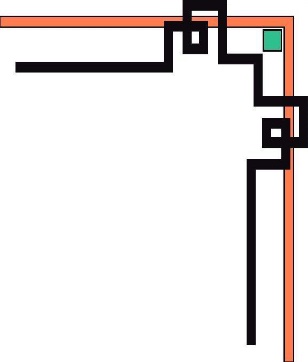
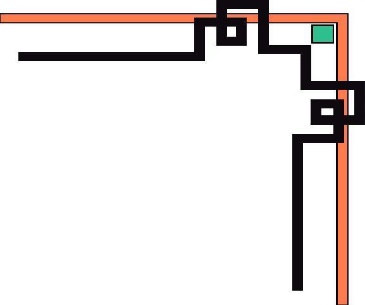
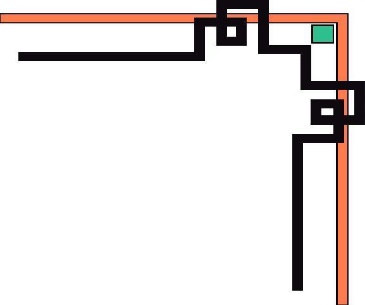
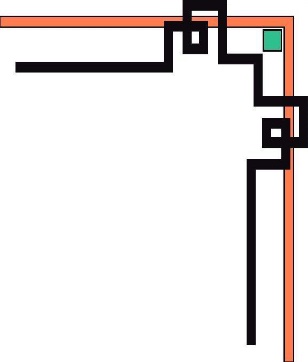
**TÀI LIỆU ĐẶC TẢ YÊU CẦU PHẦN MỀM** (Software Requirement Specification – SRS)



**CHI TIÊU HÀNG NGÀY**

Phiên bản 1.0.0

Giảng viên: Đinh Thành Nhân

Được soạn bởi: Nhóm 10

Thành viên: Trần Anh Khoa – PC02674

Cần Thơ, tháng 12 năm 2021

**Mục Lục**

[**1. Giới thiệu** 1](#_Toc90503586)

[1.1 Mục đích 1](#_Toc90503587)

[1.2 Phạm vi 1](#_Toc90503588)

[1.3 Từ viết tắt 1](#_Toc90503589)

[1.4 Tài liệu tham khảo 2](#_Toc90503590)

[**2. Phân tích** 2](#_Toc90503591)

[**2.1 Hiện trạng** 2](#_Toc90503592)

[**2.2 Yêu cầu hệ thống** 2](#_Toc90503593)

[**2.3 Các tác nhân** 3](#_Toc90503594)

[**2.4 Các chức năng của hệ thống** 3](#_Toc90503595)

[**2.5 Các ràng buộc** 5](#_Toc90503596)

[**2.6 Sơ đồ Use case** 6](#_Toc90503597)

[**2.7 Bản phân quyền Actor và Use case** 7](#_Toc90503598)

[**3. Thiết kế** 8](#_Toc90503599)

[**3.1 Thiết kế mô hình triển khai** 8](#_Toc90503600)

[**3.2 Thiết kế CSDL** 8](#_Toc90503601)

[**3.2.1 Sơ đồ quan hệ thực thể** 8](#_Toc90503602)

[**3.2.2 Sơ đồ ClassDiagram** 11](#_Toc90503603)

[**3.2.3 Thiết kế thực thể** 12](#_Toc90503604)

[**3.3 THIẾT KẾ GIAO DIỆN** 15](#_Toc90503605)

[**3.3.1 Sơ đồ tổ chức giao diện** 15](#_Toc90503606)

[**3.3.2 Thiết kế giao diện cho các chức năng nghiệp vụ** 17](#_Toc90503607)

[**3.3.3 Giao diện netbeans** 18](#_Toc90503608)

[**4. Thực hiện mã** 27](#_Toc90503609)

[**4.1 Tạo CSDL** 27](#_Toc90503610)

[**4.2 SQL truy vấn và thao tác bảng** 27](#_Toc90503611)

[**4.2.1 Tài khoản** 27](#_Toc90503612)

[**4.2.4 Danh mục chi** 30](#_Toc90503613)

[**4.2.5 Danh mục thu** 31](#_Toc90503614)

[**4.2.6 Loại ví** 32](#_Toc90503615)

[**4.2.7 Thu** 32](#_Toc90503616)

[**4.2.8 Chi** 33](#_Toc90503617)

[**4.2.9 Thống kê** 34](#_Toc90503618)

[**4.3 Lập trình JDBC** 35](#_Toc90503619)

[**4.3.1 Lớp hỗ trợ** 36](#_Toc90503620)

[**4.3.2 Model Class - Các lớp mô tả dữ liệu** 66](#_Toc90503621)

[**4.3.3 DAO Class – Các lớp truy xuất dữ liệu** 93](#_Toc90503622)

[**4.4 Viết mã ứng dụng** 151](#_Toc90503623)

[**4.4.3 Form Đăng ký** 167](#_Toc90503624)

[**4.4.4 Form Quên mật khẩu** 177](#_Toc90503625)

[**4.4.5 Form Báo cáo** 185](#_Toc90503626)

[**4.4.6 Form Người dùng** 231](#_Toc90503627)

[**4.4.7 Form Tài khoản** 248](#_Toc90503628)

[**4.4.8 Form Thu Chi** 256](#_Toc90503629)

[**4.4.9 Form Tổng quan** 283](#_Toc90503630)

[**4.4.10 Form Ví tiền** 294](#_Toc90503631)

[**4.4.11 Form giao diện chính** 306](#_Toc90503632)

[**5. Kiểm thử** 310](#_Toc90503633)

[**6. Đóng gói và triển khai** 310](#_Toc90503634)

[**6.1 HƯỚNG DẪN CHUYỂN ĐỔI JAR THÀNH EXE** 310](#_Toc90503635)

[**6.2 HƯỚNG DẪN CÀI ĐẶT TRIỂN KHAI** 311](#_Toc90503636)

[**6.3 HƯỚNG DẪN SỬ DỤNG PHẦN MỀM** 311](#_Toc90503637)

# **1. Giới thiệu**

## 1.1 Mục đích

Tài liệu đặc tả yêu cầu phần mềm này cung cấp một cái nhìn tổng quát mô tả đầy đủ về tất cả các yêu cầu, chức năng và thành phần của dự án phần mềm **chi tiêu hàng ngày.** Tài liệu này nhằm phục vụ cho các thành viên của nhóm phát triển dự án những người sẽ thực hiện và kiểm tra các hoạt động chính xác của hệ thống phần mềm.

## 1.2 Phạm vi

Tài liệu đặc tả yêu cầu phần mềm này được xây dựng nhằm phục vụ cho dự án **chi tiêu hàng ngày**.

Với việc lưu trữ hợp lý và tối giản cách ghi chép việc chi tiêu hàng ngày của từng cá nhân thì cung cấp đến cho từng cá nhân một công cụ để ghi chép và quản lý việc chi tiêu hàng ngày được đơn giản hơn. Tiết kiệm được kinh tế thay vì sử dụng sổ sách như xưa và chi phí sử dụng thấp nhưng vẫn đảm bảo tính toàn vẹn trong việc ghi chép chi tiêu.

## 1.3 Từ viết tắt

| **Thuật ngữ** | **Định nghĩa** | **Giải thích** |
| --- | --- | --- |
| CSDL | Cơ sở dữ liệu | Nơi lưu trữ thông tin và cho phép truy cập |
| DESC | Description | Mô tả |
| DEP | Dependency | Sự phụ thuộc |
| RAT | Rational | Quan hệ |
| MUST |  | Mức độ tối thiểu cần thiết để vượt qua các đợt kiểm thử |

## 1.4 Tài liệu tham khảo

* Phần mềm sổ thu chi misa - https://www.sothuchi.misa.vn/
* Phần mềm moneylover - https://www.moneylover.vn/

# **2. Phân tích**

## **2.1 Hiện trạng**

Công ty ghi chép chi tiêu cá nhân chuyên quản lý việc chi tiêu của từng khách hàng và báo cáo thống kê về khách hàng.

Việc ghi chép chi tiêu cá nhân đang được thực hiện thông qua excel thì do dữ liệu ngày càng lớn nên việc quản lý ngày càng khó khăn, dễ gặp sai sót mà không bảo mật.

## **2.2 Yêu cầu hệ thống**

Công ty mong muốn xây dựng một phần mềm để giải quyết khó khăn.

**Yêu cầu chức năng nghiệp vụ:**

* Quản lý người dùng.
* Quản lý việc thu và chi tiêu.
* Quản lý ví tiền.
* Thống kê việc thu, chi tiêu, thu chi, và tình hình tài chính hiện tại.
* Người dùng tự đăng ký sử dụng phần mềm.
* Xem thông tin tài khoản.

**Yêu cầu về bảo bảo mật**

* Tất cả mọi người dùng phải đăng nhập mới sử dụng được phần mềm.
* Admin được phép thực hiện tất cả chức năng.
* Người dùng không được quyền xóa chính tài khoản của họ, không được xem danh sách người dùng.

**Yêu cầu về môi trường công nghệ**

* Ứng dụng phải được thực với công nghệ Swing và JDBC chạy trên mọi hệ điều hành với môi trường JDK tối thiểu 1.8
* Hệ quản trị CSDL SQL Server 2018 trở lên.

## **2.3 Các tác nhân**

Hệ thống gồm có các tác nhân là: Người dùng và admin.

Khách hàng trực tiếp sử dụng phần mềm, tạo tài khoản để sử dụng những chức năng của phần mềm.

Admin sử dụng tất cả chức năng của phần mềm và chức năng người dùng để mà quản lý tài khoản người dùng.

## **2.4 Các chức năng của hệ thống**

Chức năng tài khoản

* Người dùng có nhu cầu xem thông tin tài khoản của mình 🡪 chọn tài khoản để xem thông tin của mình.

Chức năng tổng quan

* Người dùng xem tổng quan ví tiền của mình và tổng thu – chi gần đây là bao nhiêu 🡪 Chọn chức năng tổng quan.

Chức năng ví tiền

* Người dùng muốn tạo những ví tiền ảo để lưu lại số tiền hiện có của bản thân 🡪 Chọn chức năng ví tiền.
* Người dùng có thể cập nhật lại các ví tiền hiện có và có thể chọn đóng hoặc mở các ví tiền hiện có để xác định ví tiền đó của mình có còn sử dụng hay không

Chức năng thu - chi

* Người dùng có nhu cầu tạo các phiếu thu, phiếu chi để ghi lại những thu nhập và chi tiêu của mình 🡪 Chọn chức năng thu-chi.
* Người dùng có thể sửa lại hay xóa những phiếu thu, phiếu chi đó khi người dùng muốn.

Chức năng người dùng

* Admin đăng nhập vào hệ thống 🡪 Toàn quyền sử dụng hệ thống theo yêu cầu, tạo tài khoản cho người dùng mới, cập nhật, xóa tài khoản và tất cả các chức năng trong hệ thống.
* Người dùng đăng nhập vào hệ thống được phép sử dụng hết chức năng riêng chức năng quản lý người dùng thì chỉ có admin sử dụng.
* Người dùng khi quên mật khẩu 🡪 Sử dụng chức năng đổi mật khẩu để thay đổi mật khẩu mới.
* Quản lý khi quên mật khẩu 🡪 Sử dụng chức năng đổi mật khẩu để thay đổi mật khẩu mới.

Chức năng báo cáo

- Khi người dùng chọn chức năng báo cáo, hệ thống sẽ hiện ra những chức năng nhỏ bao gồm: Tình hình thu chi, phân tích chi tiêu, phân tích thu, tài chính hiện tại.

- Đối với chức năng tình hình thu chi: Hệ thống sẽ tổng hợp lại tất cả việc thu – chi theo từng năm từng tháng từng ngày để cho người dùng xem được mình đã thu và chi ra bao nhiêu và tỉ số thu chi đó là nhiều hay ít.

- Đối với chức năng phân tích chi tiêu: Hệ thống sẽ tổng hợp lại tất cả việc chi tiêu theo từng năm từng tháng từng ngày để cho người dùng xem được mình đã chi ra bao nhiêu.

- Đối với chức năng phân tích thu: Hệ thống sẽ tổng hợp lại tất cả việc thu theo từng năm từng tháng từng ngày để cho người dùng xem được mình đã thu vào bao nhiêu.

- Đối với chức năng tài chính hiện tại: Hệ thống sẽ tổng hợp tổng số tiền có trong tất cả các ví của người dùng và chia ra từng ví hiện còn bao nhiêu.

- Đối với từng chức năng đều có chức năng xuất file pdf để người dùng xem kĩ hơn.

## **2.5 Các ràng buộc**

* Hệ thống ràng buộc bởi yêu cầu của khách hàng: Khi khách hàng có yêu cầu sử dụng chức năng của phần mềm🡪 hệ thống mới cho phép các chức năng thực hiện yêu cầu của khách hàng
* Ràng buộc về dữ liệu đầu vào so với dữ liệu trong CSDL: Hàng hóa khách hàng chọn, được nhân viên nhập mã của các hàng hóa đó🡪 tất cả thông tin về hàng hóa được hiển thị lên giao diện.
* Ràng buộc về dữ liệu đầu vào so với dữ liệu trong CSDL: Mỗi người dùng cần có một mã người dùng riêng biệt để dễ dàng quản lý.
* Phần mềm chỉ được cài đặt trên hệ thống máy tính của người dùng, khi đó hệ thống phần mềm mới kết nối được CSDL của người dùng. Hoặc nếu không cài đặt trên hệ thống máy tính của người dùng mà cài đặt trên máy tính khác🡪 buộc phải có CSDL của người dùng thì mới thực hiện đầy đủ được các thao tác nghiệp vụ.

Để có thể hình dung rõ hơn về các tác nhân cũng như yêu cầu chức năng của hệ thống bằng cách mô hình hóa bằng sơ đồ Use Cases, các sơ đồ sẽ được trình bày phía sau.

## **2.6 Sơ đồ Use case**

Use case là sơ đồ tổng quan về mặt chức năng và phân vai trò người sử dụng. Dựa vào yêu cầu hệ thống của khách hàng, chúng ta có thể phác thảo sơ đồ use case như sau.



## **2.7 Bản phân quyền Actor và Use case**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Actor** | **Mô tả Actor** |
| 1 | Admin | Người dùng Admin |
| 2 | Người dùng | Người dùng |
|  |  |  |
|  |  |  |

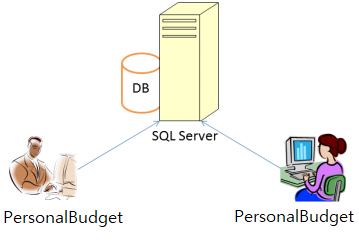
# **3. Thiết kế**

## **3.1 Thiết kế mô hình triển khai**

Ứng dụng phần mềm được xây dựng để phục vụ cho nhiều người dùng nhưng cơ sở dữ liệu thì lưu trữ tập trung.

Cần một máy cài SQL Server 2018.

Máy này cần hệ điều hành window 10 trở lên và cần JDK 1.8+. Máy người dùng cần cài phần mềm PersonalBudget.

****

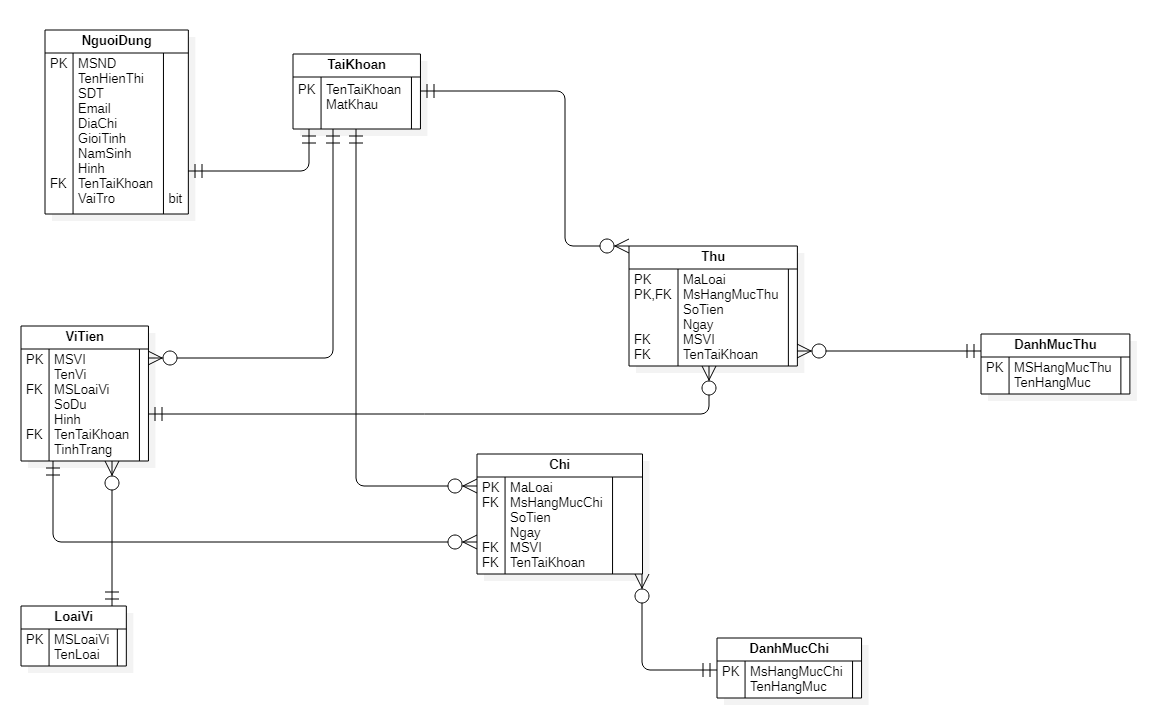
## **3.2 Thiết kế CSDL**

### **3.2.1 Sơ đồ quan hệ thực thể**

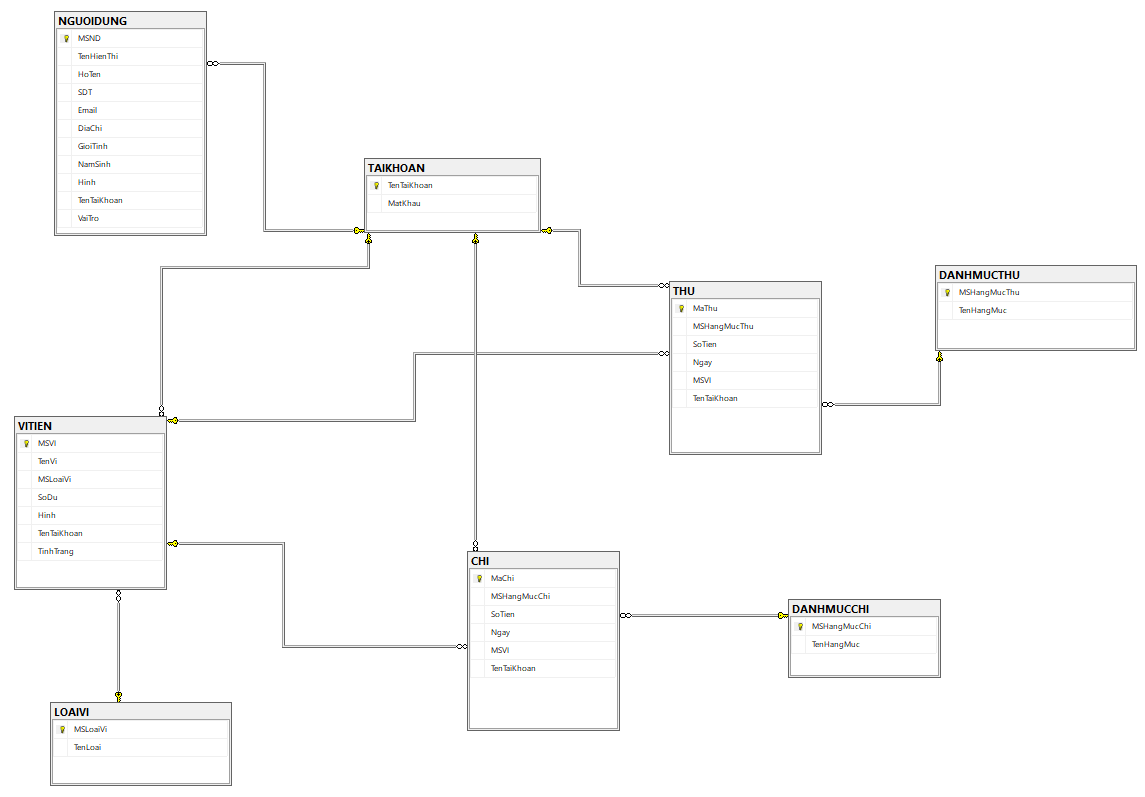
Trong hệ thống đã được mô tả trên chúng ta thấy có các thực thể sau.

* Người dùng: là người quản trị cần có thông tin đăng nhập đồng thời được ghi nhận mỗi khi người Dùng sử dụng chức năng để lưu.
* Tài khoản: Quản lý tài khoản để đăng nhập phần mềm.
* Ví tiền: Quản lý những ví tiền mà người dùng hay admin tạo ra.
* Loại ví: Quản lý những loại ví của ví tiền.
* Thu: Quản lý những thu nhập của người dùng hay admin tạo ra.
* Chi: Quản lý những chi tiêu của người dùng hay admin tạo ra.
* Danh mục thu: Quản lí những hạng mục thu cho quản lý thu.
* Danh mục chi: Quản lí những hạng mục thu cho quản lý chi tiêu.

Sau khi phân tích chúng ta có thể thấy các thực thể có mối quan hệ với nhau như sơ đồ CSDL và ERD sau:

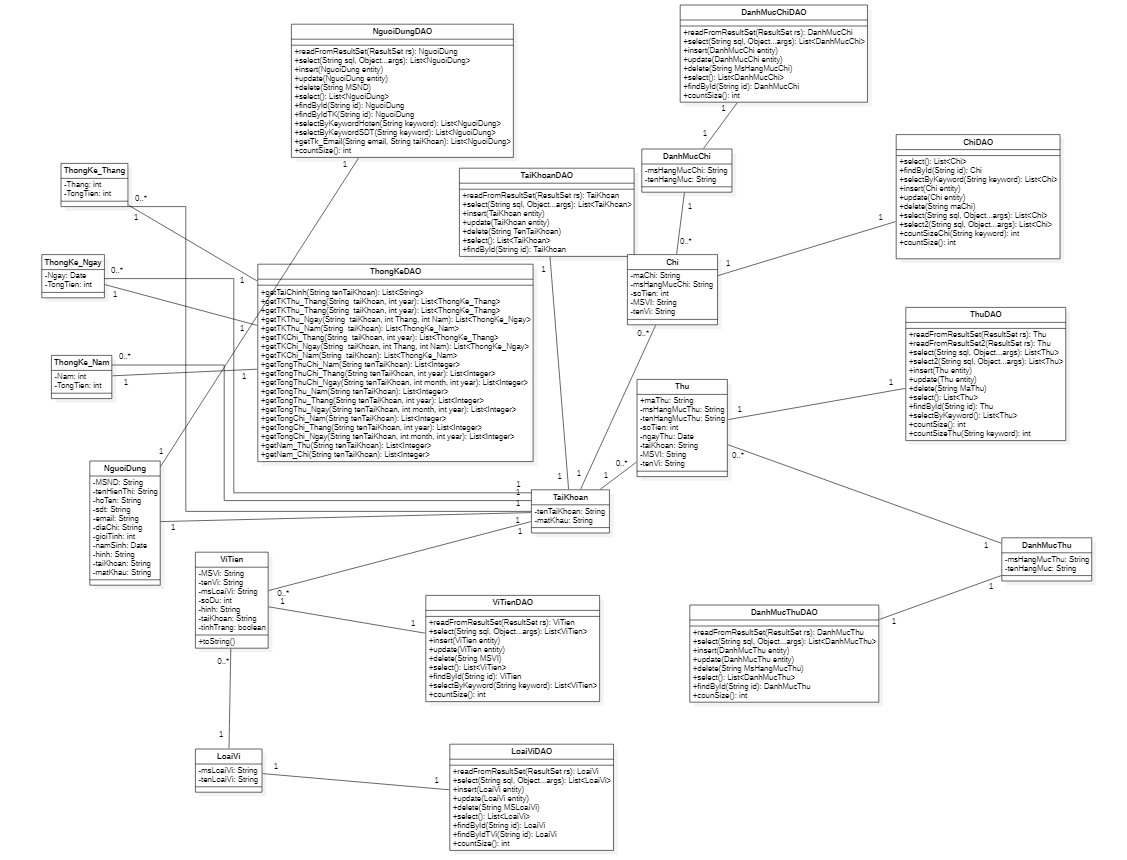


Hình 3.1 ERD diagram



Hình 3.2 CDSL

## **3.2.2 Sơ đồ ClassDiagram**



Hình 4 Chức năng quản lí nhân viên

## **3.2.3 Thiết kế thực thể**

Bảng CHI:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| MaChi | CHAR(50) | KHÓA CHÍNH | Mã chi |
| MSHangMucChi | CHAR(10) | NOT NULL | Mã số hạng mục chi |
| SoTien | MONEY | NOT NULL | Số tiền chi |
| Ngay | DATE | NOT NULL | Ngày chi |
| MSVI | CHAR(10) | KHÓA NGOẠI | Mã số ví |
| TenTaiKhoan | NVARCHAR(25) | KHÓA NGOẠI | Tên tài khoản |

Bảng DANHMUCCHI:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| MSHangMucChi | CHAR(10) | KHÓA CHÍNH | Mã số hạng mục chi |
| TenHangMuc | NVARCHAR(50) | NOT NULL | Tên hạng mục |

Bảng DANHMUCTHU:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tên cột | | Kiểu dữ liệu | Ràng buộc | | | Ghi chú |
| MSHangMucThu | CHAR(10) | | | KHÓA CHÍNH | Mã số hạng mục thu | |
| TenHangMuc | NVARCHAR(50) | | | NOT NULL | Tên hạng mục | |

Bảng LOAIVI:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| MSLoaiVi | CHAR(10) | KHÓA CHÍNH | Mã số loại ví |
| TenLoai | NVARCHAR(50) | NOT NULL | Tên loại ví |

Bảng TAIKHOAN:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| TenTaiKhoan | NVARCHAR(25) | KHÓA CHÍNH | Tên tài khoản |
| MatKhau | NVARCHAR (20) | NOT NULL | Mật khẩu |

Bảng THU

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| MaThu | CHAR(50) | KHÓA CHÍNH | Mã thu |
| MSHangMucThu | CHAR(10) | KHÓA NGOẠI | Mã số hạng mục thu |
| SoTien | MONEY | NOT NULL | Số tiền thu |
| Ngay | DATE | NOT NULL | Ngày thu |
| MSVI | CHAR(10) | KHÓA NGOẠI | Mã số ví |
| TenTaiKhoan | NVARCHAR(25) | KHÓA NGOẠI | Tên tài khoản |

Bảng VITIEN

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| MSVI | CHAR(10) | KHÓA CHÍNH | Mã số ví |
| TenVi | NVARCHAR(50) | NOT NULL | Tên ví |
| MSLoaiVi | CHAR(10) | KHÓA NGOẠI | Mã số loại ví |
| SoDu | MONEY | NOT NULL | Số dư |
| Hinh | NVARCHAR(50) | NULL | Hình |
| TenTaiKhoan | NVARCHAR(25) | KHÓA NGOẠI | Tên tài khoản |
| TinhTrang | BIT | NOT NULL | Tình trạng |

Bảng NGUOIDUNG

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| MSND | CHAR(15) | KHÓA CHÍNH | Mã số người dùng |
| TenHienThi | NVARCHAR(50) | NOT NULL | Tên hiển thị |
| HoTen | NVARCHAR(50) | NOT NULL | Họ tên |
| NamSinh | DATE | NOT NULL | Năm sinh |
| GioiTinh | INT | NOT NULL | Giới tính |
| DiaChi | NVARCHAR(50) | NOT NULL | Địa chỉ |
| Email | NVARCHAR(50) | UNIQUE | Email |
| SDT | NVARCHAR (15) | UNIQUE | Số điện thoại |
| Hinh | NVARCHAR(50) | NULL | Hình |
| TenTaiKhoan | NVARCHAR(25) | KHÓA NGOẠI | Tên tài khoản |
| VaiTro | BIT | NOT NULL | Vai trò |

## **3.3 THIẾT KẾ GIAO DIỆN**

### **3.3.1 Sơ đồ tổ chức giao diện**

Giao diện chính là một cửa sổ chứa menu chính và toolbar. Thông qua đó để đi đến các giao diện thành viên để thực hiện các chức năng trong hệ thống.

Theo yêu cầu thì mọi người dùng phải đăng nhập trước khi sử dụng ứng dụng nên form đăng nhập xuất hiện trước để yêu cầu đăng nhập.

Ngoài ra mỗi ứng dụng trong thời gian khởi động cần có một màn hình chào cùng với thanh tiến trình để người có cảm giác ứng dụng đang khởi động.

Giao diện chính

Tài khoản

Thu-Chi

Tổng hợp báo cáo

Màn hình chào

Đăng nhập

Đăng kí

Đổi mật khẩu

Đăng xuất

Quản lí người dùng

Ví tiền

Giới thiệu

Hướng dẫn

Tổng quan

### **3.3.2 Thiết kế giao diện cho các chức năng nghiệp vụ**

Trong phần thiết kế giao diện các bạn nên sử dụng gói hình ảnh được cung cấp sẵn trên tài nguyên học tập của LMS. Nếu các bạn không muốn sử dụng gói hình ảnh này thì bạn có thể chọn gói hình ảnh phù hợp của các bạn. Lưu ý quan trọng:

Các bạn phải đặt tên đúng như hướng dẫn vì các tên này ảnh hưởng đến việc viết mã sau này. Nếu các bạn đặt tên khác thì phải chỉnh mã sau này theo tên mà các bạn đã đặt.

