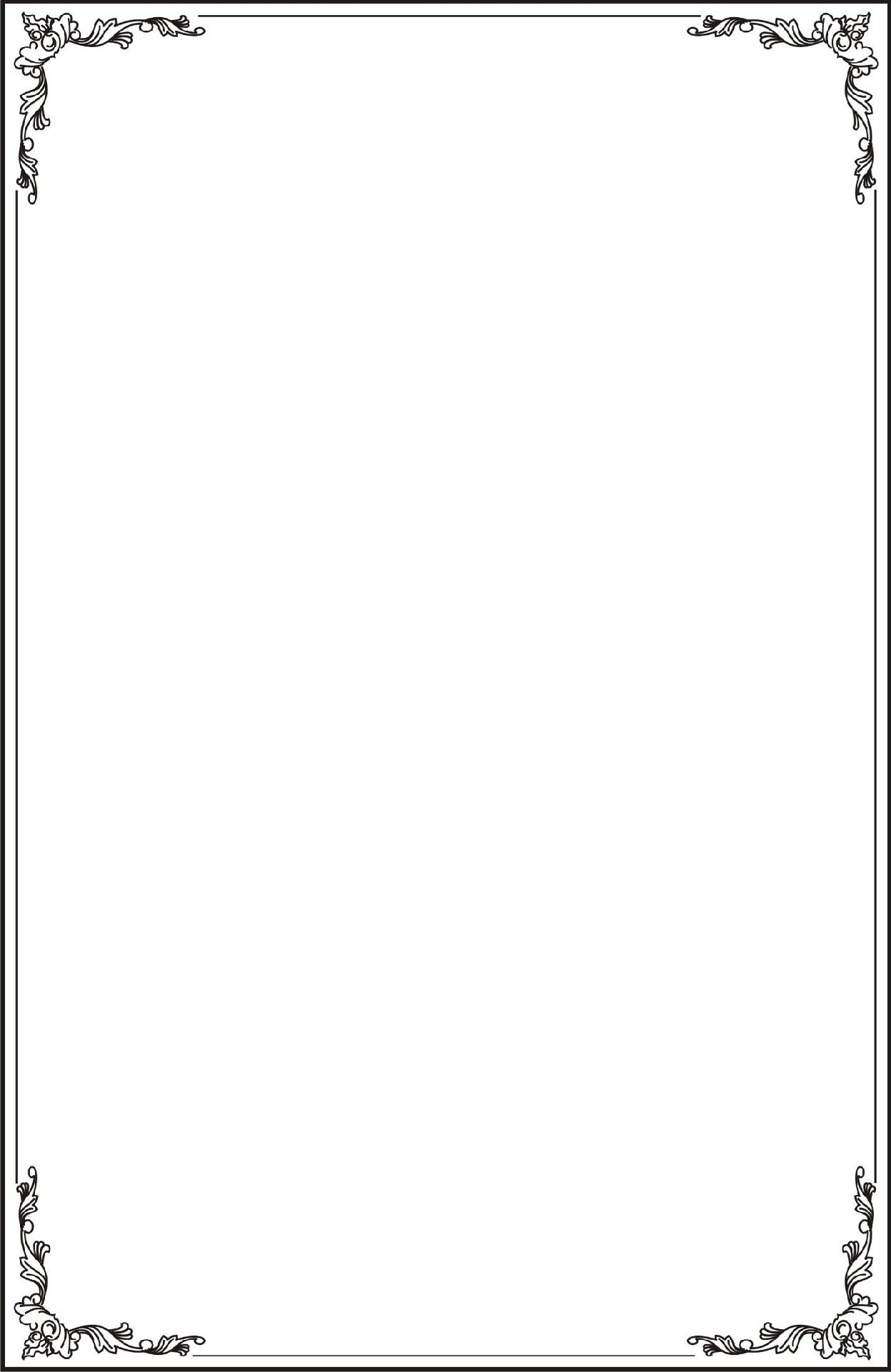
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



**KHOA CÔNG NGHỆ THÔNG TIN**

**BÀI TẬP LỚN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**Đề tài**: Xây dựng ứng dụng quản lý quán cà phê

**LỚP 01 – NHÓM 13**

|  |  |
| --- | --- |
| **Tên thành viên** | **Mã số sinh viên** |
| Lê Duy Quyết | B21DCCN642 |
| Lê Ngọc Thảo Vân | B21DCCN786 |
| Phạm Văn Tiến | B21DCCN708 |
| Cao Hồng Đức | B21DCCN234 |
| Nguyễn Thái Minh | B21DCCN090 |

Hà Nội, 2023

**MỤC LỤC**

[I. Giới thiệu 2](#_Toc151617222)

[A. Phát biểu bài toán và các chức năng dự định phát triển 2](#_Toc151617223)

[1. Bài toán 2](#_Toc151617224)

[2. Các chức năng ứng dụng 2](#_Toc151617225)

[B. Kỹ thuật, công nghệ sử dụng 3](#_Toc151617226)

[1. Phương pháp 3](#_Toc151617227)

[2. Ngôn ngữ sử dụng 3](#_Toc151617228)

[3. Công cụ 3](#_Toc151617229)

[II. Nội dung chính 4](#_Toc151617230)

[A. Xây dựng cơ sở dữ liệu 4](#_Toc151617231)

[B. Thiết kế các package của ứng dụng 5](#_Toc151617232)

[1. Package Util 5](#_Toc151617233)

[2. Package Obj 5](#_Toc151617234)

[3. Package DAO và DAO.impl 6](#_Toc151617235)

[4. Package GUI 19](#_Toc151617236)

[5. Package team13.quanlyquancaphe 60](#_Toc151617237)

[III. Tài liệu tham khảo 60](#_Toc151617238)

# Giới thiệu

## Phát biểu bài toán và các chức năng dự định phát triển

### Bài toán

Xây dựng một ứng dụng quản lý quán cà phê nhằm tối ưu hóa quá trình quản lý, giúp chủ quán và nhân viên thực hiện công việc một cách hiệu quả và thuận lợi. Ứng dụng này sẽ cung cấp các chức năng cơ bản để quản lý các hoạt động hàng ngày của quán cà phê.

### Các chức năng ứng dụng

***Hệ thống đăng nhập***

* Cho phép nhân viên và quản lý đăng nhập vào hệ thống

***Trang chủ***

***Danh mục***

* Quản lý danh sách danh mục trong quán cà phê
* Cho phép thêm mới, sửa đổi, xóa, tìm kiếm các danh mục đối với quản lý, nhân viên chỉ có thể tìm kiếm.

***Sản phẩm***

* Quản lý thông tin chi tiết về sản phẩm: tên sản phẩm, thuộc danh mục nào, giá và trạng thái.
* Cho phép thêm mới, sửa đổi, xóa, tìm kiếm sản phẩm đối với quản lý, nhân viên chỉ có thể tìm kiếm.

***Khu vực & Bàn ăn***

* Cho phép quản lý các khu vực và bàn ăn trong quán cà phê đối với quản lý.
* Cho phép theo dõi trạng thái và đặt bàn đối với quản lý và nhân viên.

***Hóa đơn***

* Quản lý hóa đơn bán hàng
* Cho phép xem, xóa, tìm kiếm và xuất hóa đơn đối với quản lý, nhân viên không có chức năng xoá.

***Người dùng***

* Cho phép quản lý thông tin về người dùng đối với quản lý

## Kỹ thuật, công nghệ sử dụng

### Phương pháp

* Sử dụng các kiến thức về lập trình hướng đối tượng (OOP)
* Sử dụng Java Design Pattern: DAO Pattern

Ý tưởng là thay vì có logic giao tiếp trực tiếp với cơ sở dữ liệu, hệ thống file, dịch vụ web hoặc bất kỳ cơ chế lưu trữ nào mà ứng dụng cần sử dụng, chúng ta sẽ để logic này sẽ giao tiếp qua lớp trung gian DAO. Lớp DAO này sau đó giao tiếp với hệ thống lưu trữ, hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu (tìm kiếm, thêm, xóa, sửa,…).



