
public class ActionEvent  
    extends AWTEvent
```

#### **1.2.3.4. Lớp `InputEvent`**

Lớp `InputEvent` là lớp sự kiện gốc (root class) cho tất cả sự kiện liên quan tới đầu vào (cấp độ thành phần). Các sự kiện liên quan tới đầu vào (input event) được phân phối bởi `Listener` trước khi chúng được xử lý một cách thông thường bởi source, nơi chúng sinh ra. Điều này cho phép các `Listener` và các lớp thành phần con có thể "consume" sự kiện để mà source sẽ không xử lý chúng theo phương thức mặc định. Cú pháp khai báo của lớp `java.awt.event.InputEvent` là:

```
public abstract class InputEvent  
    extends ComponentEvent
```

#### **1.2.3.5. Lớp `KeyEvent`**

Sự kiện liên quan tới phím (Key Event) được tạo ra khi bạn nhập ký tự. Có ba kiểu key event, mà được biểu diễn bởi các hằng nguyên, chúng là:

- `KEY_PRESSED`
- `KEY_RELEASED`
- `KEY_TYPED`

Cú pháp khai báo của lớp `java.awt.event.KeyEvent` như sau:

```
public class KeyEvent  
    extends InputEvent
```

#### **1.2.3.6. Lớp `MouseEvent`**

Sự kiện này chỉ một hoạt động liên quan tới chuột xảy ra trong một thành phần. Sự kiện tầm thấp này được tạo bởi một đối tượng `Component` cho các sự kiện liên quan tới

chuột và di chuyển chuột, chẳng hạn như một nút chuột được nhấn, được nhả ra, được click (nhấn và nhả ra), di chuyển chuột, kéo chuột, ...

Cú pháp khai báo cho lớp `java.awt.event.MouseEvent`:

```
public class MouseEvent  
    extends InputEvent
```

### 1.3. Lý thuyết JDBC với MySQL

**Khái niệm:** **JDBC** (Java Database Connectivity) là một API tiêu chuẩn dùng để tương tác với các loại cơ sở dữ liệu quan hệ. **JDBC** có một tập hợp các class và các Interface dùng cho ứng dụng **Java** có thể nói chuyện với các cơ sở dữ liệu.

Các thành phần của JDBC Api về cơ bản bao gồm:

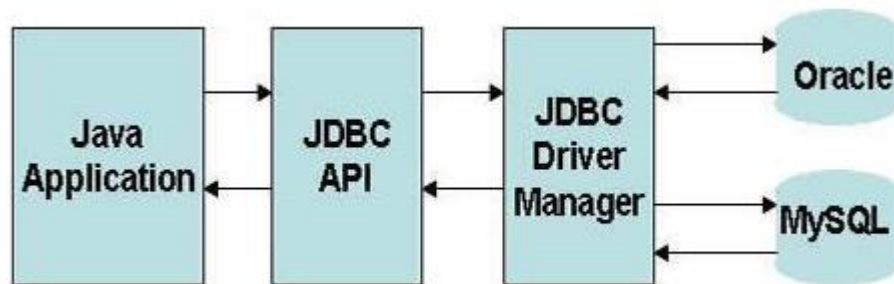
**DriverManager:** Là một class, nó dùng để quản lý danh sách các Driver (database drivers).

**Driver:** Là một Interface, nó dùng để liên kết các liên lạc với cơ sở dữ liệu, điều khiển các liên lạc với database. Một khi Driver được tải lên, lập trình viên không cần phải gọi nó một cách cụ thể.

**Connection:** Là một Interface với tất cả các method cho việc liên lạc với database. Nó mô tả nội dung liên lạc. tất cả các thông tin liên lạc với cơ sở dữ liệu là thông qua chỉ có đối tượng *Connection*.

**Statement:** Là một Interface, gói gọn một câu lệnh SQL gửi tới cơ sở dữ liệu được phân tích, tổng hợp, lập kế hoạch và thực hiện.