Trong lập trình giao diện người ta thường qui ước tiếp đầu ngữ của tên các thành phần giao diện như sau

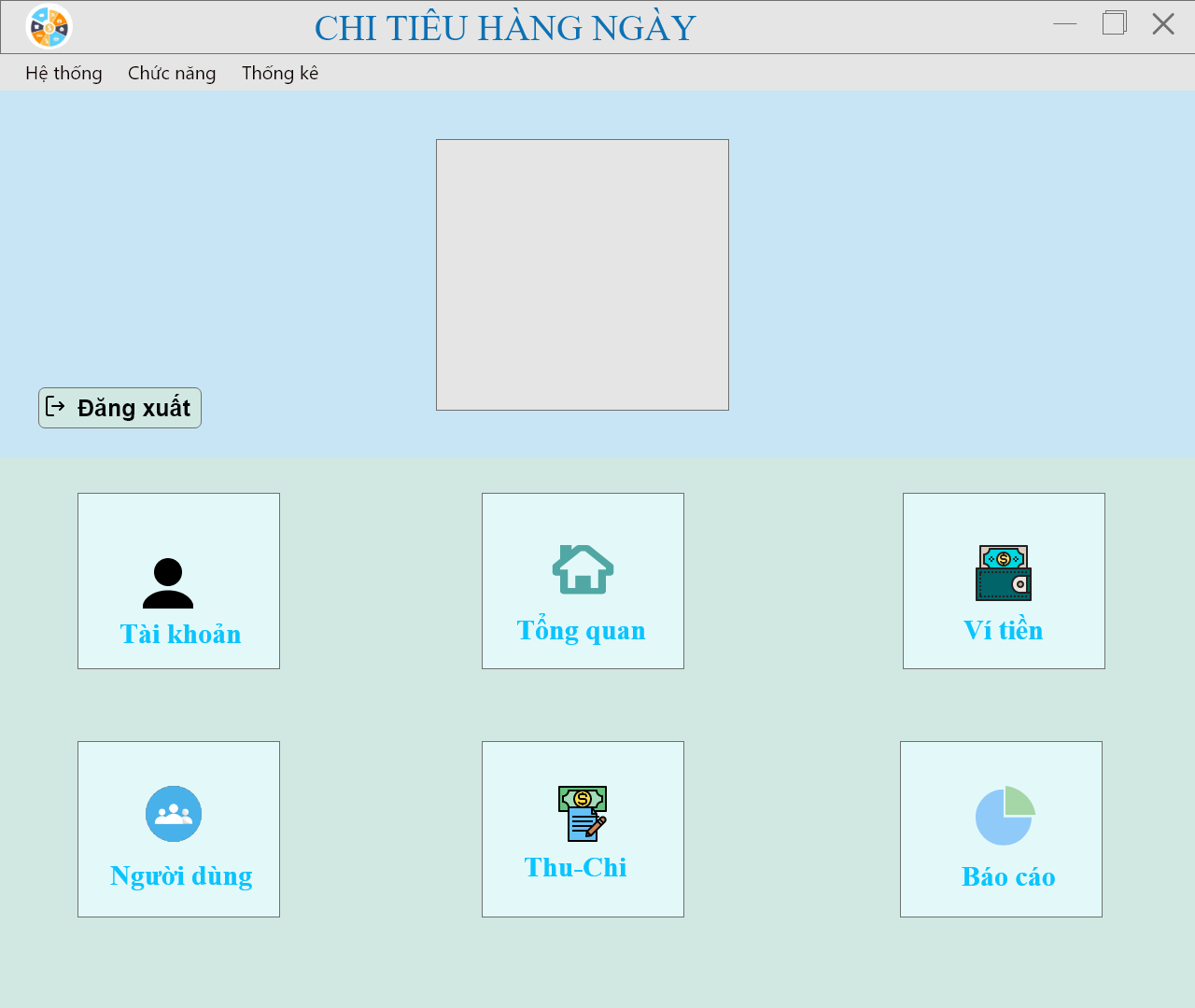
|  |  |  |
| --- | --- | --- |
| Control | Prefix | Mô tả |
| JLabel | lbl | Nhãn |
| JButton | btn | Nút |
| JTextField, JTextArea, JPassword, JTextPane | txt | Các loại ô nhập |
| JRadioButton | rdo | Hộp chọn |
| JCheckBox | chk | Hộp kiểm |
| ButtonGroup | bgr | Nhóm các button |
| JComboBox | cbo | Hộp chọn xổ xuống |
| JListBox | lst | Hộp chọn nhiều |
| JTree | tree | Cây |
| JMenu | mnu | Mục lục |
| JMenuItem | mni | Mục menu |
| JPanel | pnl | Khung chứa |
| JTabbedPane | tab |  |
| JTable | tbl |  |
| ………. |  |  |

### **3.3.3 Giao diện netbeans**

**Giao diện đăng nhập**



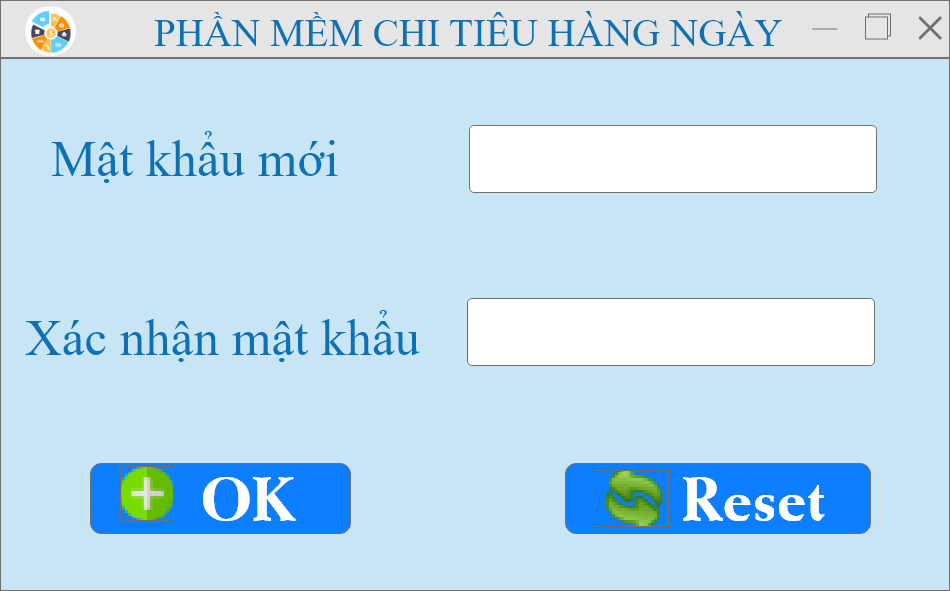
**Giao diện màn hình chính**



**Giao diện đăng ký**



**Giao diện mật khẩu**

**Giao diện tổng quan**



**Giao diện ví tiền**





**Giao diện tài khoản**



**Giao diện thu chi**

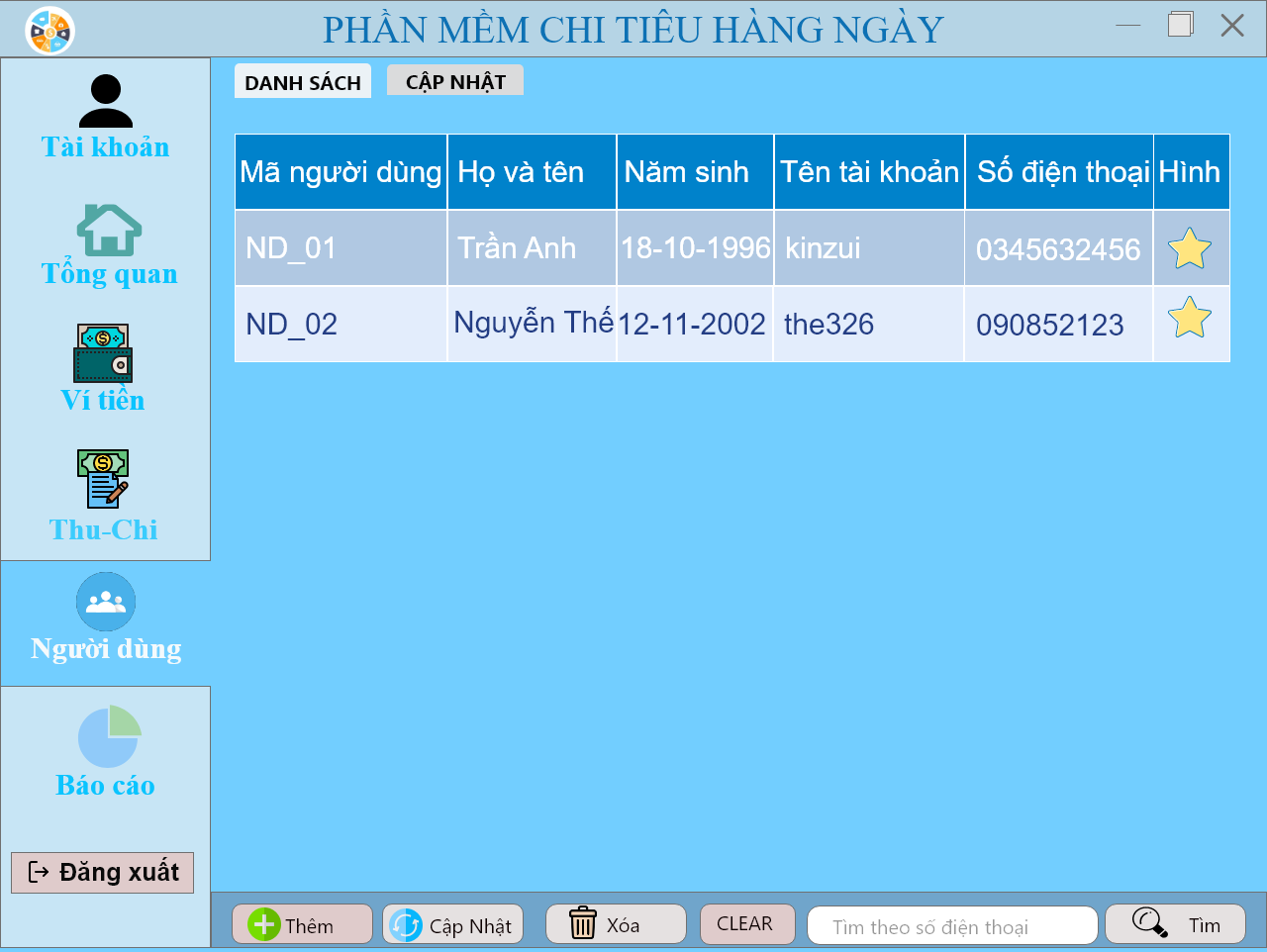






**Giao diện người dùng - Admin**

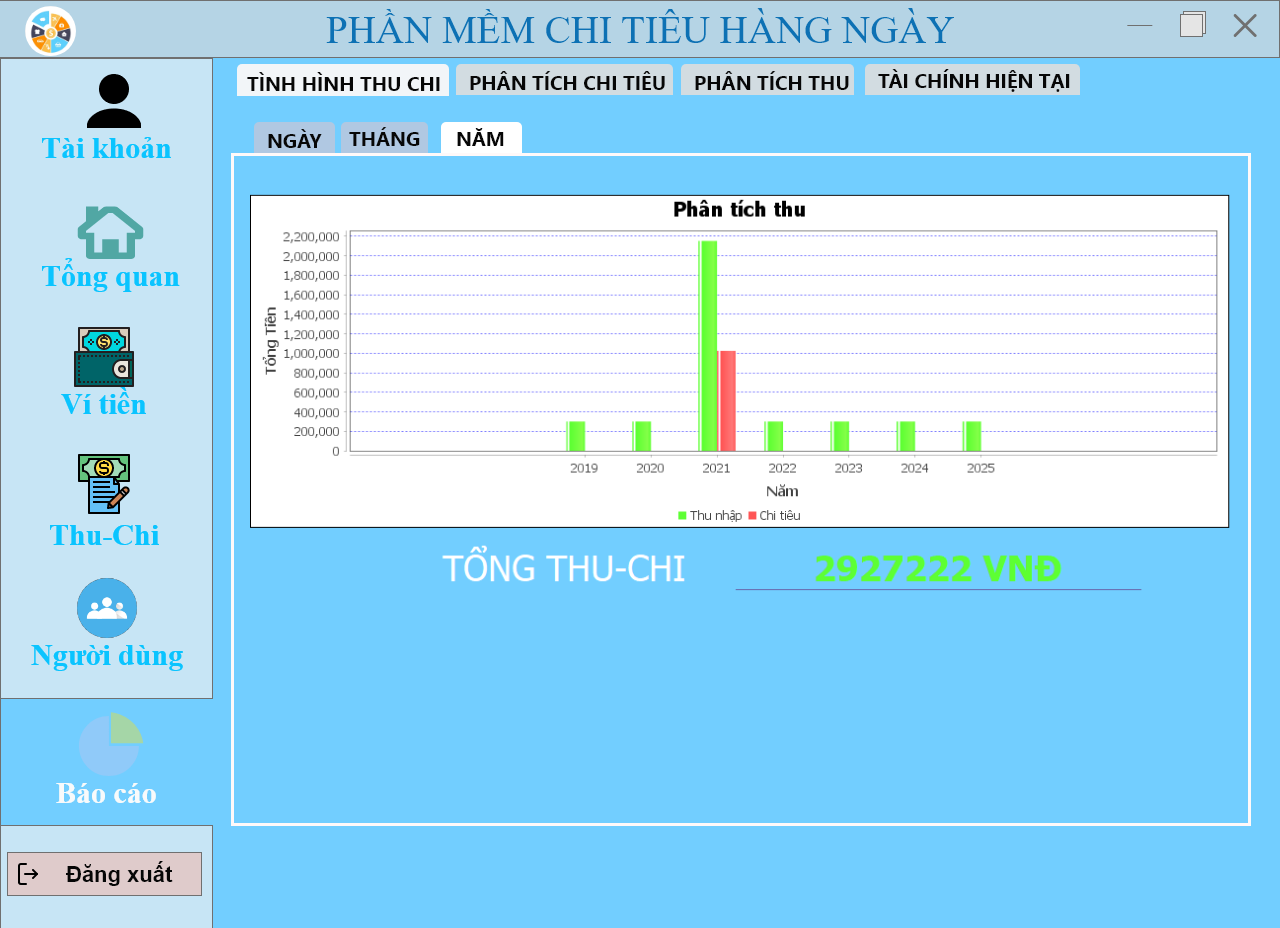




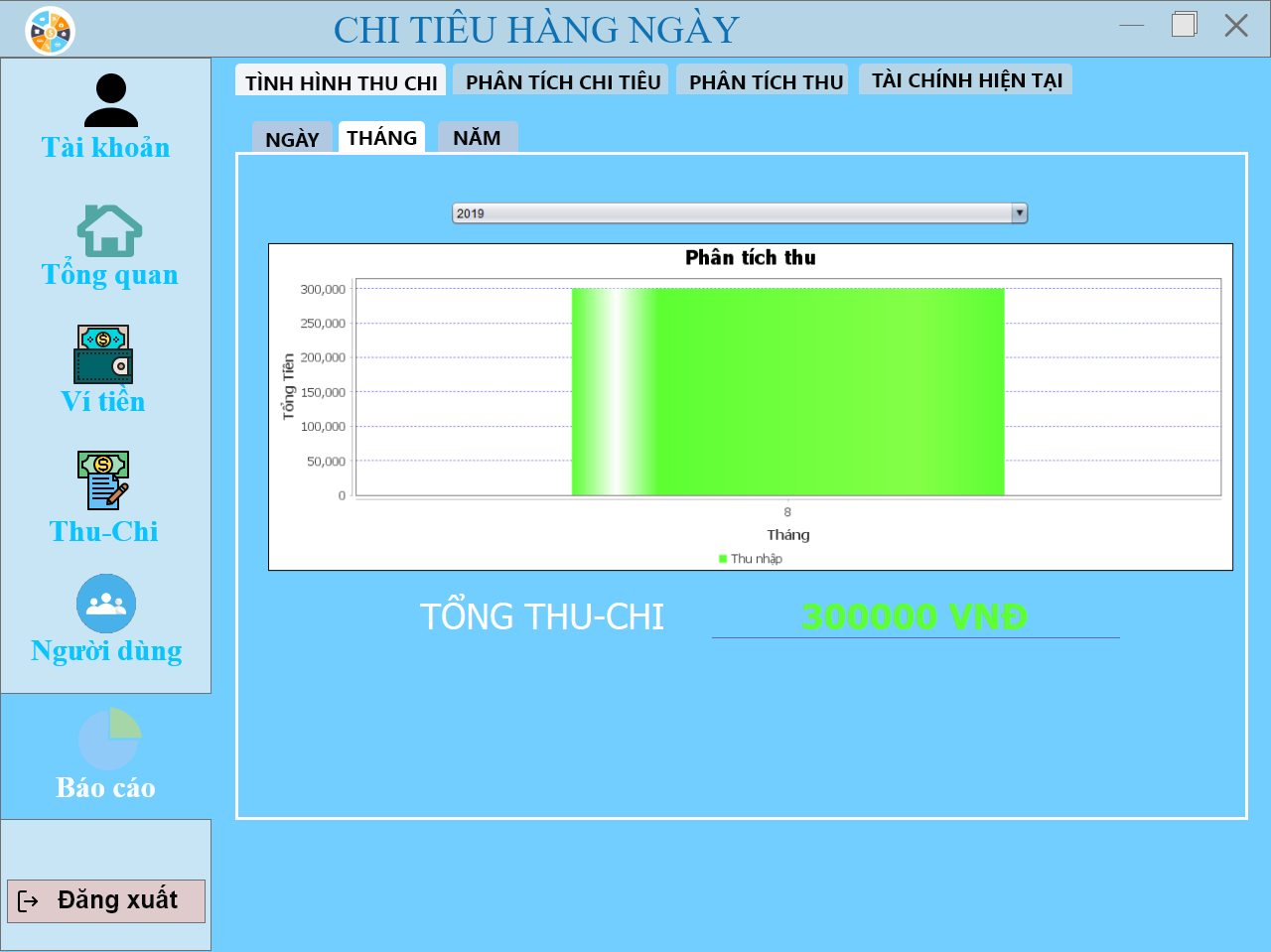
**Giao diện người dùng – Người dùng**



**Giao diện báo cáo**

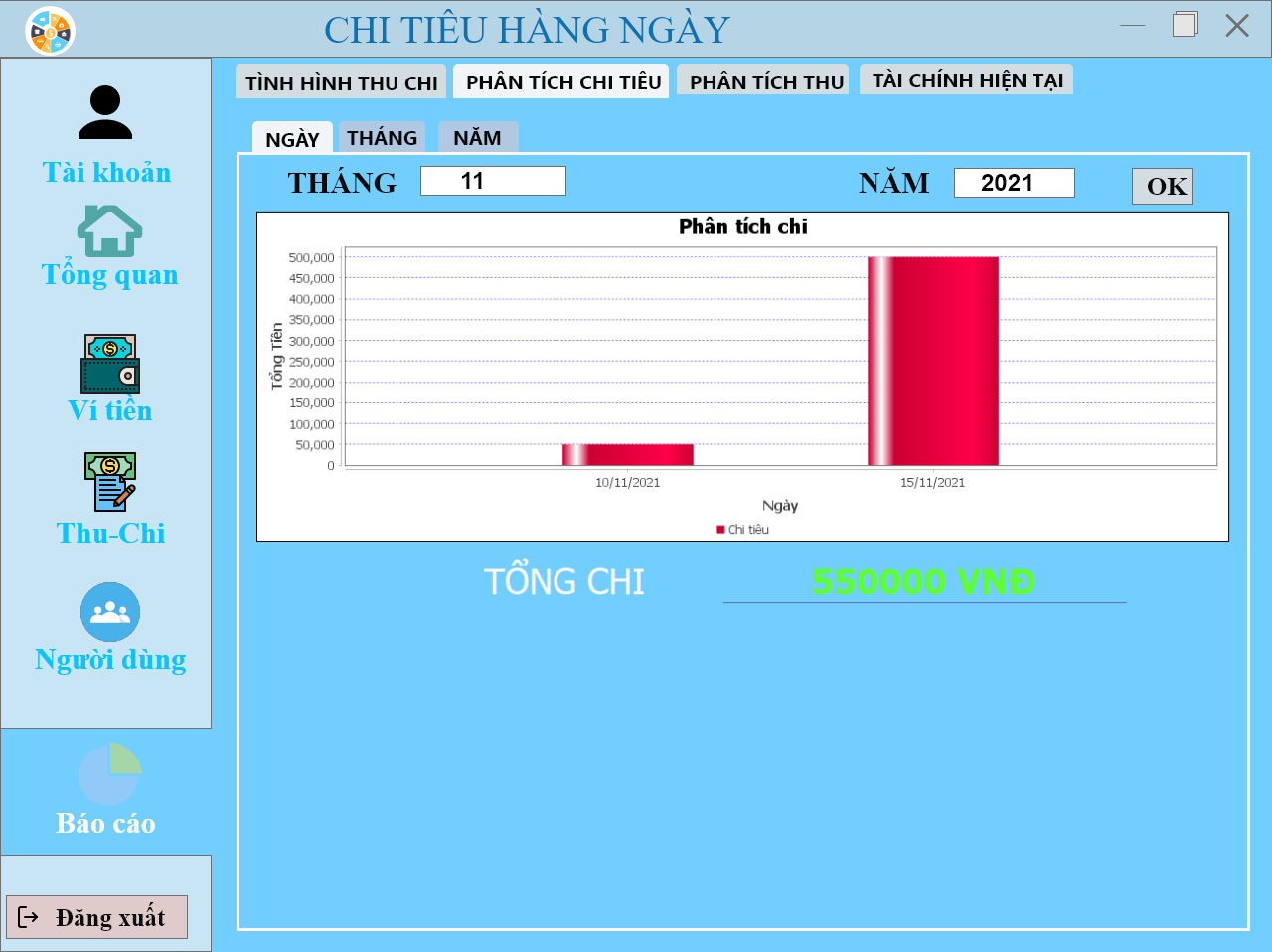






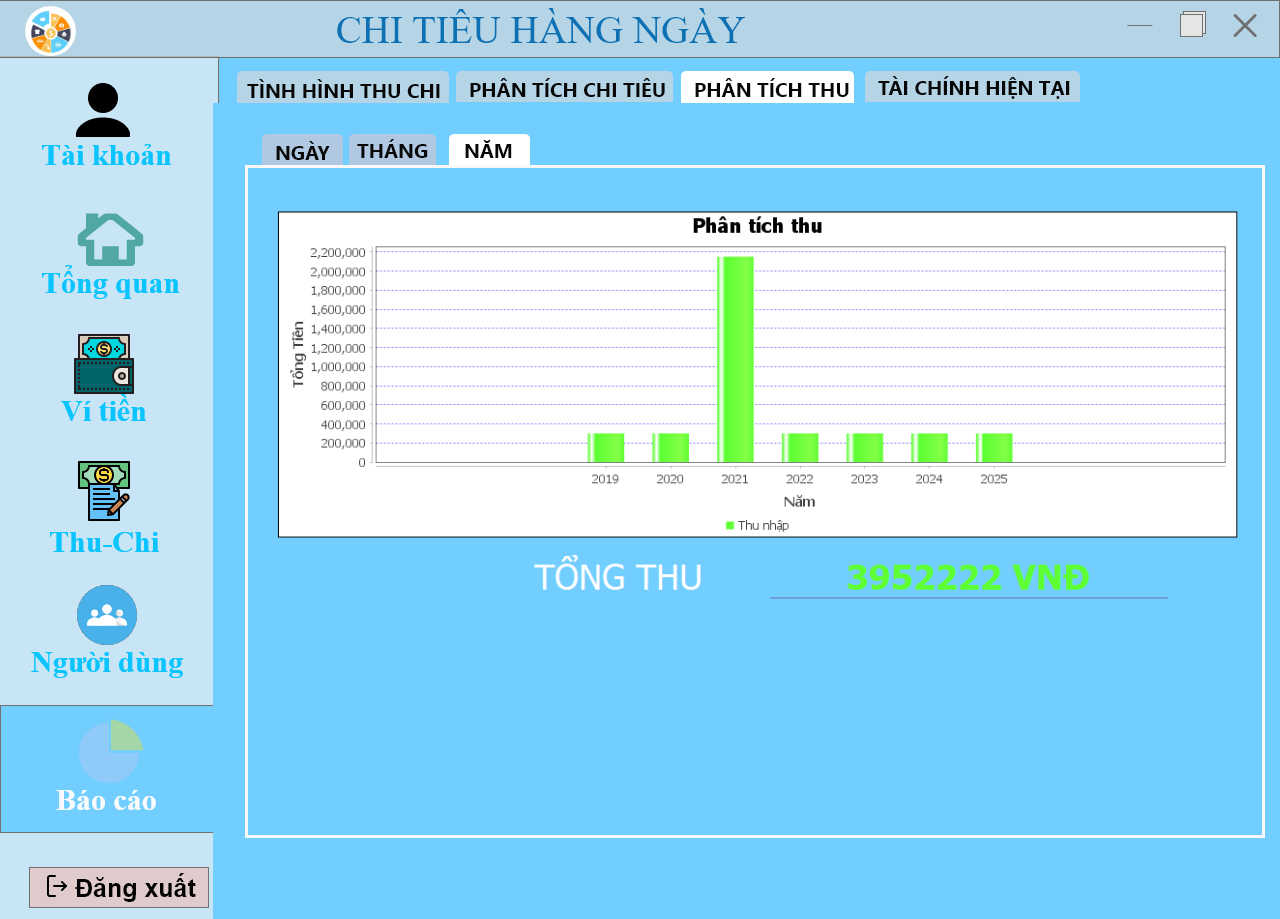


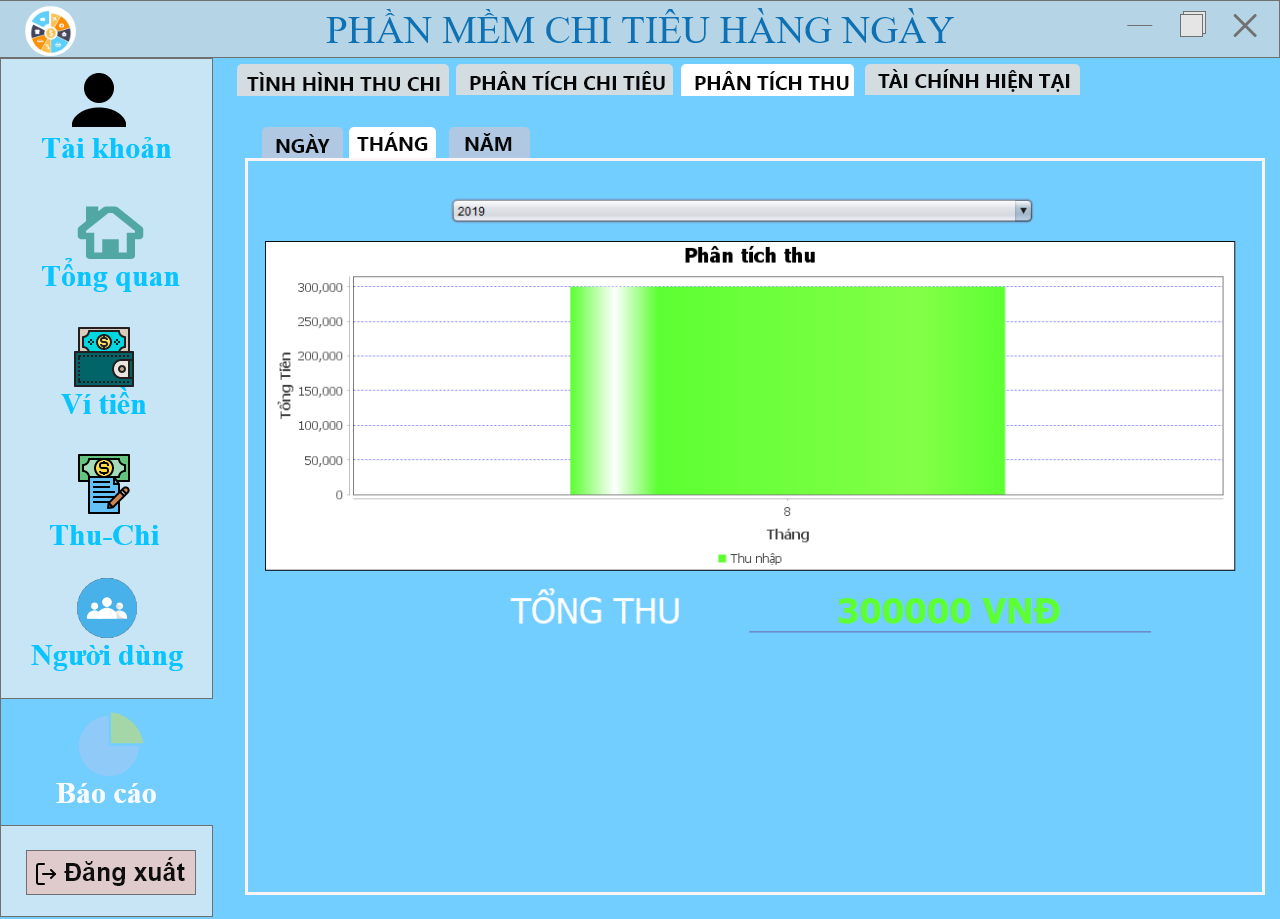












# **4. Thực hiện mã**

## **4.1 Tạo CSDL**

Do việc thiết kế CSDL của nhóm sử dụng phương thức tạo bảng bằng cách design là công cụ SQL Server đã hỗ trợ sẵn, sử dụng công cụ này có thể giúp nhóm giảm nhẹ được số lượng công việc và làm 1 cách nhanh chóng.

## **4.2 SQL truy vấn và thao tác bảng**

### **4.2.1 Tài khoản**

|  |  |
| --- | --- |
| Hành động | Cậu lệnh SQL |
| Thêm mới | INSERT INTO TAIKHOAN (TenTaiKhoan, MatKhau) VALUES (?, ?) |
| Cập nhật theo tài khoản | UPDATE TAIKHOAN Set MatKhau = ? where TenTaiKhoan = ? |
| Xóa theo tài khoản | DELETE FROM TAIKHOAN WHERE TenTaiKhoan like ? |
| Truy vấn tất cả | SELECT \* FROM TAIKHOAN |
| Truy vấn theo tài khoản | SELECT \* FROM TAIKHOAN WHERE TenTaiKhoan like ? |

**4.2.2 Người dùng**

|  |  |
| --- | --- |
| Hành động | Cậu lệnh SQL |
| Thêm mới | INSERT INTO TaiKhoan (TenTaiKhoan, MatKhau) VALUES (?, ?)  INSERT INTO NguoiDung (MSND, TenHienThi, HoTen, SDT, Email, DiaChi, GioiTinh, NamSinh, Hinh, TenTaiKhoan) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?) |
| Cập nhật theo tài khoản và mã | UPDATE TaiKhoan SET MatKhau= ? WHERE TenTaiKhoan = ?  UPDATE NguoiDung Set TenHienThi = ?, HoTen = ?, SDT = ?, Email = ?, DiaChi = ?, GioiTinh = ?, NamSinh = ?, Hinh = ?, TenTaiKhoan = ? where MSND = ? |
| Xóa theo mã | DELETE FROM NGUOIDUNG WHERE MSND = ? |
| Truy vấn tất cả | SELECT NGUOIDUNG.MSND, NGUOIDUNG.TenHienThi, NGUOIDUNG.HoTen,NGUOIDUNG.SDT, NGUOIDUNG.Email, NGUOIDUNG.DiaChi, NGUOIDUNG.GioiTinh, NGUOIDUNG.NamSinh, NGUOIDUNG.Hinh, NGUOIDUNG.TenTaiKhoan, TAIKHOAN.MatKhau FROM NGUOIDUNG , TAIKHOAN WHERE NGUOIDUNG.TenTaiKhoan = TAIKHOAN.TenTaiKhoan |
| Truy vấn theo mã | SELECT NGUOIDUNG.MSND, NGUOIDUNG.TenHienThi, NGUOIDUNG.HoTen, NGUOIDUNG.SDT, NGUOIDUNG.Email, NGUOIDUNG.DiaChi, NGUOIDUNG.GioiTinh, NGUOIDUNG.NamSinh, NGUOIDUNG.Hinh, NGUOIDUNG.TenTaiKhoan, TAIKHOAN.MatKhau FROM NGUOIDUNG, TAIKHOAN "  WHERE NGUOIDUNG.TenTaiKhoan = TAIKHOAN.TenTaiKhoan and NGUOIDUNG.MSND = ? |
| Truy vấn theo số điện thoại | SELECT NGUOIDUNG.MSND, NGUOIDUNG.TenHienThi, NGUOIDUNG.HoTen, NGUOIDUNG.SDT, NGUOIDUNG.Email, NGUOIDUNG.DiaChi, NGUOIDUNG.GioiTinh, NGUOIDUNG.NamSinh, NGUOIDUNG.Hinh, NGUOIDUNG.TenTaiKhoan, TAIKHOAN.MatKhau FROM NGUOIDUNG, TAIKHOAN "  WHERE NGUOIDUNG.TenTaiKhoan = TAIKHOAN.TenTaiKhoan and NGUOIDUNG.SDT like ? |
| Gọi Stored procedure tìm tài khoản và email | call sp\_timTaiKhoan\_Email (?,?) |

**4.2.3 Ví tiền**

|  |  |
| --- | --- |
| Hành động | Cậu lệnh SQL |
| Thêm mới | INSERT INTO VITIEN (MSVI, TenVi, MSLoaiVi, SoDu, hinh, TenTaiKhoan, TinhTrang) VALUES (?, ?, ?, ?, ?, ?, ?) |
| Cập nhật theo mã | UPDATE VITIEN Set TenVi = ?, MSLoaiVi = ?, SoDu = ?, hinh = ?, TenTaiKhoan = ?, TinhTrang = ? where MSVI = ? |
| Cập nhật tiền theo mã | UPDATE VITIEN Set SoDu = ? where MSVI = ? |
| Xóa theo mã | DELETE FROM VITIEN WHERE MSVI = ? |
| Truy vấn tất cả | select VITIEN.MSVI, VITIEN.TenVi, VITIEN.MSLoaiVi, LOAIVI.TenLoai, VITIEN.SoDu, VITIEN.Hinh,VITIEN.TenTaiKhoan, VITIEN.TinhTrang from VITIEN , LOAIVI where VITIEN.MSLoaiVi = LOAIVI.MSLoaiVi |
| Truy vấn theo tài khoản | select VITIEN.MSVI, VITIEN.TenVi, VITIEN.MSLoaiVi, LOAIVI.TenLoai, VITIEN.SoDu, VITIEN.Hinh,VITIEN.TenTaiKhoan, VITIEN.TinhTrang from VITIEN , LOAIVI "  WHERE VITIEN.MSLoaiVi = LOAIVI.MSLoaiVi and TenTaiKhoan like ? |

### **4.2.4 Danh mục chi**

|  |  |
| --- | --- |
| Hành động | Cậu lệnh SQL |
| Thêm mới | INSERT INTO DANHMUCCHI (MSHangMucChi, TenHangMuc) VALUES (?, ?) |
| Cập nhật theo mã | UPDATE DANHMUCCHI Set TenHangMuc = ? where MSHangMucChi = ? |
| Xóa theo mã | DELETE FROM DANHMUCCHI WHERE MSHangMucChi like ? |
| Truy vấn tất cả | SELECT \* FROM DANHMUCCHI |
| Truy vấn theo mã | SELECT \* FROM DANHMUCCHI WHERE MSHangMucChi = ? |

### **4.2.5 Danh mục thu**

|  |  |
| --- | --- |
| Hành động | Cậu lệnh SQL |
| Thêm mới | INSERT INTO DANHMUCTHU (MSHangMucThu, TenHangMuc) VALUES (?, ?) |
| Cập nhật theo mã | UPDATE DANHMUCTHU Set TenHangMuc = ? where MSHangMucThu = ? |
| Xóa theo mã | DELETE FROM DANHMUCTHU WHERE MSHangMucThu like ? |
| Truy vấn tất cả | SELECT \* FROM DANHMUCTHU |
| Truy vấn theo mã | SELECT \* FROM DANHMUCTHU WHERE MSHangMucThu = ? |