### Ngôn ngữ sử dụng

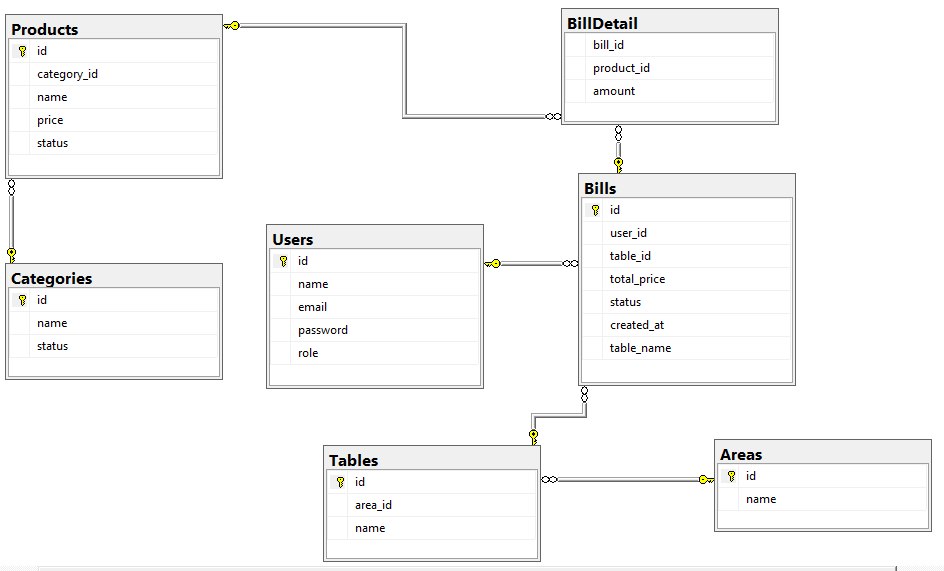
* Java và SQL

### Công cụ

* Apache Netbeans IDE.
* SQL Server và SQL Server Management Studio Management Studio (SSMS).
* Thư viện Java AWT, Swing và API JDBC.

# Nội dung chính

## Xây dựng cơ sở dữ liệu



* Products bao gồm: id (khóa chính), category\_id (khóa ngoại tham chiếu đến bảng Categories), name, price và status.
* Categories bao gồm: id (khóa chính), name và status.
* BillDetail bao gồm: bill\_id (khóa chính), product\_id (khóa ngoại tham chiếu đến bảng Products) và amount.
* Bills bao gồm: id (khóa chính), user\_id(khóa ngoại tham chiếu đến bảng Users), table\_id(khóa ngoại tham chiếu đến bảng Tables), total\_price, status. creat\_at và table\_name.
* Users bao gồm: id (khóa chính), name, email, password và role.
* Tables bao gồm: id (khóa chính), area\_id((khóa ngoại tham chiếu đến bảng Areas) và name.
* Areas bao gồm: id (khóa chính) và name.

## Thiết kế các package của ứng dụng

### Package Util

* *Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

  Mô tả được tạo tự độngCommon*: Class cung cấp 2 phương thức: Kiểm tra 1 đối tượng (String, List, Set hoặc Map) có null hoặc rỗng không và kiểm tra một chuỗi có thể chuyển thành số nguyên không.
* *DbUtil*: Thiết lập kết nối đến cơ sở dữ liệu SQL Server trong Java sử dụng JDBC, cung cấp phương thức getConnection() trả về một đối tượng Connection đến cơ sở dữ liệu.
* *WrapLayout*: Là 1 lớp kế thừa của FlowLayout trong Java Swing được thiết kế để hỗ trợ việc xếp các thành phần (components) trong GUI theo chiều ngang và hỗ trợ xuống dòng khi không đủ không gian để đặt tất cả thành phần trong 1 dòng duy nhất.

### Package Obj

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự độngĐịnh nghĩa các đối tượng trong ứng dụng và thuộc tính, phương thức tương ứng:

* *Area*: có 2 thuộc tính mã khu vực và tên khu vực, có hàm tạo, các phương thức get/ set, toString.
* *Bill*: có các thuộc tính: mã hoá đơn, mã người dùng, mã bàn, tổng tiền hoá đơn, trạng thái thanh toán, thời gian tạo, tên người dùng và tên bàn; có hàm tạo, các phương thức get/ set.
* *BillDetail*: có các thuộc tính: mã hoá đơn, mã hàng, tên hàng, số lượng, đơn giá của mỗi loại hàng; có hàm tạo, các phương thức get/ set.
* *Category*: có các thuộc tính: mã danh mục, tên danh mục và trạng thái hoạt động; có hàm tạo, các phương thức get/ set, toString.
* *Product*: có các thuộc tính: mã hàng hoá, tên hàng, đơn giá, trạng thái hoạt động, mã danh mục và tên danh mục; có hàm tạo, các phương thức get/ set.
* *Table*: có các thuộc tính: mã bàn, mã khu vực và tên bàn; có hàm tạo, các phương thức get/ set, toString.
* *User*: có các thuộc tính: mã, tên, email, mật khẩu và vai trò (Admin/ Nhân viên) của người dùng; có hàm tạo, các phương thức get/ set.

### Package DAO và DAO.impl

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

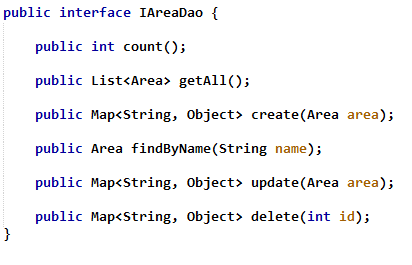
Mô tả được tạo tự độngPackage DAO gồm các interfaces để định nghĩa các phương thức trừu tượng triển khai truy cập dữ liệu cho từng đối tượng trong ứng dụng.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

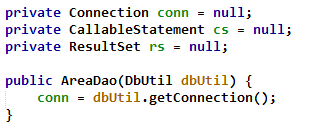
Mô tả được tạo tự độngPackage DAO.impl gồm các lớp triển khai chi tiết các phương thức của từng interface được định nghĩa trong DAO, các lớp này sẽ thao tác trực tiếp với cơ sở dữ liệu.

#### Interface IAreaDao và Class AreaDao

Tương tác với cơ sở dữ liệu về khu vực.



Kết nối AreaDao với CSDL



Các phương thức:

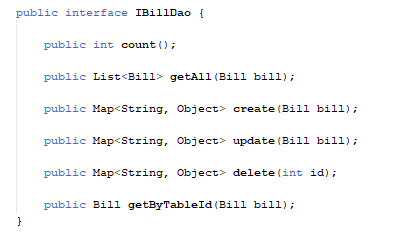
Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

* Hàm count(): Đếm số khu vực
* Hàm getAll(): lấy dữ liệu của tất cả các khu vực trong database
* Hàm create(): Tạo mới khu vực có các thuộc tính tương đương với đối tượng area truyền vào.
* Hàm findByName(): Tìm khu vực bằng tên (người dùng nhập tên)
* Hàm update(): Cập nhật khu vực thành các thuộc tính tương ứng với đối tượng area truyền vào.
* Hàm delete(): Xóa khu vực dựa trên mã khu vực.

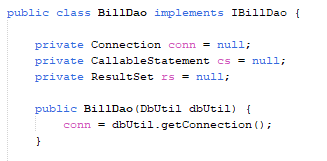
#### Interface IBillDao và Class BillDao

Tương tác với cơ sở dữ liệu về hoá đơn.



Khai báo các phương thức trong BillDao:

* Constructor BillDao: Phương thức khởi tạo lớp BillDao nhận vào DbUtil để kết nối với cơ sở dữ liệu



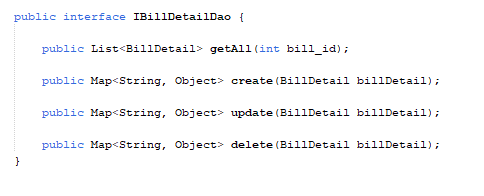
Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

* Phương thức getAll(): Phương thức được sử dụng để lấy danh sách các hoá đơn từ cơ sở dữ liệu dựa trên các thuộc tính trùng với đối tượng bill được đưa vào và thời gian truy xuất.
* Phương thức getByTableId(): Phương thức dùng để đọc dữ liệu của một hóa đơn có table\_id trùng với tham số được đưa vào

#### Interface IBillDetailDao và Class BillDetailDao

Tương tác với cơ sở dữ liệu về dữ liệu chi tiết của các hoá đơn.



Khai báo các phương thức trong BillDetailDao:

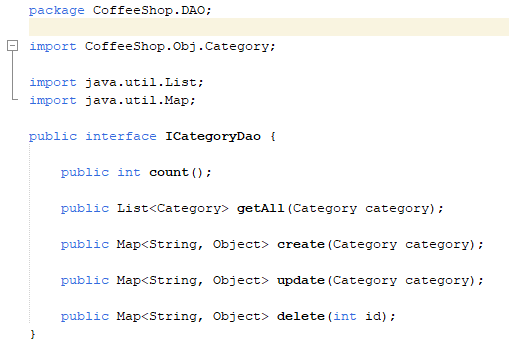
Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

* Phương thức getAll(): Phương thức được sử dụng để lấy dữ liệu chi tiết hoá đơn từ cơ sở dữ liệu dựa trên mã hoá đơn
* Phương thức create(): Phương thức được sử dụng để dữ liệu chi tiết vào 1 hoá đơn (ở đây là thêm sản phẩm vào hoá đơn đang đặt)
* Phương thức update(): Phương thức được sử dụng để cập nhật thông tin chi tiết của 1 hoá đơn (ở đây là thay đổi số lượng của 1 sản phẩm trong hoá đơn đang đặt)
* Phương thức delete(): Phương thức được sử dụng để xóa thông tin chi tiết của 1 hoá đơn (ở đây là xoá 1 sản phẩm trong hoá đơn đang đặt)

#### Interface ICategoryDao và Class CategoryDao

Tương tác với cơ sở dữ liệu về danh mục các sản phẩm của quán cà phê.

