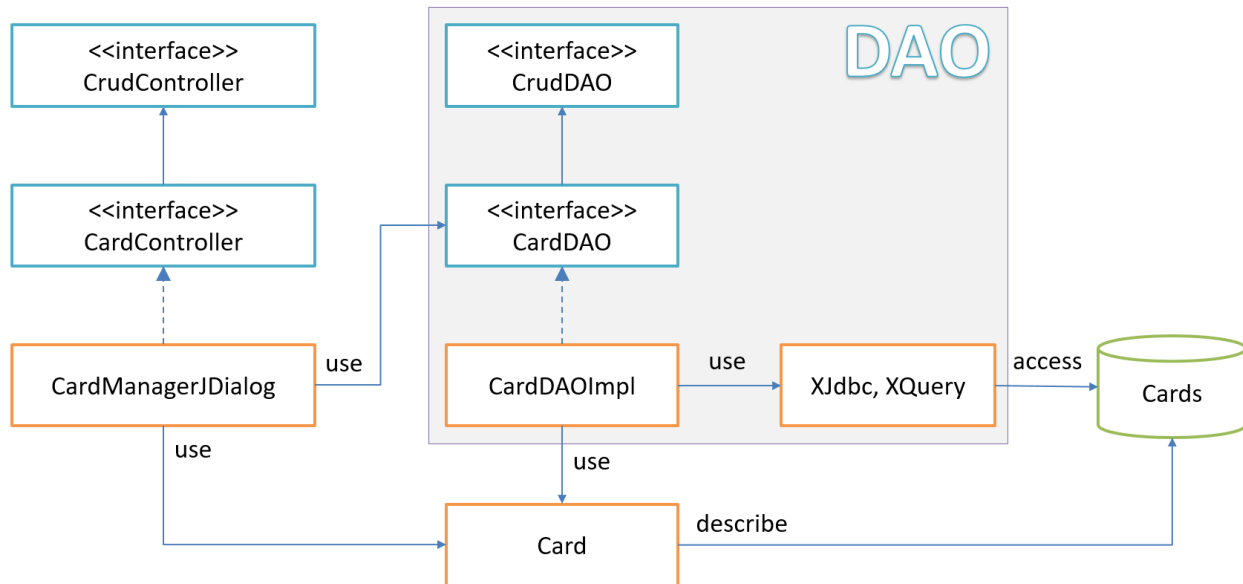


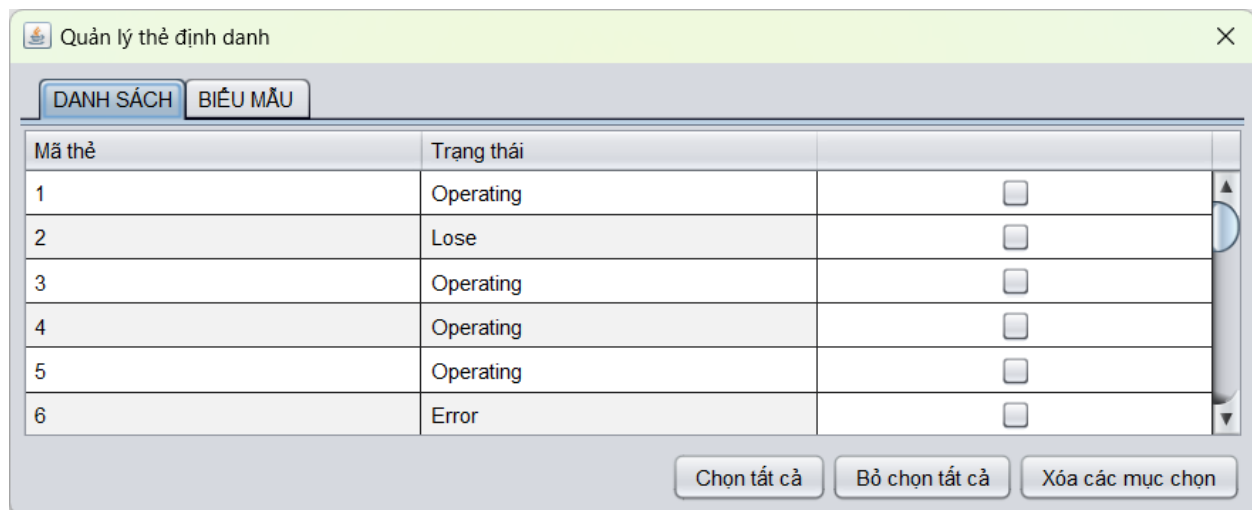
BÀI 1: QUẢN LÝ LOẠI ĐỒ UỐNG

Thực hiện theo hướng dẫn của file Lab3.1-CategoryManager.doc