**ResultSet:** *ResultSet* đại diện cho tập hợp các bản ghi lấy do thực hiện truy vấn.



Các class sử dụng để kết nối

## Class DBConnection

```
public class DBConnection {
    static public Connection getConnection() throws ClassNotFoundException, SQLException {
        Class.forName("com.mysql.cj.jdbc.Driver");
        String url="jdbc:mysql://localhost:3306/ThuVien";
        String pw="123456@abc";
        String user="root";
        Connection con=DriverManager.getConnection(url, user, pw);
        return con;
    }
}
```

Các lệnh thao tác thêm, sửa xoá

```
public boolean Insert(Account acc) throws ClassNotFoundException, SQLException {
    Connection con=DBConnection.getConnection();
    String sql = "INSERT INTO Account VALUES(?, ?, ?, ?)";
    PreparedStatement pstmt = con.prepareStatement(sql);
    pstmt.setString(1, acc.getUsername());
    pstmt.setString(2, acc.getPassword());
    pstmt.setString(3, acc.getTypeUser());
    pstmt.setInt(4, acc.getMaNV());
    return pstmt.executeUpdate()>0?true:false;
}
```

```
public boolean Update(NXB nxb) throws ClassNotFoundException, SQLException {
    Connection con=DBConnection.getConnection();
    String sql = "UPDATE NXB SET TenNXB=?, DiaChi=?, SoDT=? where MaNXB=?";
    PreparedStatement pstmt = con.prepareStatement(sql);

    pstmt.setString(1, nxb.getTenNXB());
    pstmt.setString(2, nxb.getDiaChi());
    pstmt.setString(3, nxb.getSoDT());
    pstmt.setString(4, nxb.getMaNXB());
    if( pstmt.executeUpdate()>0)
        return true;
    return false;
}
```

```
public boolean Delete(int manv) throws ClassNotFoundException, SQLException{
    Connection connection = DBConnection.getConnection();
    String sqlString ="Delete from account where manv = ?";
    PreparedStatement preparedStatement = connection.prepareStatement(sqlString);
    preparedStatement.setInt(1, manv);
    int rs=preparedStatement.executeUpdate();
    if(rs>0)
        return true;
    return false;
}
```



## Gọi Procedure

```
public ResultSet GetThongKeMuon() {  
    try {  
        Connection con = DBConnection.getConnection();  
        String sql = "call thongkemuon()";  
        PreparedStatement pstmt = con.prepareStatement(sql);  
        return pstmt.executeQuery();  
    } catch (Exception e) {  
        e.printStackTrace();  
        return null;  
    }  
}
```

## **PHẦN 2: GIỚI THIỆU HỆ THỐNG**

### **2.1. Phần mềm quản lý thư viện là gì?**

Phần mềm quản lý thư viện hay còn gọi là ứng dụng quản lý thư viện. Công cụ này được thiết kế ra để chuẩn hóa và nâng cao hiệu quả quản lý. Nhất là trong việc quản lý các tài liệu, số liệu, sổ sách thư viện. Nhằm đảm bảo quản lý khoa học, gọn gàng theo trình tự nhất định. Giúp bạn có thể theo dõi được mọi hoạt động của thư viện. Ứng dụng dành cho các trường học, nhà sách, thư viện,... Tính đến thời điểm này, phần mềm đã được ứng dụng rộng rãi khắp mọi nơi. Trở thành cánh tay đắc lực vô cùng cần thiết trong quản lý. Nhận được nhiều phản hồi tích cực và đánh giá cao từ phía khách hàng.

### **2.2. Ưu điểm của hệ thống quản lý thư viện**

Nhằm khắc phục những hạn chế theo cách quản lý truyền thống. Phần mềm quản lý thư viện được thiết kế với nhiều ưu điểm vượt trội:

Chi phí đầu tư ban đầu thấp nhưng hiệu quả lâu dài. Sử dụng công cụ quản lý thư viện giúp tiết kiệm được một khoản chi phí lớn.

Không mất thời gian theo dõi, tìm kiếm, kiểm kê. Mang lại hiệu quả công việc cao hơn.

Phần mềm giúp chuẩn hóa và nâng cao công tác quản lý. Lưu thông sách, tài liệu,...

Hướng đến xây dựng một thư viện chuẩn hóa, hiện đại. Nhằm giảm thiểu tối đa công việc phải quản lý thủ công mất thời gian. Giúp quản lý nhanh chóng, dễ dàng các hoạt động nhập, xuất kho, theo dõi mượn trả sách...

Thiết kế phần mềm thân thiện, thao tác đơn giản, dễ sử dụng, linh hoạt. Không đòi hỏi người dùng phải hiểu biết quá nhiều về tin học

Người dùng có thể nâng cấp phần mềm với nhiều tính năng thông minh, hiện đại

### **2.3. Chức năng của phần mềm quản lý thư viện**

Chức năng hệ thống quản lý kho sách dễ dàng

Với số lượng sách đồ sộ như vậy, không thể quản lý bằng thủ công. Phần mềm quản lý thư viện cho phép tạo ra các phiếu nhập kho. Từ những phiếu nhập kho sẽ giúp nhân

viên thư viện thống kê chi tiết nhất. Từ số lượng các đầu sách nhập kho trong mỗi đợt. Đến danh mục các loại sách, tác giả, nhà xuất bản...

Đồng thời công cụ còn giúp quản lý các lần xuất thanh lý sách, tài liệu thư viện. Chẳng hạn như thống kê ngày, tháng, nội dung, tên sách, số liệu, giá tiền. Cho phép phân loại từng mục sách riêng để sắp xếp vào vị trí phù hợp. Ngoài ra còn hỗ trợ tra cứu nhanh dễ dàng, tiện lợi.

Chức năng quản lý danh sách độc giả, quản lý mượn, trả sách

Ứng dụng quản lý thư viện trường học tích hợp đa chức năng. Cho phép quản lý thông tin độc giả như: họ tên, địa chỉ, số điện thoại,... Quản lý độc giả đã mượn, trả sách một cách chi tiết nhất.

Chức năng tra cứu vị trí sách

Ứng dụng thư viện hỗ trợ quản lý chi tiết vị trí của từng đầu sách/ tài liệu. Giúp tạo vị trí sắp xếp theo đúng sơ đồ của thư viện trên phần mềm. Người quản lý có thể nhập thông tin sách vào từng vị trí, từng ngăn, từng tầng. Khi cần tra cứu, chỉ cần tìm kiếm trên phần mềm rất dễ dàng.

Chức năng báo cáo thống kê

Hỗ trợ thống kê số lượng sách còn trong kho. Thống kê tình trạng mượn, trả sách của khách hàng.

Chức năng phân quyền quản trị

Ứng dụng quản lý thư viện cho phép tạo ra các tài khoản đăng nhập. Phân quyền quản lý từng module cho các nhân viên thư viện.