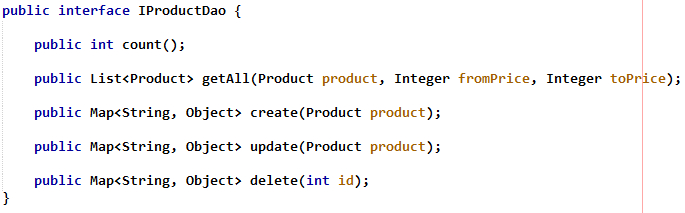


Các phương thức trong CategoryDao:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

#### Interface IProductDao và Class ProductDao

Tương tác với cơ sở dữ liệu về sản phẩm.

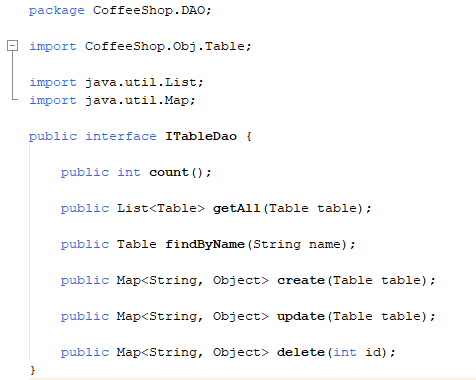
Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự độngCác phương thức trong ProductDao:

Hàm getAll(): Lấy sản phẩm có trong danh mục với điều kiện lấy tương ứng các tham số truyền vào (thuộc tính của product và khoảng giá).

#### Interface ITableDao và Class TableDao

Tương tác với cơ sở dữ liệu về bàn trong quán.



Các phương thức trong TableDao:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

#### Interface IUserDao và Class UserDao

Tương tác với cơ sở dữ liệu về người dùng (gồm nhân viên và quản lý (admin)).

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

* auth(String email, String password): Kiểm tra đăng nhập bằng cách gọi procedure “sp\_checkUser”. Trả về một đối tượng User nếu đăng nhập thành công (có email và password trong cơ sở dữ liệu), ngược lại trả về null.

### Package GUI

#### Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình Mô tả được tạo tự độngPackage GUI.user

Gồm 1 Jpanel PnlUser và 3 Jdialog JDDeleteUser, JDModifyUser và JDSearchUser định nghĩa giao diện và xử lí tương tác của người dùng.

##### JDDeleteUser

Chức năng: Định nghĩa một cửa sổ dialog cho việc xác nhận và xoá một người dùng, bao gồm các phương thức như:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự động

* Constructor JDDeleteUser: Nhận một Frame cha, giá trị boolean cho biết cửa sổ có là modal hay không, một đối tượng DbUtil để tạo kết nối đến cơ sở dữ liệu, một callback để xử lý sự kiện sau khi xoá người dùng, và thông tin về người dùng cần xoá.
* btnDeleteActionPerformed: Xử lý sự kiện khi nút "Xoá" được nhấn. Gọi phương thức delete UserDao để thực hiện xoá người dùng. Hiển thị thông báo và gọi callback nếu xoá thành công.
* btnCancelActionPerformed: Xử lý sự kiện khi nút "Huỷ" được nhấn. Thực hiện đóng cửa sổ dialog.

Giao diện:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, Hình chữ nhật

Mô tả được tạo tự động

##### JDModifyUser

Chức năng: định nghĩa một cửa sổ dialog cho phép thêm mới hoặc sửa đổi thông tin người dùng, bao gồm các phương thức như:

* Constructor JDModifyUser: Thiết lập giao diện và điều chỉnh hiển thị dựa trên việc có đối tượng user hay không, nếu có thì tức đây là sửa đổi thông tin người dùng, nếu là null sẽ là thêm mới người dùng.
* loadingData: Nạp dữ liệu người dùng vào trường dữ liệu nếu đây là sửa đổi thông tin.
* btnModifyActionPerformed: Xử lý sự kiện khi người dùng nhấn “Thêm mới” hoặc “Sửa đổi”. Kiểm tra các dữ liệu đầu vào (tên, email, password) và báo lỗi nếu có. Nếu dữ liệu hợp lệ, tiến hành tạo 1 đối tượng User mới và thực hiện create() hoặc update() thông qua UserDao. Cuối cùng là hiển thị thông báo kết quả và gọi callback để cập nhật lại.

Giao diện:

|  |  |
| --- | --- |
| Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số  Mô tả được tạo tự độngJDSearchUser |  |

Chức năng: định nghĩa một cửa sổ dialog cho phép tìm kiếm thông tin người dùng dựa trên tên, email và quyền.

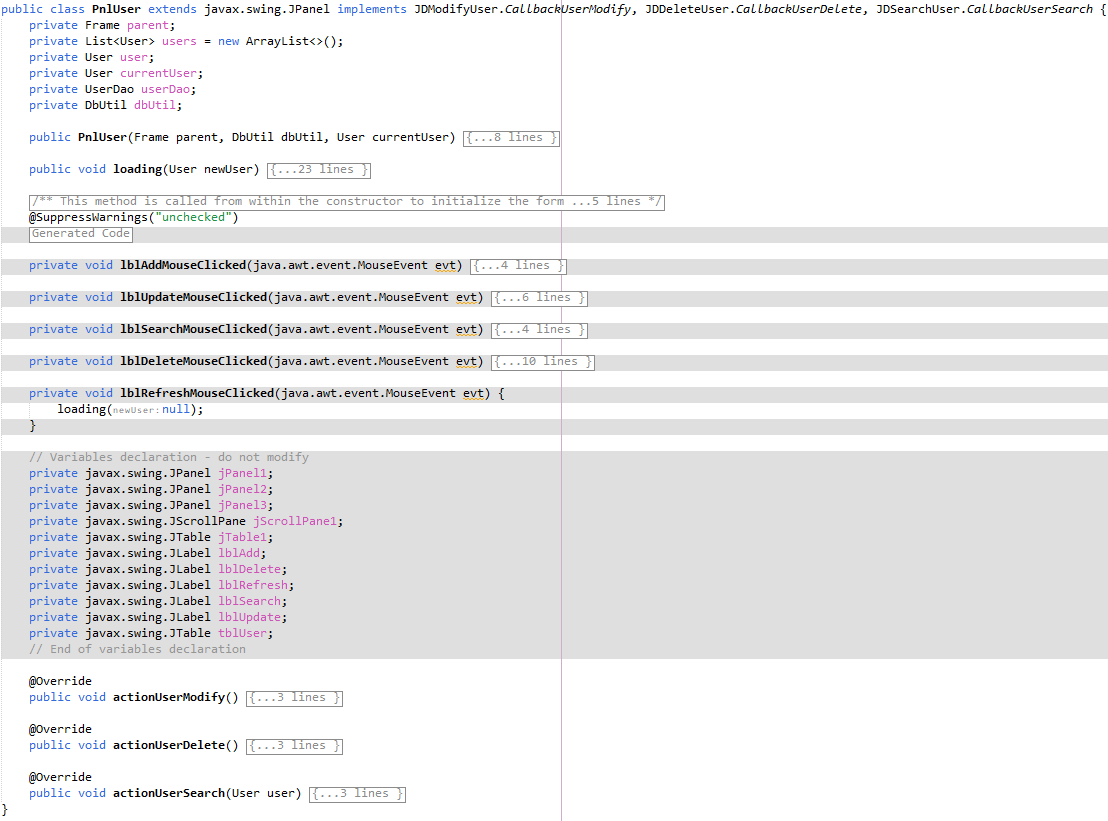
Giao diện:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

##### PnlUser

Chức năng: Jpanel hiển thị và quản lý thông tin người dùng, bao gồm các phương thức như:

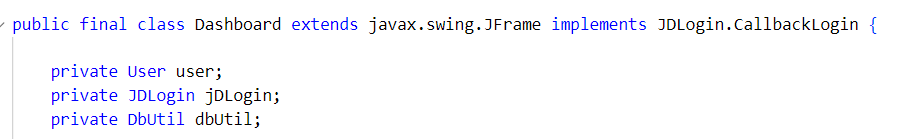


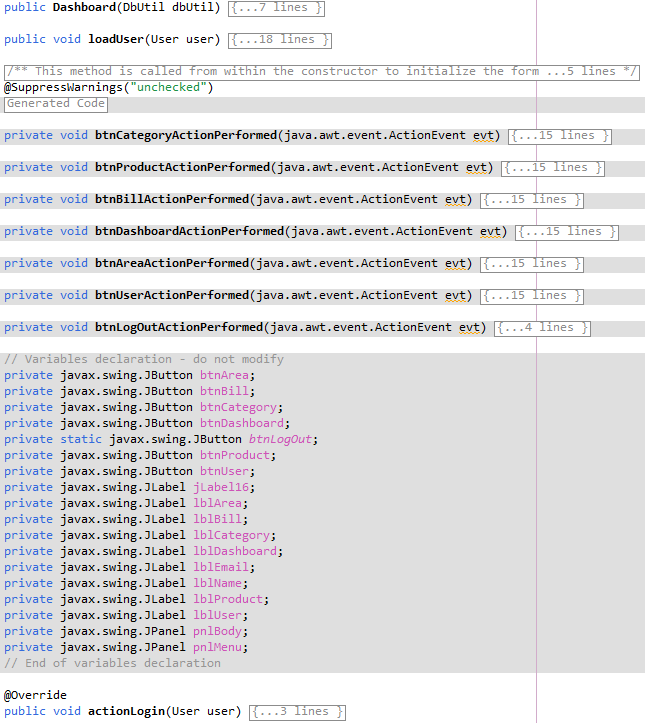
* Constructor: Khởi tạo một đối tượng PnlUser làm giao diện hiển thị danh sách người dùng. Gọi hàm loading(null) để hiển thị danh sách người dùng khi panel được khởi tạo.
* loading: Hiển thị danh sách người dùng sử dụng getAll() trong UserDao. Tham số truyền vào newUser nếu khác null sẽ hiển thị kết quả tìm kiếm.
* Các phương thức lblAddMouseClicked(), lblUpdateMouseClicked(), lblSearchMouseClicked() và lblDeleteMouseClicked(): Xử lý sự kiện khi người dùng nhấn thêm, cập nhật, sửa và xoá. Gọi đến các Jdialog tương ứng để thực hiện.
* actionUserModify(), actionUserDelete(), actionUserSearch(): Ghi đè các phương thức callback từ các dialog (JDModifyUser, JDDeleteUser, JDSearchUser) để cập nhật lại danh sách người dùng sau khi thực hiện thêm mới, sửa đổi, xoá, hoặc tìm kiếm.

#### Package GUI.home

##### Dashboard

Quản lý giao diện và chuyển hướng giữa các chức năng của ứng dụng quản lý quán cà phê.

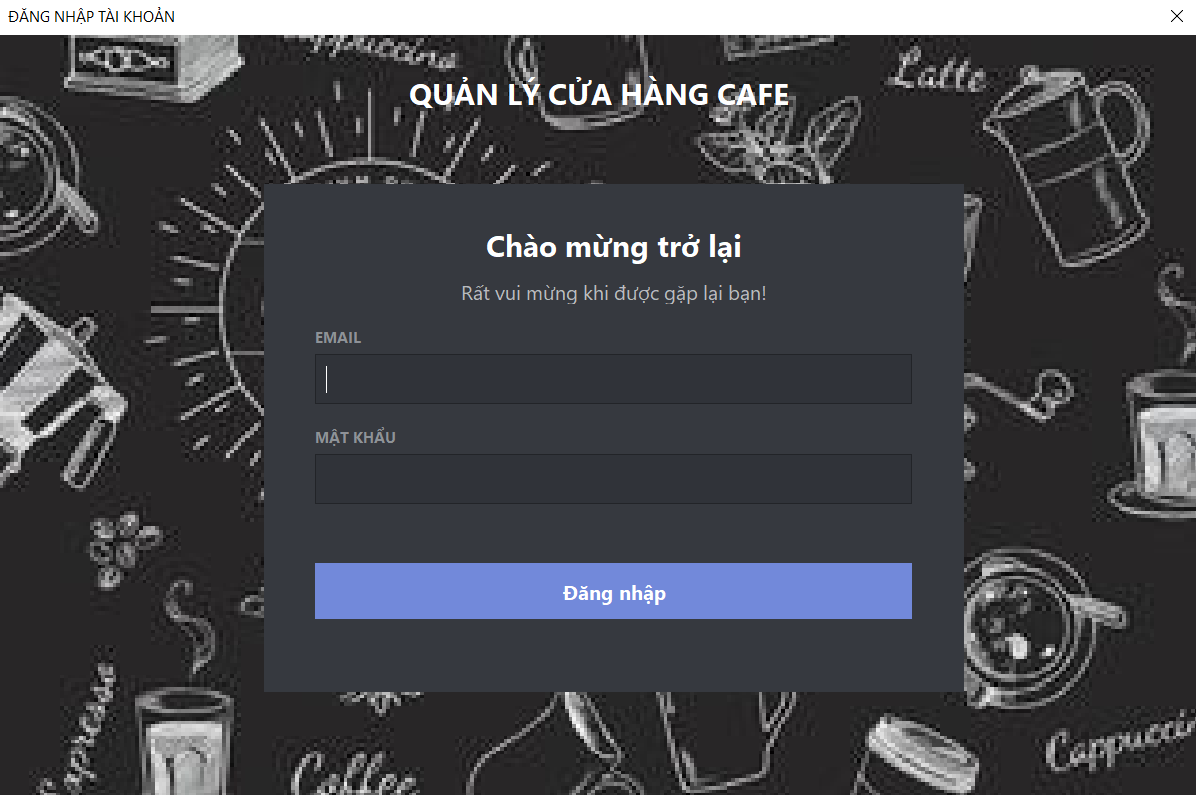
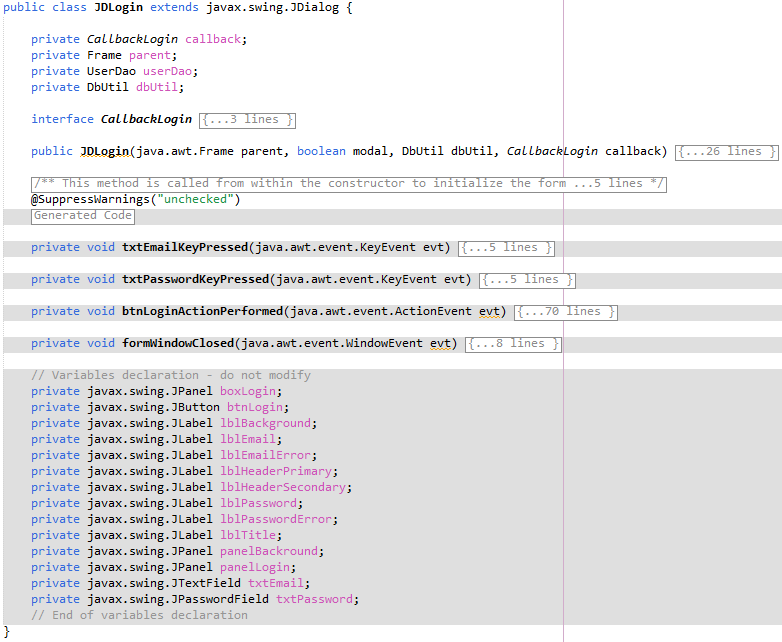


Khai báo các phương thức:

* Phương thức loadUser(): Phương thức này dùng để cập nhật thông tin người dùng khi đăng nhập thành công
* Phương thức btnCategoryActionPerformed(): Phương thức dùng để hiển thị danh mục trên giao diện và thay đổi màu nền của các nhãn để đánh dấu rằng người dùng đang xem “Danh mục”.
* Phương thức btProductActionPerformed(): Phương thức dùng để hiển thị danh sách sản phẩm trên giao diện và thay đổi màu nền của các nhãn để đánh dấu rằng người dùng đang xem “Sản phẩm”.
* Phương thức btnBillActionPerformed(): Phương thức dùng để hiển thị hóa đơn trên giao diện và thay đổi màu nền của các nhãn để đánh dấu rằng người dùng đang xem “Hóa đơn”.
* Phương thức btnDashboardActionPerformed(): Phương thức dùng để hiển thị trang chủ trên giao diện và thay đổi màu nền của các nhãn để đánh dấu rằng người dùng đang xem “Trang chủ”.
* Phương thức btnAreaActionPerformed(): Phương thức dùng để hiển thị nội dung quản lý khu vực & bàn ăn và thay đổi màu nền của các nhãn để đánh dấu rằng người dùng đang xem “Khu vực & bàn ăn”.
* Phương thức btnLogOutActionPerformed(): ẩn cửa sổ chính của ứng dụng và hiển thị cửa sổ đăng nhập khi nhấn vào “Đăng xuất”.
* Phương thức btnUserActionPerformed(): Phương thức dùng để hiển thị trang quản lý người dùng và thay đổi màu nền của các nhãn để đánh dấu rằng người dùng đang xem mục “Người dùng”.

##### JDLogin

Hiển thị cửa sổ đăng nhập



Khai báo các phương thức:

* Constructor JDLogin: phương thức khởi tạo của lớp JDLogin nhận các tham số parent, modal, dbUtil, callback để khởi tạo và đặt vị trí các thành phần giao diện.
* Phương thức txtEmailKeyPressed() và txtPasswordKeyPressed(): khi người dùng nhấn phím Enter khi đang nhập dữ liệu vào ô Email hoặc Mật khẩu thì sẽ thực hiện hành động đăng nhập.
* Phương thức btnLoginActionPerformed(): Lấy thông tin email và mật khẩu từ các trường văn bản và kiểm tra tính hợp lệ của chúng. Nếu thông tin không hợp lệ thì hiển thị thông báo lỗi và thay đổi màu sắc của khung văn bản. Nếu thông tin hợp lệ thì sử dụng “userDao” để xác thực thông tin đăng nhập. Nếu thông tin chính xác, thực hiện hành động đăng nhập thông qua “callback” và ẩn cửa sổ hiện tại, hiển thị cửa số chính và xóa thông tin đăng nhập đã nhập.
* Phương thức formWindowClosed(): khi đóng cửa sổ, phương thức này sẽ được gọi để đóng kết nối đến cơ sở dữ liệu và kết thúc chương trình.