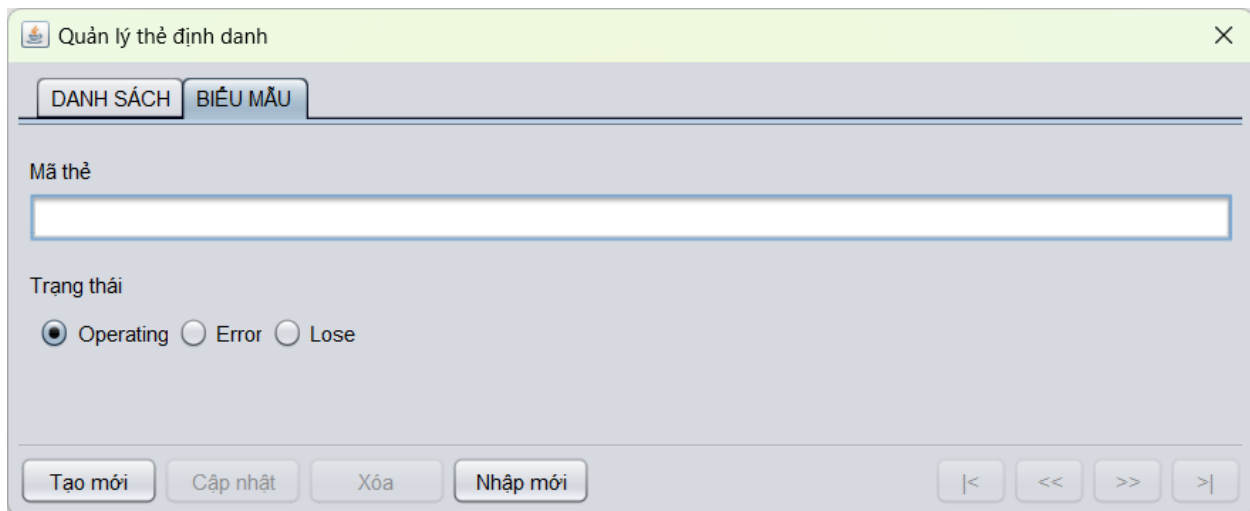
BÀI 2: QUẢN LÝ THẺ ĐỊNH DANH



Hình 1: Sơ đồ tổ chức



Hình 2.1: Giao diện – Danh sách

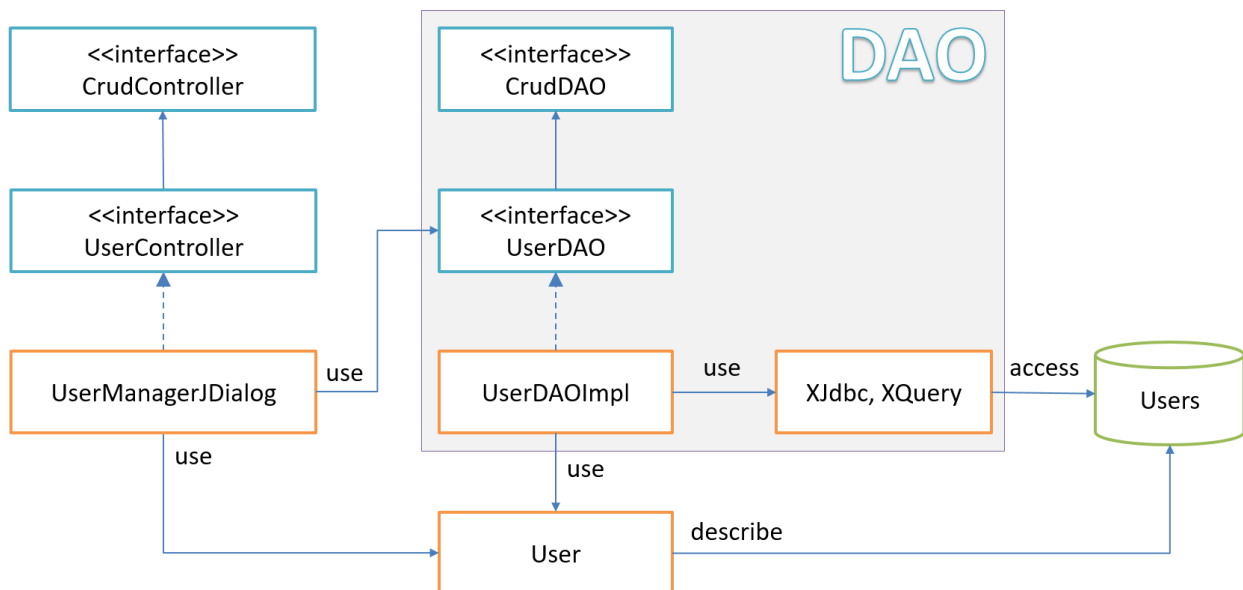