## 2.4. Lời kết

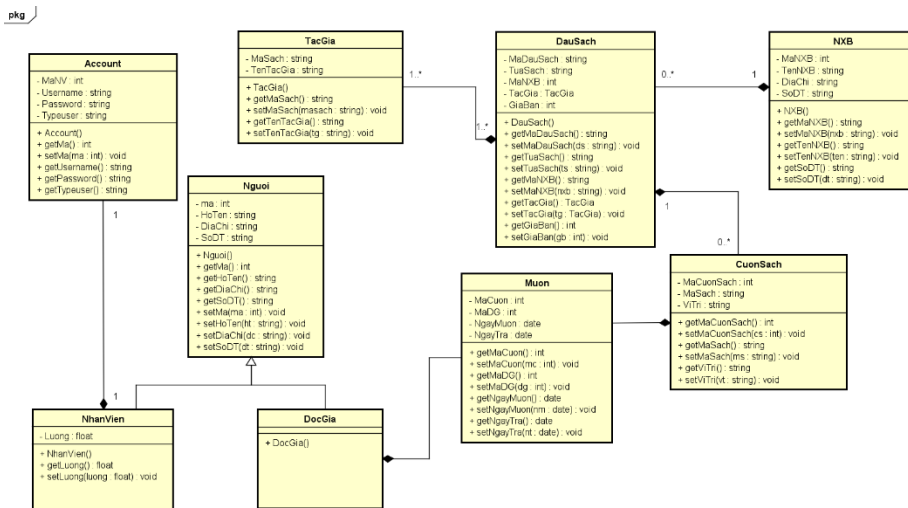
Sự góp mặt của những **phần mềm quản lý thư viện** hiện nay rất hữu ích. Giúp các trường học, thư viện, cơ sở lưu trữ sách giải quyết triệt để mọi sai sót trong quản lý. Đồng thời mang đến hiệu quả cao, tiết kiệm nhất.

## PHẦN 3: THIẾT KẾ HỆ THỐNG

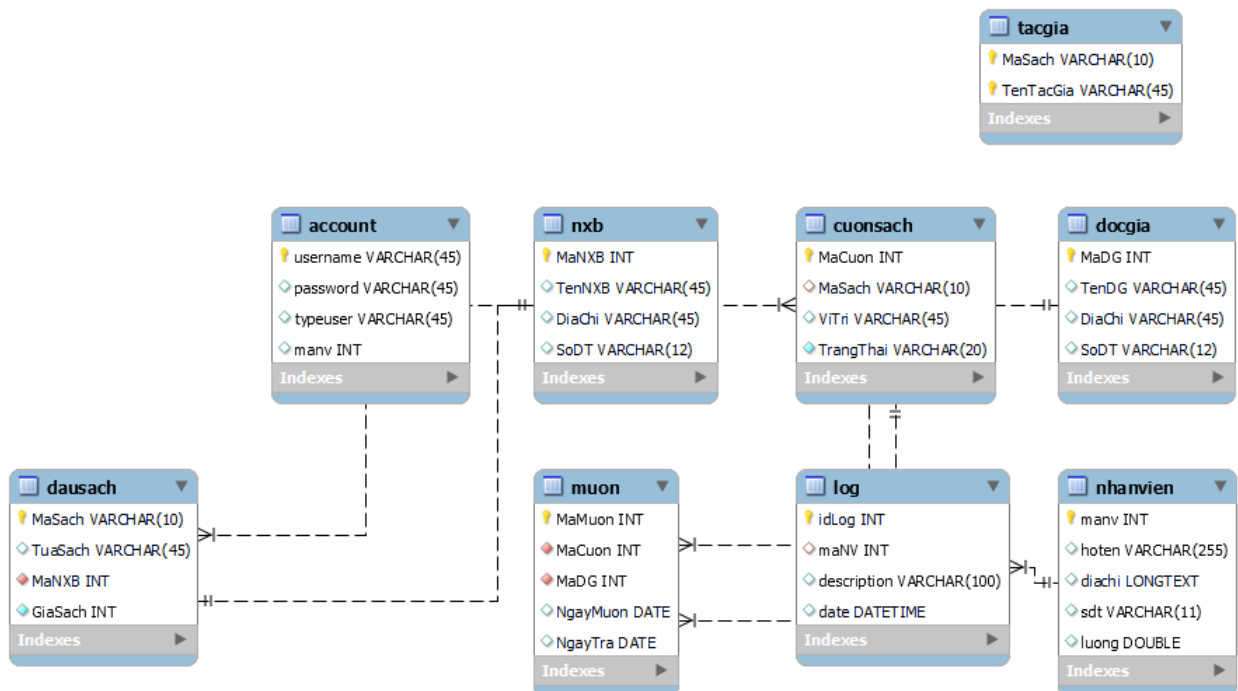
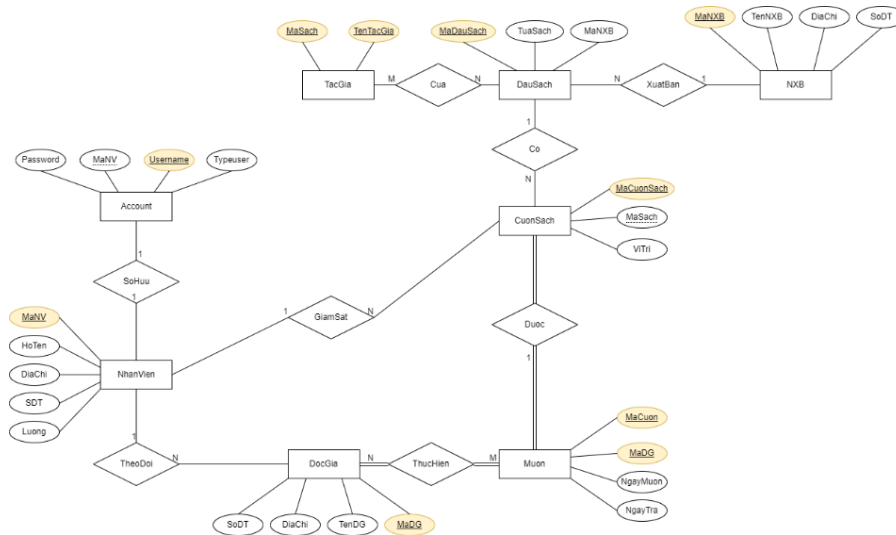
### 3.1. Sơ đồ chức năng



### 3.2. Sơ đồ lớp



### 3.3. Thiết kế cơ sở dữ liệu



Bảng **account** chứa thông tin đăng nhập bao gồm: username, password, typeuser, mã nhân viên.