##### PnlHome

Hiển thị tổng quan số lượng về các mục khác nhau: Danh mục, Sản phẩm, Khu vực & bàn ăn, Hóa đơn, Người dùng





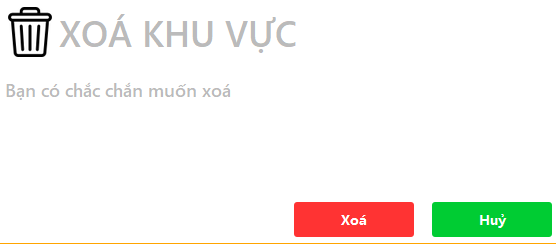
Khai báo các phương thức:

* Constructor PnlHome: phương thức khởi tạo của lớp PnlHome nhận các tham số parent, dbUtil để khởi tạo và đặt vị trí các thành phần giao diện.
* Phương thức loading(): Phương thức có tác dụng cập nhật giao diện với thông tin về số lượng người dùng, danh mục, sản phẩm, khu vực & bàn, hóa đơn từ cơ sở dữ liệu.

#### Package GUI.area

Gồm 1 Jpanel PnlArea và 3 Jdialog JDDeleteArea, JDModifyArea định nghĩa giao diện và xử lí tương tác của người dùng đối với khu vực.

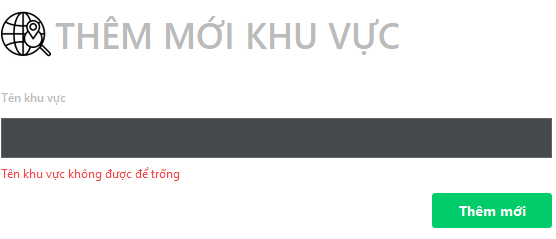
##### JDDeleteArea



Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động

##### JDModifyArea



Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

Các phương thức:

* Phương thức loadData(): Sửa đổi giao diện nếu là hành động “Sửa đổi khu vực” và lấy dữ liệu từ khu vực được người dùng chọn.
* Phương thức btnModifyActionPerformed(): Phương thức giải quyết sự kiện cho nút “Thêm mới” hoặc “Sửa đổi”.
* Phương thức txtNameKeyPressed(): làm phím Enter có chức năng như khi nhấn chuột vào nút “Thêm mới” hoặc “Sửa đổi”.

##### PnlArea



Các phương thức:

* Phương thức PnlArea(): tạo lựa chọn làm mới, nếu là admin thì sẽ có thêm chức năng thêm mới, sửa đổi và xóa.
* Phương thức loading(): tải dữ liệu khu vực và in trong 1 JTabbedPane
* Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

  Mô tả được tạo tự độngPhương thức addTab(): thêm 1 tab mới (khu vực) vào JTabbedPane (chứa nhiều khu vực)
* Phương thức makeTextPanel(): Tạo và trả về một JPanel khu vực được sử dụng để chứa nội dung của mỗi tab trong JTabbedPane (chứa 1 danh sách khu vực)
* Phương thức makeTable(): Tạo và hiển thị một bàn trong giao diện người dùng, được đặt trong một JPanel, thông tin bàn và hóa đơn sẽ hiển thị khi người dùng click vào giao diện bàn
* Phương thức lblAddMouseClicked(): Thêm sự kiện trong nút “Thêm mới” để thêm khu vực
* Phương thức lblUpdateMouseClicked(): Thêm sự kiện trong nút “Sửa đổi” để sửa đổi khu vực
* Phương thức tabbedPaneMouseClicked (): Thêm sự kiện cho các nút trong thanh TabbedPane để thay đổi lựa chọn các khu vực đang có
* Phương thức lblDeleteMouseClicked(): Thêm sự kiện trong nút “Xóa” để xóa khu vực
* Phương thức lblRefreshMouseClicked (): Thêm sự kiện trong nút “Làm mới” để hiện thị lại toàn bộ dữ liệu ban đầu
* Phương thức jMenuItemDeleteActionPerformed (): Thêm sự kiện xóa bàn trong 1 khu vực
* Phương thức jMenuItemEditActionPerformed (): Thêm sự kiện cập nhật bàn trong 1 khu vực

#### Package GUI.bill

##### ExportBill

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, số

Mô tả được tạo tự độngXuất file dữ liệu bill dưới dạng file văn bản

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

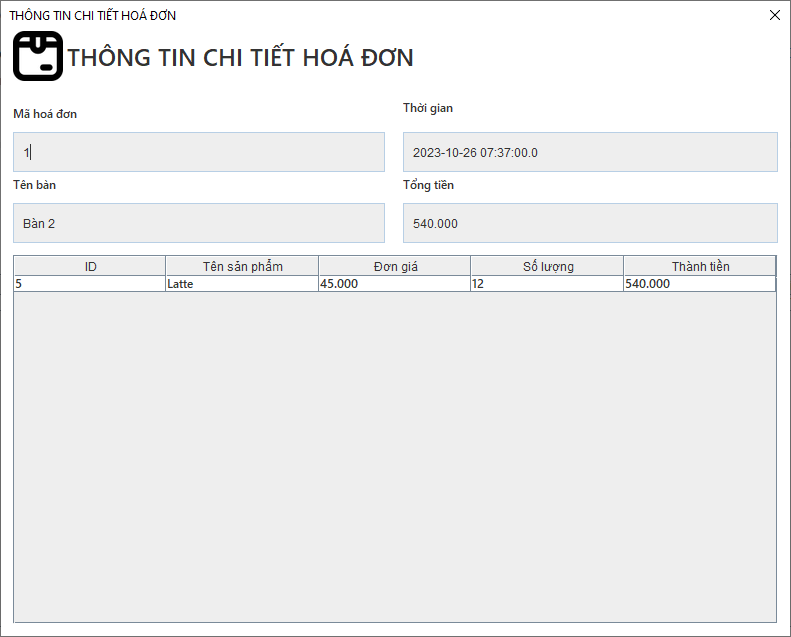
Mô tả được tạo tự độngKhai báo các thành phần thuộc tính và phương thức:

Khai báo các phương thức:

* Constructor ExportBill: Khởi tạo lớp ExportBill nhận các tham số parent, modal, dbUtil, bill để khởi tạo các thuộc tính và thực hiện các thao tác xuất hoá đơn dựa trên dữ liệu và tùy chọn người dùng
* Phương thức createFileName(): Tạo tên File dựa trên thời gian hiện tại
* Phương thức printRow(String[] rowData): Phương thức trả về một dòng trong bảng hóa đơn để xuất ra file với các cột được căn chỉnh
* Phương thức padString(String str, int length): Phương thức này dùng để thêm ký tự vào chuỗi để đạt được độ dài length

##### JDBill

Hiển thị thông tin chi tiết hóa đơn



Khai báo các thành phần thuộc tính và phương thức:

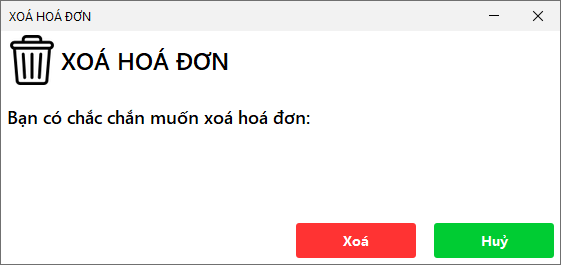
Ảnh có chứa văn bản, ảnh chụp màn hình, số, phần mềm

Mô tả được tạo tự động

* Constructor JDBill: Phương thức khởi tạo của lớp JDBill nhận các tham số parent, modal, dbUtil, callback, bill để khởi tạo và đặt vị trí các thành phần giao diện
* Phương thức loading(): Phương thức này dùng để hiển thị dữ liệu chi tiết của hóa đơn từ cơ sở dữ liệu và hiển thị lên bảng tblBillDetail

##### JDDeleteBill

Hiển thị bảng xác nhận xóa hóa đơn



Khai báo các thành phần thuộc tính và phương thức:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

##### JDSearchBill

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự độngHiển thị bảng tìm kiếm hóa đơn

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự độngKhai báo các thành phần thuộc tính và phương thức:

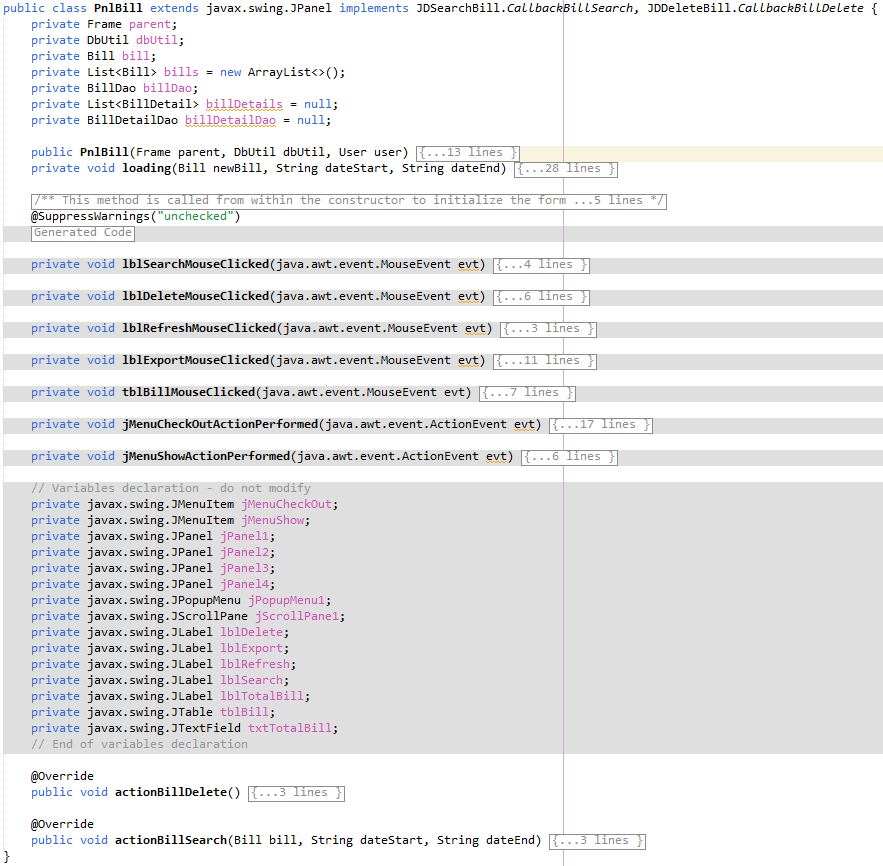
* Constructor JDSearchBill: Phương thức khởi tạo của lớp JDSearchBill nhận các tham số parent, modal, dbUtil, callback để khởi tạo và đặt vị trí các thành phần giao diện
* Phương thức btnSearchActionPerformed(): Phương thức dùng để lấy thông tin người dùng nhập vào từ các ô, kiểm tra tính hợp lệ của từng trường nhập vào và hiển thị danh sách các Bill thỏa mãn nếu nhập vào hợp lệ.

##### PnlBill

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự độngHiển thị danh sách các hóa đơn được tạo và các nút truy cập tiện ích liên quan tới hóa đơn

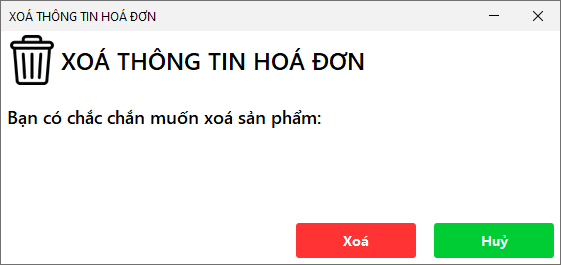
Khai báo các thành phần thuộc tính và phương thức:

* Constructor PnlBill: Phương thức khởi tạo của lớp PnlBill nhận các tham số parent, dbUtil, user để khởi tạo và đặt vị trí các thành phần giao diện
* Phương thức loading(): Phương thức dùng để biểu diễn danh sách các bill theo tham số được truyền vào và tính toán tổng tiền thu về của các hoá đơn hiển trị trên bảng.
* Phương thức tblBillMouseClicked(): Hiển thị 1 popupMenu sau khi người dùng chuột phải vào 1 hàng trên bảng danh sách hoá đơn, cho phép người dùng chọn “Xem” để xem chi tiết hoá đơn và “Thanh toán” nếu như hoá đơn này chưa được thanh toán.
* Phương thức actionBillDelete(): Dùng để hiển thị lại toàn bộ danh sách các bill bằng việc gọi tới hàm loading() với tham số null sau khi thực hiện xoá 1 hoá đơn.
* Phương thức actionBillSearch(): Dùng để hiển thị danh sách các bill thỏa mãn bill được truyền vào và theo khoảng thời gian bằng hàm loading()

#### Package GUI.billDetail

##### JDDeleteBillDetail

Hiển thị bảng xác nhận xoá thông tin chi tiết trong hoá đơn (ở đây là xoá 1 sản phẩm trong hoá đơn đang đặt)



Khai báo các thành phần thuộc tính và phương thức:

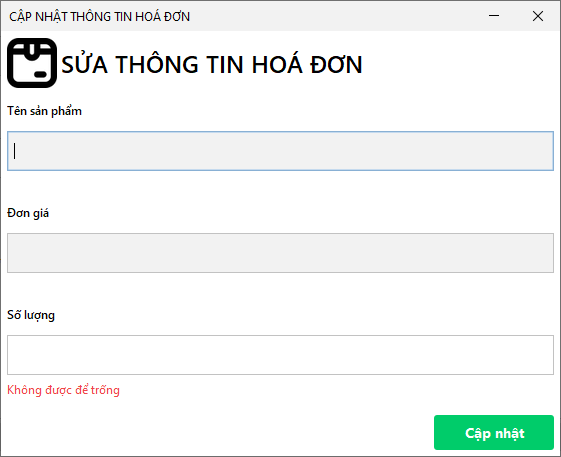
Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

Khai báo các phương thức:

* Constructor JDDeleteBillDetail: Phương thức khởi tạo của lớp JDDeleteBill nhận các tham số parent, modal, dbUtil, callback, billDetail để khởi tạo và đặt vị trí các thành phần giao diện
* Phương thức btnDeleteActionPerformed(): Phương thức này dùng để xóa một chi tiết hóa đơn khi người dùng ấn vào nút xóa và hiển thị thông báo xóa thành công
* Phương thức btnCancelActionPerformed(): Phương thức dùng hàm dispose() để đóng cửa sổ khi người dùng nhấn nút hủy

##### JDModifyBillDetail

Hiển thị bảng xác nhận cập nhật chi tiết thông tin trong hoá đơn (ở đây là sửa lại số lượng của 1 sản phẩm trong hoá đơn đang đặt)

Khai báo các thành phần thuộc tính và phương thức:

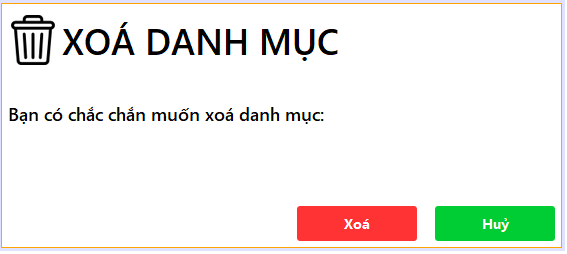
* Constructor JDModifyBillDetail: Phương thức khởi tạo của lớp JDDeleteBill nhận các tham số parent, modal, dbUtil, callback, billDetail để khởi tạo và đặt vị trí các thành phần giao diện
* Phương thức btnModifyActionPerformed(): Phương thức này dùng để thay đổi một chi tiết hóa đơn khi người dùng ấn vào nút xóa và hiển thị thông báo xóa thành công

#### Package GUI.category

Gồm 1 Jpanel PnlCategory và 3 Jdialog JDDeleteCategory, JDModifyCategory và JDSearchCategory định nghĩa giao diện và xử lý tương tác của người dùng đối với danh mục.