### **4.2.6 Loại ví**

|  |  |
| --- | --- |
| Hành động | Cậu lệnh SQL |
| Thêm mới | INSERT INTO LOAIVI (MSLoaiVi, TenLoai) VALUES (?, ?) |
| Cập nhật theo mã | UPDATE LOAIVI Set TenLoai = ? WHERE MSLoaiVi = ? |
| Xóa theo mã | DELETE FROM LOAIVI WHERE MSLoaiVi = ? |
| Truy vấn tất cả | SELECT \* FROM LOAIVI |
| Truy vấn theo mã | SELECT \* FROM LOAIVI WHERE MSLoaiVi = ? |
| Truy vấn theo tên ví | SELECT \* FROM LOAIVI WHERE TenVi = ? |

### **4.2.7 Thu**

|  |  |
| --- | --- |
| Hành động | Cậu lệnh SQL |
| Thêm mới | INSERT INTO THU (MaThu, MsHangMucThu, SoTien, Ngay, TenTaiKhoan, MSVI) VALUES (?, ?, ?, ?, ?, ?) |
| Cập nhật theo mã | UPDATE THU Set MsHangMucThu = ?, SoTien= ?, Ngay= ?, TenTaiKhoan = ?, MSVI=? WHERE MaThu = ? |
| Xóa theo mã | DELETE FROM THU WHERE MaThu = ? |
| Truy vấn tất cả | SELECT \* FROM THU |
| Truy vấn theo mã | SELECT THU.MaThu, THU.MSHangMucThu, DANHMUCTHU.TenHangMuc, THU.SoTien, THU.Ngay, THU.TenTaiKhoan, THU.MSVI, VITIEN.TenVi from (THU INNER JOIN DANHMUCTHU ON THU.MSHangMucThu = DANHMUCTHU.MSHangMucThu inner join VITIEN ON THU.MSVI = VITIEN.MSVI)" WHERE THU.MaThu = ? |
| Truy vấn theo tài khoản | SELECT THU.MaThu, THU.MSHangMucThu, DANHMUCTHU.TenHangMuc, THU.SoTien, THU.Ngay, THU.TenTaiKhoan, THU.MSVI, VITIEN.TenVi from (THU INNER JOIN DANHMUCTHU ON THU.MSHangMucThu = DANHMUCTHU.MSHangMucThu inner join VITIEN ON THU.MSVI = VITIEN.MSVI) WHERE THU.TenTaiKhoan like ? |

### **4.2.8 Chi**

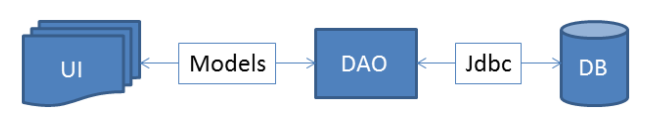
|  |  |
| --- | --- |
| Hành động | Cậu lệnh SQL |
| Thêm mới | INSERT INTO CHI (MaChi, MsHangMucChi, SoTien, Ngay, TenTaiKhoan, MSVI) VALUES (?, ?, ?, ?, ?, ?) |
| Cập nhật theo mã | UPDATE CHI Set MsHangMucChi = ?, SoTien= ?, Ngay= ?, TenTaiKhoan = ?, MSVI=? WHERE MaChi = ? |
| Xóa theo mã | DELETE FROM CHI WHERE MaChi = ? |
| Truy vấn tất cả | SELECT \* FROM CHI |
| Truy vấn theo mã | select CHI.MaChi, CHI.MSHangMucChi, DANHMUCCHI.TenHangMuc, CHI.SoTien, CHI.Ngay, CHI.TenTaiKhoan, CHI.MSVI, VITIEN.TenVi from (CHI INNER JOIN DANHMUCCHI ON CHI.MSHangMucChi = DANHMUCCHI.MSHangMucChi INNER JOIN VITIEN ON CHI.MSVI = VITIEN.MSVI) " WHERE CHI.MaChi like ? |
| Truy vấn theo tài khoản | SELECT CHI.MaChi, CHI.MSHangMucChi, DANHMUCCHI.TenHangMuc, CHI.SoTien, CHI.Ngay, CHI.TenTaiKhoan, CHI.MSVI, VITIEN.TenVi from (CHI INNER JOIN DANHMUCCHI ON CHI.MSHangMucChi = DANHMUCCHI.MSHangMucChi INNER JOIN VITIEN ON CHI.MSVI = VITIEN.MSVI) WHERE CHI.TenTaiKhoan like ? |

### **4.2.9 Thống kê**

|  |  |
| --- | --- |
| Hành động | Cậu lệnh SQL |
| Gọi Stored procedure thống kê thu theo tháng | call sp\_ThongKeThu\_Thang (?,?) |
| Gọi Stored procedure thống kê thu theo ngày | call sp\_ThongKeThu\_Ngay (?,?,?) |
| Gọi Stored procedure thống kê thu theo năm | call sp\_ThongKeThu\_Nam (?) |
| Gọi Stored procedure thống kê chi theo tháng | call sp\_ThongKeChi\_Thang (?,?) |
| Gọi Stored procedure thống kê chi theo ngày | call sp\_ThongKeChi\_Ngay (?,?,?) |
| Gọi Stored procedure thống kê chi theo năm | call sp\_ThongKeChi\_Nam (?) |
| Gọi Stored procedure tổng thu chi theo năm | call sp\_TongThuChi\_Nam (?) |
| Gọi Stored procedure tổng thu chi theo tháng | call sp\_TongThuChi\_Thang (?,?) |
| Gọi Stored procedure tổng thu chi theo ngày | call sp\_TongThuChi\_Ngay (?,?, ?) |
| Gọi Stored procedure tổng thu theo năm | call sp\_TongThu\_Nam (?) |
| Gọi Stored procedure tổng thu theo tháng | call sp\_TongThu\_Thang (?,?) |
| Gọi Stored procedure tổng thu theo ngày | call sp\_TongThu\_Ngay (?,?, ?) |
| Gọi Stored procedure tổng chi theo năm | call sp\_TongChi\_Nam (?) |
| Gọi Stored procedure tổng chi theo tháng | call sp\_TongChi\_Thang (?,?) |
| Gọi Stored procedure tổng chi theo ngày | call sp\_TongChi\_Ngay (?,?, ?) |
| Gọi Stored procedure năm thu | call sp\_SoNam\_Thu (?) |
| Gọi Stored procedure năm chi | call sp\_SoNam\_Chi (?) |

### **4.3 Lập trình JDBC**

Trong dự án này chúng ta chọn mô hình lập trình với CSDL như hình sau:

****

Trong đó:

* UI: các thành phần giao diện.
* DB: cơ sở dữ liệu.
* Jdbc: là lớp tiện ích cung cấp các hàm tiện ích làm việc với CSDL thông qua câu.lệnh sql hoặc lời gọi thủ tục lưu.
* Model: là các lớp mô tả dữ liệu theo cấu trúc các bảng trong CSDL.
* DAO: là các lớp thao tác và truy vấn dữ liệu. Nó có nhiệm vụ chuyển đổi Model sang SQL và ngược lại.

Theo mô hình này thì lập trình giao diện làm việc với CSDL thông qua các lớp model. Model như là một gói dữ liệu được sử dụng để giao tiếp giữa giao diện và các thành phần DAO. DAO mới thực sự là lập trình CSDL nó làm nhiệm vụ nhận model từ giao diện, chuyển đổi sang SQL và sử dụng Jdbc để thực hiện các thao tác dữ liệu. Ngược lại thì truy vấn dữ liệu từ CSDL, chuyển đổi thành model để hiển thị lên các thành phần giao diện.

*Việc tách thành các phần riêng biết giúp quản lý dự án tốt hơn, dễ bảo trì nâng cấp hơn.*

**4.3.1 Lớp hỗ trợ**

Các lớp hỗ trợ chứa các phương thức tiện ích giúp chúng ta thực hiện các xử lý trong ứng dụng đơn giản hơn, ngắn gọn hơn, dễ hiểu hơn. Trong phần này chúng ta xây dựng các lớp tiện ích như sau.

4.3.1.1 Lớp tiện ích DateHelper

Chứa các phương thức chuyển đổi thời gian và chuỗi giúp lập trình sau này khi xử lý thời gian dễ dàng hơn.

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Helper;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  /\*\*  \*  \* @author Admin  \*/  public class dateHelper {  public static final SimpleDateFormat DATE\_FORMATER = new SimpleDateFormat("dd/MM/yyyy");  //chuuyển String sang Date  /\*  @param date truyền vào date kiểu String  @param pattern truyền vào kiểu  return trả về date kiểu Date  \*/  public static Date toDate(String date,String...pattern){  try {  if(pattern.length>0)DATE\_FORMATER.applyPattern(pattern[0]);  if(date==null)return dateHelper.now();  return DATE\_FORMATER.parse(date);  } catch (ParseException ex) {  throw new RuntimeException(ex);  }  }    //chuyển Date sang String  /\*  @param date chuyền vào date kiểu date  @param pattern định dạng date  return date kiểu String đã định theo dạng pattern  \*/  public static String toString(Date date, String...pattern){  if(pattern.length>0)DATE\_FORMATER.applyPattern(pattern[0]);  if(date==null)date=dateHelper.now();  return DATE\_FORMATER.format(date);  }  //lấy giờ hiện tại  public static Date now() {  return new Date(); //new Date lấy giờ hiện tại  }    //thêm 1 số ngày vào mốc thời gian  /\*  @param date kiểu Date  @param days số ngày thêm, kiểu int  return date kiểu Date đã thêm số ngày  \*/  public static Date addDays(Date date, int days){  date.setTime(date.getTime() + days\*24\*60\*60\*1000);  return date;  }    //thêm 1 số ngày vào mốc thời gian hiện tại  /\*  @param days số ngày thêm, kiểu int  return date kiểu Date đã thêm số ngày vào date hiện tại  \*/  public static Date add(int days){  Date now = now();  now.setTime(now.getTime() + days\*24\*60\*60\*1000);  return now;  }  } |

4.3.1.2 Lớp tiện ích JDBCHelper

Chứa các phương thức thực hiện các câu lệnh thao tác và truy vấn được gọn nhẹ hơn.

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Helper;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.logging.Level;  import java.util.logging.Logger;  /\*\*  \*  \* @author Admin  \*/  public class jdbcHelper {  public static String driver="com.microsoft.sqlserver.jdbc.SQLServerDriver";  public static String dburl="jdbc:sqlserver://localhost:1433;databaseName=QuanLyChiTieu";  public static String username="sa";  public static String password="1";    //nạp driver  static{  try {  Class.forName(driver);  } catch (ClassNotFoundException ex) {  Logger.getLogger(jdbcHelper.class.getName()).log(Level.SEVERE, null, ex);  }  }    //nạp truyền giá trị đối số vào prepareStatement  //có thể statment không có đối số  //prepareStatement có thể là prepareStatement hoặc CallableStatements  /\*  @param sql câu lệnh sql statement (có đối số hoặc không)  @param args mảng đối số của câu lệnh sql (có hoặc không)  return pstmt là PrepareStatement đã được truyền đối số  \*/  public static PreparedStatement preparedStatement(String sql,Object...args)throws SQLException{  Connection con=DriverManager.getConnection(dburl, username, password);  PreparedStatement pstmt=null;  if(sql.startsWith("{"))pstmt=con.prepareCall(sql); //có thể gán biến kiểu PreparedStatement là prepareCall (CallableStatement)  else pstmt=con.prepareStatement(sql);  for(int i=0; i<args.length;i++){  pstmt.setObject(i+1, args[i]);    }  return pstmt;  }    //thao tác (INSERT, UPDATE, DELETE)  //thực thi prepareStatement (đã được truyền đối số ở hàm trên)  //khi làm chỉ cần gọi hàm này, từ trong hàm này nó sẽ gọi ra hàm preparedStatement ở trên  /\*  @param sql (String) câu lệnh sql statement (có đối số hoặc không)  @param args mảng đối số của câu lệnh sql (có hoặc không)  \*/  public static void executeUpdate(String sql,Object...args){  try {  PreparedStatement pstmt= preparedStatement(sql, args);  try{  pstmt.executeUpdate();  }finally{  pstmt.getConnection().close(); //đóng Connection từ statement  }  } catch (SQLException ex) {  throw new RuntimeException(ex);  }  }    //thao tác truy vấn (SELECT)  //thực thi prepareStatement (đã được truyền đối số ở hàm trên)  //khi làm chỉ cần gọi hàm này, từ trong hàm này nó sẽ gọi ra hàm preparedStatement ở trên  /\*  @param sql (String) câu lệnh sql statement (có đối số hoặc không)  @param args mảng đối số của câu lệnh sql (có hoặc không)  \*/  public static ResultSet executeQuery(String sql, Object...args){  try {  PreparedStatement pstmt=preparedStatement(sql, args);  try{  return pstmt.executeQuery();  }finally{  //pstmt.getConnection().close();  }  } catch (SQLException ex) {  throw new RuntimeException(ex); //throw các lỗi khi chạy CT, VD không có return khi try bị lỗi  }  }  } |

4.3.1.3 Lớp tiện ích CheckHelper

Chứa các phương thức thực hiện các câu lệnh kiểm tra lỗi của phần mềm

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Helper;  import static java.awt.Color.red;  import static java.awt.Color.white;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  import javafx.scene.control.RadioButton;  import javax.swing.ButtonGroup;  import javax.swing.JPasswordField;  import javax.swing.JRadioButton;  import javax.swing.JTextArea;  import javax.swing.JTextField;  /\*\*  \*  \* @author Admin  \*/  public class checkHelper {    //pass từ 6-16 kí tự  public static boolean checkPass(JPasswordField txt) {  txt.setBackground(white);  if (txt.getPassword().length > 7 && txt.getPassword().length < 17) {  return true;  } else {  txt.setBackground(red);  txt.requestFocus();  DialogHelper.alert(txt.getRootPane(), txt.getText() + " phải có từ 8-16 kí tự.");  return false;  }  }    public static boolean checkPass2(JTextField txt) {  txt.setBackground(white);  if (txt.getText().length() > 7 && txt.getText().length() < 17) {  return true;  } else {  txt.setBackground(red);  txt.requestFocus();  DialogHelper.alert(txt.getRootPane(), txt.getText() + " phải có từ 8-16 kí tự.");  return false;  }  }      public static boolean checkXNPass(JPasswordField txtXNPass, JPasswordField txtPass) {  txtXNPass.setBackground(white);  txtPass.setBackground(white);  if (txtXNPass.getText().equals(txtPass.getText())) {  return true;  } else {  txtXNPass.setBackground(red);  txtPass.setBackground(red);    DialogHelper.alert(txtXNPass.getRootPane(), "Mật khẩu và xác nhận mật khẩu không giống nhau!");  return false;  }  }    public static boolean checkXNPass2(JTextField txtXNPass, JTextField txtPass) {  txtXNPass.setBackground(white);  txtPass.setBackground(white);  if (txtXNPass.getText().equals(txtPass.getText())) {  return true;  } else {  txtXNPass.setBackground(red);  txtPass.setBackground(red);    DialogHelper.alert(txtXNPass.getRootPane(), "Mật khẩu và xác nhận mật khẩu không giống nhau!");  return false;  }  }    public static boolean isValidDate(String inDate) {  if (inDate == null) {  return false;  }    SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy");  if (inDate.trim().length() != dateFormat.toPattern().length()) {  return false;  }  dateFormat.setLenient(false);  try {    dateFormat.parse(inDate.trim());  } catch (ParseException pe) {  return false;  }  return true;  }  //định dạng dd/MM/yyyy (hoặc d/M/yyyy nếu là số 0 đứng trước)  public static boolean checkDate(JTextField txt) {  txt.setBackground(white);  String id = txt.getText();  if (isValidDate(id)) {  return true;  } else {  txt.setBackground(red);  txt.requestFocus();  DialogHelper.alert(txt.getRootPane(), txt.getText() + " không đúng định dạng dd/MM/yyyy");  return false;  }  }  //gồm các ký tự chữ đấu cách  //từ 3-25 kí tự  public static boolean checkName(JTextField txt) {  txt.setBackground(white);  String id = txt.getText();  Pattern p = Pattern.compile("\\p{L}{3,25}");  Matcher matcher = p.matcher(id);      if (matcher.find()) {  return true;  } else {  txt.setBackground(red);  txt.requestFocus();  DialogHelper.alert(txt.getRootPane(), txt.getText() + " phải là tên tiếng việt hoặc không đấu\ntừ 3-25 kí tự");  return false;  }  }  public static boolean checkSoDu(JTextField txt) {  txt.setBackground(white);  String id = txt.getText();  Pattern p = Pattern.compile("[a-zA-Z]");  Matcher matcher = p.matcher(id);      if (matcher.find()) {  txt.setBackground(red);  txt.requestFocus();  DialogHelper.alert(txt.getRootPane(), txt.getText() + " không có dấu và từ 5-25 kí tự");  return false;    } else {  return true;  }  }    public static boolean checkSoAm(JTextField txt) {  txt.setBackground(white);  int id = Integer.parseInt(txt.getText());      if (id<=0) {  txt.setBackground(red);  txt.requestFocus();  DialogHelper.alert(txt.getRootPane(), "Không được nhập số âm");  return false;    } else {  return true;  }  }    //từ 3-25 kí tự  public static boolean checkTaiKhoan(JTextField txt) {  txt.setBackground(white);  String id = txt.getText();  Pattern p = Pattern.compile("[a-zA-Z0-9]{5,25}");  Matcher matcher = p.matcher(id);      if (matcher.find()) {  return true;  } else {  txt.setBackground(red);  txt.requestFocus();  DialogHelper.alert(txt.getRootPane(), txt.getText() + " không có dấu và từ 5-25 kí tự");  return false;  }  }  public static boolean checkKituMK(JTextField txt) {  txt.setBackground(white);  String id = txt.getText();  Pattern p = Pattern.compile("[a-zA-Z0-9]{8,25}");  Matcher matcher = p.matcher(id);      if (matcher.find()) {  return true;  } else {  txt.setBackground(red);  txt.requestFocus();  DialogHelper.alert(txt.getRootPane(), txt.getText() + " không có dấu và từ 8-16 kí tự");  return false;  }  }  //gồm 10 số  //các đầu 3 số của nhà mạng  public static boolean checkSDT(JTextField txt) {  txt.setBackground(white);  String id = txt.getText();  String rgx = "(0)[0-9]{9}";  if (id.matches(rgx)) {  return true;  } else {  txt.setBackground(red);  txt.requestFocus();  DialogHelper.alert(txt.getRootPane(), txt.getText() + " phải gồm 10 số và bắt đầu bằng số 0\n.");  return false;  }  }    public static boolean checkThang(JTextField txt) {  txt.setBackground(white);  String id = txt.getText();  int id2 = Integer.parseInt(id);  if (id2>=1 && id2<=31) {  return true;  } else {  txt.setBackground(red);  txt.requestFocus();  DialogHelper.alert(txt.getRootPane(), txt.getText() + " phải gồm 1 số và nhỏ hơn hoặc bằng 31 \n.");  return false;  }  }    public static boolean checkNam(JTextField txt) {  txt.setBackground(white);  String id = txt.getText();  int id2 = Integer.parseInt(id);  if (id2>2018) {  return true;  } else {  txt.setBackground(red);  txt.requestFocus();  DialogHelper.alert(txt.getRootPane(), txt.getText() + " phải gồm 4 số và bắt đầu là từ năm 2019 \n.");  return false;  }  }    public static boolean checkEmail(JTextField txt) {  txt.setBackground(white);  String id = txt.getText();  String rgx = "^[a-zA-Z][a-zA-Z0-9\_\\.]{2,32}@[a-zA-Z0-9]{2,10}(\\.[a-zA-Z0-9]{2,4}){1,2}$";  if (id.matches(rgx)) {  return true;  } else {  txt.setBackground(red);  txt.requestFocus();  DialogHelper.alert(txt.getRootPane(), txt.getText() + " không đúng định dạng");  return false;  }  }      //bất kì kí tự nào  //từ 3-255 kí tự  public static boolean checkDiaChi(JTextField txt) {  txt.setBackground(white);  String id = txt.getText();  String rgx = ".{3,255}";  if (id.matches(rgx)) {  return true;  } else {  txt.setBackground(red);  txt.requestFocus();  DialogHelper.alert(txt.getRootPane(), txt.getText() + " phải từ 3-255 kí tự.");  return false;  }  }  public static boolean checkNullText(JTextField txt) {  txt.setBackground(white);  if (txt.getText().trim().length() > 0) {  return true;  } else {  txt.setBackground(red);  txt.requestFocus();  DialogHelper.alert(txt.getRootPane(), "Không được để trống " + txt.getName());  return false;  }  }  public static boolean checkNullPass(JPasswordField txt) {  txt.setBackground(white);  if (txt.getPassword().length > 0) {  return true;  } else {  txt.setBackground(red);  txt.requestFocus();  DialogHelper.alert(txt.getRootPane(), "Không được để trống " + txt.getName());  return false;  }  }    public static boolean checkRdo(JRadioButton rdo1, JRadioButton rdo2, JRadioButton rdo3) {    if(rdo1.isSelected() || rdo2.isSelected() || rdo3.isSelected()){  return true;  }  else{    rdo1.requestFocus();  DialogHelper.alert(rdo1.getRootPane(),"Không được để trống giới tính");  return false;  }  }      public static boolean checkRdo2(JRadioButton rdo1, JRadioButton rdo2) {    if(rdo1.isSelected() || rdo2.isSelected() ){  return true;  }  else{    rdo1.requestFocus();  DialogHelper.alert(rdo1.getRootPane(),"Không được để trống vai trò");  return false;  }  }  } |

4.3.1.4 ShareHelper

Chứa các phương thức thực hiện các câu lệnh kiểm tra tài khoản và đăng nhập của phần mềm

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Helper;  import java.awt.Image;  import java.io.File;  import java.nio.file.Files;  import java.nio.file.Path;  import java.nio.file.Paths;  import java.nio.file.StandardCopyOption;  import javax.swing.ImageIcon;  import Model.TaiKhoan;  import javax.swing.JLabel;  /\*\*  \*  \* @author Admin  \*/  public class ShareHelper {  /\*\*  \* Ảnh biểu tượng của ứng dụng, xuất hiện trên mọi cửa sổ  \*/    public static final Image APP\_ICON;  static{  // Tải biểu tượng ứng dụng  //Images là thư mục con của src    String file = "/Images/logog.png";  APP\_ICON = new ImageIcon(ShareHelper.class.getResource(file)).getImage();  }    /\*\*  \* Sao chép file logo chuyên đề vào thư mục logo  \* @param file là đối tượng file ảnh  \* @return chép được hay không  \* \*/    public static boolean saveLog(File file){  File dir = new File("Images");  // Tạo thư mục nếu chưa tồn tại  if(!dir.exists()){  dir.mkdirs();  }  File newFile = new File(dir, file.getName());    try {  // Copy vào thư mục logos (đè nếu đã tồn tại)  Path source = Paths.get(file.getAbsolutePath());  Path destination = Paths.get(newFile.getAbsolutePath());  Files.copy(source, destination, StandardCopyOption.REPLACE\_EXISTING);  return true;  }  catch (Exception ex) {  return false;  }  }  /\*\*  \* Đọc hình ảnh logo chuyên đề  \* @param fileName là tên file logo  \* @return ảnh đọc được  \*/    public static ImageIcon readLogo(String fileName, JLabel lblHinh){  File path = new File("Images", fileName);    return new ImageIcon( new ImageIcon(path.getAbsolutePath()).getImage().getScaledInstance(lblHinh.getWidth(), lblHinh.getHeight(), Image.SCALE\_SMOOTH));    }  public static ImageIcon readLogo2(String fileName, int Width, int Height){  File path = new File("Images", fileName);    return new ImageIcon( new ImageIcon(path.getAbsolutePath()).getImage().getScaledInstance(Width, Height, Image.SCALE\_SMOOTH));    }  /\*\*  \* Đối tượng này chứa thông tin người sử dụng sau khi đăng nhập  \*/    public static TaiKhoan USER = null;  /\*\*  \* Xóa thông tin của người sử dụng khi có yêu cầu đăng xuất  \*/    public static void logoff(){  ShareHelper.USER =null;  }    /\*\*  \* Kiểm tra xem đăng nhập hay chưa  \* @return đăng nhập hay chưa  \*/    public static boolean authenticated(){  return ShareHelper.USER != null;  }  } |

4.3.1.5 SendMail\_LayMaHelper

Chứa các phương thức để gửi mail

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Helper;  import java.util.Properties;  import java.util.Random;  import javax.mail.Message;  import javax.mail.PasswordAuthentication;  import javax.mail.Session;  import javax.mail.Transport;  import javax.mail.internet.InternetAddress;  import javax.mail.internet.MimeMessage;  import javax.swing.JOptionPane;  /\*\*  \*  \* @author Admin  \*/  public class SendMail\_LayMaHelper {  public String layma(){  /\*  :: la double colon, dùng cho việc đi tắt và giúp cho biến kia gọi lên phương thức tham chiếu đó; như StringBuilder::new thì sẽ tạo một new StringBuilder()  tránh viết new StringBuilder()vì rằng một void không thể thành một biến ở trong một hàm được, nên sử dụng :: để tạo đường tắt để có thể sử dụng nhanh phương thức  tránh việc gọi lên rườm rà  \*/  int leftLimit = 48; // letter '0'  int rightLimit = 122; // letter 'z'  int len = 10;  Random random = new Random();  String generatedString = random.ints(leftLimit, rightLimit + 1)  .limit(len)  .collect(StringBuilder::new, StringBuilder::appendCodePoint, StringBuilder::append)  .toString();    return generatedString;  }    public void sendMail(String Email, String maXacNhan){  // TODO add your handling code here:  try{  String taiKhoan = "windnunbe@gmail.com";  String matKhau = "tumottoichin";  //Tạo đối tượng Properties và chỉ định thông tin host, port  Properties prop = new Properties();    prop.setProperty("mail.user", taiKhoan);  prop.setProperty("mail.password", matKhau);  prop.put("mail.smtp.host", "smtp.gmail.com");  prop.put("mail.smtp.port", "587");  prop.put("mail.smtp.auth", "true");  prop.put("mail.smtp.starttls.enable", "true");    //Tạo đối tượng session  Session session = Session.getDefaultInstance(prop, new javax.mail.Authenticator() {  @Override  protected PasswordAuthentication getPasswordAuthentication() {  return new PasswordAuthentication(taiKhoan, matKhau);  }  });      String subject = "Mã kích hoạt để tạo lại mật khẩu mới";  String body = "Mã kích hoạt: " + maXacNhan;    MimeMessage message = new MimeMessage(session);  message.setFrom(new InternetAddress(taiKhoan));  message.setRecipients  (  Message.RecipientType.TO,  InternetAddress.parse(Email)  );  message.setSubject(subject);  message.setText(body);  Transport.send(message);//gọi phương thức send để gửi mess di  JOptionPane.showMessageDialog(null,"Đã gởi mail thành công","Thông báo",JOptionPane.INFORMATION\_MESSAGE);  } catch (Exception e) {  JOptionPane.showMessageDialog(null, "Gởi mail thất bại","Thông báo",JOptionPane.ERROR\_MESSAGE);    }  }  } |

4.3.1.2 DialogHelper

Chứa các phương thức hiển thị hộp thoại.

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Helper;  import java.awt.Component;  import javax.swing.JOptionPane;  /\*\*  \*  \* @author Admin  \*/  public class DialogHelper {  /\*\*  \* Hiển thị thông báo cho người dùng  \* @param parent là cửa sổ chứa thông báo  \* @param message là thông báo  \*/    public static void alert(Component parent, String message){  JOptionPane.showMessageDialog(parent, message, "Chi Tiêu hàng ngày", JOptionPane.INFORMATION\_MESSAGE);  }    /\*\*  \* Hiển thị thông báo và yêu cầu người dùng xác nhận  \* @param parent là cửa sổ chứa thông báo  \* @param message là câu hỏi yes/no  \* @return là kết quả nhận được true/false  \*/    public static boolean confirm(Component parent, String message){  int result = JOptionPane.showConfirmDialog(parent, message,  "Chi Tiêu hàng ngày",  JOptionPane.YES\_NO\_OPTION, JOptionPane.QUESTION\_MESSAGE);    return result ==JOptionPane.YES\_OPTION;  }    /\*\*  \* Hiển thị thông báo yêu cầu nhập dữ liệu  \* @param parent là cửa sổ chứa thông báo  \* @param message là thông báo nhắc nhở nhập  \* @return là kết quả nhận được từ người sử dụng nhập vào  \*/    public static String prompt(Component parent, String message) {  return JOptionPane.showInputDialog(parent, message,  "Chi Tiêu hàng ngày", JOptionPane.INFORMATION\_MESSAGE);  }  } |

### **4.3.2 Model Class - Các lớp mô tả dữ liệu**

Để tránh các sai sót về kiểu dữ liệu khi làm việc với CSDL, chúng ta cần phải xây dựng các lớp mô tả dữ liệu với kiểu phù hợp. Điều này giúp làm việc với CSDL chặt chẽ hơn, dễ hơn.