Bảng **nxb** chứa thông tin của nhà xuất bản bao gồm: Mã nhà xuất bản, địa chỉ, số điện thoại.

Bảng **dausach** chứa thông tin của mỗi loại sách bao gồm: Mã sách, tựa sách, mã nhà xuất bản, giá sách.

Bảng **cuonsach** chứa thông tin của mỗi cuốn sách bao gồm: Mã cuốn sách, mã sách, vị trí cuốn sách, trạng thái(mượn, chưa mượn)

Bảng **docgia** chứa thông tin cá nhân của độc giả bao gồm: Mã độc giả, tên độc giả, địa chỉ, số điện thoại.

Bảng **muon** chứa thông tin về việc mượn trả sách bao gồm: Mã mượn, mã cuốn, mã độc giả, ngày mượn, ngày trả.

Bảng **nhanvien** chứa thông tin của nhân viên bao gồm: Mã nhân viên, họ tên, địa chỉ, số điện thoại, lương.

Bảng **log** chứa thông tin lịch sử làm việc của nhân viên bao gồm: Mã lịch sử, mã nhân viên, mô tả, ngày.

### 3.4. Thiết kế giao diện

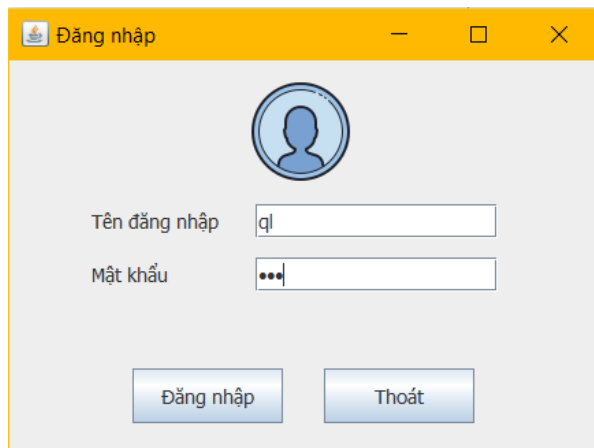
Tổ chức dữ liệu với các đối tượng Đầu sách, Cuốn sách, Nhà xuất bản, Tác giả, Độc giả, Nhân viên, Mỗi đối tượng đều có CRUD đầy đủ

Cách xây dựng giao diện cho chương trình:

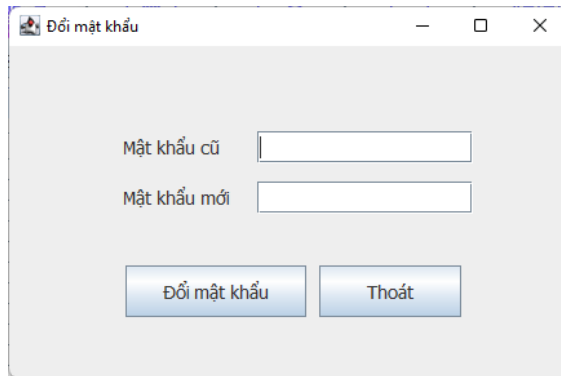
- Sử dụng Frame để có giao diện chính
- Những chức năng cho các đối tượng sẽ được xây dựng trên 1 Panel hoặc frame. Mỗi khi thực hiện gọi thao tác thì form panel sẽ xuất hiện
- Mỗi button đều được gắn cho một sự kiện nhất định

#### 3.4.1. Tính năng chung:

\* Frame Đăng nhập

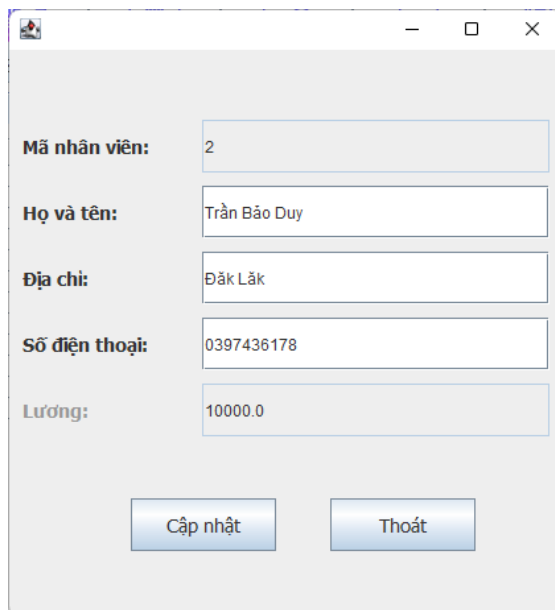


\*Frame Đổi mật khẩu



The dialog box titled "Đổi mật khẩu" contains two text input fields. The first field is labeled "Mật khẩu cũ" (Old password) and the second is labeled "Mật khẩu mới" (New password). Below the fields are two buttons: "Đổi mật khẩu" (Change password) and "Thoát" (Exit).