Hình 2.2: Giao diện – Biểu mẫu

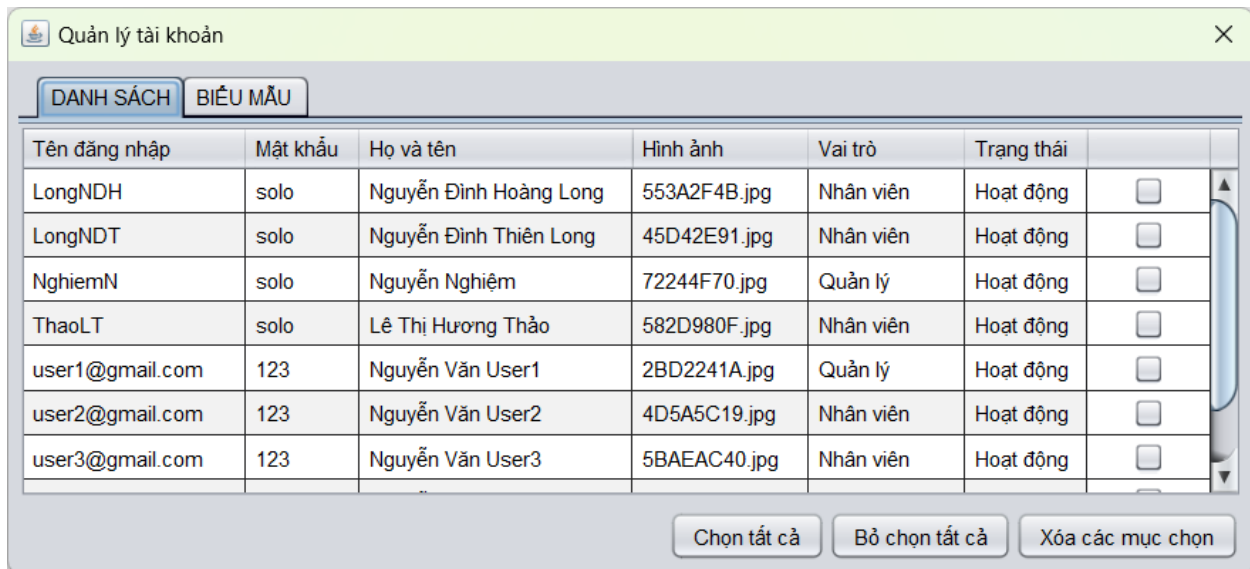
Hướng dẫn: Thực hiện tương tự như bài 1 để hoàn thành chức năng Quản lý thẻ định vị