4.3.2.1 TaiKhoan

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Model;  /\*\*  \*  \* @author Admin  \*/  public class TaiKhoan {  private String tenTaiKhoan;  private String matKhau;  public String getTenTaiKhoan() {  return tenTaiKhoan;  }  public void setTenTaiKhoan(String tenTaiKhoan) {  this.tenTaiKhoan = tenTaiKhoan;  }  public String getMatKhau() {  return matKhau;  }  public void setMatKhau(String matKhau) {  this.matKhau = matKhau;  }      } |

4.3.2.2 NguoiDung

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Model;  import java.util.Date;  /\*\*  \*  \* @author Admin  \*/  public class NguoiDung {  private String MSND;  private String tenHienThi;  private String hoTen;  private String sdt;  private String email;  private String diaChi;  private int gioiTinh;  private Date namSinh;  private String hinh;  private String taiKhoan;  private String matKhau;  public String getMSND() {  return MSND;  }  public void setMSND(String MSND) {  this.MSND = MSND;  }  public String getTenHienThi() {  return tenHienThi;  }  public void setTenHienThi(String tenHienThi) {  this.tenHienThi = tenHienThi;  }  public String getHoTen() {  return hoTen;  }  public void setHoTen(String hoTen) {  this.hoTen = hoTen;  }  public String getSdt() {  return sdt;  }  public void setSdt(String sdt) {  this.sdt = sdt;  }  public String getEmail() {  return email;  }  public void setEmail(String email) {  this.email = email;  }  public String getDiaChi() {  return diaChi;  }  public void setDiaChi(String diaChi) {  this.diaChi = diaChi;  }  public int getGioiTinh() {  return gioiTinh;  }  public void setGioiTinh(int gioiTinh) {  this.gioiTinh = gioiTinh;  }  public Date getNamSinh() {  return namSinh;  }  public void setNamSinh(Date namSinh) {  this.namSinh = namSinh;  }  public String getHinh() {  return hinh;  }  public void setHinh(String hinh) {  this.hinh = hinh;  }  public String getTaiKhoan() {  return taiKhoan;  }  public void setTaiKhoan(String taiKhoan) {  this.taiKhoan = taiKhoan;  }  public String getMatKhau() {  return matKhau;  }  public void setMatKhau(String matKhau) {  this.matKhau = matKhau;  }      } |

4.3.2.3 ViTien

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Model;  /\*\*  \*  \* @author Admin  \*/  public class ViTien {  private String MSVi;  private String tenVi;  private String msLoaiVi;  private String tenLoai;  private int soDu;  private String hinh;  private String taiKhoan;  private boolean tinhTrang;  @Override  public String toString() {  return tenVi;  }        public String getMSVi() {  return MSVi;  }  public void setMSVi(String MSVi) {  this.MSVi = MSVi;  }  public String getTenVi() {  return tenVi;  }  public void setTenVi(String tenVi) {  this.tenVi = tenVi;  }  public String getMsLoaiVi() {  return msLoaiVi;  }  public void setMsLoaiVi(String msLoaiVi) {  this.msLoaiVi = msLoaiVi;  }  public String getTenLoai() {  return tenLoai;  }  public void setTenLoai(String tenLoai) {  this.tenLoai = tenLoai;  }  public int getSoDu() {  return soDu;  }  public void setSoDu(int soDu) {  this.soDu = soDu;  }  public String getHinh() {  return hinh;  }  public void setHinh(String hinh) {  this.hinh = hinh;  }  public String getTaiKhoan() {  return taiKhoan;  }  public void setTaiKhoan(String taiKhoan) {  this.taiKhoan = taiKhoan;  }    public boolean isTinhTrang() {  return tinhTrang;  }  public void setTinhTrang(boolean tinhTrang) {  this.tinhTrang = tinhTrang;  }    public String toStringTinhTrang(){  String tinhTrang ="Đóng";  if(isTinhTrang()==true){  tinhTrang = "Mở";  }  return tinhTrang;  }      } |

4.3.2.4 LoaiVi

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Model;  /\*\*  \*  \* @author Admin  \*/  public class ViTien {  private String MSVi;  private String tenVi;  private String msLoaiVi;  private String tenLoai;  private int soDu;  private String hinh;  private String taiKhoan;  private boolean tinhTrang;  @Override  public String toString() {  return tenVi;  }        public String getMSVi() {  return MSVi;  }  public void setMSVi(String MSVi) {  this.MSVi = MSVi;  }  public String getTenVi() {  return tenVi;  }  public void setTenVi(String tenVi) {  this.tenVi = tenVi;  }  public String getMsLoaiVi() {  return msLoaiVi;  }  public void setMsLoaiVi(String msLoaiVi) {  this.msLoaiVi = msLoaiVi;  }  public String getTenLoai() {  return tenLoai;  }  public void setTenLoai(String tenLoai) {  this.tenLoai = tenLoai;  }  public int getSoDu() {  return soDu;  }  public void setSoDu(int soDu) {  this.soDu = soDu;  }  public String getHinh() {  return hinh;  }  public void setHinh(String hinh) {  this.hinh = hinh;  }  public String getTaiKhoan() {  return taiKhoan;  }  public void setTaiKhoan(String taiKhoan) {  this.taiKhoan = taiKhoan;  }    public boolean isTinhTrang() {  return tinhTrang;  }  public void setTinhTrang(boolean tinhTrang) {  this.tinhTrang = tinhTrang;  }    public String toStringTinhTrang(){  String tinhTrang ="Đóng";  if(isTinhTrang()==true){  tinhTrang = "Mở";  }  return tinhTrang;  }      } |

4.3.2.5 Thu

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Model;  import java.util.Date;  /\*\*  \*  \* @author Admin  \*/  public class Thu {  private String maThu;  private String msHangMucThu;  private String tenHangMucThu;  private int soTien;  private Date ngayThu;  private String taiKhoan;  private String MSVI;  private String tenVi;    public String getMaThu() {  return maThu;  }  public void setMaThu(String maThu) {  this.maThu = maThu;  }  public String getMsHangMucThu() {  return msHangMucThu;  }  public void setMsHangMucThu(String msHangMucThu) {  this.msHangMucThu = msHangMucThu;  }  public int getSoTien() {  return soTien;  }  public void setSoTien(int soTien) {  this.soTien = soTien;  }  public Date getNgayThu() {  return ngayThu;  }  public void setNgayThu(Date ngayThu) {  this.ngayThu = ngayThu;  }  public String getTaiKhoan() {  return taiKhoan;  }  public void setTaiKhoan(String taiKhoan) {  this.taiKhoan = taiKhoan;  }    public String getMSVI() {  return MSVI;  }  public void setMSVI(String MSVI) {  this.MSVI = MSVI;  }  public String getTenHangMucThu() {  return tenHangMucThu;  }  public void setTenHangMucThu(String tenHangMucThu) {  this.tenHangMucThu = tenHangMucThu;  }  public String getTenVi() {  return tenVi;  }  public void setTenVi(String tenVi) {  this.tenVi = tenVi;  }      } |

4.3.2.6 Chi

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Model;  import java.util.Date;  /\*\*  \*  \* @author Admin  \*/  public class Chi {  private String maChi;  private String msHangMucChi;  private String tenHangMucChi;  private int soTien;  private Date ngayChi;  private String taiKhoan;  private String MSVI;  private String tenVi;  public String getMaChi() {  return maChi;  }  public void setMaChi(String maChi) {  this.maChi = maChi;  }  public String getMsHangMucChi() {  return msHangMucChi;  }  public void setMsHangMucChi(String msHangMucChi) {  this.msHangMucChi = msHangMucChi;  }  public int getSoTien() {  return soTien;  }  public void setSoTien(int soTien) {  this.soTien = soTien;  }  public Date getNgayChi() {  return ngayChi;  }  public void setNgayChi(Date ngayChi) {  this.ngayChi = ngayChi;  }  public String getTaiKhoan() {  return taiKhoan;  }  public void setTaiKhoan(String taiKhoan) {  this.taiKhoan = taiKhoan;  }    public String getMSVI() {  return MSVI;  }  public void setMSVI(String MSVI) {  this.MSVI = MSVI;  }  public String getTenHangMucChi() {  return tenHangMucChi;  }  public void setTenHangMucChi(String tenHangMucChi) {  this.tenHangMucChi = tenHangMucChi;  }  public String getTenVi() {  return tenVi;  }  public void setTenVi(String tenVi) {  this.tenVi = tenVi;  }      } |

4.3.2.7 DanhMucThu

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Model;  /\*\*  \*  \* @author Admin  \*/  public class DanhMucThu {  private String msHangMucThu;  private String tenHangMuc;  public String getMsHangMucThu() {  return msHangMucThu;  }  public void setMsHangMucThu(String msHangMucThu) {  this.msHangMucThu = msHangMucThu;  }  public String getTenHangMuc() {  return tenHangMuc;  }  public void setTenHangMuc(String tenHangMuc) {  this.tenHangMuc = tenHangMuc;  }    @Override  public String toString() {  return tenHangMuc;  }      } |

4.3.2.8 DanhMucChi

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Model;  /\*\*  \*  \* @author Admin  \*/  public class DanhMucChi {  private String msHangMucChi;  private String tenHangMuc;  @Override  public String toString() {  return tenHangMuc;  }    public String getMsHangMucChi() {  return msHangMucChi;  }  public void setMsHangMucChi(String msHangMucChi) {  this.msHangMucChi = msHangMucChi;  }  public String getTenHangMuc() {  return tenHangMuc;  }  public void setTenHangMuc(String tenHangMuc) {  this.tenHangMuc = tenHangMuc;  }      } |

4.3.2.9 ThongKe\_Nam

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Model;  /\*\*  \*  \* @author Admin  \*/  public class ThongKe\_Nam {  private int Nam;  private int TongTien;  public ThongKe\_Nam(int Nam, int TongTien) {  this.Nam = Nam;  this.TongTien = TongTien;  }  public ThongKe\_Nam() {    }  public int getNam() {  return Nam;  }  public void setNam(int Nam) {  this.Nam = Nam;  }  public int getTongTien() {  return TongTien;  }  public void setTongTien(int TongTien) {  this.TongTien = TongTien;  }  } |

4.3.2.10 ThongKeNgay

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Model;  import java.util.Date;  /\*\*  \*  \* @author Admin  \*/  public class ThongKe\_Ngay {  private Date Ngay;  private int TongTien;  public ThongKe\_Ngay(Date Ngay, int TongTien) {  this.Ngay = Ngay;  this.TongTien = TongTien;  }  public ThongKe\_Ngay() {    }  public Date getNgay() {  return Ngay;  }  public int getTongTien() {  return TongTien;  }  public void setNgay(Date Ngay) {  this.Ngay = Ngay;  }  public void setTongTien(int TongTien) {  this.TongTien = TongTien;  }      } |

**4.3.2.11 ThongKeThang**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Model;  /\*\*  \*  \* @author Admin  \*/  public class ThongKe\_Thang {  private int Thang;  private int TongTien;  public ThongKe\_Thang(int Ngay, int TongTien) {  this.Thang = Ngay;  this.TongTien = TongTien;  }  public ThongKe\_Thang() {    }  public int getNgay() {  return Thang;  }  public int getTongTien() {  return TongTien;  }  public void setNgay(int Ngay) {  this.Thang = Ngay;  }  public void setTongTien(int TongTien) {  this.TongTien = TongTien;  }  } |

### **4.3.3 DAO Class – Các lớp truy xuất dữ liệu**

DAO (Data Access Object) là các lớp giúp giúp làm việc với các bảng trong CSDL. Mỗi lớp được định nghĩa một cách đầy đủ nhất về các thao tác và truy vấn dữ liệu sẽ được sử dụng trong ứng dụng. DAO sau đó sẽ được sử dụng trong các điều khiển sự kiện của giao diện để làm việc với CSDL một cách thuận tiện.

Trong phạm vi ứng dụng này chúng ta cần xây dựng mỗi DAO chứa tối thiểu các hành động sau

4.3.3.1 TaiKhoanDAO

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package DAO;  import Helper.jdbcHelper;  import Model.TaiKhoan;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  /\*\*  \*  \* @author Admin  \*/  public class TaiKhoanDAO {  public TaiKhoan readFromResultSet(ResultSet rs) throws SQLException{  TaiKhoan model=new TaiKhoan();  model.setTenTaiKhoan(rs.getString("TenTaiKhoan"));  model.setMatKhau(rs.getString("MatKhau"));    return model;  }    //thực hiện truy vấn lấy về 1 tập ResultSet rồi điền tập ResultSet đó vào 1 List  public List<TaiKhoan> select(String sql,Object...args){  List<TaiKhoan> list=new ArrayList<>();  try {  ResultSet rs=null;    rs=jdbcHelper.executeQuery(sql, args);  while(rs.next()){  TaiKhoan model = readFromResultSet(rs);  list.add(model);  }  rs.close();    } catch (SQLException ex) {    throw new RuntimeException();  }  return list;  }    /\*\*  \* Thêm mới thực thể vào CSDL  \* @param entity là thực thể chứa thông tin bản ghi mới  \*/  public void insert(TaiKhoan entity) {      String sql="INSERT INTO TAIKHOAN (TenTaiKhoan, MatKhau) VALUES (?, ?)";  jdbcHelper.executeUpdate(sql,    entity.getTenTaiKhoan(),  entity.getMatKhau());    }  /\*\*  \* Cập nhật thực thể vào CSDL  \* @param entity là thực thể chứa thông tin bản ghi cần cập nhật  \*/  public void update(TaiKhoan entity) {    String sql="UPDATE TAIKHOAN Set MatKhau = ? where TenTaiKhoan = ? ";  jdbcHelper.executeUpdate(sql,    entity.getMatKhau(),  entity.getTenTaiKhoan());  }  /\*\*  \* Xóa bản ghi khỏi CSDL  \* @param TenTaiKhoan là mã của bản ghi cần xóa  \*/  public void delete(String TenTaiKhoan) {    String sql="DELETE FROM TAIKHOAN WHERE TenTaiKhoan = ?";  jdbcHelper.executeUpdate(sql, TenTaiKhoan);  }  /\*\*  \* Truy vấn tất cả các các thực thể  \* @return danh sách các thực thể  \*/  public List<TaiKhoan> select() {  String sql="SELECT \* FROM TAIKHOAN" ;  return select(sql); //trong 1 class có thể có 2 method trùng tên (nhưng param khác nhau)  }  /\*\*  \* Truy vấn thực thể theo mã id  \* @param id là mã của bản ghi được truy vấn  \* @return thực thể chứa thông tin của bản ghi  \*/  public TaiKhoan findById(String id) {  String sql="SELECT \* FROM TAIKHOAN "  + "WHERE TenTaiKhoan like ? ";  List<TaiKhoan> list=select(sql, id);  return list.size()>0?list.get(0):null; //có thể trả về là null  }  } |

4.3.3.2 NguoiDungDAO

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package DAO;  import Helper.jdbcHelper;  import Model.NguoiDung;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  /\*\*  \*  \* @author Admin  \*/  public class NguoiDungDAO {    public NguoiDung readFromResultSet(ResultSet rs) throws SQLException{  NguoiDung model=new NguoiDung();  model.setMSND(rs.getString("MSND"));  model.setTenHienThi(rs.getString("TenHienThi"));  model.setHoTen(rs.getString("HoTen"));  model.setSdt(rs.getString("SDT"));  model.setEmail(rs.getString("Email"));  model.setDiaChi(rs.getString("DiaChi"));  model.setGioiTinh(rs.getInt("GioiTinh"));  model.setNamSinh(rs.getDate("NamSinh"));  model.setHinh(rs.getString("Hinh"));  model.setTaiKhoan(rs.getString("TenTaiKhoan"));  model.setMatKhau(rs.getString("MatKhau"));  return model;  }      //thực hiện truy vấn lấy về 1 tập ResultSet rồi điền tập ResultSet đó vào 1 List  public List<NguoiDung> select(String sql,Object...args){  List<NguoiDung> list=new ArrayList<>();  try {  ResultSet rs=null;    rs=jdbcHelper.executeQuery(sql, args);  while(rs.next()){  NguoiDung model = readFromResultSet(rs);  list.add(model);  }  rs.close();    } catch (SQLException ex) {    throw new RuntimeException();  }  return list;  }    /\*\*  \* Thêm mới thực thể vào CSDL  \* @param entity là thực thể chứa thông tin bản ghi mới  \*/  public void insert(NguoiDung entity) {    String sql\_1 = "INSERT INTO TaiKhoan (TenTaiKhoan, MatKhau) VALUES (?, ?)";  jdbcHelper.executeUpdate(sql\_1,  entity.getTaiKhoan(),  entity.getMatKhau()  );    String sql\_2="INSERT INTO NguoiDung (MSND, TenHienThi, HoTen, SDT, Email, DiaChi, GioiTinh, NamSinh, Hinh, TenTaiKhoan) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";  jdbcHelper.executeUpdate(sql\_2,  entity.getMSND(),  entity.getTenHienThi(),  entity.getHoTen(),  entity.getSdt(),  entity.getEmail(),  entity.getDiaChi(),  entity.getGioiTinh(),  entity.getNamSinh(),  entity.getHinh(),  entity.getTaiKhoan());      }  /\*\*  \* Cập nhật thực thể vào CSDL  \* @param entity là thực thể chứa thông tin bản ghi cần cập nhật  \*/  public void update(NguoiDung entity) {      String sql\_1 = "UPDATE TaiKhoan SET MatKhau= ? WHERE TenTaiKhoan = ?";  jdbcHelper.executeUpdate(sql\_1,  entity.getMatKhau(),  entity.getTaiKhoan()  );      String sql\_2="UPDATE NguoiDung Set TenHienThi = ?, HoTen = ?, SDT = ?, Email = ?, DiaChi = ?, GioiTinh = ?, NamSinh = ?, Hinh = ?, TenTaiKhoan = ? where MSND = ? ";  jdbcHelper.executeUpdate(sql\_2,    entity.getTenHienThi(),  entity.getHoTen(),  entity.getSdt(),  entity.getEmail(),  entity.getDiaChi(),  entity.getGioiTinh(),  entity.getNamSinh(),  entity.getHinh(),  entity.getTaiKhoan(),  entity.getMSND()  );  }  /\*\*  \* Xóa bản ghi khỏi CSDL  \* @param MSND là mã của bản ghi cần xóa  \*/  public void delete(String MSND) {    String sql="DELETE FROM NGUOIDUNG WHERE MSND = ?";  jdbcHelper.executeUpdate(sql, MSND);  }  /\*\*  \* Truy vấn tất cả các các thực thể  \* @return danh sách các thực thể  \*/  public List<NguoiDung> select() {  String sql="SELECT NGUOIDUNG.MSND, NGUOIDUNG.TenHienThi, NGUOIDUNG.HoTen,NGUOIDUNG.SDT, NGUOIDUNG.Email, NGUOIDUNG.DiaChi, NGUOIDUNG.GioiTinh, NGUOIDUNG.NamSinh, NGUOIDUNG.Hinh, NGUOIDUNG.TenTaiKhoan, TAIKHOAN.MatKhau FROM NGUOIDUNG , TAIKHOAN WHERE NGUOIDUNG.TenTaiKhoan = TAIKHOAN.TenTaiKhoan";  return select(sql); //trong 1 class có thể có 2 method trùng tên (nhưng param khác nhau)  }  /\*\*  \* Truy vấn thực thể theo mã id  \* @param id là mã của bản ghi được truy vấn  \* @return thực thể chứa thông tin của bản ghi  \*/  public NguoiDung findById(String id) {  String sql="SELECT NGUOIDUNG.MSND, NGUOIDUNG.TenHienThi, NGUOIDUNG.HoTen, NGUOIDUNG.SDT, NGUOIDUNG.Email, NGUOIDUNG.DiaChi, NGUOIDUNG.GioiTinh, NGUOIDUNG.NamSinh, NGUOIDUNG.Hinh, NGUOIDUNG.TenTaiKhoan, TAIKHOAN.MatKhau FROM NGUOIDUNG, TAIKHOAN "  + "WHERE NGUOIDUNG.TenTaiKhoan = TAIKHOAN.TenTaiKhoan and NGUOIDUNG.MSND = ?";  List<NguoiDung> list=select(sql, id);  return list.size()>0?list.get(0):null; //có thể trả về là null  }    public NguoiDung findByIdTK(String id) {  String sql="SELECT NGUOIDUNG.MSND, NGUOIDUNG.TenHienThi, NGUOIDUNG.HoTen, NGUOIDUNG.SDT, NGUOIDUNG.Email, NGUOIDUNG.DiaChi, NGUOIDUNG.GioiTinh, NGUOIDUNG.NamSinh, NGUOIDUNG.Hinh, NGUOIDUNG.TenTaiKhoan, TAIKHOAN.MatKhau FROM NGUOIDUNG, TAIKHOAN "  + "WHERE NGUOIDUNG.TenTaiKhoan = TAIKHOAN.TenTaiKhoan and NGUOIDUNG.TenTaiKhoan like ?";  List<NguoiDung> list=select(sql, id);  return list.size()>0?list.get(0):null; //có thể trả về là null  }      public int countSize(){  return select().size();  }    public List<NguoiDung> selectByKeywordSDT(String keyword) {  String sql="SELECT NGUOIDUNG.MSND, NGUOIDUNG.TenHienThi, NGUOIDUNG.HoTen, NGUOIDUNG.SDT, NGUOIDUNG.Email, NGUOIDUNG.DiaChi, NGUOIDUNG.GioiTinh, NGUOIDUNG.NamSinh, NGUOIDUNG.Hinh, NGUOIDUNG.TenTaiKhoan, TAIKHOAN.MatKhau FROM NGUOIDUNG, TAIKHOAN "  + "WHERE NGUOIDUNG.TenTaiKhoan = TAIKHOAN.TenTaiKhoan and NGUOIDUNG.SDT like ?";  return select(sql, "%"+keyword+"%");  }    public List<NguoiDung> getTk\_Email(String email, String taiKhoan) {  List<NguoiDung> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_timTaiKhoan\_Email (?,?)}";  rs = jdbcHelper.executeQuery(sql, email, taiKhoan);  while (rs.next()) {  NguoiDung model = readFromResultSet(rs);  list.add(model);  }    rs.getStatement().getConnection().close();    } catch (SQLException e) {  throw new RuntimeException(e);  }  return list;  }    } |

4.3.3.3 ViTienDAO

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package DAO;  import Helper.jdbcHelper;  import Model.ViTien;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  /\*\*  \*  \* @author Admin  \*/  public class ViTienDAO {  public ViTien readFromResultSet(ResultSet rs) throws SQLException{  ViTien model=new ViTien();  model.setMSVi(rs.getString("MSVI"));  model.setTenVi(rs.getString("TenVi"));  model.setMsLoaiVi(rs.getString("MSLoaiVi"));  model.setTenLoai(rs.getString("TenLoai"));  model.setSoDu(rs.getInt("SoDu"));  model.setHinh(rs.getString("Hinh"));  model.setTaiKhoan(rs.getString("TenTaiKhoan"));  model.setTinhTrang(rs.getBoolean("TinhTrang"));  return model;  }    //thực hiện truy vấn lấy về 1 tập ResultSet rồi điền tập ResultSet đó vào 1 List  public List<ViTien> select(String sql,Object...args){  List<ViTien> list=new ArrayList<>();  try {  ResultSet rs=null;    rs=jdbcHelper.executeQuery(sql, args);  while(rs.next()){  ViTien model = readFromResultSet(rs);  list.add(model);  }  rs.close();    } catch (SQLException ex) {    throw new RuntimeException();  }  return list;  }    /\*\*  \* Thêm mới thực thể vào CSDL  \* @param entity là thực thể chứa thông tin bản ghi mới  \*/  public void insert(ViTien entity) {      String sql="INSERT INTO VITIEN (MSVI, TenVi, MSLoaiVi, SoDu, hinh, TenTaiKhoan, TinhTrang) VALUES (?, ?, ?, ?, ?, ?, ?)";  jdbcHelper.executeUpdate(sql,  entity.getMSVi(),  entity.getTenVi(),  entity.getMsLoaiVi(),  entity.getSoDu(),  entity.getHinh(),  entity.getTaiKhoan(),  entity.isTinhTrang());      }  /\*\*  \* Cập nhật thực thể vào CSDL  \* @param entity là thực thể chứa thông tin bản ghi cần cập nhật  \*/  public void update(ViTien entity) {    String sql="UPDATE VITIEN Set TenVi = ?, MSLoaiVi = ?, SoDu = ?, hinh = ?, TenTaiKhoan = ?, TinhTrang = ? where MSVI = ? ";  jdbcHelper.executeUpdate(sql,    entity.getTenVi(),  entity.getMsLoaiVi(),  entity.getSoDu(),  entity.getHinh(),  entity.getTaiKhoan(),  entity.isTinhTrang(),  entity.getMSVi()  );  }    public void updateTien(int tongTien, String msVi) {    String sql="UPDATE VITIEN Set SoDu = ? where MSVI = ? ";  jdbcHelper.executeUpdate(sql,    tongTien,  msVi    );  }        /\*\*  \* Xóa bản ghi khỏi CSDL  \* @param MSVI là mã của bản ghi cần xóa  \*/  public void delete(String MSVI) {    String sql="DELETE FROM VITIEN WHERE MSVI = ?";  jdbcHelper.executeUpdate(sql, MSVI);  }  /\*\*  \* Truy vấn tất cả các các thực thể  \* @return danh sách các thực thể  \*/  public List<ViTien> select() {  String sql="select VITIEN.MSVI, VITIEN.TenVi, VITIEN.MSLoaiVi, LOAIVI.TenLoai, VITIEN.SoDu, VITIEN.Hinh,VITIEN.TenTaiKhoan, VITIEN.TinhTrang from VITIEN , LOAIVI where VITIEN.MSLoaiVi = LOAIVI.MSLoaiVi" ;  return select(sql); //trong 1 class có thể có 2 method trùng tên (nhưng param khác nhau)  }  /\*\*  \* Truy vấn thực thể theo mã id  \* @param id là mã của bản ghi được truy vấn  \* @return thực thể chứa thông tin của bản ghi  \*/  public ViTien findById(String id) {  String sql="select VITIEN.MSVI, VITIEN.TenVi, VITIEN.MSLoaiVi, LOAIVI.TenLoai, VITIEN.SoDu, VITIEN.Hinh,VITIEN.TenTaiKhoan, VITIEN.TinhTrang from VITIEN , LOAIVI "  + "WHERE VITIEN.MSLoaiVi = LOAIVI.MSLoaiVi and VITIEN.MSVI = ? ";  List<ViTien> list=select(sql, id);  return list.size()>0?list.get(0):null; //có thể trả về là null  }    public List<ViTien> selectByKeyword(String keyword) {  String sql="select VITIEN.MSVI, VITIEN.TenVi, VITIEN.MSLoaiVi, LOAIVI.TenLoai, VITIEN.SoDu, VITIEN.Hinh,VITIEN.TenTaiKhoan, VITIEN.TinhTrang from VITIEN , LOAIVI "  + "WHERE VITIEN.MSLoaiVi = LOAIVI.MSLoaiVi and TenTaiKhoan like ? ";  return select(sql, "%"+keyword+"%");  }    public int countSize(String keyword){  return selectByKeyword(keyword).size();  }    public int countSize2(){  return select().size();  }  } |

4.3.3.4 LoaiViDAO

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package DAO;  import Helper.jdbcHelper;  import Model.LoaiVi;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  /\*\*  \*  \* @author Admin  \*/  public class LoaiViDAO {    public LoaiVi readFromResultSet(ResultSet rs) throws SQLException{  LoaiVi model=new LoaiVi();  model.setMsLoaiVi(rs.getString("MSLoaiVi"));  model.setTenLoaiVi(rs.getString("TenLoai"));    return model;  }    //thực hiện truy vấn lấy về 1 tập ResultSet rồi điền tập ResultSet đó vào 1 List  public List<LoaiVi> select(String sql,Object...args){  List<LoaiVi> list=new ArrayList<>();    try {  ResultSet rs=null;    rs=jdbcHelper.executeQuery(sql, args);  while(rs.next()){  LoaiVi model = readFromResultSet(rs);  list.add(model);    }  rs.close();    } catch (SQLException ex) {    throw new RuntimeException();  }  return list;  }    /\*\*  \* Thêm mới thực thể vào CSDL  \* @param entity là thực thể chứa thông tin bản ghi mới  \*/  public void insert(LoaiVi entity) {      String sql="INSERT INTO LOAIVI (MSLoaiVi, TenLoai) VALUES (?, ?)";  jdbcHelper.executeUpdate(sql,    entity.getMsLoaiVi(),  entity.getTenLoaiVi());    }  /\*\*  \* Cập nhật thực thể vào CSDL  \* @param entity là thực thể chứa thông tin bản ghi cần cập nhật  \*/  public void update(LoaiVi entity) {    String sql="UPDATE LOAIVI Set TenLoai = ? where MSLoaiVi = ? ";  jdbcHelper.executeUpdate(sql,    entity.getTenLoaiVi(),  entity.getMsLoaiVi());  }  /\*\*  \* Xóa bản ghi khỏi CSDL  \* @param MSVI là mã của bản ghi cần xóa  \*/  public void delete(String MSLoaiVi) {    String sql="DELETE FROM LOAIVI WHERE MSLoaiVi = ?";  jdbcHelper.executeUpdate(sql, MSLoaiVi);  }  /\*\*  \* Truy vấn tất cả các các thực thể  \* @return danh sách các thực thể  \*/  public List<LoaiVi> select() {  String sql="SELECT \* FROM LOAIVI" ;  return select(sql); //trong 1 class có thể có 2 method trùng tên (nhưng param khác nhau)  }  /\*\*  \* Truy vấn thực thể theo mã id  \* @param id là mã của bản ghi được truy vấn  \* @return thực thể chứa thông tin của bản ghi  \*/  public LoaiVi findById(String id) {  String sql="SELECT \* FROM LOAIVI "  + "WHERE MSLoaiVi = ? ";  List<LoaiVi> list=select(sql, id);  return list.size()>0?list.get(0):null; //có thể trả về là null  }    public LoaiVi findByIdTVi(String id) {  String sql="SELECT \* FROM LOAIVI "  + "WHERE TenVi = ? ";  List<LoaiVi> list=select(sql, id);  return list.size()>0?list.get(0):null; //có thể trả về là null  }    public int countSize(){  return select().size();  }    } |

4.3.3.5 ThuDAO

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Model;  import java.util.Date;  /\*\*  \*  \* @author Admin  \*/  public class Thu {  private String maThu;  private String msHangMucThu;  private String tenHangMucThu;  private int soTien;  private Date ngayThu;  private String taiKhoan;  private String MSVI;  private String tenVi;            public String getMaThu() {  return maThu;  }  public void setMaThu(String maThu) {  this.maThu = maThu;  }  public String getMsHangMucThu() {  return msHangMucThu;  }  public void setMsHangMucThu(String msHangMucThu) {  this.msHangMucThu = msHangMucThu;  }  public int getSoTien() {  return soTien;  }  public void setSoTien(int soTien) {  this.soTien = soTien;  }  public Date getNgayThu() {  return ngayThu;  }  public void setNgayThu(Date ngayThu) {  this.ngayThu = ngayThu;  }  public String getTaiKhoan() {  return taiKhoan;  }  public void setTaiKhoan(String taiKhoan) {  this.taiKhoan = taiKhoan;  }    public String getMSVI() {  return MSVI;  }  public void setMSVI(String MSVI) {  this.MSVI = MSVI;  }  public String getTenHangMucThu() {  return tenHangMucThu;  }  public void setTenHangMucThu(String tenHangMucThu) {  this.tenHangMucThu = tenHangMucThu;  }  public String getTenVi() {  return tenVi;  }  public void setTenVi(String tenVi) {  this.tenVi = tenVi;  }      } |

4.3.3.6 ChiDAO

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package DAO;  import Helper.jdbcHelper;  import Model.Chi;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  /\*\*  \*  \* @author Admin  \*/  public class ChiDAO {  public Chi readFromResultSet(ResultSet rs) throws SQLException{  Chi model=new Chi();  model.setMaChi(rs.getString("MaChi"));  model.setMsHangMucChi(rs.getString("MsHangMucChi"));  model.setTenHangMucChi(rs.getString("TenHangMuc"));  model.setSoTien(rs.getInt("SoTien"));  model.setNgayChi(rs.getDate("Ngay"));  model.setTaiKhoan(rs.getString("TenTaiKhoan"));  model.setMSVI(rs.getString("MSVI"));  model.setTenVi(rs.getString("TenVi"));  return model;  }    public Chi readFromResultSet2(ResultSet rs) throws SQLException{  Chi model=new Chi();  model.setMaChi(rs.getString("MaChi"));  model.setMsHangMucChi(rs.getString("MsHangMucChi"));  model.setSoTien(rs.getInt("SoTien"));  model.setNgayChi(rs.getDate("Ngay"));  model.setTaiKhoan(rs.getString("TenTaiKhoan"));  model.setMSVI(rs.getString("MSVI"));  return model;  }  //thực hiện truy vấn lấy về 1 tập ResultSet rồi điền tập ResultSet đó vào 1 List  public List<Chi> select(String sql,Object...args){  List<Chi> list=new ArrayList<>();  try {  ResultSet rs=null;    rs=jdbcHelper.executeQuery(sql, args);  while(rs.next()){  Chi model = readFromResultSet(rs);  list.add(model);  }  rs.close();    } catch (SQLException ex) {    throw new RuntimeException();  }  return list;  }    public List<Chi> select2(String sql,Object...args){  List<Chi> list=new ArrayList<>();  try {  ResultSet rs=null;    rs=jdbcHelper.executeQuery(sql, args);  while(rs.next()){  Chi model = readFromResultSet2(rs);  list.add(model);  }  rs.close();    } catch (SQLException ex) {    throw new RuntimeException();  }  return list;  }  /\*\*  \* Thêm mới thực thể vào CSDL  \* @param entity là thực thể chứa thông tin bản ghi mới  \*/  public void insert(Chi entity) {      String sql="INSERT INTO CHI (MaChi, MsHangMucChi, SoTien, Ngay, TenTaiKhoan, MSVI) VALUES (?, ?, ?, ?, ?, ?)";  jdbcHelper.executeUpdate(sql,    entity.getMaChi(),  entity.getMsHangMucChi(),  entity.getSoTien(),  entity.getNgayChi(),  entity.getTaiKhoan(),  entity.getMSVI());    }    /\*\*  \* Cập nhật thực thể vào CSDL  \* @param entity là thực thể chứa thông tin bản ghi cần cập nhật  \*/  public void update(Chi entity) {    String sql="UPDATE CHI Set MsHangMucChi = ?, SoTien= ?, Ngay= ?, TenTaiKhoan = ?, MSVI=? WHERE MaChi = ? ";  jdbcHelper.executeUpdate(sql,    entity.getMsHangMucChi(),  entity.getSoTien(),  entity.getNgayChi(),  entity.getTaiKhoan(),  entity.getMSVI(),  entity.getMaChi());  }  /\*\*  \* Xóa bản ghi khỏi CSDL  \* @param MsHanMuc là mã của bản ghi cần xóa  \*/  public void delete(String MaChi) {    String sql="DELETE FROM CHI WHERE MaChi = ?";  jdbcHelper.executeUpdate(sql, MaChi);  }  /\*\*  \* Truy vấn tất cả các các thực thể  \* @return danh sách các thực thể  \*/  public List<Chi> select() {  String sql="SELECT \* FROM CHI";    return select2(sql); //trong 1 class có thể có 2 method trùng tên (nhưng param khác nhau)  }  /\*\*  \* Truy vấn thực thể theo mã id  \* @param id là mã của bản ghi được truy vấn  \* @return thực thể chứa thông tin của bản ghi  \*/  public Chi findById(String id) {  String sql="SELECT CHI.MaChi, CHI.MSHangMucChi, DANHMUCCHI.TenHangMuc, CHI.SoTien, CHI.Ngay, CHI.TenTaiKhoan, CHI.MSVI, VITIEN.TenVi from (CHI INNER JOIN DANHMUCCHI ON CHI.MSHangMucChi = DANHMUCCHI.MSHangMucChi INNER JOIN VITIEN ON CHI.MSVI = VITIEN.MSVI) "  + "WHERE CHI.MaChi like ?";    List<Chi> list=select(sql, id);  return list.size()>0?list.get(0):null; //có thể trả về là null  }      public List<Chi> selectByKeyword(String keyword) {  String sql="SELECT CHI.MaChi, CHI.MSHangMucChi, DANHMUCCHI.TenHangMuc, CHI.SoTien, CHI.Ngay, CHI.TenTaiKhoan, CHI.MSVI, VITIEN.TenVi from (CHI INNER JOIN DANHMUCCHI ON CHI.MSHangMucChi = DANHMUCCHI.MSHangMucChi INNER JOIN VITIEN ON CHI.MSVI = VITIEN.MSVI) "  + "WHERE CHI.TenTaiKhoan like ?";  return select(sql, "%"+keyword+"%");  }    public int countSizeChi(String keyword){  return selectByKeyword(keyword).size();  }    public int countSize(){  return select().size();  }  } |