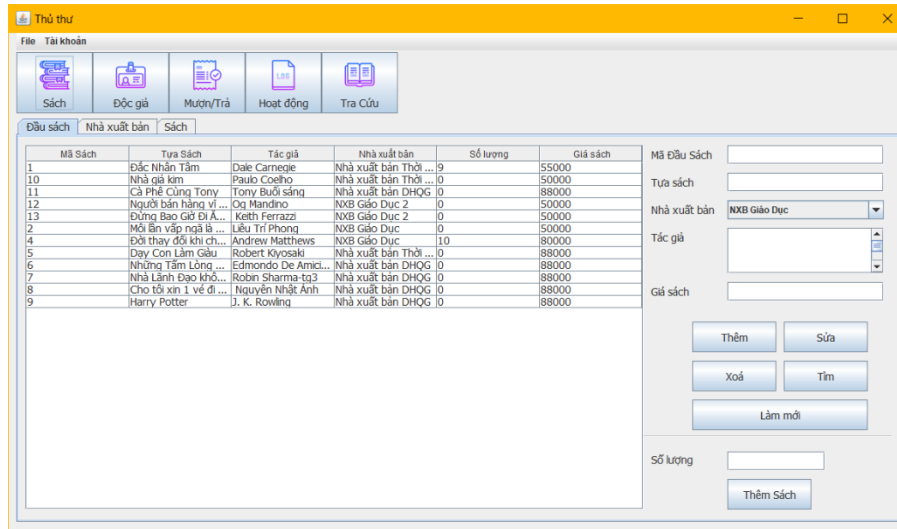
\* Frame Thông tin cá nhân



The form titled "Thông tin cá nhân" contains five text input fields. The first field is labeled "Mã nhân viên:" (Employee ID) and contains the value "2". The second field is labeled "Họ và tên:" (Full name) and contains the value "Trần Bảo Duy". The third field is labeled "Địa chỉ:" (Address) and contains the value "Đắk Lắk". The fourth field is labeled "Số điện thoại:" (Phone number) and contains the value "0397436178". The fifth field is labeled "Lương:" (Salary) and contains the value "10000.0". Below the fields are two buttons: "Cập nhật" (Update) and "Thoát" (Exit).

Mỗi nhân viên sẽ được cấp một tài khoản để đăng nhập. Dựa vào tên đăng nhập và mật khẩu sẽ tự mở Frame phù hợp với phân quyền (Quyền **Quản lý** và **Thủ thư**)

### 3.4.2. Frame của Thủ thư



Gồm các chức năng: Sách, Độc giả, Mượn/Trả sách, Hoạt động và Tra cứu.

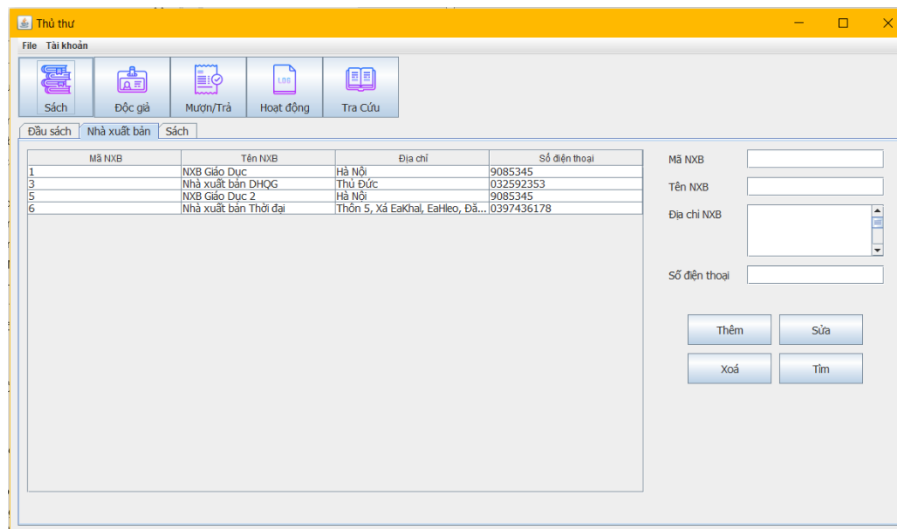
#### 3.4.2.1. Sách

Ở đây chúng ta có thể thấy được 3 tab nhỏ là Đầu Sách, Độc giả, Nhà Xuất bản.

##### 3.4.2.1.1. Đầu Sách

Ở đây chúng ta có thể thực hiện các thao tác tìm, thêm, sửa, xóa đầu sách.