BÀI 3: QUẢN LÝ NGƯỜI SỬ DỤNG

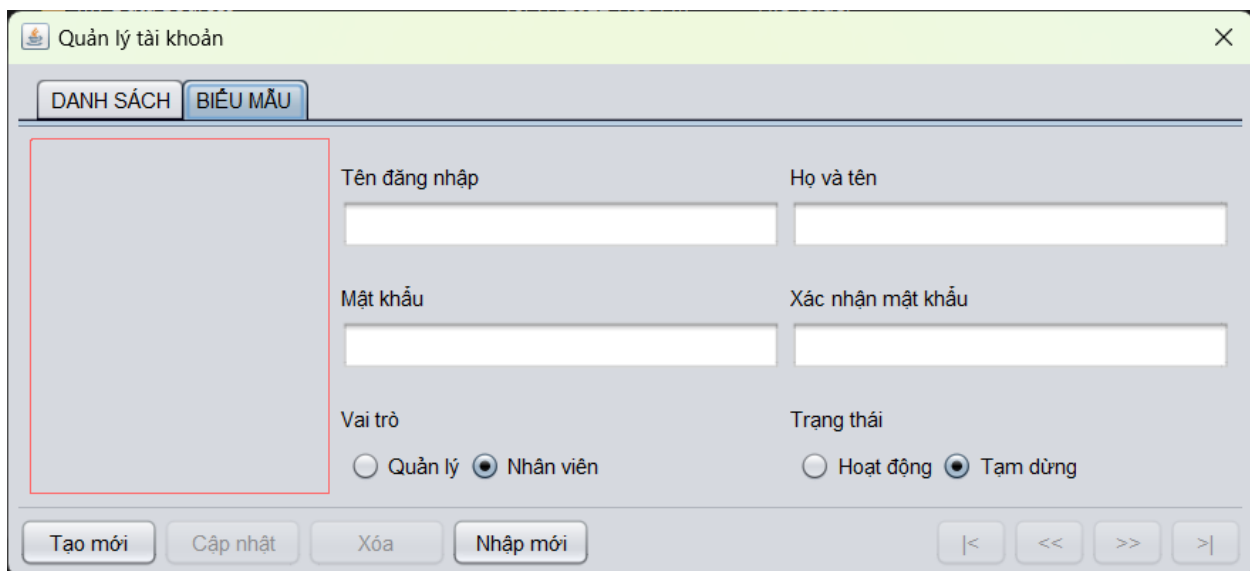
Thực hiện tương tự như bài 1 để hoàn thành chức năng Quản lý người sử dụng