4.3.3.7 DanhMucThuDAO

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package DAO;  import Helper.jdbcHelper;  import Model.DanhMucThu;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  /\*\*  \*  \* @author Admin  \*/  public class DanhMucThuDAO {  public DanhMucThu readFromResultSet(ResultSet rs) throws SQLException{  DanhMucThu model=new DanhMucThu();  model.setMsHangMucThu(rs.getString("MSHangMucThu"));  model.setTenHangMuc(rs.getString("TenHangMuc"));    return model;  }    //thực hiện truy vấn lấy về 1 tập ResultSet rồi điền tập ResultSet đó vào 1 List  public List<DanhMucThu> select(String sql,Object...args){  List<DanhMucThu> list=new ArrayList<>();  try {  ResultSet rs=null;    rs=jdbcHelper.executeQuery(sql, args);  while(rs.next()){  DanhMucThu model = readFromResultSet(rs);  list.add(model);  }  rs.close();    } catch (SQLException ex) {    throw new RuntimeException();  }  return list;  }    /\*\*  \* Thêm mới thực thể vào CSDL  \* @param entity là thực thể chứa thông tin bản ghi mới  \*/  public void insert(DanhMucThu entity) {      String sql="INSERT INTO DANHMUCTHU (MSHangMucThu, TenHangMuc) VALUES (?, ?)";  jdbcHelper.executeUpdate(sql,    entity.getMsHangMucThu(),  entity.getTenHangMuc());    }  /\*\*  \* Cập nhật thực thể vào CSDL  \* @param entity là thực thể chứa thông tin bản ghi cần cập nhật  \*/  public void update(DanhMucThu entity) {    String sql="UPDATE DANHMUCTHU Set TenHangMuc = ? where MSHangMucThu = ? ";  jdbcHelper.executeUpdate(sql,    entity.getTenHangMuc(),  entity.getMsHangMucThu());  }  /\*\*  \* Xóa bản ghi khỏi CSDL  \* @param MSVI là mã của bản ghi cần xóa  \*/  public void delete(String MsHangMucThu) {    String sql="DELETE FROM DANHMUCTHU WHERE MSHangMucThu like ?";  jdbcHelper.executeUpdate(sql, MsHangMucThu);  }  /\*\*  \* Truy vấn tất cả các các thực thể  \* @return danh sách các thực thể  \*/  public List<DanhMucThu> select() {  String sql="SELECT \* FROM DANHMUCTHU" ;  return select(sql); //trong 1 class có thể có 2 method trùng tên (nhưng param khác nhau)  }  /\*\*  \* Truy vấn thực thể theo mã id  \* @param id là mã của bản ghi được truy vấn  \* @return thực thể chứa thông tin của bản ghi  \*/  public DanhMucThu findById(String id) {  String sql="SELECT \* FROM DANHMUCTHU "  + "WHERE MSHangMucThu = ? ";  List<DanhMucThu> list=select(sql, id);  return list.size()>0?list.get(0):null; //có thể trả về là null  }    public int countSize(){  return select().size();  }  } |

4.3.3.8 DanhMucChiDAO

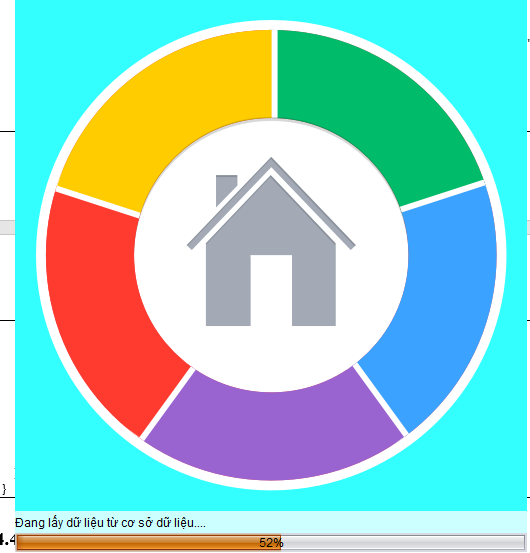
|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package DAO;  import Helper.jdbcHelper;  import Model.DanhMucChi;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  /\*\*  \*  \* @author Admin  \*/  public class DanhMucChiDAO {  public DanhMucChi readFromResultSet(ResultSet rs) throws SQLException{  DanhMucChi model=new DanhMucChi();  model.setMsHangMucChi(rs.getString("MSHangMucChi"));  model.setTenHangMuc(rs.getString("TenHangMuc"));    return model;  }    //thực hiện truy vấn lấy về 1 tập ResultSet rồi điền tập ResultSet đó vào 1 List  public List<DanhMucChi> select(String sql,Object...args){  List<DanhMucChi> list=new ArrayList<>();  try {  ResultSet rs=null;    rs=jdbcHelper.executeQuery(sql, args);  while(rs.next()){  DanhMucChi model = readFromResultSet(rs);  list.add(model);  }  rs.close();    } catch (SQLException ex) {    throw new RuntimeException();  }  return list;  }    /\*\*  \* Thêm mới thực thể vào CSDL  \* @param entity là thực thể chứa thông tin bản ghi mới  \*/  public void insert(DanhMucChi entity) {      String sql="INSERT INTO DANHMUCCHI (MSHangMucChi, TenHangMuc) VALUES (?, ?)";  jdbcHelper.executeUpdate(sql,    entity.getMsHangMucChi(),  entity.getTenHangMuc());    }  /\*\*  \* Cập nhật thực thể vào CSDL  \* @param entity là thực thể chứa thông tin bản ghi cần cập nhật  \*/  public void update(DanhMucChi entity) {    String sql="UPDATE DANHMUCCHI Set TenHangMuc = ? where MSHangMucChi = ? ";  jdbcHelper.executeUpdate(sql,    entity.getTenHangMuc(),  entity.getMsHangMucChi());  }  /\*\*  \* Xóa bản ghi khỏi CSDL  \* @param MSVI là mã của bản ghi cần xóa  \*/  public void delete(String MsHangMucChi) {    String sql="DELETE FROM DANHMUCCHI WHERE MSHangMucChi like ?";  jdbcHelper.executeUpdate(sql, MsHangMucChi);  }  /\*\*  \* Truy vấn tất cả các các thực thể  \* @return danh sách các thực thể  \*/  public List<DanhMucChi> select() {  String sql="SELECT \* FROM DANHMUCCHI" ;  return select(sql); //trong 1 class có thể có 2 method trùng tên (nhưng param khác nhau)  }  /\*\*  \* Truy vấn thực thể theo mã id  \* @param id là mã của bản ghi được truy vấn  \* @return thực thể chứa thông tin của bản ghi  \*/  public DanhMucChi findById(String id) {  String sql="SELECT \* FROM DANHMUCCHI "  + "WHERE MSHangMucChi = ? ";  List<DanhMucChi> list=select(sql, id);  return list.size()>0?list.get(0):null; //có thể trả về là null  }    public int countSize(){  return select().size();  }  } |

4.3.3.9 ThongKeDAO

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package DAO;  import Helper.checkHelper;  import Helper.jdbcHelper;  import Model.ThongKe\_Nam;  import Model.ThongKe\_Ngay;  import Model.ThongKe\_Thang;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  /\*\*  \*  \* @author Admin  \*/  public class ThongKeDAO {      public ThongKe\_Ngay readFromResultSet(ResultSet rs) throws SQLException{  ThongKe\_Ngay model=new ThongKe\_Ngay();  model.setNgay(rs.getDate("NGAY"));  model.setTongTien(rs.getInt("TONGTIEN"));    return model;  }    public ThongKe\_Thang readFromResultSet2(ResultSet rs) throws SQLException{  ThongKe\_Thang model=new ThongKe\_Thang();  model.setNgay(rs.getInt("THANG"));  model.setTongTien(rs.getInt("TONGTIEN"));    return model;  }    public ThongKe\_Nam readFromResultSet3(ResultSet rs) throws SQLException{  ThongKe\_Nam model=new ThongKe\_Nam();  model.setNam(rs.getInt("NAM"));  model.setTongTien(rs.getInt("TONGTIEN"));    return model;  }    public List<String> getTaiChinh(String tenTaiKhoan) {  List<String> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_TaiChinhHT (?)}";  rs = jdbcHelper.executeQuery(sql, tenTaiKhoan);  while (rs.next()) {  String a= rs.getString("TongTien");      list.add(a);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  throw new RuntimeException(ex);  }  return list;  }    public List<ThongKe\_Thang> getTKThu\_Thang(String taiKhoan, int year) {  List<ThongKe\_Thang> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_ThongKeThu\_Thang (?,?)}";  rs = jdbcHelper.executeQuery(sql, taiKhoan,year);  while (rs.next()) {  ThongKe\_Thang model= readFromResultSet2(rs);    list.add(model);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  ex.printStackTrace();  throw new RuntimeException(ex);  }  return list;  }  public List<ThongKe\_Ngay> getTKThu\_Ngay(String taiKhoan, int Thang, int Nam) {  List<ThongKe\_Ngay> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_ThongKeThu\_Ngay (?,?,?)}";  rs = jdbcHelper.executeQuery(sql, taiKhoan,Thang, Nam);  while (rs.next()) {  ThongKe\_Ngay model= readFromResultSet(rs);    list.add(model);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  ex.printStackTrace();  throw new RuntimeException(ex);  }  return list;  }    public List<ThongKe\_Nam> getTKThu\_Nam(String taiKhoan) {  List<ThongKe\_Nam> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_ThongKeThu\_Nam (?)}";  rs = jdbcHelper.executeQuery(sql, taiKhoan);  while (rs.next()) {  ThongKe\_Nam model= readFromResultSet3(rs);    list.add(model);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  ex.printStackTrace();  throw new RuntimeException(ex);  }  return list;  }        public List<ThongKe\_Thang> getTKChi\_Thang(String taiKhoan, int year) {  List<ThongKe\_Thang> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_ThongKeChi\_Thang (?,?)}";  rs = jdbcHelper.executeQuery(sql, taiKhoan,year);  while (rs.next()) {  ThongKe\_Thang model= readFromResultSet2(rs);    list.add(model);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  ex.printStackTrace();  throw new RuntimeException(ex);  }  return list;  }  public List<ThongKe\_Ngay> getTKChi\_Ngay(String taiKhoan, int Thang, int Nam) {  List<ThongKe\_Ngay> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_ThongKeChi\_Ngay (?,?,?)}";  rs = jdbcHelper.executeQuery(sql, taiKhoan,Thang, Nam);  while (rs.next()) {  ThongKe\_Ngay model= readFromResultSet(rs);    list.add(model);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  ex.printStackTrace();  throw new RuntimeException(ex);  }  return list;  }    public List<ThongKe\_Nam> getTKChi\_Nam(String taiKhoan) {  List<ThongKe\_Nam> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_ThongKeChi\_Nam (?)}";  rs = jdbcHelper.executeQuery(sql, taiKhoan);  while (rs.next()) {  ThongKe\_Nam model= readFromResultSet3(rs);    list.add(model);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  ex.printStackTrace();  throw new RuntimeException(ex);  }  return list;  }      //Tong thu chi  public List<Integer> getTongThuChi\_Nam(String tenTaiKhoan) {  List<Integer> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_TongThuChi\_Nam (?)}";  rs = jdbcHelper.executeQuery(sql, tenTaiKhoan);  while (rs.next()) {  int a= rs.getInt("TongTien");      list.add(a);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  throw new RuntimeException(ex);  }  return list;  }        public List<Integer> getTongThuChi\_Thang(String tenTaiKhoan, int year) {  List<Integer> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_TongThuChi\_Thang (?,?)}";  rs = jdbcHelper.executeQuery(sql, tenTaiKhoan, year);  while (rs.next()) {  int a= rs.getInt("TongTien");      list.add(a);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  throw new RuntimeException(ex);  }  return list;  }    public List<Integer> getTongThuChi\_Ngay(String tenTaiKhoan, int month, int year) {  List<Integer> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_TongThuChi\_Ngay (?,?, ?)}";  rs = jdbcHelper.executeQuery(sql, tenTaiKhoan, month, year);  while (rs.next()) {  int a= rs.getInt("TongTien");      list.add(a);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  throw new RuntimeException(ex);  }  return list;  }  //  //Tong thu    public List<Integer> getTongThu\_Nam(String tenTaiKhoan) {  List<Integer> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_TongThu\_Nam (?)}";  rs = jdbcHelper.executeQuery(sql, tenTaiKhoan);  while (rs.next()) {  int a= rs.getInt("TongTien");      list.add(a);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  throw new RuntimeException(ex);  }  return list;  }    public List<Integer> getTongThu\_NamNay(String tenTaiKhoan) {  List<Integer> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_TongThu\_NamNay (?)}";  rs = jdbcHelper.executeQuery(sql, tenTaiKhoan);  while (rs.next()) {  int a= rs.getInt("TongTien");      list.add(a);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  throw new RuntimeException(ex);  }  return list;  }    public List<Integer> getTongThu\_Thang(String tenTaiKhoan, int year) {  List<Integer> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_TongThu\_Thang (?,?)}";  rs = jdbcHelper.executeQuery(sql, tenTaiKhoan, year);  while (rs.next()) {  int a= rs.getInt("TongTien");    list.add(a);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  throw new RuntimeException(ex);  }  return list;  }    public List<Integer> getTongThu\_Ngay(String tenTaiKhoan, int month, int year) {  List<Integer> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_TongThu\_Ngay (?,?, ?)}";  rs = jdbcHelper.executeQuery(sql, tenTaiKhoan, month, year);  while (rs.next()) {  int a= rs.getInt("TongTien");      list.add(a);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  throw new RuntimeException(ex);  }  return list;  }    public List<Integer> getTongThu\_ThangNay(String tenTaiKhoan) {  List<Integer> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_TongThu\_ThangNay (?)}";  rs = jdbcHelper.executeQuery(sql, tenTaiKhoan);  while (rs.next()) {  int a= rs.getInt("TongTien");    list.add(a);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  throw new RuntimeException(ex);  }  return list;  }    public List<Integer> getTongThu\_NgayNay(String tenTaiKhoan) {  List<Integer> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_TongThu\_NgayHomNay (?)}";  rs = jdbcHelper.executeQuery(sql, tenTaiKhoan);  while (rs.next()) {  int a= rs.getInt("TongTien");      list.add(a);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  throw new RuntimeException(ex);  }  return list;  }    ///  //Tong chi  public List<Integer> getTongChi\_Nam(String tenTaiKhoan) {  List<Integer> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_TongChi\_Nam (?)}";  rs = jdbcHelper.executeQuery(sql, tenTaiKhoan);  while (rs.next()) {  int a= rs.getInt("TongTien");      list.add(a);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  throw new RuntimeException(ex);  }  return list;  }    public List<Integer> getTongChi\_NamNay(String tenTaiKhoan) {  List<Integer> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_TongChi\_NamNay (?)}";  rs = jdbcHelper.executeQuery(sql, tenTaiKhoan);  while (rs.next()) {  int a= rs.getInt("TongTien");      list.add(a);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  throw new RuntimeException(ex);  }  return list;  }    public List<Integer> getTongChi\_Thang(String tenTaiKhoan, int year) {  List<Integer> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_TongChi\_Thang (?,?)}";  rs = jdbcHelper.executeQuery(sql, tenTaiKhoan, year);  while (rs.next()) {  int a= rs.getInt("TongTien");    list.add(a);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  throw new RuntimeException(ex);  }  return list;  }    public List<Integer> getTongChi\_Ngay(String tenTaiKhoan, int month, int year) {  List<Integer> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_TongChi\_Ngay (?,?, ?)}";  rs = jdbcHelper.executeQuery(sql, tenTaiKhoan, month, year);  while (rs.next()) {  int a= rs.getInt("TongTien");      list.add(a);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  throw new RuntimeException(ex);  }  return list;  }    public List<Integer> getTongChi\_ThangNay(String tenTaiKhoan) {  List<Integer> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_TongChi\_ThangNay (?)}";  rs = jdbcHelper.executeQuery(sql, tenTaiKhoan);  while (rs.next()) {  int a= rs.getInt("TongTien");    list.add(a);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  throw new RuntimeException(ex);  }  return list;  }    public List<Integer> getTongChi\_NgayNay(String tenTaiKhoan) {  List<Integer> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_TongChi\_NgayHomNay (?)}";  rs = jdbcHelper.executeQuery(sql, tenTaiKhoan);  while (rs.next()) {  int a= rs.getInt("TongTien");      list.add(a);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  throw new RuntimeException(ex);  }  return list;  }  //    public List<Integer> getNam\_Thu(String tenTaiKhoan) {  List<Integer> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_SoNam\_Thu (?)}";  rs = jdbcHelper.executeQuery(sql, tenTaiKhoan);  while (rs.next()) {  int a= rs.getInt("Nam");      list.add(a);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  throw new RuntimeException(ex);  }  return list;  }    public List<Integer> getNam\_Chi(String tenTaiKhoan) {  List<Integer> list = new ArrayList<>();  try {  ResultSet rs = null;    String sql = "{call sp\_SoNam\_Chi (?)}";  rs = jdbcHelper.executeQuery(sql, tenTaiKhoan);  while (rs.next()) {  int a= rs.getInt("Nam");      list.add(a);  }    rs.getStatement().getConnection().close();    } catch (SQLException ex) {  throw new RuntimeException(ex);  }  return list;  }  } |

**4.4 Viết mã ứng dụng**

**4.4.1 Màn hình chào (JDialog)**



MÔ TẢ HOẠT ĐỘNG

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc** |
|  |  |
| Constructor | intt() |
|  |  |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

**/\***

**\* To change this license header, choose License Headers in Project Properties.**

**\* To change this template file, choose Tools | Templates**

**\* and open the template in the editor.**

**\*/**

**package Jdialog\_GT\_DN;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import javax.swing.Timer;**

**/\*\***

**\***

**\* @author Admin**

**\*/**

**public class Loading extends javax.swing.JDialog {**

**public int valueProgress;**

**/\*\***

**\* Creates new form Loading**

**\*/**

**public Loading(java.awt.Frame parent, boolean modal) {**

**super(parent,modal);**

**initComponents();**

**intt();**

**}**

**public void intt(){**

**new Timer(30, new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**int value = jProgessBarLoad.getValue();**

**if(value <100){**

**jProgessBarLoad.setValue(value +1);**

**valueProgress = jProgessBarLoad.getValue();**

**if(value ==0){**

**lblStatus.setText("Khởi động ứng dụng....");**

**}**

**if(value >0 && value<30){**

**lblStatus.setText("Đang kết nối với cơ sở dữ liệu....");**

**}**

**if(value >30 && value<60){**

**lblStatus.setText("Đang lấy dữ liệu từ cơ sở dữ liệu....");**

**}**

**if(value >60 && value<80){**

**lblStatus.setText("Lấy thành công đang load dữ liệu....");**

**}**

**if(value>80 && value<98){**

**lblStatus.setText("Đang khởi tạo chương trình...");**

**}**

**if(value>98){**

**lblStatus.setText("Khởi tạo thành công");**

**}**

**}**

**else{**

**Loading.this.dispose();**

**}**

**}**

**}).start();**

**}**

**/\*\***

**\* This method is called from within the constructor to initialize the form.**

**\* WARNING: Do NOT modify this code. The content of this method is always**

**\* regenerated by the Form Editor.**

**\*/**

**@SuppressWarnings("unchecked")**

**// <editor-fold defaultstate="collapsed" desc="Generated Code">**

**private void initComponents() {**

**jPanel1 = new javax.swing.JPanel();**

**lblStatus = new javax.swing.JLabel();**

**jProgessBarLoad = new javax.swing.JProgressBar();**

**jLabel1 = new javax.swing.JLabel();**

**setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);**

**setUndecorated(true);**

**jPanel1.setLayout(new java.awt.BorderLayout());**

**lblStatus.setBackground(new java.awt.Color(204, 255, 255));**

**lblStatus.setOpaque(true);**

**jPanel1.add(lblStatus, java.awt.BorderLayout.CENTER);**

**jProgessBarLoad.setBackground(new java.awt.Color(255, 102, 102));**

**jProgessBarLoad.setToolTipText("");**

**jProgessBarLoad.setStringPainted(true);**

**jPanel1.add(jProgessBarLoad, java.awt.BorderLayout.PAGE\_END);**

**jLabel1.setBackground(new java.awt.Color(51, 255, 255));**

**jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Images/logo\_title.png"))); // NOI18N**

**jLabel1.setOpaque(true);**

**jPanel1.add(jLabel1, java.awt.BorderLayout.PAGE\_START);**

**javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());**

**getContentPane().setLayout(layout);**

**layout.setHorizontalGroup(**

**layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)**

**.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)**

**);**

**layout.setVerticalGroup(**

**layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)**

**.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, 553, Short.MAX\_VALUE)**

**);**

**pack();**

**setLocationRelativeTo(null);**

**}// </editor-fold>**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**public static void main(String args[]) {**

**/\* Set the Nimbus look and feel \*/**

**//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">**

**/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.**

**\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html**

**\*/**

**try {**

**for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {**

**if ("Nimbus".equals(info.getName())) {**

**javax.swing.UIManager.setLookAndFeel(info.getClassName());**

**break;**

**}**

**}**

**} catch (ClassNotFoundException ex) {**

**java.util.logging.Logger.getLogger(Loading.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (InstantiationException ex) {**

**java.util.logging.Logger.getLogger(Loading.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (IllegalAccessException ex) {**

**java.util.logging.Logger.getLogger(Loading.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

**java.util.logging.Logger.getLogger(Loading.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**}**

**//</editor-fold>**

**/\* Create and display the form \*/**

**java.awt.EventQueue.invokeLater(new Runnable() {**

**public void run(){**

**Loading dialog = new Loading(new javax.swing.JFrame(), true);**

**dialog.addWindowListener(new java.awt.event.WindowAdapter() {**

**@Override**

**public void windowClosing(java.awt.event.WindowEvent e) {**

**System.exit(0);**

**}**

**});**

**dialog.setVisible(true);**

**}**

**});**

**}**

**// Variables declaration - do not modify**

**private javax.swing.JLabel jLabel1;**

**private javax.swing.JPanel jPanel1;**

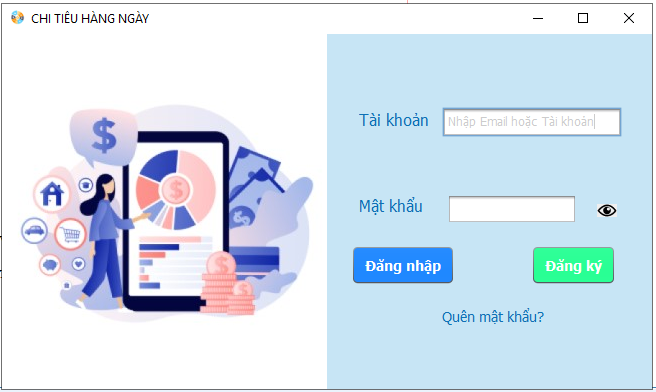
**private javax.swing.JProgressBar jProgessBarLoad;**

**private javax.swing.JLabel lblStatus;**

**// End of variables declaration**

**}**

**4.4.2 Form đăng nhập**



MÔ TẢ HOẠT ĐỘNG

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc** |
| Constructor | intt() |
| btnDangNhap | if(check()){  login();  } |
| btnDangKy | new DangKy().setVisible(true);  this.setVisible(false); |
| txtUsername | txtlUsername.setForeground(Color.black);  if(txtlUsername.getText().equalsIgnoreCase("Nhập Email hoặc Tài khoản")){  txtlUsername.setText("");  }  if(txtlUsername.getText().isEmpty()){  txtlUsername.setForeground(new Color(109,109,109));  txtlUsername.setText("Nhập Email hoặc Tài khoản");  } |
| lblShow | lblShow.setVisible(false);  lblHide.setVisible(true);  txtPass.setEchoChar('\u0000'); |
| lblHide | lblShow.setVisible(true);  lblHide.setVisible(false);  txtPass.setEchoChar('\u25cf'); |
| formWindowClosing | close(); |
| lblQuenMK | new QuenMatKhau\_1().setVisible(true); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

public class Login extends javax.swing.JFrame {

TaiKhoanDAO tkDAO = new TaiKhoanDAO();

/\*\*

\* Creates new form Login

\*/

public Login() {

initComponents();

init();

}

public void init(){

this.setIconImage(ShareHelper.APP\_ICON);

this.setTitle("CHI TIÊU HÀNG NGÀY");

lblHide.setVisible(false);

tatMoDialog();

}

public void tatMoDialog(){

new Loading(this,true).setVisible(true);

}

public boolean check() {

if(Helper.checkHelper.checkNullText(txtlUsername) && Helper.checkHelper.checkNullPass(txtPass)){

return true;

}

else{

return false;

}

}

public void login(){

String userName =txtlUsername.getText();

String matKhau = new String(txtPass.getPassword());

try {

TaiKhoan taikhoan = tkDAO.findById(userName);

if(taikhoan != null){

String matKhau2 = taikhoan.getMatKhau();

if(matKhau.equals(matKhau2)){

ShareHelper.USER = taikhoan;

DialogHelper.alert(this, "Đăng nhập thành công");

this.dispose();

new Main().setVisible(true);

this.dispose();

}

else{

DialogHelper.alert(this, "Sai mật khẩu!");

txtPass.setBackground(Color.red);

txtPass.requestFocus();

}

}

else{

DialogHelper.alert(this, "Sai tên đăng nhập");

txtlUsername.setBackground(Color.red);

txtlUsername.requestFocus();

}

} catch (Exception e) {

}

}

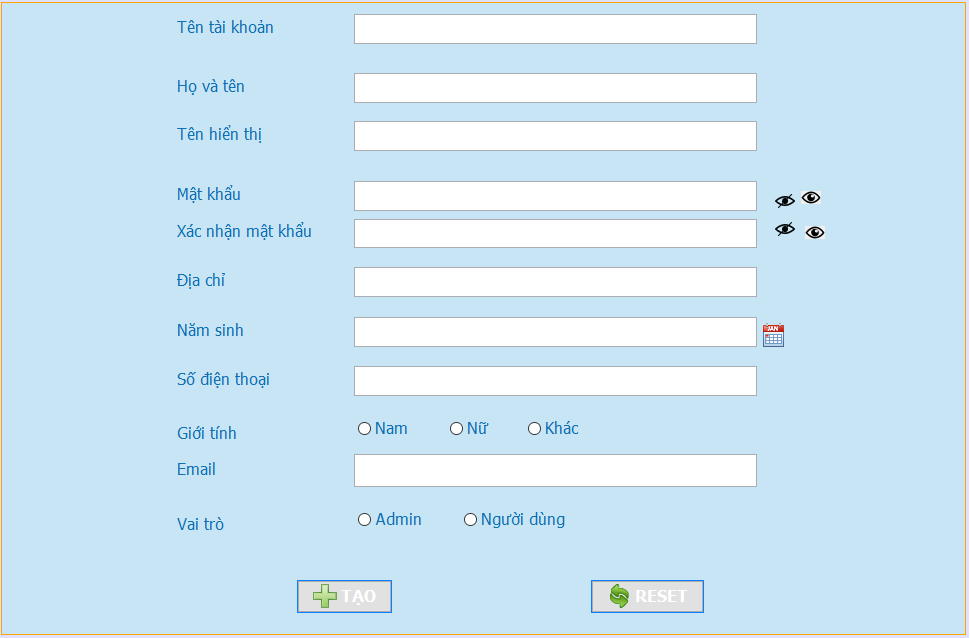
public void close(){

DialogHelper.alert(this, "Cám ơn đã sử dụng phần mềm! \n Chúc bạn một ngày tốt lành!!");

System.exit(0);

}

### **4.4.3 Form Đăng ký**



MÔ TẢ HOẠT ĐỘNG

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc** |
| Constructor | init() |
| btnTao | if(check()){  if(checkTrungTenTaiKhoan\_Email(txtTaiKhoan, txtEmail)){  insert();  this.dispose();  new Login().setVisible(true);  }  } |
| btnReset | clear(); |
| lblClose\_MKMousePressed | lblClose\_MK.setVisible(false);  lblOpen\_MK.setVisible(true);  txtMatKhau.setEchoChar('\u0000'); |
| lblClose\_XNMousePressed | lblClose\_XN.setVisible(false);  lblOpen\_XN.setVisible(true);  txtXacNhanMatKhau.setEchoChar('\u0000'); |
| lblOpen\_MKMousePressed | lblClose\_MK.setVisible(true);  lblOpen\_MK.setVisible(false);  txtMatKhau.setEchoChar('\u25cf'); |
| lblOpen\_XNMousePressed | lblClose\_XN.setVisible(true);  lblOpen\_XN.setVisible(false);  txtXacNhanMatKhau.setEchoChar('\u25cf'); |
| formWindowClosing | CloseJframe(); |

Hướng dẫn cài đặt mã nguồn:

public class DangKy extends javax.swing.JFrame {

/\*

gmail: windnunbe@gmail.com

password: tumottoichin

\*/

/\*\*

\* Creates new form DangKy

\*/

public NguoiDungDAO nguoiDungDAO = new NguoiDungDAO();

public String maND=null;

public DangKy() {

initComponents();

init();

}

public void init(){

lblOpen\_MK.setVisible(false);

lblOpen\_XN.setVisible(false);

setIconImage(ShareHelper.APP\_ICON);

this.setTitle("CHI TIÊU HÀNG NGÀY");

}

public void CloseJframe(){

DialogHelper.alert(this, "Cám ơn đã sử dụng phần mềm! \n Chúc bạn một ngày tốt lành!!");

System.exit(0);

}

public void MinimizeJframe(){

setExtendedState(ICONIFIED);

}

public void MaximaizeJframe(){

if(this.getExtendedState() != NguoiDungJFrame.MAXIMIZED\_BOTH){

this.setExtendedState(NguoiDungJFrame.MAXIMIZED\_BOTH);

}

else{

this.setExtendedState(NguoiDungJFrame.NORMAL);

}

}

public NguoiDung getModelThem(){

NguoiDung model = new NguoiDung();

if(nguoiDungDAO.countSize()<9){

model.setMSND("ND\_0"+String.valueOf(nguoiDungDAO.countSize()+1));

}

else{

model.setMSND("ND\_"+String.valueOf(nguoiDungDAO.countSize()+1));

}

model.setDiaChi(txtDiaChi.getText());

model.setEmail(txtEmail.getText());

if(rdoNam.isSelected()){

model.setGioiTinh(1);

}

else if(rdoNu.isSelected()){

model.setGioiTinh(2);

}

else{

model.setGioiTinh(3);

}

model.setHoTen(txtHoTen.getText());

model.setMatKhau(txtMatKhau.getText());

model.setNamSinh(dateHelper.toDate(txtNamSinh.getText()));

model.setSdt(txtSDT.getText());

model.setTaiKhoan(txtTaiKhoan.getText());

model.setTenHienThi(txtTenHienThi.getText());

return model;

}

public void clear(){

txtTenHienThi.setText("");

maND = null;

txtDiaChi.setText("");

txtEmail.setText("");

txtHoTen.setText("");

txtMatKhau.setText("");

txtNamSinh.setText("");

txtSDT.setText("");

txtTaiKhoan.setText("");

txtTenHienThi.setText("");

grpGT.clearSelection();

grpVaiTro.clearSelection();

}

public void insert(){

NguoiDung model = getModelThem();

try {

nguoiDungDAO.insert(model);

clear();

DialogHelper.alert(this, "Thêm mới thành công");

} catch (Exception e) {

DialogHelper.alert(this, "Thêm mới thất bại");

}

}

public boolean checkNullRdo(JRadioButton rdo1, JRadioButton rdo2, JRadioButton rdo3){

if(rdo1.isSelected()){

return true;

}

else if (rdo2.isSelected()){

return true;

}

else if(rdo3.isSelected()){

return true;

}

else{

DialogHelper.alert(this, "Chọn giới tính " );

return false;

}

}

public boolean checkTrungTenTaiKhoan\_Email(JTextField txtTaiKhoan, JTextField email){

List<NguoiDung> nd = nguoiDungDAO.getTk\_Email(email.getText(), txtTaiKhoan.getText());

txtTaiKhoan.setBackground(white);

if(nd.isEmpty()){

txtTaiKhoan.setBackground(white);

email.setBackground(white);

return true;

}

else{

DialogHelper.alert(this, "Tài khoản hoặc email đã có trong dữ liệu!");

return false;

}

}

public boolean check(){

boolean error = false;

if (checkHelper.checkNullText(txtTaiKhoan)

&& checkHelper.checkNullText(txtHoTen)

&& checkHelper.checkNullText(txtTenHienThi)

&& checkHelper.checkNullPass(txtMatKhau)

&& checkHelper.checkNullPass(txtXacNhanMatKhau)

&& checkHelper.checkNullText(txtDiaChi)

&& checkHelper.checkNullText(txtNamSinh)

&& checkHelper.checkNullText(txtSDT)

&& checkNullRdo(rdoNam, rdoNu, rdoKhac)

&& checkHelper.checkNullText(txtEmail)) {

if (checkHelper.checkTaiKhoan(txtTaiKhoan)

&& checkHelper.checkName(txtTenHienThi)

&& checkHelper.checkName(txtHoTen)

&& checkHelper.checkPass(txtMatKhau)

&& checkHelper.checkXNPass(txtXacNhanMatKhau, txtMatKhau)

&& checkHelper.checkDiaChi(txtDiaChi)

&& checkHelper.checkDate(txtNamSinh)

&& checkHelper.checkSDT(txtSDT)

&& checkHelper.checkEmail(txtEmail))

{

error = true;