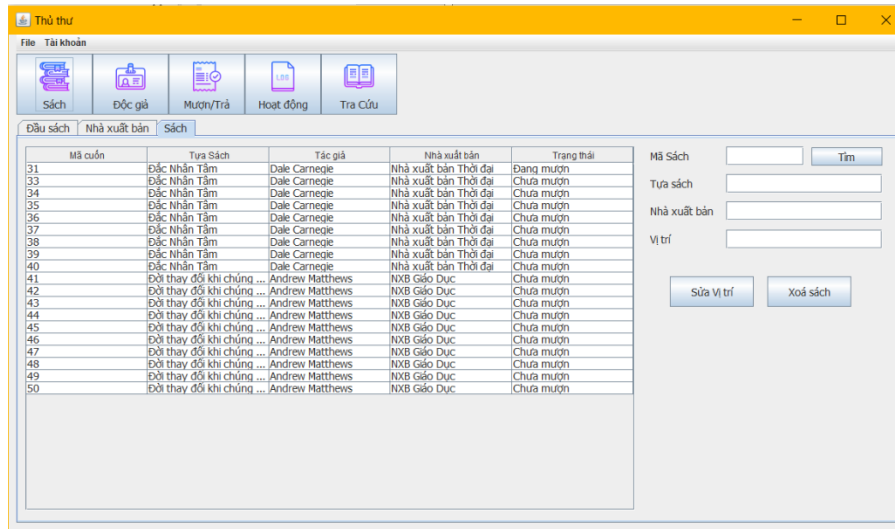
##### 3.4.2.1.2. Nhà xuất bản



Tương tự với sách, ở đây chúng ta có thể tìm, thêm, xóa, sửa cũng như xem danh sách thông tin các nhà xuất bản của những cuốn sách có trong thư viện.

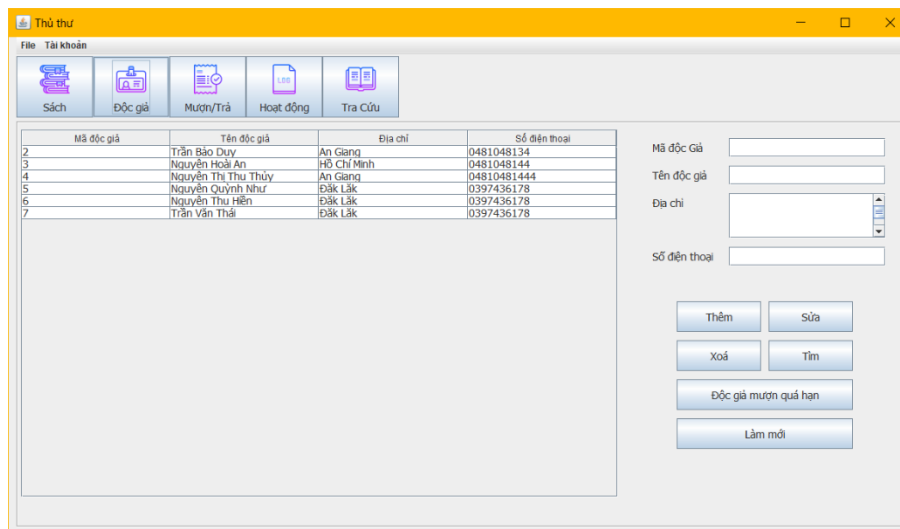


### 3.4.2.1.3. Sách



Ở đây chúng ta có thể xem chi tiết thông tin từng cuốn sách có trong thư viện, tìm, sửa vị trí và xóa chúng khỏi thư viện.

### 3.4.2.2. Độc giả



Qua thẻ này chúng ta có thể xem được danh sách thông tin các độc giả - người mượn sách của thư viện, ngoài ra có thể thêm mới, sửa thông tin, xóa và tìm thông tin độc giả.

### 3.4.2.3. Mượn trả

Mã cuốn	Tựa sách	Ngày mượn	Ngày trả
31	Đặc Nhân Tâm	2021-12-04	

Mã độc giả: 2  
Tên độc giả: Trần Bảo Duy  
Địa chỉ: An Giang  
Số điện thoại: 0481048134  
☒ Đang mượn ☐ Tất cả  
Xoá Tìm

Mã cuốn	Tựa sách	Nhà xuất bản	Tác giả
33	Đặc Nhân Tâm	Dale Carnegie	Nhà xuất bản Thời đại
34	Đặc Nhân Tâm	Dale Carnegie	Nhà xuất bản Thời đại
35	Đặc Nhân Tâm	Dale Carnegie	Nhà xuất bản Thời đại
36	Đặc Nhân Tâm	Dale Carnegie	Nhà xuất bản Thời đại
37	Đặc Nhân Tâm	Dale Carnegie	Nhà xuất bản Thời đại
38	Đặc Nhân Tâm	Dale Carnegie	Nhà xuất bản Thời đại
39	Đặc Nhân Tâm	Dale Carnegie	Nhà xuất bản Thời đại
40	Đặc Nhân Tâm	Dale Carnegie	Nhà xuất bản Thời đại
41	Đổi thay đổi khi chúng ta tha...	Andrew Matthews	NXB Giáo Dục
42	Đổi thay đổi khi chúng ta tha...	Andrew Matthews	NXB Giáo Dục
43	Đổi thay đổi khi chúng ta tha...	Andrew Matthews	NXB Giáo Dục
44	Đổi thay đổi khi chúng ta tha...	Andrew Matthews	NXB Giáo Dục
45	Đổi thay đổi khi chúng ta tha...	Andrew Matthews	NXB Giáo Dục
46	Đổi thay đổi khi chúng ta tha...	Andrew Matthews	NXB Giáo Dục

Mã sách:   
Tựa sách:   
Mượn Trả  
Gia hạn Mất sách

Chúng ta có thể theo dõi và thực hiện việc mượn trả sách, gia hạn sách đang mượn và báo mất sách, xem được thông tin về sách đang/đã mượn của độc giả từ Frame này.