Hình 1: Sơ đồ tổ chức



Hình 2.1: Giao diện – Danh sách

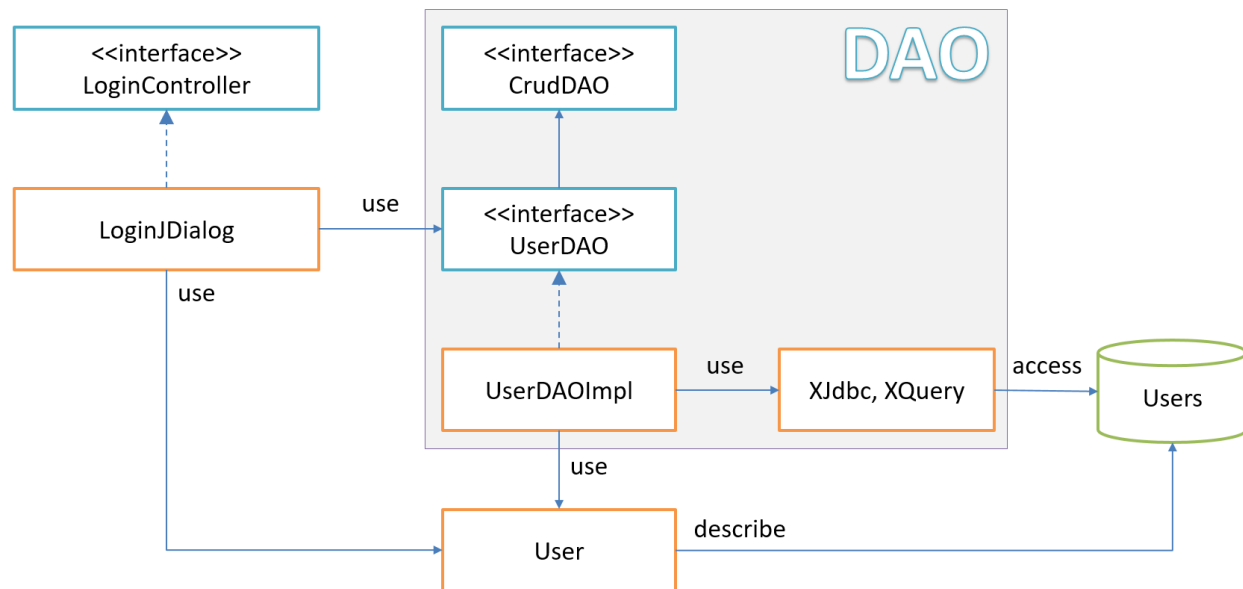


Hình 2.2: Giao diện – Biểu mẫu

Hướng dẫn: Thực hiện tương tự như bài 1 để hoàn thành chức năng Quản lý thẻ định vị

BÀI 4: ĐĂNG NHẬP

SƠ ĐỒ TỔ CHỨC



GIAO DIỆN



LOGINCONTROLLER

```

public interface LoginController {
    void open();
    void login();
    default void exit(){
        if(XDialog.confirm("Bạn muốn kết thúc?")){
            System.exit(0);
        }
    }
}

```

```
}
}
}
```

BỔ SUNG LỚP TIỆN ÍCH XAUTH VÀ HIỆU CHỈNH POLYCAFEJFRAME

- Thêm lớp tiện ích Xauth chứa biến static để duy trì và chia sẻ thông tin user sau khi đăng nhập giữa các chức năng trong ứng dụng

```
public class XAuth {
    public static User user = User.builder()
        .username("user1@gmail.com")
        .password("123")
        .enabled(true)
        .manager(true)
        .fullname("Nguyễn Văn Tèo")
        .photo("trump.png")
        .build(); // biến user này sẽ được thay thế sau khi đăng nhập
}
```

- Hiệu chỉnh phương thức init() của PolyCafeJFrame như sau để hiển thị thông tin user lên giao diện.

```
@Override
public void init() {
    this.setIconImage(XIcon.getIcon("trump-small.png").getImage());
    this.setLocationRelativeTo(null);

    this.showWelcomeJDialog(this);
    this.showLoginJDialog(this);

    XIcon.setIcon(lblPhoto, "photos/" + XAuth.user.getPhoto());
    lblFullname.setText(XAuth.user.getFullname());

    if(!XAuth.user.isManager()){
        pnlCenter.remove(pnlManager);
    }
}
```

CÀI ĐẶT MÃ CHO LOGINCONTROLLER

```
@Override
public void open() {
    this.setLocationRelativeTo(null);
}

@Override
public void login() {
    String username = txtUsername.getText();
    String password = txtPassword.getText();
}
```