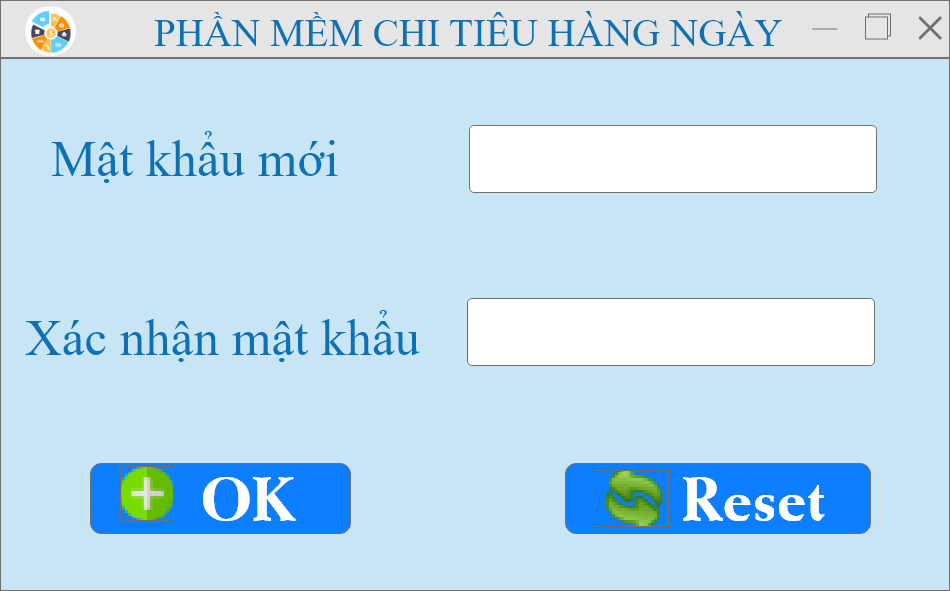
}

}

return error;

}

### **4.4.4 Form Quên mật khẩu**

MÔ TẢ HOẠT ĐỘNG:

Hình 1

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc** |
| Constructor | init() |
| btnLayMa | if(checkHelper.checkNullText(txtTaiKhoan)&&checkHelper.checkNullText(txtEmail)){  if(check()){  layMa();  }  else{  DialogHelper.alert(this, "Vui lòng nhập đúng tài khoản và email đã đăng ký");  }  } |
| btnReset | reset(); |
| btnOK | if(checkHelper.checkNullText(txtTaiKhoan) && checkHelper.checkNullText(txtEmail)&& checkHelper.checkNullText(txtMaXacNhan)){  if(check()){  btnOK();  }  else{  DialogHelper.alert(this, "Vui lòng nhập đúng tài khoản và email đã đăng ký");  }  } |

Hình 2

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc** |
| Constructor | init() |
| btnOK | if(checkHelper.checkNullText(txtMatKhauMoi)&& checkHelper.checkNullText(txtXNMatKhau)){  if(checkHelper.checkKituMK(txtMatKhauMoi) && checkHelper.checkKituMK(txtXNMatKhau))  if(checkHelper.checkXNPass2(txtXNMatKhau, txtMatKhauMoi)){  insertTK();  }    } }  } |
| btnReset | reset(); |

Hướng dẫn cài đặt mã nguồn:

Hình 1

public QuenMatKhau\_1() {

initComponents();

init();

}

public void init(){

setIconImage(ShareHelper.APP\_ICON);

setTitle("CHI TIÊU HÀNG NGÀY");

setLocationRelativeTo(null);

txtMaXacNhan.setEnabled(false);

btnOK.setEnabled(false);

}

public boolean check(){

boolean error = false;

try {

List<NguoiDung> nguoiDung = ndDao.getTk\_Email(txtEmail.getText(),txtTaiKhoan.getText());

if(nguoiDung.isEmpty()){

error=false;

}

else{

error= true;

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn");

}

return error;

}

public void btnLayMa(){

this.temp= sMLayMa.layma();

sMLayMa.sendMail(txtEmail.getText(),temp);

JOptionPane.showMessageDialog(this, "Gửi mã xác nhận thành công","Thông báo", JOptionPane.INFORMATION\_MESSAGE);

txtMaXacNhan.setEnabled(true);

}

public void btnOK(){

try {

if(txtMaXacNhan.getText().equals(this.temp)){

taiKhoan = txtTaiKhoan.getText();

JOptionPane.showMessageDialog(this, "Nhập mã xác nhận đúng","Thông báo", JOptionPane.INFORMATION\_MESSAGE);

this.setVisible(false);

new QuenMatKhau\_2().setVisible(true);

}

else{

JOptionPane.showMessageDialog(this, "Nhập mã xác nhận không đúng","Thông báo", JOptionPane.INFORMATION\_MESSAGE);

}

} catch (Exception e) {

JOptionPane.showMessageDialog(this, "Mã xác nhận bị lỗi","Thông báo", JOptionPane.INFORMATION\_MESSAGE);

}

}

public void reset(){

txtEmail.setText("");

txtMaXacNhan.setText("");

txtMaXacNhan.setEnabled(false);

txtTaiKhoan.setText("");

}

public void layMa(){

if(checkHelper.checkEmail(txtEmail)){

if(check()==true){

btnLayMa();

btnOK.setEnabled(true);

}

else{

DialogHelper.alert(this, "Vui lòng xem lại tài khoản và email có đúng khi đăng ký không?");

}

}

}

Hình 2

public void init(){

setLocationRelativeTo(null);

}

public void insertTK(){

try {

TaiKhoan tk = getModel();

tkDao.update(tk);

DialogHelper.alert(this, "Đổi mật khẩu thành công!");

this.dispose();

new Login().setVisible(true);

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn tài khoản");

}

}

public void reset(){

txtMatKhauMoi.setText("");

txtXNMatKhau.setText("");

}

public TaiKhoan getModel(){

TaiKhoan tk = new TaiKhoan();

tk.setTenTaiKhoan(QuenMatKhau\_1.taiKhoan);

tk.setMatKhau(txtMatKhauMoi.getText());

return tk;

}

### **4.4.5 Form Báo cáo**



MÔ TẢ HOẠT ĐỘNG:

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc** |
| Constructor | init() |
| btnLayMa | if(checkHelper.checkNullText(txtTaiKhoan)&&checkHelper.checkNullText(txtEmail)){  if(check()){  layMa();  }  else{  DialogHelper.alert(this, "Vui lòng nhập đúng tài khoản và email đã đăng ký");  }  } |
| btnReset | reset(); |
| btnOK | if(checkHelper.checkNullText(txtTaiKhoan) && checkHelper.checkNullText(txtEmail)&& checkHelper.checkNullText(txtMaXacNhan)){  if(check()){  btnOK();  }  else{  DialogHelper.alert(this, "Vui lòng nhập đúng tài khoản và email đã đăng ký");  }  } |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

**public TaiKhoanDAO tkDao = new TaiKhoanDAO();**

**public ThongKeDAO thongkeDao = new ThongKeDAO();**

**public ViTienDAO vtDao = new ViTienDAO();**

**public NguoiDungDAO ndDao = new NguoiDungDAO();**

**/\*\***

**\* Creates new form BaoCaoJFrame**

**\*/**

**public BaoCaoJFrame(int index) {**

**initComponents();**

**tabs.setSelectedIndex(index);**

**}**

**public void init(){**

**setIconImage(ShareHelper.APP\_ICON);**

**this.setTitle("CHI TIÊU HÀNG NGÀY");**

**setLocationRelativeTo(null);**

**txtTongThuChi\_Ngay.setEditable(false);**

**txtTongChi\_Ngay.setEditable(false);**

**txtTongThu\_Ngay.setEditable(false);**

**getTaiChinh();**

**load();**

**fillComboBoxChi();**

**fillComboBoxThu();**

**fillComboBoxThuChi();**

**showBar();**

**}**

**public void showBar(){**

**showBarChart\_Nam\_ThuChi();**

**showBarChart\_Nam\_Thu();**

**showBarChart\_Nam\_Chi();**

**showTongTien();**

**}**

**public void showTongTien(){**

**getTaiChinh\_ThuChi\_Nam();**

**getTaiChinh\_Chi\_Nam();**

**getTaiChinh\_Thu\_Nam();**

**}**

**public void CloseJframe(){**

**DialogHelper.alert(this, "Cám ơn đã sử dụng phần mềm! \n Chúc bạn một ngày tốt lành!!");**

**System.exit(0);**

**}**

**public boolean checkThuChi(JTextField txtThang, JTextField txtNam){**

**boolean error = false;**

**if(checkHelper.checkNullText(txtThang)&&checkHelper.checkNullText(txtNam)){**

**if(checkHelper.checkThang(txtThang) && checkHelper.checkNam(txtNam)){**

**error = true;**

**}**

**}**

**return error;**

**}**

**//hover cac bang ben trai**

**public void miceEntered(JPanel panel, JLabel lbl){**

**Font myFont = new Font("Tahoma", Font.BOLD, 14);**

**panel.setBackground(new Color(114,206,255));**

**lbl.setForeground(white);**

**lbl.setFont(myFont);**

**}**

**public void miceExited(JPanel panel, JLabel lbl){**

**Font myFont = new Font("Tahoma", Font.BOLD, 14);**

**panel.setBackground(new Color(199,229,245));**

**lbl.setForeground(new Color(59,204,253));**

**lbl.setFont(myFont);**

**}**

**//Get tai chinh**

**public void getTaiChinh(){**

**try {**

**List<String> taichinh = thongkeDao.getTaiChinh(ShareHelper.USER.getTenTaiKhoan());**

**String TongTien = null;**

**if(taichinh.isEmpty()){**

**DialogHelper.alert(this, "Tài khoản này không có dữ liệu tài chính hiện tại!!");**

**}**

**else{**

**for(String tc : taichinh){**

**TongTien = tc;**

**}**

**if(txtTaiChinh.getText().equalsIgnoreCase("")){**

**txtTaiChinh.setForeground(new Color(93, 255, 51));**

**txtTaiChinh.setText("0 VNĐ");**

**}**

**else{**

**txtTaiChinh.setForeground(new Color(93, 255, 51));**

**txtTaiChinh.setText(TongTien+ " VNĐ");**

**}**

**}**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**public void load(){**

**try {**

**DefaultTableModel model = (DefaultTableModel) tblTaiChinh.getModel();**

**model.setRowCount(0);**

**List<ViTien> list = vtDao.selectByKeyword(ShareHelper.USER.getTenTaiKhoan());**

**Object [] ten= {"TÊN VÍ","SỐ DƯ(VNĐ)"};**

**model.setColumnIdentifiers(ten);**

**for(ViTien vt:list){**

**Object [] rows= {**

**vt.getTenVi(),**

**vt.getSoDu(),**

**};**

**model.addRow(rows);**

**}**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn dữ liệu");**

**e.printStackTrace();**

**}**

**}**

**//fill combobox**

**public void fillComboBoxThu(){**

**try {**

**DefaultComboBoxModel model = (DefaultComboBoxModel) cboThang\_Thu.getModel();**

**model.removeAllElements();**

**List<Integer> nam = thongkeDao.getNam\_Thu(ShareHelper.USER.getTenTaiKhoan());**

**if(nam.isEmpty()){**

**DialogHelper.alert(this, "Tài khoản này chưa có khoản thu");**

**}**

**else{**

**for(Integer nT: nam){**

**model.addElement(nT.intValue());**

**}**

**cboThang\_Thu.setSelectedIndex(0);**

**}**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**public void fillComboBoxThuChi(){**

**try {**

**DefaultComboBoxModel model = (DefaultComboBoxModel) cboThang\_ThuChi.getModel();**

**model.removeAllElements();**

**List<Integer> nam = thongkeDao.getNam\_Thu(ShareHelper.USER.getTenTaiKhoan());**

**if(nam.isEmpty()){**

**DialogHelper.alert(this, "Tài khoản này chưa có khoản thu chi");**

**}**

**else{**

**for(Integer nT: nam){**

**model.addElement(nT.intValue());**

**}**

**cboThang\_ThuChi.setSelectedIndex(0);**

**}**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**public void fillComboBoxChi(){**

**try {**

**DefaultComboBoxModel model = (DefaultComboBoxModel) cboThang\_Chi.getModel();**

**model.removeAllElements();**

**List<Integer> nam = thongkeDao.getNam\_Chi(ShareHelper.USER.getTenTaiKhoan());**

**if(nam.isEmpty()){**

**DialogHelper.alert(this, "Tài khoản này chưa có khoản chi");**

**}**

**else{**

**for(Integer nT: nam){**

**model.addElement(nT.intValue());**

**}**

**cboThang\_Chi.setSelectedIndex(0);**

**}**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**//PANEL TINH HINH THU**

**public void showBarChart\_Nam\_Thu(){**

**try {**

**DefaultCategoryDataset dataset = new DefaultCategoryDataset();**

**List<ThongKe\_Nam> tk = thongkeDao.getTKThu\_Nam(ShareHelper.USER.getTenTaiKhoan());**

**for(ThongKe\_Nam tk1: tk){**

**dataset.setValue(tk1.getTongTien(), "Thu nhập", String.valueOf(tk1.getNam()));**

**}**

**JFreeChart chart = ChartFactory.createBarChart("Phân tích thu","Năm","Tổng Tiền",**

**dataset, PlotOrientation.VERTICAL, true ,true,false);**

**CategoryPlot categoryPlot = chart.getCategoryPlot();**

**categoryPlot.setRangeGridlinePaint(Color.BLUE);**

**categoryPlot.setBackgroundPaint(Color.WHITE);**

**BarRenderer renderer = (BarRenderer) categoryPlot.getRenderer();**

**renderer.setItemMargin(0);**

**CategoryAxis domain = categoryPlot.getDomainAxis();**

**domain.setLowerMargin(0.25);**

**domain.setUpperMargin(0.25);**

**Color clr3 = new Color(93, 255, 51);**

**// Color clr3 = new Color(204,0,51);**

**renderer.setSeriesPaint(0, clr3);**

**ChartPanel barChartPanel = new ChartPanel(chart);**

**panelBarChartThu\_Nam.removeAll();**

**panelBarChartThu\_Nam.add(barChartPanel);**

**panelBarChartThu\_Nam.updateUI();**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**public void showBarChart\_Thang\_Thu(){**

**try {**

**int nam = (int) cboThang\_Thu.getSelectedItem();**

**DefaultCategoryDataset dataset = new DefaultCategoryDataset();**

**List<ThongKe\_Thang> tk = thongkeDao.getTKThu\_Thang(ShareHelper.USER.getTenTaiKhoan(),nam);**

**for(ThongKe\_Thang tk1: tk){**

**dataset.setValue(tk1.getTongTien(), "Thu nhập", String.valueOf(tk1.getNgay()));**

**}**

**//**

**//**

**JFreeChart chart = ChartFactory.createBarChart("Phân tích thu","Tháng","Tổng Tiền",**

**dataset, PlotOrientation.VERTICAL, true ,true,false);**

**CategoryPlot categoryPlot = chart.getCategoryPlot();**

**categoryPlot.setRangeGridlinePaint(Color.BLUE);**

**categoryPlot.setBackgroundPaint(Color.WHITE);**

**BarRenderer renderer = (BarRenderer) categoryPlot.getRenderer();**

**renderer.setItemMargin(0);**

**CategoryAxis domain = categoryPlot.getDomainAxis();**

**domain.setLowerMargin(0.25);**

**domain.setUpperMargin(0.25);**

**Color clr3 = new Color(93, 255, 51);**

**// Color clr3 = new Color(204,0,51);**

**renderer.setSeriesPaint(0, clr3);**

**ChartPanel barpChartPanel = new ChartPanel(chart);**

**panelBarChartThu\_Thang.removeAll();**

**panelBarChartThu\_Thang.add(barpChartPanel);**

**panelBarChartThu\_Thang.updateUI();**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**public void showBarChart\_Ngay\_Thu(){**

**try {**

**int thang = Integer.parseInt(txtThang\_Thu.getText());**

**int nam = Integer.parseInt(txtNam\_Thu.getText());**

**DefaultCategoryDataset dataset = new DefaultCategoryDataset();**

**List<ThongKe\_Ngay> tk = thongkeDao.getTKThu\_Ngay(ShareHelper.USER.getTenTaiKhoan(),thang,nam);**

**for(ThongKe\_Ngay tk1: tk){**

**dataset.setValue(tk1.getTongTien(), "Thu nhập", dateHelper.toString(tk1.getNgay()));**

**}**

**//**

**//**

**JFreeChart chart = ChartFactory.createBarChart("Phân tích thu","Ngày","Tổng Tiền",**

**dataset, PlotOrientation.VERTICAL, true ,true,false);**

**CategoryPlot categoryPlot = chart.getCategoryPlot();**

**categoryPlot.setRangeGridlinePaint(Color.BLUE);**

**categoryPlot.setBackgroundPaint(Color.WHITE);**

**BarRenderer renderer = (BarRenderer) categoryPlot.getRenderer();**

**renderer.setItemMargin(0);**

**CategoryAxis domain = categoryPlot.getDomainAxis();**

**domain.setLowerMargin(0.25);**

**domain.setUpperMargin(0.25);**

**Color clr3 = new Color(93, 255, 51);**

**// Color clr3 = new Color(204,0,51);**

**renderer.setSeriesPaint(0, clr3);**

**ChartPanel barpChartPanel = new ChartPanel(chart);**

**panelBarChartThu\_Ngay.removeAll();**

**panelBarChartThu\_Ngay.add(barpChartPanel);**

**panelBarChartThu\_Ngay.validate();**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**//PANEL TINH HINH CHI TIEU**

**public void showBarChart\_Thang\_Chi(){**

**try {**

**int nam = (int) cboThang\_Chi.getSelectedItem();**

**DefaultCategoryDataset dataset = new DefaultCategoryDataset();**

**List<ThongKe\_Thang> tk = thongkeDao.getTKChi\_Thang(ShareHelper.USER.getTenTaiKhoan(),nam);**

**for(ThongKe\_Thang tk1: tk){**

**dataset.setValue(tk1.getTongTien(), "Chi tiêu", String.valueOf(tk1.getNgay()));**

**}**

**//**

**//**

**JFreeChart chart = ChartFactory.createBarChart("Phân tích chi","Tháng","Tổng Tiền",**

**dataset, PlotOrientation.VERTICAL, true ,true,false);**

**CategoryPlot categoryPlot = chart.getCategoryPlot();**

**categoryPlot.setRangeGridlinePaint(Color.BLUE);**

**categoryPlot.setBackgroundPaint(Color.WHITE);**

**BarRenderer renderer = (BarRenderer) categoryPlot.getRenderer();**

**renderer.setItemMargin(0);**

**CategoryAxis domain = categoryPlot.getDomainAxis();**

**// Color clr3 = new Color(93, 255, 51);**

**Color clr3 = new Color(204,0,51);**

**renderer.setSeriesPaint(0, clr3);**

**domain.setLowerMargin(0.25);**

**domain.setUpperMargin(0.25);**

**ChartPanel barpChartPanel = new ChartPanel(chart);**

**panelBarChartChi\_Thang.removeAll();**

**panelBarChartChi\_Thang.add(barpChartPanel);**

**panelBarChartChi\_Thang.validate();**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**public void showBarChart\_Nam\_Chi(){**

**try {**

**DefaultCategoryDataset dataset = new DefaultCategoryDataset();**

**List<ThongKe\_Nam> tk = thongkeDao.getTKChi\_Nam(ShareHelper.USER.getTenTaiKhoan());**

**for(ThongKe\_Nam tk1: tk){**

**dataset.setValue(tk1.getTongTien(), "Chi tiêu", String.valueOf(tk1.getNam()));**

**}**

**//**

**//**

**JFreeChart chart = ChartFactory.createBarChart("Phân tích chi","Năm","Tổng Tiền",**

**dataset, PlotOrientation.VERTICAL, true ,true,false);**

**CategoryPlot categoryPlot = chart.getCategoryPlot();**

**categoryPlot.setRangeGridlinePaint(Color.BLUE);**

**categoryPlot.setBackgroundPaint(Color.WHITE);**

**BarRenderer renderer = (BarRenderer) categoryPlot.getRenderer();**

**renderer.setItemMargin(0);**

**CategoryAxis domain = categoryPlot.getDomainAxis();**

**domain.setLowerMargin(0.25);**

**domain.setUpperMargin(0.25);**

**Color clr3 = new Color(204,0,51);**

**renderer.setSeriesPaint(0, clr3);**

**ChartPanel barpChartPanel = new ChartPanel(chart);**

**panelBarChartChi\_Nam.removeAll();**

**panelBarChartChi\_Nam.add(barpChartPanel);**

**panelBarChartChi\_Nam.validate();**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**public void showBarChart\_Ngay\_Chi(){**

**try {**

**int thang = Integer.parseInt(txtThang\_Chi.getText());**

**int nam = Integer.parseInt(txtNam\_Chi.getText());**

**DefaultCategoryDataset dataset = new DefaultCategoryDataset();**

**List<ThongKe\_Ngay> tk = thongkeDao.getTKChi\_Ngay(ShareHelper.USER.getTenTaiKhoan(),thang,nam);**

**for(ThongKe\_Ngay tk1: tk){**

**dataset.setValue(tk1.getTongTien(), "Chi tiêu", dateHelper.toString(tk1.getNgay()));**

**}**

**//**

**//**

**JFreeChart chart = ChartFactory.createBarChart("Phân tích chi","Ngày","Tổng Tiền",**

**dataset, PlotOrientation.VERTICAL, true ,true,false);**

**CategoryPlot categoryPlot = chart.getCategoryPlot();**

**categoryPlot.setRangeGridlinePaint(Color.BLUE);**

**categoryPlot.setBackgroundPaint(Color.WHITE);**

**BarRenderer renderer = (BarRenderer) categoryPlot.getRenderer();**

**renderer.setItemMargin(0);**

**CategoryAxis domain = categoryPlot.getDomainAxis();**

**domain.setLowerMargin(0.25);**

**domain.setUpperMargin(0.25);**

**Color clr3 = new Color(204,0,51);**

**renderer.setSeriesPaint(0, clr3);**

**ChartPanel barpChartPanel = new ChartPanel(chart);**

**panelBarChartChi\_Ngay.removeAll();**

**panelBarChartChi\_Ngay.add(barpChartPanel);**

**panelBarChartChi\_Ngay.validate();**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**//PANEL TINH HINH THU CHI**

**public void showBarChart\_Nam\_ThuChi(){**

**try {**

**DefaultCategoryDataset dataset = new DefaultCategoryDataset();**

**List<ThongKe\_Nam> tk = thongkeDao.getTKThu\_Nam(ShareHelper.USER.getTenTaiKhoan());**

**for(ThongKe\_Nam tk1: tk){**

**dataset.setValue(tk1.getTongTien(), "Thu nhập", String.valueOf(tk1.getNam()));**

**}**

**List<ThongKe\_Nam> tk2 = thongkeDao.getTKChi\_Nam("babaraus123");**

**for(ThongKe\_Nam tk1: tk2){**

**dataset.setValue(tk1.getTongTien(), "Chi tiêu", String.valueOf(tk1.getNam()));**

**}**

**JFreeChart chart = ChartFactory.createBarChart("Phân tích thu","Năm","Tổng Tiền",**

**dataset, PlotOrientation.VERTICAL, true ,true,false);**

**CategoryPlot categoryPlot = chart.getCategoryPlot();**

**categoryPlot.setRangeGridlinePaint(Color.BLUE);**

**categoryPlot.setBackgroundPaint(Color.WHITE);**

**BarRenderer renderer = (BarRenderer) categoryPlot.getRenderer();**

**renderer.setItemMargin(0);**

**CategoryAxis domain = categoryPlot.getDomainAxis();**

**domain.setLowerMargin(0.25);**

**domain.setUpperMargin(0.25);**

**Color clr3 = new Color(93, 255, 51);**

**// Color clr3 = new Color(204,0,51);**

**renderer.setSeriesPaint(0, clr3);**

**ChartPanel barChartPanel = new ChartPanel(chart);**

**panelBarChartThuChi\_Nam.removeAll();**

**panelBarChartThuChi\_Nam.add(barChartPanel);**

**panelBarChartThuChi\_Nam.updateUI();**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**public void showBarChart\_Thang\_ThuChi(){**

**try {**

**int nam = (int) cboThang\_ThuChi.getSelectedItem();**

**DefaultCategoryDataset dataset = new DefaultCategoryDataset();**

**List<ThongKe\_Thang> tk = thongkeDao.getTKThu\_Thang(ShareHelper.USER.getTenTaiKhoan(),nam);**

**for(ThongKe\_Thang tk1: tk){**

**dataset.setValue(tk1.getTongTien(), "Thu nhập", String.valueOf(tk1.getNgay()));**

**}**

**//**

**List<ThongKe\_Thang> tk2 = thongkeDao.getTKChi\_Thang(ShareHelper.USER.getTenTaiKhoan(),nam);**

**for(ThongKe\_Thang tk1: tk2){**

**dataset.setValue(tk1.getTongTien(), "Chi tiêu", String.valueOf(tk1.getNgay()));**

**}**

**JFreeChart chart = ChartFactory.createBarChart("Phân tích thu","Tháng","Tổng Tiền",**

**dataset, PlotOrientation.VERTICAL, true ,true,false);**

**CategoryPlot categoryPlot = chart.getCategoryPlot();**

**categoryPlot.setRangeGridlinePaint(Color.BLUE);**

**categoryPlot.setBackgroundPaint(Color.WHITE);**

**BarRenderer renderer = (BarRenderer) categoryPlot.getRenderer();**

**renderer.setItemMargin(0);**

**CategoryAxis domain = categoryPlot.getDomainAxis();**

**domain.setLowerMargin(0.25);**

**domain.setUpperMargin(0.25);**

**Color clr3 = new Color(93, 255, 51);**

**// Color clr3 = new Color(204,0,51);**

**renderer.setSeriesPaint(0, clr3);**

**ChartPanel barpChartPanel = new ChartPanel(chart);**

**panelBarChartThuChi\_Thang.removeAll();**

**panelBarChartThuChi\_Thang.add(barpChartPanel);**

**panelBarChartThuChi\_Thang.updateUI();**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**public void showBarChart\_Ngay\_ThuChi(){**

**try {**

**int thang = Integer.parseInt(txtThang\_ThuChi.getText());**

**int nam = Integer.parseInt(txtNam\_ThuChi.getText());**

**DefaultCategoryDataset dataset = new DefaultCategoryDataset();**

**List<ThongKe\_Ngay> tk = thongkeDao.getTKThu\_Ngay(ShareHelper.USER.getTenTaiKhoan(),thang,nam);**

**for(ThongKe\_Ngay tk1: tk){**

**dataset.setValue(tk1.getTongTien(), "Thu nhập", dateHelper.toString(tk1.getNgay()));**

**}**

**//**

**List<ThongKe\_Ngay> tk2 = thongkeDao.getTKChi\_Ngay(ShareHelper.USER.getTenTaiKhoan(),thang,nam);**

**for(ThongKe\_Ngay tk1: tk2){**

**dataset.setValue(tk1.getTongTien(), "Chi tiêu", dateHelper.toString(tk1.getNgay()));**

**}**

**//**

**JFreeChart chart = ChartFactory.createBarChart("Phân tích thu","Ngày","Tổng Tiền",**

**dataset, PlotOrientation.VERTICAL, true ,true,false);**

**CategoryPlot categoryPlot = chart.getCategoryPlot();**

**categoryPlot.setRangeGridlinePaint(Color.BLUE);**

**categoryPlot.setBackgroundPaint(Color.WHITE);**

**BarRenderer renderer = (BarRenderer) categoryPlot.getRenderer();**

**renderer.setItemMargin(0);**

**CategoryAxis domain = categoryPlot.getDomainAxis();**

**domain.setLowerMargin(0.25);**

**domain.setUpperMargin(0.25);**

**Color clr3 = new Color(93, 255, 51);**

**// Color clr3 = new Color(204,0,51);**

**renderer.setSeriesPaint(0, clr3);**

**ChartPanel barpChartPanel = new ChartPanel(chart);**

**panelBarChartThuChi\_Ngay.removeAll();**

**panelBarChartThuChi\_Ngay.add(barpChartPanel);**

**panelBarChartThuChi\_Ngay.validate();**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**public void getTaiChinh\_ThuChi\_Nam(){**

**try {**

**List<Integer> taichinh = thongkeDao.getTongThuChi\_Nam(ShareHelper.USER.getTenTaiKhoan());**

**int TongTien = 0;**

**if(taichinh.isEmpty()){**

**DialogHelper.alert(this, "Tài khoản này không có dữ liệu thu chi hiện tại!!");**

**}**

**else{**

**for(Integer tc : taichinh){**

**TongTien = tc.intValue();**

**}**

**if(TongTien<0){**

**txtTongThuChi\_Nam.setForeground(new Color(204,0,51));**

**}**

**else{**

**txtTongThuChi\_Nam.setForeground(new Color(93, 255, 51));**

**}**

**txtTongThuChi\_Nam.setEditable(false);**

**txtTongThuChi\_Nam.setText(TongTien+ " VNĐ");**

**}**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**public void getTaiChinh\_ThuChi\_Thang(){**

**try {**

**int nam = (int) cboThang\_ThuChi.getSelectedItem();**

**List<Integer> taichinh = thongkeDao.getTongThuChi\_Thang(ShareHelper.USER.getTenTaiKhoan(), nam);**

**int TongTien = 0;**

**if(taichinh.isEmpty()){**

**DialogHelper.alert(this, "Tài khoản này không có dữ liệu thu chi hiện tại!!");**

**}**

**else{**

**for(Integer tc : taichinh){**

**TongTien = tc.intValue();**

**}**

**if(TongTien<0){**

**txtTongThuChi\_Thang.setForeground(new Color(204,0,51));**

**}**

**else{**

**txtTongThuChi\_Thang.setForeground(new Color(93, 255, 51));**

**}**

**txtTongThuChi\_Thang.setEditable(false);**

**txtTongThuChi\_Thang.setText(TongTien+ " VNĐ");**

**}**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**public void getTaiChinh\_ThuChi\_Ngay(){**

**try {**

**List<Integer> taichinh = thongkeDao.getTongThuChi\_Ngay(ShareHelper.USER.getTenTaiKhoan(), Integer.parseInt(txtThang\_ThuChi.getText()),Integer.parseInt(txtNam\_ThuChi.getText()));**