##### JDDeleteCategory

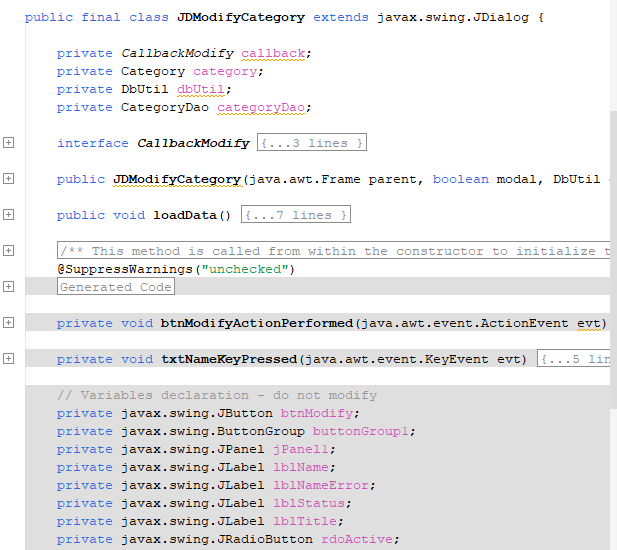
Định nghĩa một cửa sổ dialog cho việc xác nhận và xoá một danh mục



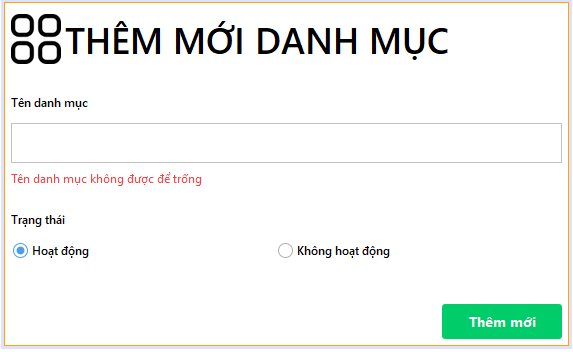


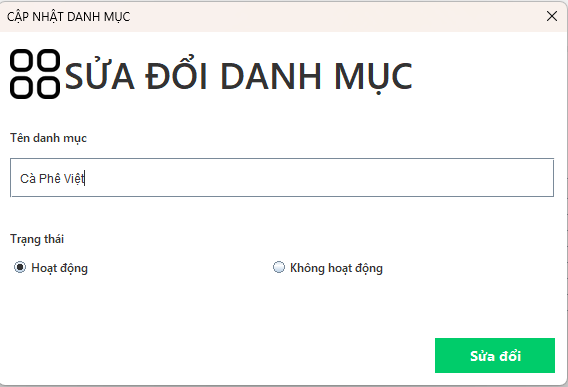
##### JDModifyCategory

Định nghĩa một cửa sổ dialog cho phép thêm mới hoặc sửa đổi thông tin danh mục



* Constructor JDModifyCategory: Thiết lập giao diện và điều chỉnh hiển thị dựa trên việc có đối tượng category hay không, nếu có thì tức đây là sửa đổi thông tin danh mục, nếu là null sẽ là thêm mới danh mục
* loadData: Nạp dữ liệu danh mục vào trường dữ liệu và sửa lại giao diện JDialog nếu đây là sửa đổi thông tin.
* btnModifyActionPerformed: Xử lý sự kiện khi người dùng nhấn “Thêm mới” hoặc “Sửa đổi”. Kiểm tra các dữ liệu đầu vào (tên) và báo lỗi nếu có. Nếu dữ liệu hợp lệ, tiến hành tạo 1 đối tượng danh mục mới và thực hiện create() hoặc update() thông qua CategoryDao. Cuối cùng là hiển thị thông báo kết quả và gọi callback để cập nhật lại.

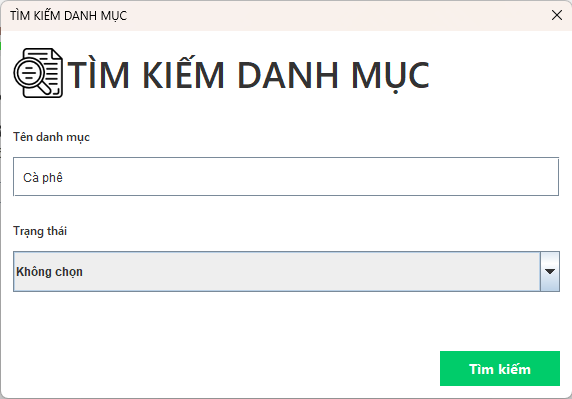




##### JDSearchCategory

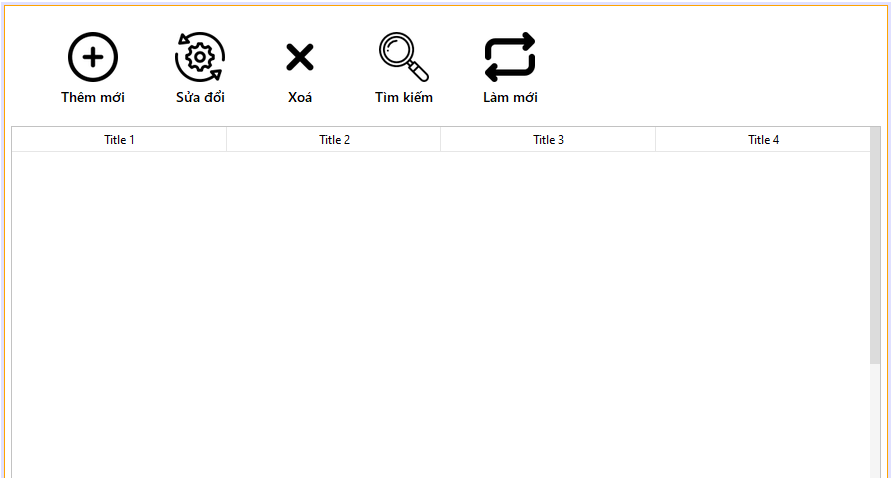
Định nghĩa một cửa sổ dialog cho tìm kiếm thông tin danh mục dựa trên tên danh mục, trạng thái

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

##### PnlCategory

Jpanel hiển thị và quản lý thông tin category



Các phương thức:

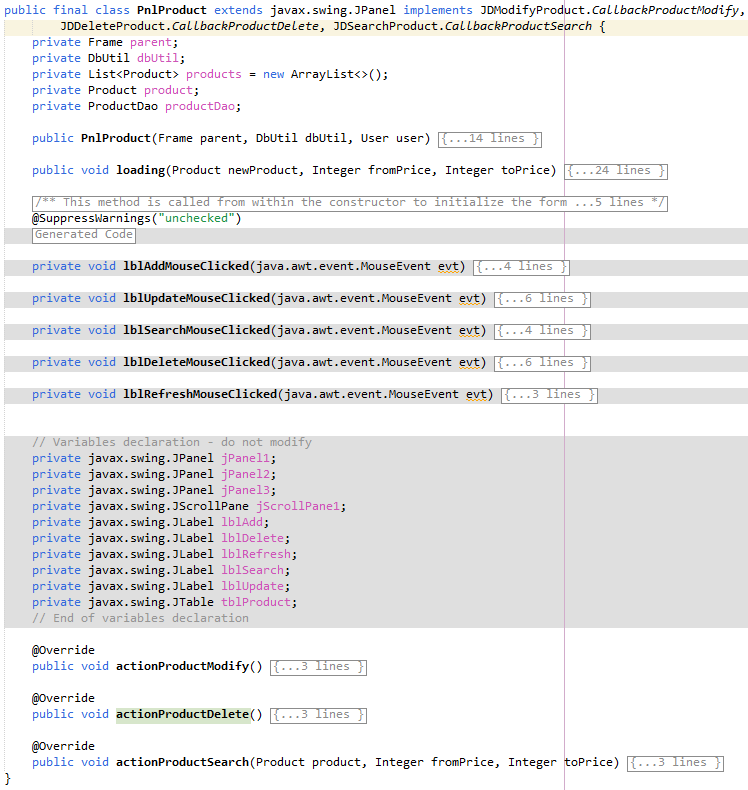
* Constructor: Khởi tạo một đối tượng PnlCategory làm giao diện hiển thị danh sách các category. Gọi hàm loading để hiển thị danh sách category khi panel được khởi tạo.
* loading: Hiển thị danh sách category sử dụng getAll() trong CategoryDao. Tham số truyền vào newCategory nếu khác null sẽ hiển thị kết quả tìm kiếm.
* Các phương thức lblAddMouseClicked, lblUpdateMouseClicked, lblSearchMouseClicked,  lblDeleteMouseClicked và lblRefreshMouseClicked : Xử lý sự kiện khi người dùng nhấn thêm, cập nhật, tìm kiếm,  xoá và làm mới. Gọi đến các Jdialog tương ứng để thực hiện.
* Các phương thức actionCategoryModify(), actionCategoryDelete(), actionCategorySearch(): Ghi đè các phương thức callback từ các dialog (JDModifyCategory, JDDeleteCategory, JDSearchCategory) để cập nhật lại danh sách category sau khi thực hiện thêm mới, sửa đổi, xoá, hoặc tìm kiếm.

#### Package GUI.product

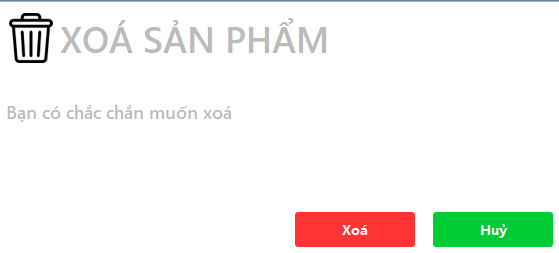
Gồm 1 Jpanel PnlProduct và 3 Jdialog JDDeleteProduct, JDModifyProduct, JDSearchProduct định nghĩa giao diện và xử lí tương tác của người dùng.