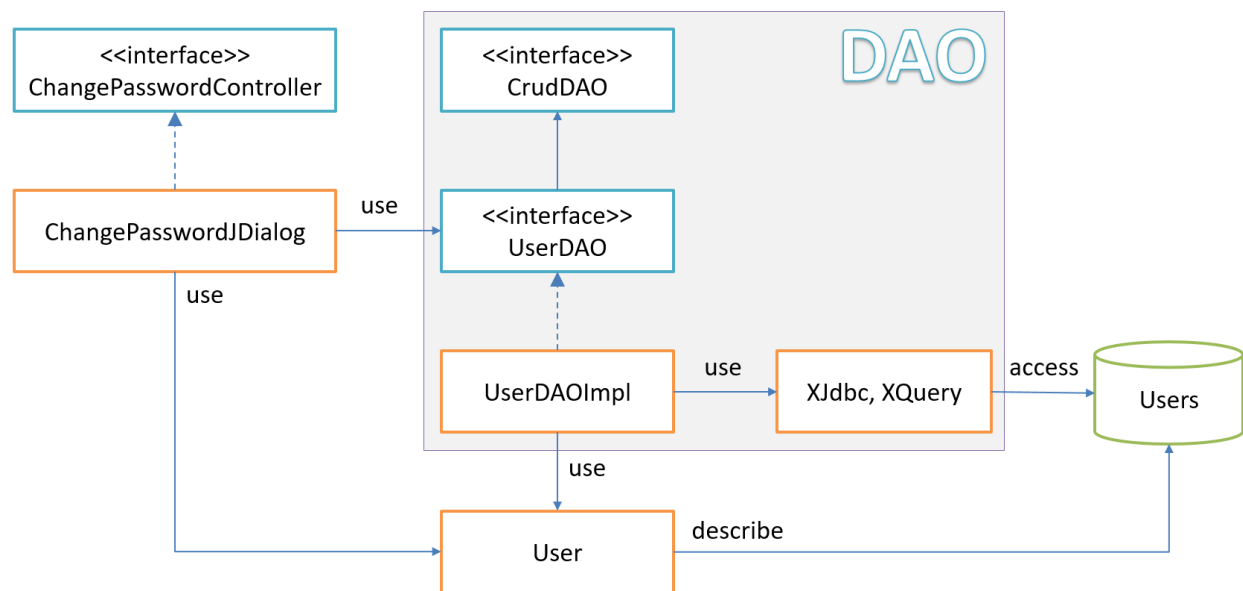
```

UserDAO dao = new UserDAOImpl();
User user = dao.findById(username);
if (user == null) {
    XDialog.alert("Sai tên đăng nhập!");
} else if (!password.equals(user.getPassword())) {
    XDialog.alert("Sai mật khẩu đăng nhập!");
} else if (!user.isEnabled()) {
    XDialog.alert("Tài khoản của bạn đang tạm dừng!");
} else {
    XAuth.user = user; // duy trì user đăng nhập
    this.dispose();
}
}

```

BÀI 5: ĐỔI MẬT KHẨU

SƠ ĐỒ TỔ CHỨC



GIAO DIỆN

CHANGEPASSWORDCONTROLLER

```
public interface ChangePasswordController {
    void open();
    void save();
    void close();
}
```

Cài đặt mã cho controller

```
UserDAO dao = new UserDAOImpl();

@Override
public void open() {
    this.setLocationRelativeTo(null);
}

@Override
public void close() {
    this.dispose();
}

@Override
public void save() {
    String username = txtUsername.getText();
    String password = txtPassword.getText();
    String newpass = txtNewpass.getText();
    String confirm = txtConfirm.getText();
}
```

```
if (!newpass.equals(confirm)) {  
    XDialog.alert("Xác nhận mật khẩu không đúng!");  
} else if (!username.equals(XAuth.user.getUsername())) {  
    XDialog.alert("Sai tên đăng nhập!");  
} else if (!password.equals(XAuth.user.getPassword())) {  
    XDialog.alert("Sai mật khẩu!");  
} else {  
    XAuth.user.setPassword(newpass);  
    dao.update(XAuth.user);  
    XDialog.alert("Đổi mật khẩu thành công!");  
}  
}
```