**int TongTien = 0;**

**if(taichinh.isEmpty()){**

**DialogHelper.alert(this, "Tài khoản này không có dữ liệu thu chi hiện tại!!");**

**}**

**else{**

**for(Integer tc : taichinh){**

**TongTien = tc.intValue();**

**}**

**if(TongTien<0){**

**txtTongThuChi\_Ngay.setForeground(new Color(204,0,51));**

**}**

**else{**

**txtTongThuChi\_Ngay.setForeground(new Color(93, 255, 51));**

**}**

**txtTongThuChi\_Ngay.setEditable(false);**

**txtTongThuChi\_Ngay.setText(TongTien+ " VNĐ");**

**}**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**public void getTaiChinh\_Thu\_Nam(){**

**try {**

**List<Integer> taichinh = thongkeDao.getTongThu\_Nam(ShareHelper.USER.getTenTaiKhoan());**

**int TongTien = 0;**

**if(taichinh.isEmpty()){**

**DialogHelper.alert(this, "Tài khoản này không có dữ liệu thu hiện tại!!");**

**}**

**else{**

**for(Integer tc : taichinh){**

**TongTien = tc.intValue();**

**}**

**if(TongTien<0){**

**txtTongThu\_Nam.setForeground(new Color(204,0,51));**

**}**

**else{**

**txtTongThu\_Nam.setForeground(new Color(93, 255, 51));**

**}**

**txtTongThu\_Nam.setEditable(false);**

**txtTongThu\_Nam.setText(TongTien+ " VNĐ");**

**}**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**public void getTaiChinh\_Thu\_Thang(){**

**try {**

**int nam = (int) cboThang\_Thu.getSelectedItem();**

**List<Integer> taichinh = thongkeDao.getTongThu\_Thang(ShareHelper.USER.getTenTaiKhoan(), nam);**

**int TongTien = 0;**

**if(taichinh.isEmpty()){**

**DialogHelper.alert(this, "Tài khoản này không có dữ liệu thu hiện tại!!");**

**}**

**else{**

**for(Integer tc : taichinh){**

**TongTien = tc.intValue();**

**}**

**if(TongTien<0){**

**txtTongThu\_Thang.setForeground(new Color(204,0,51));**

**}**

**else{**

**txtTongThu\_Thang.setForeground(new Color(93, 255, 51));**

**}**

**txtTongThu\_Thang.setEditable(false);**

**txtTongThu\_Thang.setText(TongTien+ " VNĐ");**

**}**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**public void getTaiChinh\_Thu\_Ngay(){**

**try {**

**int thang = Integer.parseInt(txtThang\_Thu.getText());**

**int nam = Integer.parseInt(txtNam\_Thu.getText());**

**List<Integer> taichinh = thongkeDao.getTongThu\_Ngay(ShareHelper.USER.getTenTaiKhoan(), thang,nam);**

**int TongTien = 0;**

**if(taichinh.isEmpty()){**

**DialogHelper.alert(this, "Tài khoản này không có dữ liệu thu hiện tại!!");**

**}**

**else{**

**for(Integer tc : taichinh){**

**TongTien = tc.intValue();**

**}**

**if(TongTien<0){**

**txtTongThu\_Ngay.setForeground(new Color(204,0,51));**

**}**

**else{**

**txtTongThu\_Ngay.setForeground(new Color(93, 255, 51));**

**}**

**txtTongThu\_Ngay.setEditable(false);**

**txtTongThu\_Ngay.setText(TongTien+ " VNĐ");**

**}**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**//Tai chinh Chi**

**public void getTaiChinh\_Chi\_Nam(){**

**try {**

**List<Integer> taichinh = thongkeDao.getTongChi\_Nam(ShareHelper.USER.getTenTaiKhoan());**

**int TongTien = 0;**

**if(taichinh.isEmpty()){**

**DialogHelper.alert(this, "Tài khoản này không có dữ liệu chi hiện tại!!");**

**}**

**else{**

**for(Integer tc : taichinh){**

**TongTien = tc.intValue();**

**}**

**if(TongTien<0){**

**txtTongChi\_Nam.setForeground(new Color(204,0,51));**

**}**

**else{**

**txtTongChi\_Nam.setForeground(new Color(93, 255, 51));**

**}**

**txtTongChi\_Nam.setEditable(false);**

**txtTongChi\_Nam.setText(TongTien+ " VNĐ");**

**}**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**public void getTaiChinh\_Chi\_Thang(){**

**try {**

**int nam = (int) cboThang\_Chi.getSelectedItem();**

**List<Integer> taichinh = thongkeDao.getTongChi\_Thang(ShareHelper.USER.getTenTaiKhoan(), nam);**

**int TongTien = 0;**

**if(taichinh.isEmpty()){**

**DialogHelper.alert(this, "Tài khoản này không có dữ liệu chi hiện tại!!");**

**}**

**else{**

**for(Integer tc : taichinh){**

**TongTien = tc.intValue();**

**}**

**if(TongTien<0){**

**txtTongChi\_Thang.setForeground(new Color(204,0,51));**

**}**

**else{**

**txtTongChi\_Thang.setForeground(new Color(93, 255, 51));**

**}**

**txtTongChi\_Thang.setEditable(false);**

**txtTongChi\_Thang.setText(TongTien+ " VNĐ");**

**}**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

**}**

**}**

**public void getTaiChinh\_Chi\_Ngay(){**

**try {**

**int thang = Integer.parseInt(txtThang\_Chi.getText());**

**int nam = Integer.parseInt(txtNam\_Chi.getText());**

**List<Integer> taichinh = thongkeDao.getTongChi\_Ngay(ShareHelper.USER.getTenTaiKhoan(), thang,nam);**

**int TongTien = 0;**

**if(taichinh.isEmpty()){**

**DialogHelper.alert(this, "Tài khoản này không có dữ liệu chi hiện tại!!");**

**}**

**else{**

**for(Integer tc : taichinh){**

**TongTien = tc.intValue();**

**}**

**if(TongTien<0){**

**txtTongChi\_Ngay.setForeground(new Color(204,0,51));**

**}**

**else{**

**txtTongChi\_Ngay.setForeground(new Color(93, 255, 51));**

**}**

**txtTongChi\_Ngay.setEditable(false);**

**txtTongChi\_Ngay.setText(TongTien+ " VNĐ");**

**}**

**} catch (Exception e) {**

**DialogHelper.alert(this, "Lỗi truy vấn");**

**e.printStackTrace();**

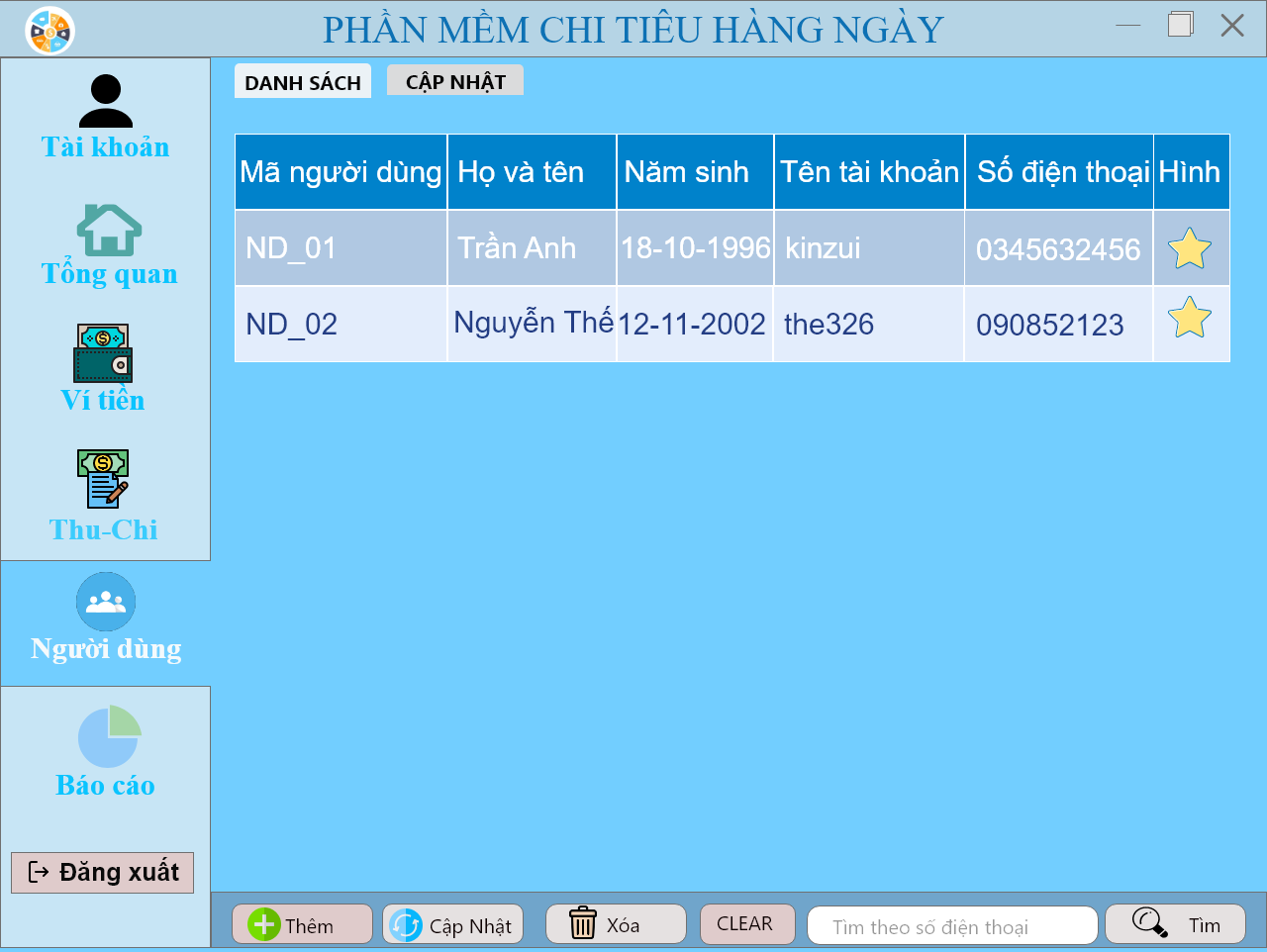
**}**

**}**

### **4.4.6 Form Người dùng**

Người dùng – admin





Người dùng – Người dùng



MÔ TẢ HOẠT ĐỘNG

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

Người dùng – admin

public int index = -1;

public NguoiDungDAO nguoiDungDAO = new NguoiDungDAO();

public String maND=null;

public TaiKhoanDAO tkDao = new TaiKhoanDAO();

/\*\*

\* Creates new form NguoiDungJFrame

\*/

public NguoiDungJFrame() {

initComponents();

init();

}

public void init(){

setIconImage(ShareHelper.APP\_ICON);

this.setTitle("CHI TIÊU HÀNG NGÀY");

setStatus(true);

}

public void setStatus(boolean status){

btnThem.setEnabled(status);

btnCapNhat.setEnabled(!status);

btnXoa.setEnabled(!status);

txtTaiKhoan.setEnabled(status);

btnTim.setEnabled(status);

boolean first = this.index > 0;

boolean last = this.index < tblNguoiDung.getRowCount() - 1;

btnFirst.setEnabled(!status && first);

btnPrev.setEnabled(!status && first);

btnLast.setEnabled(!status && last);

btnNext.setEnabled(!status && last);

}

public void CloseJframe(){

DialogHelper.alert(this, "Cám ơn đã sử dụng phần mềm! \n Chúc bạn một ngày tốt lành!!");

System.exit(0);

}

public void MinimizeJframe(){

setExtendedState(ICONIFIED);

}

public void MaximaizeJframe(){

if(this.getExtendedState() != NguoiDungJFrame.MAXIMIZED\_BOTH){

this.setExtendedState(NguoiDungJFrame.MAXIMIZED\_BOTH);

}

else{

this.setExtendedState(NguoiDungJFrame.NORMAL);

}

}

////

public void load(){

try {

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

model.setRowCount(0);

List<NguoiDung> ndlist = new ArrayList<>();

String timKiem = txtTim.getText();

ndlist= nguoiDungDAO.selectByKeywordSDT(timKiem);

if(ndlist.isEmpty()){

DialogHelper.alert(this, "Người dùng này không có trong dữ liệu");

}

for(NguoiDung nd : ndlist){

Object [] rows = {

nd.getMSND(),

nd.getHoTen(),

dateHelper.toString(nd.getNamSinh()),

nd.getTaiKhoan(),

nd.getSdt(),

nd.isVaiTro()?"Admin":"Người dùng",

nd.getHinh()

};

model.addRow(rows);

}

lblSoDem.setText(1+ "/"+ nguoiDungDAO.countSize());

} catch (Exception e) {

DialogHelper.alert(this, "Tải dữ liệu bảng thất bại");

}

}

public void setTrang(){

txtTenHienThi.setBackground(white);

txtDiaChi.setBackground(white);

txtEmail.setBackground(white);

txtHoTen.setBackground(white);

txtMatKhau.setBackground(white);

txtNamSinh.setBackground(white);

txtSDT.setBackground(white);

txtTaiKhoan.setBackground(white);

txtTenHienThi.setBackground(white);

}

public void clear(){

setTrang();

txtTenHienThi.setText("");

maND = null;

txtDiaChi.setText("");

txtEmail.setText("");

txtHoTen.setText("");

txtMatKhau.setText("");

txtNamSinh.setText("");

txtSDT.setText("");

txtTaiKhoan.setText("");

txtTenHienThi.setText("");

grpGT.clearSelection();

lblHinh.setIcon(new ImageIcon());

grpVT.clearSelection();

txtTim.setText("");

}

public NguoiDung getModelThem(){

NguoiDung model = new NguoiDung();

if(nguoiDungDAO.countSize()<9){

model.setMSND("ND\_0"+String.valueOf(nguoiDungDAO.countSize()+1));

}

else{

model.setMSND("ND\_"+String.valueOf(nguoiDungDAO.countSize()+1));

}

model.setDiaChi(txtDiaChi.getText());

model.setEmail(txtEmail.getText());

if(rdoNam.isSelected()){

model.setGioiTinh(1);

}

if(rdoNu.isSelected()){

model.setGioiTinh(2);

}

if(rdoKhac.isSelected()){

model.setGioiTinh(3);

}

model.setHinh(lblHinh.getToolTipText());

model.setHoTen(txtHoTen.getText());

model.setMatKhau(txtMatKhau.getText());

model.setNamSinh(dateHelper.toDate(txtNamSinh.getText()));

model.setSdt(txtSDT.getText());

model.setTaiKhoan(txtTaiKhoan.getText());

model.setTenHienThi(txtTenHienThi.getText());

if(rdoAdmin.isSelected()){

model.setVaiTro(true);

}

if(rdoNguoiDung.isSelected()){

model.setVaiTro(false);

}

return model;

}

public NguoiDung getModel(){

NguoiDung model = new NguoiDung();

model.setMSND(maND);

model.setDiaChi(txtDiaChi.getText());

model.setEmail(txtEmail.getText());

if(rdoNam.isSelected()){

model.setGioiTinh(1);

}

if(rdoNu.isSelected()){

model.setGioiTinh(2);

}

if(rdoKhac.isSelected()){

model.setGioiTinh(3);

}

if(rdoAdmin.isSelected()){

model.setVaiTro(true);

}

if(rdoNguoiDung.isSelected()){

model.setVaiTro(false);

}

model.setHinh(lblHinh.getToolTipText());

model.setHoTen(txtHoTen.getText());

model.setMatKhau(txtMatKhau.getText());

model.setNamSinh(dateHelper.toDate(txtNamSinh.getText()));

model.setSdt(txtSDT.getText());

model.setTaiKhoan(txtTaiKhoan.getText());

model.setTenHienThi(txtTenHienThi.getText());

return model;

}

public void setModel(NguoiDung entity){

maND= entity.getMSND();

txtTenHienThi.setText(entity.getTenHienThi());

txtDiaChi.setText(entity.getDiaChi());

txtEmail.setText(entity.getEmail());

txtHoTen.setText(entity.getHoTen());

txtMatKhau.setText(entity.getMatKhau());

txtNamSinh.setText(dateHelper.toString(entity.getNamSinh()));

txtSDT.setText(entity.getSdt());

txtTaiKhoan.setText(entity.getTaiKhoan());

if(entity.getGioiTinh()==1){

rdoNam.setSelected(true);

}

else if(entity.getGioiTinh() == 2){

rdoNu.setSelected(true);

}

else{

rdoKhac.setSelected(true);

}

if(entity.isVaiTro()){

rdoAdmin.setSelected(true);

}

else{

rdoNguoiDung.setSelected(true);

}

if(entity.getHinh()== null){

DialogHelper.alert(this, "Người dùng này chưa đặt ảnh đại diện");

}else{

lblHinh.setIcon(ShareHelper.readLogo(entity.getHinh(), lblHinh));

lblHinh.setToolTipText(entity.getHinh());

}

}

public void selectImage(){

if(fileChooser.showOpenDialog(this)== JFileChooser.APPROVE\_OPTION){

File file = fileChooser.getSelectedFile();

if(ShareHelper.saveLog(file)){

lblHinh.setIcon(ShareHelper.readLogo(file.getName(), lblHinh));

lblHinh.setToolTipText(file.getName());

}

}

}

public void edit(){

try {

//lay ma CD tu index

String maND = (String) tblNguoiDung.getValueAt(index, 0);

NguoiDung nd = nguoiDungDAO.findById(maND);

lblSoDem.setText((this.index+1)+ "/"+ nguoiDungDAO.countSize());

if(nd != null){

setStatus(false);

setModel(nd);

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu");

e.printStackTrace();

}

}

public boolean checkTrungTenTaiKhoan\_Email(JTextField txtTaiKhoan, JTextField email){

List<NguoiDung> nd = nguoiDungDAO.getTk\_Email(email.getText(), txtTaiKhoan.getText());

txtTaiKhoan.setBackground(white);

if(nd.isEmpty()){

txtTaiKhoan.setBackground(white);

email.setBackground(white);

return true;

}

else{

DialogHelper.alert(this, "Tài khoản hoặc email đã có trong dữ liệu!");

return false;

}

}

//hover cac bang ben trai

public void miceEntered(JPanel panel, JLabel lbl){

Font myFont = new Font("Tahoma", Font.BOLD, 14);

panel.setBackground(new Color(114,206,255));

lbl.setForeground(white);

lbl.setFont(myFont);

}

public void miceExited(JPanel panel, JLabel lbl){

Font myFont = new Font("Tahoma", Font.BOLD, 14);

panel.setBackground(new Color(199,229,245));

lbl.setForeground(new Color(59,204,253));

lbl.setFont(myFont);

}

public boolean check(){

boolean error = false;

if(checkHelper.checkNullText(txtTenHienThi)

&& checkHelper.checkNullText(txtHoTen)

&& checkHelper.checkNullText(txtTaiKhoan)

&& checkHelper.checkNullText(txtMatKhau)

&& checkHelper.checkNullText(txtSDT)

&& checkHelper.checkNullText(txtEmail)

&& checkHelper.checkNullText(txtDiaChi)

&& checkHelper.checkNullText(txtNamSinh)

&& checkHelper.checkRdo(rdoNam, rdoNu, rdoKhac)

&& checkHelper.checkRdo2(rdoAdmin,rdoNguoiDung)){

if(checkHelper.checkName(txtTenHienThi)

&& checkHelper.checkName(txtHoTen)

&& checkHelper.checkTaiKhoan(txtTaiKhoan)

&& checkHelper.checkPass2(txtMatKhau)

&& checkHelper.checkSDT(txtSDT)

&& checkHelper.checkEmail(txtEmail)

&& checkHelper.checkDiaChi(txtDiaChi)

&& checkHelper.checkDate(txtNamSinh))

{

error = true;

}

}

return error;

}

public boolean checkChinhMinh(JTextField txt){

TaiKhoan tk=tkDao.findById(txt.getText());

if (tk.getTenTaiKhoan().equals(USER.getTenTaiKhoan())) {

DialogHelper.alert(this, "Bạn không được xóa chính mình.");

return false;

} else {

return true;

}

}

//cac nút

public void insert(){

NguoiDung model = getModelThem();

try {

nguoiDungDAO.insert(model);

load();

clear();

tabs.setSelectedIndex(0);

DialogHelper.alert(this, "Thêm mới thành công");

} catch (Exception e) {

DialogHelper.alert(this, "Thêm mới thất bại");

e.printStackTrace();

}

}

public void update(){

NguoiDung model = getModel();

try {

nguoiDungDAO.update(model);

load();

DialogHelper.alert(this, "Cập nhật thành công!");

clear();

tabs.setSelectedIndex(0);

setStatus(true);

} catch (Exception e) {

DialogHelper.alert(this, "Cập nhật thất bại");

}

}

public void delete(){

if(DialogHelper.confirm(this, "Bạn có thực sự muốn xóa người dùng này chứ?")){

try {

if(checkChinhMinh(txtTaiKhoan)){

nguoiDungDAO.delete(maND);

load();

DialogHelper.alert(this, "Xóa thành công!");

clear();

tabs.setSelectedIndex(0);

setStatus(true);

}

} catch (Exception e) {

DialogHelper.alert(this, "Xóa thất bại");

}

}

}

public void find(){

try {

setStatus(false);

if(checkHelper.checkSDT(txtTim)){

load();

this.index=-1;

clear();

tabs.setSelectedIndex(0);

DialogHelper.alert(this, "Tìm thấy người dùng");

}

} catch (Exception e) {

DialogHelper.alert(this, "Tìm thất bại");

}

}

Người dùng – Người dùng

//hover cac bang ben trai

public void miceEntered(JPanel panel, JLabel lbl){

Font myFont = new Font("Tahoma", Font.BOLD, 14);

panel.setBackground(new Color(114,206,255));

lbl.setForeground(white);

lbl.setFont(myFont);

}

public void miceExited(JPanel panel, JLabel lbl){

Font myFont = new Font("Tahoma", Font.BOLD, 14);

panel.setBackground(new Color(199,229,245));

lbl.setForeground(new Color(59,204,253));

lbl.setFont(myFont);

}

public void CloseJframe(){

DialogHelper.alert(this, "Cám ơn đã sử dụng phần mềm! \n Chúc bạn một ngày tốt lành!!");

System.exit(0);

}

### **4.4.7 Form Tài khoản**



MÔ TẢ HOẠT ĐỘNG

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

public NguoiDungDAO ndDAO = new NguoiDungDAO();

public String taiKhoan = null;

public String maND = null;

/\*\*

\* Creates new form TaiKhoanJFrame

\*/

public TaiKhoanJFrame() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

public void init(){

setIconImage(ShareHelper.APP\_ICON);

this.setTitle("CHI TIÊU HÀNG NGÀY");

setLocationRelativeTo(null);

getHinh();

showInfo();

lblHide.setVisible(false);

DialogHelper.alert(this, "Không thể thay đổi tên tài khoản");

txtTaiKhoan.setEnabled(false);

}

public void CloseJframe(){

DialogHelper.alert(this, "Cám ơn đã sử dụng phần mềm! \n Chúc bạn một ngày tốt lành!!");

System.exit(0);

}

//hover cac bang ben trai

public void miceEntered(JPanel panel, JLabel lbl){

Font myFont = new Font("Tahoma", Font.BOLD, 14);

panel.setBackground(new Color(114,206,255));

lbl.setForeground(white);

lbl.setFont(myFont);

}

public void miceExited(JPanel panel, JLabel lbl){

Font myFont = new Font("Tahoma", Font.BOLD, 14);

panel.setBackground(new Color(199,229,245));

lbl.setForeground(new Color(59,204,253));

lbl.setFont(myFont);

}

public void getHinh(){

try {

taiKhoan = ShareHelper.USER.getTenTaiKhoan();

NguoiDung timTk = ndDAO.findByIdTK(taiKhoan);

if(timTk.getHinh()!=null){

lblHinh.setIcon(ShareHelper.readLogo(timTk.getHinh(), lblHinh));

lblHinh.setToolTipText(timTk.getHinh());

}

else{

DialogHelper.alert(this, "Tài khoản này chưa đặt ảnh đại diện");

lblHinh.setToolTipText("");

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn hình ảnh");

e.printStackTrace();

}

}

public void setModel(NguoiDung entity){

maND = entity.getMSND();

lblLoiChao.setText("Xin chào "+entity.getTenHienThi()+" !!!");

txtTenHienThi.setText(entity.getTenHienThi());

txtDiaChi.setText(entity.getDiaChi());

txtEmail.setText(entity.getEmail());

txtHoTen.setText(entity.getHoTen());

txtMatKhau.setText(entity.getMatKhau());

txtNamSinh.setText(dateHelper.toString(entity.getNamSinh()));

txtSDT.setText(entity.getSdt());

txtTaiKhoan.setText(entity.getTaiKhoan());

if(entity.getGioiTinh()==1){

rdoNam.setSelected(true);

}

else if(entity.getGioiTinh() == 2){

rdoNu.setSelected(true);

}

else{

rdoKhac.setSelected(true);

}

if(entity.getHinh() == null){

lblHinh.setText("");

DialogHelper.alert(this, "Người dùng này chưa đặt ảnh đại diện");

}else{

lblHinh.setIcon(ShareHelper.readLogo(entity.getHinh(), lblHinh));

lblHinh.setToolTipText(entity.getHinh());

}

if(entity.isVaiTro()){

rdoAdmin.setSelected(true);

}

else{

rdoNguoiDung.setSelected(true);

}

}

public NguoiDung getModel(){

NguoiDung nd = new NguoiDung();

nd.setDiaChi(txtDiaChi.getText());

nd.setEmail(txtEmail.getText());

if(rdoNam.isSelected()){

nd.setGioiTinh(1);

}

if(rdoNu.isSelected()){

nd.setGioiTinh(2);

}

if(rdoKhac.isSelected()){

nd.setGioiTinh(3);

}

nd.setHinh(lblHinh.getToolTipText());

nd.setHoTen(txtHoTen.getText());

nd.setMSND(maND);

nd.setMatKhau(String.valueOf(txtMatKhau.getPassword()));

nd.setNamSinh(dateHelper.toDate(txtNamSinh.getText()));

nd.setSdt(txtSDT.getText());

nd.setTaiKhoan(txtTaiKhoan.getText());

nd.setTenHienThi(txtTenHienThi.getText());

if(rdoAdmin.isSelected()){

nd.setVaiTro(true);

}

if(rdoNguoiDung.isSelected()){

nd.setVaiTro(false);

}

return nd;

}

public void showInfo(){

NguoiDung nd = ndDAO.findByIdTK(taiKhoan);

setModel(nd);

}

public void update(){

try {

NguoiDung nd = getModel();

ndDAO.update(nd);

DialogHelper.alert(this, "Cập nhật thành công");

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn");

}

}

public void selectImage(){

if(fileChooser.showOpenDialog(this)== JFileChooser.APPROVE\_OPTION){

File file = fileChooser.getSelectedFile();

if(ShareHelper.saveLog(file)){

lblHinh.setIcon(ShareHelper.readLogo(file.getName(), lblHinh));

lblHinh.setToolTipText(file.getName());

}

}

}

public boolean check(){

boolean error = false;

if(checkHelper.checkNullText(txtTenHienThi)

&& checkHelper.checkNullText(txtHoTen)

&& checkHelper.checkNullText(txtTaiKhoan)

&& checkHelper.checkNullPass(txtMatKhau)

&& checkHelper.checkNullText(txtSDT)

&& checkHelper.checkNullText(txtEmail)

&& checkHelper.checkNullText(txtDiaChi)

&& checkHelper.checkNullText(txtNamSinh)){

if(checkHelper.checkName(txtTenHienThi)

&& checkHelper.checkName(txtHoTen)

&& checkHelper.checkTaiKhoan(txtTaiKhoan)

&& checkHelper.checkPass(txtMatKhau)

&& checkHelper.checkSDT(txtSDT)

&& checkHelper.checkEmail(txtEmail)

&& checkHelper.checkDiaChi(txtDiaChi)

&& checkHelper.checkDate(txtNamSinh))

{

error = true;

}

}

return error;

}

### **4.4.8 Form Thu Chi**



MÔ TẢ HOẠT ĐỘNG

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

ChiDAO chiDao = new ChiDAO();

ThuDAO thuDao = new ThuDAO();

DanhMucThuDAO dmThuDao = new DanhMucThuDAO();

DanhMucChiDAO dmChiDao = new DanhMucChiDAO();

LoaiViDAO lviDao = new LoaiViDAO();

ViTienDAO vtDao = new ViTienDAO();

TaiKhoanDAO tkDao = new TaiKhoanDAO();

NguoiDungDAO ndDao = new NguoiDungDAO();

public String tenLoaiMoi=null;

public String msVI = null;

public int index = -1;

public int index2 = -1;

public String maThu= null;

public String maChi = null;

public int soBanDau = 0;

/\*\*

\* Creates new form ThuChiJFrame

\*/

public ThuChiJFrame() {

initComponents();

}

public void init(){

setIconImage(ShareHelper.APP\_ICON);

this.setTitle("CHI TIÊU HÀNG NGÀY");

setStatus(true);

}

public void setStatus(boolean status){

btnThem.setEnabled(status);

btnCapNhat.setEnabled(!status);

btnXoa.setEnabled(!status);

boolean first = this.index > 0;

boolean last = this.index < tblThu.getRowCount() - 1;

boolean first2 = this.index2 > 0;

boolean last2 = this.index2< tblChi.getRowCount() - 1;

btnFirst.setEnabled(!status && first);

btnPrev.setEnabled(!status && first);

btnLast.setEnabled(!status && last);

btnNext.setEnabled(!status && last);

btnFirst.setEnabled(!status && first2);

btnPrev.setEnabled(!status && first2);

btnLast.setEnabled(!status && last2);

btnNext.setEnabled(!status && last2);

}

public void load(){

try {

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

DefaultTableModel model2 = (DefaultTableModel) tblThu.getModel();

model.setRowCount(0);

model2.setRowCount(0);

List<Chi> list = chiDao.selectByKeyword(ShareHelper.USER.getTenTaiKhoan());

List<Thu> listThu = thuDao.selectByKeyword(ShareHelper.USER.getTenTaiKhoan());

// List<Chi> list = chiDao.selectByKeyword("babaraus123");

// List<Thu> listThu = thuDao.selectByKeyword("babaraus123");

for(Chi vt:list){

Object [] rows= {

vt.getMaChi(),

dateHelper.toString(vt.getNgayChi()),

vt.getTenHangMucChi(),

vt.getTenVi(),

vt.getSoTien()

};

model.addRow(rows);

}

for(Thu vt:listThu){

Object [] rows= {

vt.getMaThu(),

dateHelper.toString(vt.getNgayThu()),

vt.getTenHangMucThu(),

vt.getTenVi(),

vt.getSoTien()

};

model2.addRow(rows);

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu");

e.printStackTrace();

}

}

//fillCombobox

public void fillCombobox(){

try {

DefaultComboBoxModel modelThuChi = (DefaultComboBoxModel) cboThuChi.getModel();

modelThuChi.removeAllElements();

modelThuChi.addElement("Thu");

modelThuChi.addElement("Chi");

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu");

}

}

public void fillComboboxThu(){

try {

DefaultComboBoxModel model = (DefaultComboBoxModel) cboHangMuc.getModel();

model.removeAllElements();

List<DanhMucThu> list = dmThuDao.select();

for(DanhMucThu lvi : list){

model.addElement(lvi);

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu");

}

}

public void fillComboboxChi(){

try {

DefaultComboBoxModel model = (DefaultComboBoxModel) cboHangMuc.getModel();

model.removeAllElements();

List<DanhMucChi> list = dmChiDao.select();

for(DanhMucChi lvi : list){

model.addElement(lvi);

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu");

}

}

public void fillComboboxTenVi(){

try {

DefaultComboBoxModel model = (DefaultComboBoxModel) cboTenVi.getModel();

model.removeAllElements();

List<ViTien> list = vtDao.selectByKeyword(ShareHelper.USER.getTenTaiKhoan());

// List<ViTien> list = vtDao.selectByKeyword("babaraus123");

for(ViTien vt : list){

model.addElement(vt);

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu");

e.printStackTrace();

}

}

///

public void miceEntered(JPanel panel, JLabel lbl){

Font myFont = new Font("Tahoma", Font.BOLD, 14);

panel.setBackground(new Color(114,206,255));

lbl.setForeground(white);

lbl.setFont(myFont);

}

public void miceExited(JPanel panel, JLabel lbl){

Font myFont = new Font("Tahoma", Font.BOLD, 14);

panel.setBackground(new Color(199,229,245));

lbl.setForeground(new Color(59,204,253));

lbl.setFont(myFont);

}

public void CloseJframe(){

DialogHelper.alert(this, "Cám ơn đã sử dụng phần mềm! \n Chúc bạn một ngày tốt lành!!");

System.exit(0);

}

//check loi

public boolean check(){

boolean error = false;

if(checkHelper.checkNullText(txtSodu)

&& checkHelper.checkNullText(txtDate)){

if(checkHelper.checkSoDu(txtSodu) && checkHelper.checkSoAm(txtSodu)

&& checkHelper.checkDate(txtDate))

{

error = true;

}

}

return error;

}

///

public void btnThemCBO(){

try {

tenLoaiMoi= DialogHelper.prompt(this, "Vui lòng nhập hạng mục thêm mới?");

if(cboThuChi.getSelectedIndex()==0){

DanhMucThu dmThu = getModelDMThu();

dmThuDao.insert(dmThu);

fillComboboxThu();

DialogHelper.alert(this, "Thêm hạng mục mới thành công");

}

else{

DanhMucChi dmChi = getModelDMChi();

dmChiDao.insert(dmChi);

fillComboboxChi();

DialogHelper.alert(this, "Thêm hạng mục mới thành công");

}

} catch (Exception e) {

DialogHelper.alert(this, "Thêm hạng mục mới thất bại");

}

}

//

// //get set

public DanhMucChi getModelDMChi(){

DanhMucChi dmChi = new DanhMucChi();

if(dmChiDao.countSize()<9){

dmChi.setMsHangMucChi("HMC\_0"+String.valueOf(dmChiDao.countSize()+1));

}

else{

dmChi.setMsHangMucChi("HMC\_"+String.valueOf(dmChiDao.countSize()+1));

}

dmChi.setTenHangMuc(tenLoaiMoi);

return dmChi;

}

public DanhMucThu getModelDMThu(){

DanhMucThu dmThu = new DanhMucThu();

if(dmThuDao.countSize()<9){

dmThu.setMsHangMucThu("HMT\_0"+String.valueOf(dmThuDao.countSize()+1));

}

else{

dmThu.setMsHangMucThu("HTM\_"+String.valueOf(dmThuDao.countSize()+1));

}

dmThu.setTenHangMuc(tenLoaiMoi);

return dmThu;

}

public Thu getModelThemThu(){

Thu model = new Thu();

if(thuDao.countSize()<9){

model.setMaThu("LT\_0"+String.valueOf(thuDao.countSize()+1));

}

else{

model.setMaThu("LT\_"+String.valueOf(thuDao.countSize()+1));

}

DanhMucThu dThu = (DanhMucThu) cboHangMuc.getSelectedItem();

ViTien vt = (ViTien) cboTenVi.getSelectedItem();

model.setMsHangMucThu(dThu.getMsHangMucThu());

model.setNgayThu(dateHelper.toDate(txtDate.getText()));

model.setMSVI(vt.getMSVi());

// model.setTaiKhoan("babaraus123");

model.setTaiKhoan(ShareHelper.USER.getTenTaiKhoan());

model.setSoTien(Integer.parseInt(txtSodu.getText()));

return model;

}

public Chi getModelThemChi(){

Chi model = new Chi();

if(thuDao.countSize()<9){

model.setMaChi("MC\_0"+String.valueOf(chiDao.countSize()+1));

}

else{

model.setMaChi("MC\_"+String.valueOf(chiDao.countSize()+1));

}

DanhMucChi dChi = (DanhMucChi) cboHangMuc.getSelectedItem();

ViTien vt = (ViTien) cboTenVi.getSelectedItem();

model.setMsHangMucChi(dChi.getMsHangMucChi());

model.setNgayChi(dateHelper.toDate(txtDate.getText()));

model.setMSVI(vt.getMSVi());

// model.setTaiKhoan("babaraus123");

model.setTaiKhoan(ShareHelper.USER.getTenTaiKhoan());

model.setSoTien(Integer.parseInt(txtSodu.getText()));

return model;

}

public Thu getModelThu(){

Thu model = new Thu();

model.setMaThu(maThu);

DanhMucThu dThu = (DanhMucThu) cboHangMuc.getSelectedItem();