### 3.4.2.4. Hoạt động

Mã Nhân Viên: 2  
☐ Theo thời gian Từ: 2021-12-05 Đến: 2021-12-05 Lọc mới

ID Log	Tên nhân viên	Ghi chú	Thời gian
2	Trần Bảo Duy	Thêm đầu sách (Mã:10)	2021-12-07
7	Trần Bảo Duy	Thêm đầu sách (Mã:6)	2021-12-10
8	Trần Bảo Duy	Thêm đầu sách (Mã:7)	2021-12-10
9	Trần Bảo Duy	Thêm đầu sách (Mã:8)	2021-12-10
10	Trần Bảo Duy	Thêm đầu sách (Mã:9)	2021-12-10
11	Trần Bảo Duy	Thêm đầu sách (Mã:11)	2021-12-10
12	Trần Bảo Duy	Thêm đầu sách (Mã:12)	2021-12-10
13	Trần Bảo Duy	Thêm đầu sách (Mã:13)	2021-12-10
14	Trần Bảo Duy	Thêm nhà xuất bản (Mã NXB:nuil)	2021-12-10

Frame này sẽ ghi lại lịch sử làm việc của nhân viên đang phụ trách trực tại thư viện. Có thể phân loại thao ngày. Frame này sẽ xuất hiện ở cả Quản lý và Thủ thư, trong đó quyền thủ thư chỉ xem được hoạt động của bản thân, còn quyền quản lý có thể xem được tất cả

### 3.4.2.5. Tra cứu

Mã Sách	Tên Sách	Tác giả	Nhà xuất bản	Số lượng	Giá sách
1	Đắc Nhân Tâm	Dale Carnegie	Nhà xuất bản Thời đại	9	56000
10	Nhà giả kim	Paulo Coelho	Nhà xuất bản Thời đại	9	56000
5	Đay Con Lười Giàu	Robert Kiyosaki	Nhà xuất bản Thời đại	0	88000

Frame này giúp thủ thư thuận tiện hơn trong việc tra cứu thông tin của tác giả, nhà xuất bản.

### 3.4.3. Frame của Quản lý:

Mã nhân viên	Tên nhân viên	Địa chỉ	Số điện thoại	Lương
1	Lê Văn Cường	Đắk Lắk	0389365835	10000.0
2	Trần Bảo Duy	Đắk Lắk	0397436178	10000.0
21	Lê Di Hân	Đắk Lắk	0398110398	7792429.0
22	Bùi Ngọc Dũng	An Giang	0398110398	7792429.0

Ở frame này chúng ta có thể thấy có 7 tab là Nhân viên, Thống kê, Hoạt động, Sách, Độc giả, Mượn/Trả, Tra cứu. Trong đó 3 tab Nhân viên, Thống kê, Hoạt động là của riêng biệt quyền **Quản lý**, các tab còn lại là giống với trên quyền **Thủ thư**

#### 3.4.3.1. Nhân viên

Ở tab này chúng ta có thể thấy có 2 tab nhỏ là Danh sách nhân viên và Thông tin cá nhân.

#### 3.4.3.1.1. Danh sách nhân viên

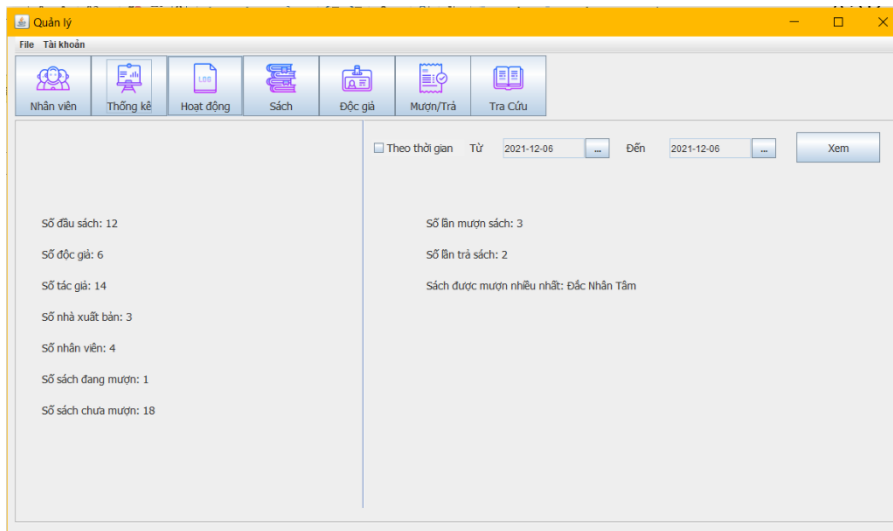
Ở tab nhỏ này chúng ta có thể xem được danh sách các nhân viên đang làm tại thư viện, có thể thực hiện các hoạt động thêm, xóa, sửa, tìm kiếm thông tin của nhân viên.

#### 3.4.3.1.2. Thông tin cá nhân

Ở tab này chúng ta có thể xem được những thông tin cá nhân của mình bao gồm:

- Mã nhân viên.
- Tên nhân viên.
- Địa chỉ.
- Số điện thoại.
- Lương.