ViTien vt = (ViTien) cboTenVi.getSelectedItem();

model.setMsHangMucThu(dThu.getMsHangMucThu());

model.setNgayThu(dateHelper.toDate(txtDate.getText()));

model.setMSVI(vt.getMSVi());

// model.setTaiKhoan("babaraus123");

model.setTaiKhoan(ShareHelper.USER.getTenTaiKhoan());

model.setSoTien(Integer.parseInt(txtSodu.getText()));

return model;

}

//

public Chi getModelChi(){

Chi model = new Chi();

model.setMaChi(maChi);

DanhMucChi dChi = (DanhMucChi) cboHangMuc.getSelectedItem();

ViTien vt = (ViTien) cboTenVi.getSelectedItem();

model.setMsHangMucChi(dChi.getMsHangMucChi());

model.setNgayChi(dateHelper.toDate(txtDate.getText()));

model.setMSVI(vt.getMSVi());

// model.setTaiKhoan("babaraus123");

model.setTaiKhoan(ShareHelper.USER.getTenTaiKhoan());

model.setSoTien(Integer.parseInt(txtSodu.getText()));

return model;

}

public void setModelThu(Thu entity){

maThu = entity.getMaThu();

txtSodu.setText(String.valueOf(entity.getSoTien()));

txtDate.setText(dateHelper.toString(entity.getNgayThu()));

cboHangMuc.getModel().setSelectedItem(dmThuDao.findById(entity.getMsHangMucThu()));

cboTenVi.getModel().setSelectedItem(vtDao.findById(entity.getMSVI()));

soBanDau = Integer.parseInt(txtSodu.getText());

cboTenVi.setSelectedItem(entity.getTenVi());

}

public void setModelChi(Chi entity){

maChi = entity.getMaChi();

txtSodu.setText(String.valueOf(entity.getSoTien()));

txtDate.setText(dateHelper.toString(entity.getNgayChi()));

cboHangMuc.getModel().setSelectedItem(dmChiDao.findById(entity.getMsHangMucChi()));

cboTenVi.getModel().setSelectedItem(vtDao.findById(entity.getMSVI()));

soBanDau = Integer.parseInt(txtSodu.getText());

}

public void editThu(){

try {

//lay ma tu index

cboThuChi.setSelectedIndex(0);

String maThu = (String) tblThu.getValueAt(this.index, 0);

Thu vt1 = thuDao.findById(maThu);

if(vt1 != null){

setStatus(false);

setModelThu(vt1);

}

// lblSoDem.setText(this.index+1+"/"+thuDao.countSizeThu("babaraus123"));

lblSoDem.setText(this.index+1+"/"+thuDao.countSizeThu(ShareHelper.USER.getTenTaiKhoan()));

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu");

e.printStackTrace();

}

}

public void editChi(){

try {

//lay ma tu index

cboThuChi.setSelectedIndex(1);

String maChi = (String) tblChi.getValueAt(this.index2, 0);

Chi vt1 = chiDao.findById(maChi);

if(vt1 != null){

setModelChi(vt1);

setStatus(false);

}

// lblSoDem.setText(this.index2+1+"/"+chiDao.countSizeChi("babaraus123"));

lblSoDem.setText(this.index2+1+"/"+chiDao.countSizeChi(ShareHelper.USER.getTenTaiKhoan()));

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu");

e.printStackTrace();

}

}

public void tinhTienChi() {

ViTien vt = (ViTien) cboTenVi.getSelectedItem();

int tongTienVi = vt.getSoDu();

int tongTien = 0;

tongTien = tongTienVi - Integer.parseInt(txtSodu.getText());

vtDao.updateTien(tongTien, vt.getMSVi());

if(vt.getSoDu()<=0){

DialogHelper.alert(this, "Số dư trong ví đã âm \n Vui lòng thay đổi kế hoạch chi");

}

}

public void tinhTienThu() {

ViTien vt = (ViTien) cboTenVi.getSelectedItem();

int tongTienVi = vt.getSoDu();

int tongTien = 0;

tongTien = tongTienVi + Integer.parseInt(txtSodu.getText());

vtDao.updateTien(tongTien, vt.getMSVi());

}

public void tinhTienChi\_CapNhat() {

ViTien vt = (ViTien) cboTenVi.getSelectedItem();

int tongTienVi = vt.getSoDu();

int soDu = Integer.parseInt(txtSodu.getText());

int tongTien = 0;

if(soDu >soBanDau){

tongTien = tongTienVi -(soDu - soBanDau);

vtDao.updateTien(tongTien, vt.getMSVi());

}

else if(soDu < soBanDau){

tongTien = tongTienVi + (soBanDau- soDu);

vtDao.updateTien(tongTien, vt.getMSVi());

}

else{

}

if(vt.getSoDu()<=0){

DialogHelper.alert(this, "Số dư trong ví đã âm ");

}

}

public void tinhTienThu\_CapNhat() {

ViTien vt = (ViTien) cboTenVi.getSelectedItem();

int tongTienVi = vt.getSoDu();

int soDu = Integer.parseInt(txtSodu.getText());

int tongTien = 0;

if(soDu >soBanDau){

tongTien = tongTienVi +(soDu - soBanDau);

vtDao.updateTien(tongTien, vt.getMSVi());

}

else if(soDu < soBanDau){

tongTien = tongTienVi - (soBanDau- soDu);

vtDao.updateTien(tongTien, vt.getMSVi());

}

else{

}

if(vt.getSoDu()<=0){

DialogHelper.alert(this, "Số dư trong ví đã âm ");

}

}

public void tinhTienThu\_Xoa() {

ViTien vt = (ViTien) cboTenVi.getSelectedItem();

int tongTienVi = vt.getSoDu();

int tongTien = 0;

tongTien = tongTienVi - Integer.parseInt(txtSodu.getText());

vtDao.updateTien(tongTien, vt.getMSVi());

if(vt.getSoDu()<=0){

DialogHelper.alert(this, "Số dư trong ví đã âm \n Vui lòng thay đổi kế hoạch chi");

}

}

public void tinhTienChi\_Xoa() {

ViTien vt = (ViTien) cboTenVi.getSelectedItem();

int tongTienVi = vt.getSoDu();

int tongTien = 0;

tongTien = tongTienVi + Integer.parseInt(txtSodu.getText());

vtDao.updateTien(tongTien, vt.getMSVi());

}

//cac nút

public void insertThu(){

Thu model = getModelThemThu();

try {

ViTien vt = (ViTien) cboTenVi.getSelectedItem();

if(vt.isTinhTrang()){

thuDao.insert(model);

tinhTienThu();

load();

clear();

DialogHelper.alert(this, "Thêm mới thành công");

}

else{

DialogHelper.alert(this, "Ví bạn chọn bị trong tình trạng đóng nên không được sử dụng");

clear();

}

} catch (Exception e) {

DialogHelper.alert(this, "Thêm mới thất bại");

e.printStackTrace();

}

}

public void insertChi(){

Chi model = getModelThemChi();

try {

ViTien vt = (ViTien) cboTenVi.getSelectedItem();

if(vt.isTinhTrang()){

chiDao.insert(model);

tinhTienChi();

load();

clear();

DialogHelper.alert(this, "Thêm mới thành công");

}

else{

DialogHelper.alert(this, "Ví bạn chọn bị trong tình trạng đóng nên không được sử dụng");

clear();

}

} catch (Exception e) {

DialogHelper.alert(this, "Thêm mới thất bại");

e.printStackTrace();

}

}

public void updateThu(){

Thu model = getModelThu();

try {

ViTien vt = (ViTien) cboTenVi.getSelectedItem();

if(vt.isTinhTrang()){

thuDao.update(model);

tinhTienThu\_CapNhat();

load();

clear();

DialogHelper.alert(this, "Cập nhật thành công!");

tabs.setSelectedIndex(0);

}

else{

DialogHelper.alert(this, "Ví bạn chọn bị trong tình trạng đóng nên không được sử dụng");

clear();

}

} catch (Exception e) {

DialogHelper.alert(this, "Cập nhật thất bại");

}

}

public void updateChi(){

Chi model = getModelChi();

try {

ViTien vt = (ViTien) cboTenVi.getSelectedItem();

if(vt.isTinhTrang()){

chiDao.update(model);

tinhTienChi\_CapNhat();

load();

clear();

DialogHelper.alert(this, "Cập nhật thành công!");

tabs.setSelectedIndex(0);

}

else{

DialogHelper.alert(this, "Ví bạn chọn bị trong tình trạng đóng nên không được sử dụng");

clear();

}

} catch (Exception e) {

DialogHelper.alert(this, "Cập nhật thất bại");

}

}

public void deleteThu(){

if(DialogHelper.confirm(this, "Bạn có thực sự muốn xóa phiếu thu này chứ?")){

try {

String xoaMaThu = String.valueOf(tblThu.getValueAt(this.index, 0));

ViTien vt = (ViTien) cboTenVi.getSelectedItem();

if(vt.isTinhTrang()){

thuDao.delete(xoaMaThu);

tinhTienThu\_Xoa();

load();

DialogHelper.alert(this, "Xóa thành công!");

clear();

tabs.setSelectedIndex(0);

}

else{

DialogHelper.alert(this, "Ví bạn chọn bị trong tình trạng đóng nên không được sử dụng");

clear();

}

} catch (Exception e) {

DialogHelper.alert(this, "Xóa thất bại");

e.printStackTrace();

}

}

}

public void deleteChi(){

if(DialogHelper.confirm(this, "Bạn có thực sự muốn xóa phiếu chi này chứ?")){

try {

String xoaMaChi = String.valueOf(tblChi.getValueAt(this.index2, 0));

ViTien vt = (ViTien) cboTenVi.getSelectedItem();

if(vt.isTinhTrang()){

chiDao.delete(xoaMaChi);

tinhTienChi\_Xoa();

load();

DialogHelper.alert(this, "Xóa thành công!");

clear();

tabs.setSelectedIndex(0);

}

else{

DialogHelper.alert(this, "Ví bạn chọn bị trong tình trạng đóng nên không được sử dụng");

clear();

}

} catch (Exception e) {

DialogHelper.alert(this, "Xóa thất bại");

e.printStackTrace();

}

}

}

public void clear(){

cboThuChi.setSelectedIndex(0);

cboHangMuc.setSelectedIndex(0);

cboTenVi.setSelectedIndex(0);

txtSodu.setText("");

txtDate.setText("");

maChi= null;

maThu=null;

}

### **4.4.9 Form Tổng quan**



MÔ TẢ HOẠT ĐỘNG

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

public class TongQuanJFrame extends javax.swing.JFrame {

public ThongKeDAO thongkeDao = new ThongKeDAO();

public TaiKhoanDAO tkDao = new TaiKhoanDAO();

public ViTienDAO vtDao = new ViTienDAO();

public NguoiDungDAO ndDao = new NguoiDungDAO();

/\*\*

\* Creates new form TongQuanJFrame

\*/

public TongQuanJFrame() {

initComponents();

}

public void init(){

setIconImage(ShareHelper.APP\_ICON);

this.setTitle("CHI TIÊU HÀNG NGÀY");

taiChinh();

load();

}

public void load(){

try {

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

model.setRowCount(0);

List<ViTien> list = vtDao.selectByKeyword(ShareHelper.USER.getTenTaiKhoan());

if(list.isEmpty()){

DialogHelper.alert(this, "Tài khoản này chưa tạo ví tiền nào!!");

}

for(ViTien vt:list){

Object [] rows= {

vt.getTenVi(),

vt.getSoDu()+" VNĐ",

};

model.addRow(rows);

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu");

e.printStackTrace();

}

}

public void taiChinh(){

getTaiChinh();

getTaiChinh\_Chi\_Nam();

getTaiChinh\_Chi\_Ngay();

getTaiChinh\_Chi\_Thang();

getTaiChinh\_Thu\_Nam();

getTaiChinh\_Thu\_Ngay();

getTaiChinh\_Thu\_Thang();

}

public void CloseJframe(){

DialogHelper.alert(this, "Cám ơn đã sử dụng phần mềm! \n Chúc bạn một ngày tốt lành!!");

System.exit(0);

}

//hover cac bang ben trai

public void miceEntered(JPanel panel, JLabel lbl){

Font myFont = new Font("Tahoma", Font.BOLD, 14);

panel.setBackground(new Color(114,206,255));

lbl.setForeground(white);

lbl.setFont(myFont);

}

public void miceExited(JPanel panel, JLabel lbl){

Font myFont = new Font("Tahoma", Font.BOLD, 14);

panel.setBackground(new Color(199,229,245));

lbl.setForeground(new Color(59,204,253));

lbl.setFont(myFont);

}

//panel lam viec

public void getTaiChinh(){

List<String> taichinh = thongkeDao.getTaiChinh(ShareHelper.USER.getTenTaiKhoan());

try {

String TongTien = null;

if(taichinh.isEmpty()){

DialogHelper.alert(this, "Tài khoản này không có dữ liệu tài chính hiện tại!!");

}

else{

for(String tc : taichinh){

TongTien = tc;

}

txtTongTien.setForeground(new Color(93, 255, 51));

txtTongTien.setText(TongTien+ " VNĐ");

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn");

e.printStackTrace();

}

}

public void getTaiChinh\_Thu\_Nam(){

try {

List<Integer> taichinh = thongkeDao.getTongThu\_NamNay(ShareHelper.USER.getTenTaiKhoan());

int TongTien = 0;

if(taichinh.isEmpty()){

DialogHelper.alert(this, "Tài khoản này chưa có khoản thu nào!!");

}

else{

for(Integer tc : taichinh){

TongTien = tc.intValue();

}

if(TongTien<0){

txtThu\_Nam2.setForeground(new Color(204,0,51));

}

else{

txtThu\_Nam2.setForeground(new Color(93, 255, 51));

}

txtThu\_Nam2.setText(TongTien+ " VNĐ");

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn");

e.printStackTrace();

}

}

public void getTaiChinh\_Thu\_Thang(){

try {

List<Integer> taichinh = thongkeDao.getTongThu\_ThangNay(ShareHelper.USER.getTenTaiKhoan());

int TongTien = 0;

if(taichinh.isEmpty()){

DialogHelper.alert(this, "Tài khoản này chưa có khoản thu nào!!");

}

else{

for(Integer tc : taichinh){

TongTien = tc.intValue();

}

if(TongTien<0){

txtThu\_Thang2.setForeground(new Color(204,0,51));

}

else{

txtThu\_Thang2.setForeground(new Color(93, 255, 51));

}

txtThu\_Thang2.setText(TongTien+ " VNĐ");

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn");

e.printStackTrace();

}

}

public void getTaiChinh\_Thu\_Ngay(){

try {

List<Integer> taichinh = thongkeDao.getTongThu\_NgayNay(ShareHelper.USER.getTenTaiKhoan());

int TongTien = 0;

if(taichinh.isEmpty()){

DialogHelper.alert(this, "Tài khoản này chưa có khoản thu nào!!");

}

else{

for(Integer tc : taichinh){

TongTien = tc.intValue();

}

if(TongTien<0){

txtThu\_Nay2.setForeground(new Color(204,0,51));

}

else{

txtThu\_Nay2.setForeground(new Color(93, 255, 51));

}

txtThu\_Nay2.setText(TongTien+ " VNĐ");

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn");

e.printStackTrace();

}

}

//Tai chinh Chi

public void getTaiChinh\_Chi\_Nam(){

try {

List<Integer> taichinh = thongkeDao.getTongChi\_NamNay(ShareHelper.USER.getTenTaiKhoan());

int TongTien = 0;

if(taichinh.isEmpty()){

DialogHelper.alert(this, "Tài khoản này chưa có khoản chi nào!!");

}

else{

for(Integer tc : taichinh){

TongTien = tc.intValue();

}

if(TongTien<0){

txtChi\_Nam2.setForeground(new Color(204,0,51));

}

else{

txtChi\_Nam2.setForeground(new Color(93, 255, 51));

}

txtChi\_Nam2.setText(TongTien+ " VNĐ");

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn");

e.printStackTrace();

}

}

public void getTaiChinh\_Chi\_Thang(){

try {

List<Integer> taichinh = thongkeDao.getTongChi\_ThangNay(ShareHelper.USER.getTenTaiKhoan());

int TongTien = 0;

if(taichinh.isEmpty()){

DialogHelper.alert(this, "Tài khoản này chưa có khoản chi nào!!");

}

else{

for(Integer tc : taichinh){

TongTien = tc.intValue();

}

if(TongTien<0){

txtChi\_Thang2.setForeground(new Color(204,0,51));

}

else{

txtChi\_Thang2.setForeground(new Color(93, 255, 51));

}

txtChi\_Thang2.setText(TongTien+ " VNĐ");

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn");

e.printStackTrace();

}

}

public void getTaiChinh\_Chi\_Ngay(){

try {

List<Integer> taichinh = thongkeDao.getTongChi\_NgayNay(ShareHelper.USER.getTenTaiKhoan());

int TongTien = 0;

if(taichinh.isEmpty()){

DialogHelper.alert(this, "Tài khoản này chưa có khoản chi nào!!");

}

else{

for(Integer tc : taichinh){

TongTien = tc.intValue();

}

if(TongTien<0){

txtChi\_Nay2.setForeground(new Color(204,0,51));

}

else{

txtChi\_Nay2.setForeground(new Color(93, 255, 51));

}

txtChi\_Nay2.setText(TongTien+ " VNĐ");

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn");

e.printStackTrace();

}

}

### **4.4.10 Form Ví tiền**



MÔ TẢ HOẠT ĐỘNG

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

public int index = -1;

public ViTienDAO vtDao = new ViTienDAO();

public LoaiViDAO lviDao = new LoaiViDAO();

public String tenLoaiMoi=null;

public ThuDAO thuDao = new ThuDAO();

public NguoiDungDAO ndDao = new NguoiDungDAO();

/\*\*

\* Creates new form ViTienJFrame\_2

\*/

public ViTienJFrame\_2() {

initComponents();

init();

}

public void init(){

setIconImage(ShareHelper.APP\_ICON);

this.setTitle("CHI TIÊU HÀNG NGÀY");

setStatus(true);

}

public void CloseJframe(){

DialogHelper.alert(this, "Cám ơn đã sử dụng phần mềm! \n Chúc bạn một ngày tốt lành!!");

System.exit(0);

}

public boolean check(){

boolean error = false;

if(checkHelper.checkNullText(txtTenVi)&&checkHelper.checkNullText(txtSodu)

&& checkHelper.checkRdo2(rdoMo, rdoDong)){

if(checkHelper.checkName(txtTenVi) && checkHelper.checkSoDu(txtSodu)

&& checkHelper.checkSoAm(txtSodu)){

error = true;

}

}

return error;

}

public void setStatus(boolean status){

btnThem.setEnabled(status);

btnCapNhat.setEnabled(!status);

rdoDong.setEnabled(!status);

boolean first = this.index > 0;

boolean last = this.index < tblViTien.getRowCount() - 1;

btnFirst.setEnabled(!status && first);

btnPrev.setEnabled(!status && first);

btnLast.setEnabled(!status && last);

btnNext.setEnabled(!status && last);

}

public void load(){

try {

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

model.setRowCount(0);

List<ViTien> list = vtDao.selectByKeyword(ShareHelper.USER.getTenTaiKhoan());

for(ViTien vt:list){

Object [] rows= {

vt.getMSVi(),

vt.getHinh(),

vt.getTenVi(),

vt.getTenLoai(),

vt.getSoDu(),

vt.toStringTinhTrang()

};

model.addRow(rows);

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu");

e.printStackTrace();

}

}

public void fillCombobox(){

try {

DefaultComboBoxModel model = (DefaultComboBoxModel) cboLoaiVi.getModel();

model.removeAllElements();

List<LoaiVi> list = lviDao.select();

for(LoaiVi lvi : list){

model.addElement(lvi);

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu");

}

}

public LoaiVi getModelLV(){

LoaiVi loaiVi = new LoaiVi();

if(lviDao.countSize()<9){

loaiVi.setMsLoaiVi("LV\_0"+String.valueOf(lviDao.countSize()+1));

}

else{

loaiVi.setMsLoaiVi("LV\_"+String.valueOf(lviDao.countSize()+1));

}

loaiVi.setTenLoaiVi(tenLoaiMoi);

return loaiVi;

}

public ViTien getModelThem(){

ViTien model = new ViTien();

if(vtDao.countSize2()<9){

model.setMSVi("VI\_0"+String.valueOf(vtDao.countSize2()+1));

}

else{

model.setMSVi("VI\_"+String.valueOf(vtDao.countSize2()+1));

}

model.setTenVi(txtTenVi.getText());

model.setHinh(lblHinh.getToolTipText());

LoaiVi lvi = (LoaiVi) cboLoaiVi.getSelectedItem();

model.setMsLoaiVi(lvi.getMsLoaiVi());

model.setTenLoai(lvi.getTenLoaiVi());

model.setSoDu(Integer.parseInt(txtSodu.getText()));

model.setTinhTrang(rdoMo.isSelected()?true:false);

model.setTaiKhoan(ShareHelper.USER.getTenTaiKhoan());

return model;

}

public ViTien getModel(){

ViTien model = new ViTien();

model.setMSVi(lblMaSo.getText());

model.setTenVi(txtTenVi.getText());

model.setHinh(lblHinh.getToolTipText());

LoaiVi lvi = (LoaiVi) cboLoaiVi.getSelectedItem();

model.setMsLoaiVi(lvi.getMsLoaiVi());

model.setTenLoai(lvi.getTenLoaiVi());

model.setSoDu(Integer.parseInt(txtSodu.getText()));

model.setTinhTrang(rdoMo.isSelected()?true:false);

model.setTaiKhoan(ShareHelper.USER.getTenTaiKhoan());

return model;

}

public void setModel(ViTien entity){

lblMaSo.setText(entity.getMSVi());

txtTenVi.setText(entity.getTenVi());

cboLoaiVi.setSelectedItem(entity.getTenLoai());

txtSodu.setText(String.valueOf(entity.getSoDu()));

if(entity.getHinh() == null){

DialogHelper.alert(this, "Người dùng này chưa đặt ảnh ví");

}else{

lblHinh.setIcon(ShareHelper.readLogo(entity.getHinh(), lblHinh));

lblHinh.setToolTipText(entity.getHinh());

}

if(entity.isTinhTrang()==true){

rdoMo.setSelected(true);

}

else{

rdoDong.setSelected(true);

}

}

public void selectImage(){

if(fileChooser.showOpenDialog(this)== JFileChooser.APPROVE\_OPTION){

File file = fileChooser.getSelectedFile();

if(ShareHelper.saveLog(file)){

lblHinh.setIcon(ShareHelper.readLogo(file.getName(), lblHinh));

lblHinh.setToolTipText(file.getName());

}

}

}

public void edit(){

try {

//lay ma CD tu index

String maVi = (String) tblViTien.getValueAt(index, 0);

ViTien vt1 = vtDao.findById(maVi);

if(vt1 !=null){

setStatus(false);

setModel(vt1);

}

lblSoDem.setText(this.index+1+"/"+vtDao.countSize(ShareHelper.USER.getTenTaiKhoan()));

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu");

e.printStackTrace();

}

}

public void btnThemCBO(){

try {

tenLoaiMoi= DialogHelper.prompt(this, "Vui lòng nhập loại ví muốn thêm mới?");

LoaiVi lvi = getModelLV();

lviDao.insert(lvi);

fillCombobox();

DialogHelper.alert(this, "Thêm loại ví mới thành công");

} catch (Exception e) {

DialogHelper.alert(this, "Thêm loại ví mới thất bại");

}

}

public boolean checkTrungTenVi(JTextField txtTaiKhoan){

ViTien vt = vtDao.findById(txtTaiKhoan.getText());

txtTaiKhoan.setBackground(white);

if(vt == null){

txtTaiKhoan.setBackground(white);

return true;

}

else{

txtTaiKhoan.setBackground(Color.red);

txtTaiKhoan.requestFocus();

DialogHelper.alert(this, "Tài khoản "+ txtTaiKhoan.getText()+" đã có trong dữ liệu!");

return false;

}

}

//hover cac bang ben trai

public void miceEntered(JPanel panel, JLabel lbl){

Font myFont = new Font("Tahoma", Font.BOLD, 14);

panel.setBackground(new Color(114,206,255));

lbl.setForeground(white);

lbl.setFont(myFont);

}

public void miceExited(JPanel panel, JLabel lbl){

Font myFont = new Font("Tahoma", Font.BOLD, 14);

panel.setBackground(new Color(199,229,245));

lbl.setForeground(new Color(59,204,253));

lbl.setFont(myFont);

}

//cac nút

public void insert(){

ViTien model = getModelThem();

try {

vtDao.insert(model);

load();

clear();

DialogHelper.alert(this, "Thêm mới thành công");

} catch (Exception e) {

DialogHelper.alert(this, "Thêm mới thất bại");

e.printStackTrace();

}

}

public void update(){

ViTien model = getModel();

try {

vtDao.update(model);

load();

clear();

DialogHelper.alert(this, "Cập nhật thành công!");

tabs.setSelectedIndex(0);

} catch (Exception e) {

DialogHelper.alert(this, "Cập nhật thất bại");

}

}

public void clear(){

lblMaSo.setText("");

txtTenVi.setText("");

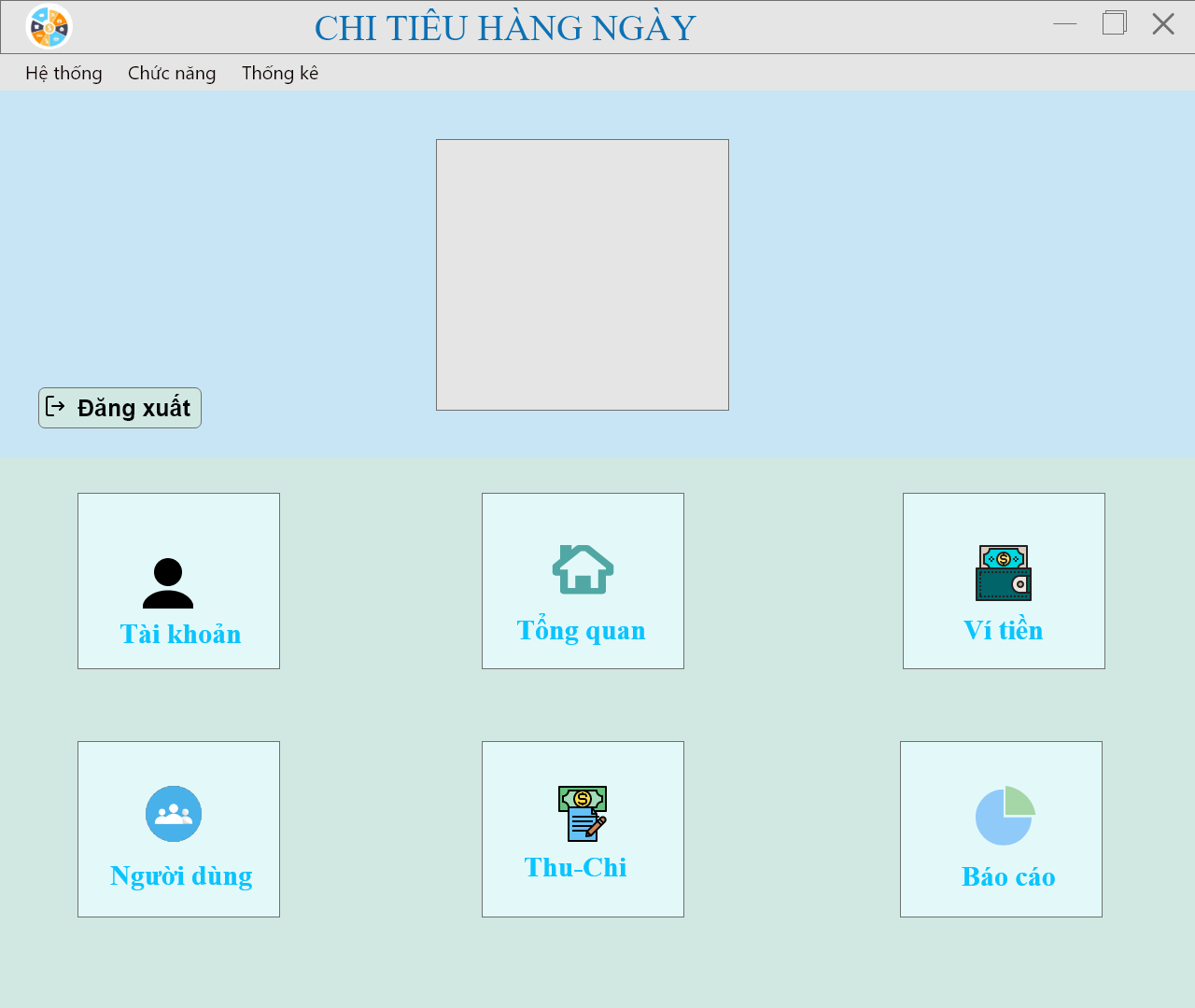
cboLoaiVi.setSelectedIndex(0);

txtSodu.setText("");

grpTT.clearSelection();

}

### **4.4.11 Form giao diện chính**



MÔ TẢ HOẠT ĐỘNG

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

public void init(){

setIconImage(ShareHelper.APP\_ICON);

this.setTitle("CHI TIÊU HÀNG NGÀY");

setLocationRelativeTo(this);

getHinh2();

}

public void CloseJframe(){

if(DialogHelper.confirm(this, "Bạn có muốn thoát khỏi ứng dụng không?")){

System.exit(0);

}

}

public void MinimizeJframe(){

setExtendedState(ICONIFIED);

}

public void MaximaizeJframe(){

if(this.getExtendedState() != NguoiDungJFrame.MAXIMIZED\_BOTH){

this.setExtendedState(NguoiDungJFrame.MAXIMIZED\_BOTH);

}

else{

this.setExtendedState(NguoiDungJFrame.NORMAL);

}

}

public void getHinh2(){

try {

String taiKhoan = ShareHelper.USER.getTenTaiKhoan();

NguoiDung timTk = ndDAO.findByIdTK(taiKhoan);

if(timTk.getHinh()==null){

DialogHelper.alert(this, "Tài khoản này chưa đặt ảnh đại diện");

}

else{

lblHinhDaiDien.setIcon(ShareHelper.readLogo(timTk.getHinh(), lblHinhDaiDien));

lblHinhDaiDien.setToolTipText(timTk.getHinh());

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn hình ảnh");

e.printStackTrace();

}

}

public void miceEntered(JPanel panel, JLabel lbl){

Font myFont = new Font("Tahoma", Font.BOLD, 14);

panel.setBackground(new Color(114,206,255));

lbl.setForeground(white);

lbl.setFont(myFont);

}

public void miceExited(JPanel panel, JLabel lbl){

Font myFont = new Font("Tahoma", Font.BOLD, 14);

panel.setBackground(new Color(227,249,249));

lbl.setForeground(new Color(59,204,253));

lbl.setFont(myFont);

}

public void exit(){

if(DialogHelper.confirm(this, "Bạn có muốn thoát khỏi ứng dụng không?")){

System.exit(0);

}

}

public void logoff(){

ShareHelper.logoff();

this.dispose();

new Login().setVisible(true);

}

public void tongQuan(){

this.dispose();

new TongQuanJFrame().setVisible(true);

}

public void taiKhoan(){

this.dispose();

new TaiKhoanJFrame().setVisible(true);

}

public void viTien(){

this.dispose();

new ViTienJFrame\_2().setVisible(true);

}

public void baoCao(){

this.dispose();

new BaoCaoJFrame(0).setVisible(true);

}

public void nguoiDung(){

this.dispose();

try {

NguoiDung nd = ndDAO.findByIdTK(ShareHelper.USER.getTenTaiKhoan());

if(nd.isVaiTro()){

new NguoiDungJFrame().setVisible(true);

}

else{

new NguoiDungJFrame\_2().setVisible(true);

}

} catch (Exception e) {

e.printStackTrace();

}

}

public void thuChi(){

this.dispose();

new ThuChiJFrame().setVisible(true);

}

public void openThongKe(int index) {

try {

if (ShareHelper.authenticated()) {

new BaoCaoJFrame(index).setVisible(true);

} else {

DialogHelper.alert(this, "Vui lòng đăng nhập!");

}

} catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu");

e.printStackTrace();

}

}

**5. Kiểm thử**

Do quá trình kiểm thử rất nhiều nên em đã tách ra 1 file riêng (test case).

**6. Đóng gói và triển khai**

**6.1 HƯỚNG DẪN CHUYỂN ĐỔI JAR THÀNH EXE**

* Sử dụng phần mềm EXE4J để chuyển đổi jar sang exe.
* Sử dụng phần mềm InnoSetup để đóng gói.

**6.2 HƯỚNG DẪN CÀI ĐẶT TRIỂN KHAI**

* Cài SQL Server 2018 trở lên.
* Tạo CSDL QuanLyChiTieu bằng cách chạy file db.sql.
* Cài đặt JDK 1.8 trở lên.
* Chạy file ChiTieu setup.

**6.3 HƯỚNG DẪN SỬ DỤNG PHẦN MỀM**

* Đăng nhập với tài khoản admin, username: admin và password:admin333.
* Cấp các tài khoản có vai trò là admin.
* Đăng nhập với tài khoản người dùng, username: babaraus123 và password: 123456789.
* Xem hướng dẫn sử dụng phần mềm theo đường link này: https://youtu.be/WK40gDUs7aU