#### 3.4.3.2. Thống kê

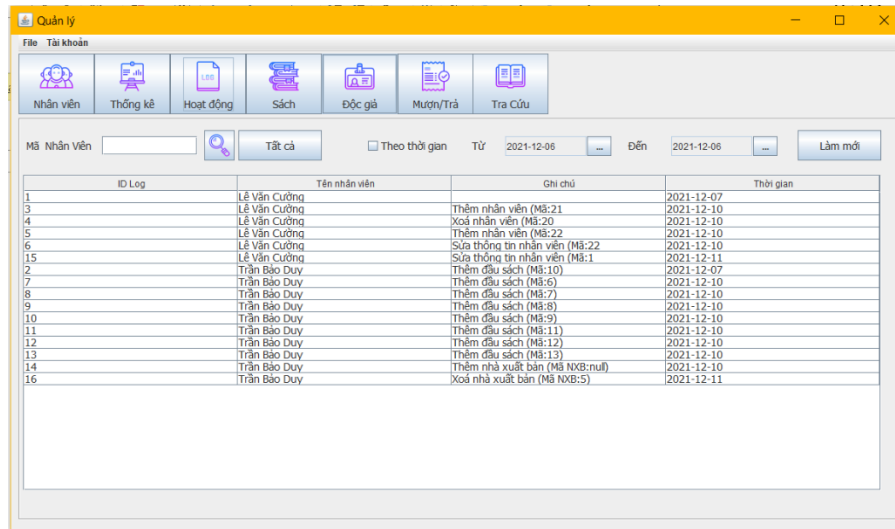


Ở frame này chúng ta có thể thấy được những số liệu thống kê về:

- Số đầu sách.
- Số độc giả.
- Số tác giả
- Số nhà xuất bản.
- Số nhân viên.
- Số sách đang mượn.

- Sổ sách chưa mượn.
- Thống kê số lần mượn trả sách theo thời gian.

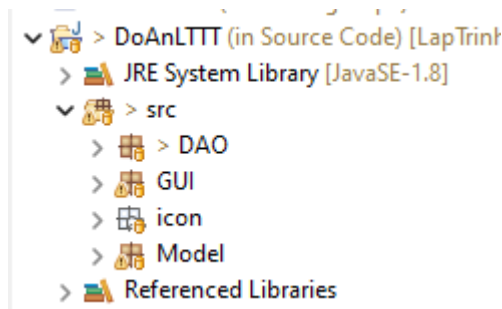
### 3.4.3.3. Hoạt động



Ở đây chúng ta có thể theo dõi được tất cả các lịch sử làm việc của thư viện theo ngày và theo nhân viên thực hiện.

## 3.5. Xây dựng cấu trúc Project

Cấu trúc thư mục



- Thư mục DAO chứa các class liên quan đến kết nối Database và là cầu nối giữa giao diện và Database
- Thư mục GUI chứa các class tạo nên giao diện cho toàn bộ Project
- Thư mục icon chứa các file icon sử dụng trong Project
- Thư mục Model chứa các class xây dựng nên các đối tượng được tương tác trong Project

## **PHẦN 4. KẾT QUẢ**

### **4.1. Kết quả chung**

Thông qua môn học và việc làm đồ án, dưới đây là kết quả mà nhóm em đã đạt được.

- Nắm vững những nguyên lý, cách tiếp cận và phương pháp lập trình hướng đối tượng với Java.

- Nắm vững các cấu trúc dữ liệu và các giải thuật thích hợp với những dữ liệu có trong ngôn ngữ lập trình hướng đối tượng Java.

- Sử dụng nhuần nhuyễn phần mềm Eclipse trong việc thực hiện các bài tập và các đồ án.

- Hiểu rõ và viết tốt các chương trình thể hiện tính chất căn bản nhất của hướng đối tượng với Java.

- Hiểu rõ và sử dụng được thư viện lập trình giao diện Jframe trên Java.

- Hiểu rõ và sử dụng được thư viện JDBC để thực hiện việc kết nối với hệ cơ sở dữ liệu MySQL để lưu trữ dữ liệu trên Java.

- Nâng cao tinh thần làm việc trong nhóm, cùng nghiên cứu và trao đổi giải quyết vấn đề trong quá trình học và thực hành.

- Đánh giá và lựa chọn các mô hình đã học để phù hợp cho bài toán thực tế.

- Vận dụng kỹ thuật lập trình hướng đối tượng với Java, lập trình giao diện Jframe, sử dụng công cụ JDBC để kết nối cơ sở dữ liệu để giải quyết vấn đề trong thực tế.

### **4.2. Hạn chế**

Dưới đây là những hạn chế của đồ án của nhóm em.

- Báo cáo chưa thể chọn theo tuần, theo tháng, theo quý, theo năm,...

- Quản lý cơ sở vật chất của thư viện.

- Hướng đăng nhập chưa tối ưu.

- Chưa lưu được báo cáo dưới dạng file.

### **4.3. Hướng cải tiến**

Trong quá trình làm đồ án thì đây là những hướng cải tiến để đồ án được hoàn thiện và phát triển trong tương lai.

- Cải tiến giao diện.
- Cải thiện các hạn chế.
- Sử dụng trên nhiều nền tảng như web, di động,...
- Tích hợp việc đăng nhập cho độc giả để theo dõi sách.
- Ứng dụng AI để gợi ý sách cho độc giả.

## 5. TÀI LIỆU THAM KHẢO

- Lập trình hướng đối tượng(OOPs) trong java(<https://viettuts.vn/java/tong-quan-java-oops>)