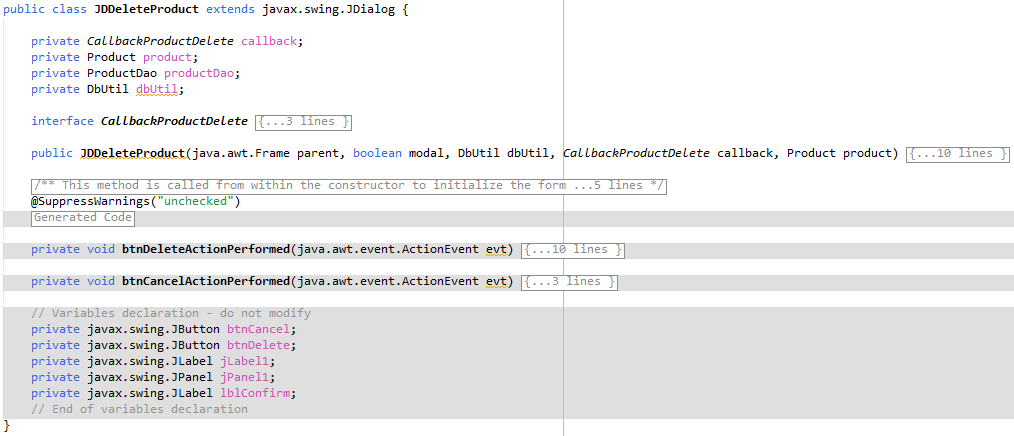
##### PnlProduct





##### JDDeleteProduct





Các phương thức:

* Phương thức btnDeleteActionPerformed(): Phương thức dùng để xóa một Product khỏi cơ sở dữ liệu khi người dùng ấn vào nút xóa trên cửa sổ JDDeleteProduct, hiển thị thông báo xóa thành công hay không thành công sau khi hoàn thành.
* Phương thức btnCancelActionPerformed(): Phương thức đóng cửa sổ khi người dùng tương tác với nút “Huỷ” bằng hàm dispose()

##### JDModifyProduct

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động

* Phương thức loadCategory(): nạp dữ liệu các danh mục đang hoạt động vào 1 DefaultComboBoxModel cho người dùng lựa chọn.
* Phương thức loadData(): nạp dữ liệu Product đang được chọn vào các trường nhập nếu thực hiện sửa đổi sản phẩm.
* Phương thức btnModifyActionPerformed (): Phương thức dùng để tạo sự kiện cho nút “Thêm mới” hoặc “Sửa đổi”.

##### Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, màn hình Mô tả được tạo tự độngJDSearchProduct

Ảnh có chứa văn bản, ảnh chụp màn hình, số, phần mềm

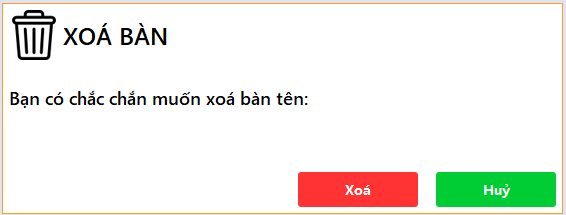
Mô tả được tạo tự động

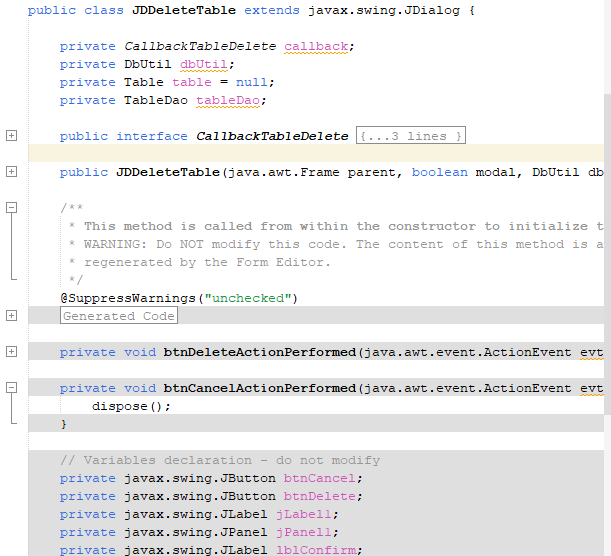
Các phương thức:

* Phương thức loadCategory(): tải danh sách Categories trong cơ sở dữ liệu, và hiển thị trong JcomboBox, mặc định là “Không chọn”.
* Phương thức loadStatus(): thêm 3 lựa chọn "Không chọn", "Hoạt động" và "Không hoạt động" vào JComboBox
* Phương thức btnModifyActionPerformed(): tạo sự kiện click chuột tìm sản phẩm có trong cơ sở dữ liệu, tham số gồm tên, khoảng giá, tên danh mục và trạng thái hoạt động

#### Package GUI.table

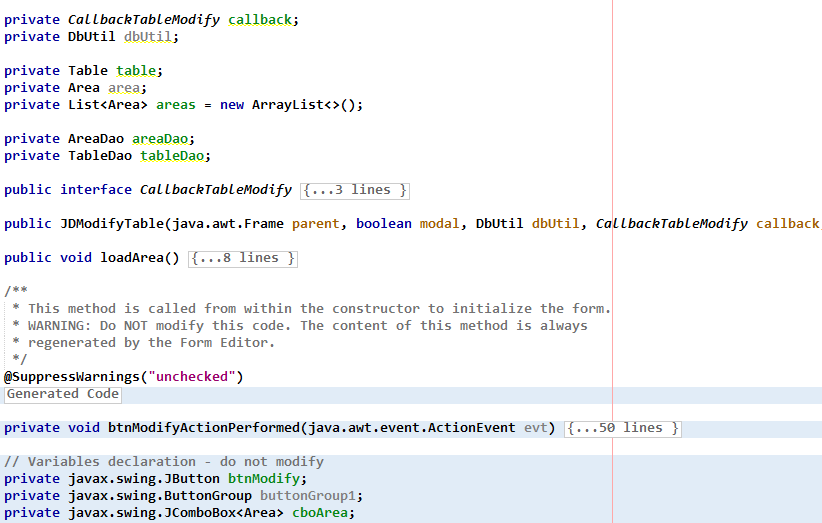
##### JDDeleteTable





##### JDModifyTable

|  |  |
| --- | --- |
|  |  |

Các phương thức:

* loadArea(): Đưa dữ liệu khu vực vào JcomboBox cho người dùng lựa chọn.
* btnModifyActionPerformed: Xử lý sự kiện khi nút "Sửa đổi" được nhấn. Hiển thị thông báo và gọi callback nếu sửa thành công.

##### JDTable

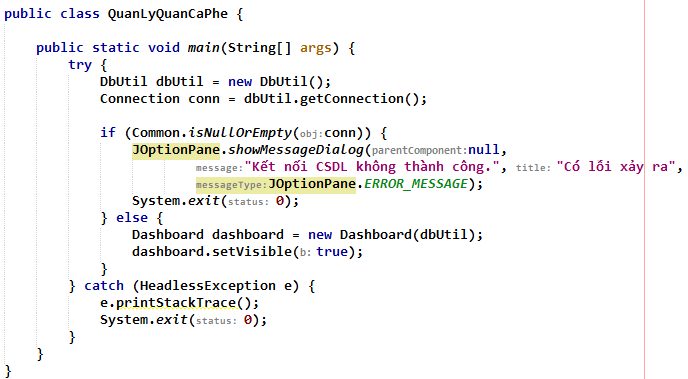


Các phương thức:

* Constructor JDTable: Nhận một Frame cha, giá trị boolean cho biết cửa sổ có là modal hay không, một đối tượng DbUtil để tạo kết nối đến cơ sở dữ liệu, một callback để xử lý sự kiện sau khi xoá người dùng, thông tin về tài khoản user đang sử dụng và tên bàn đang đặt.
* loadingProduct(): Thêm danh sách sản phẩm đang hoạt động cho người dùng lựa chọn vào Jtable.
* loadingBill(): Tạo thông tin hoá đơn khi ấn nút “Đặt bàn”.
* loadingBillDetail(): Hiển thị thông tin chi tiết sản phẩm trong hoá đơn người dùng đang đặt và cập nhật tổng giá trị hoá đơn.
* btnBookActionPerformed: Xử lý sự kiện khi nút "Đặt bàn" được nhấn.
* btnAddProductActionPerformed: Xử lý sự kiện khi nút "Thêm sản phẩm" được nhấn.
* btnCheckoutActionPerformed: Xử lý sự kiện khi nút "Thanh toán" được nhấn.
* menuItemEditActionPerformed và menuItemDeleteActionPerformed: Xử lý sự kiện khi người dùng chọn sản phẩm đã được đặt và tiến hành sửa đổi hoặc xoá nó khỏi hoá đơn.
* btnDeleteBillActionPerformed: Xử lý sự kiện khi nút "Hủy đơn" được nhấn.

### Package team13.quanlyquancaphe

Chỉ có class Main để khởi chạy chương trình, tiến hành kết nối thử đến cơ sở dữ liệu, nếu thành công sẽ tạo và hiển thị giao diện chính (Dashboard) của ứng dụng.



# Tài liệu tham khảo

Java Wrap Layout: <https://tips4java.wordpress.com/2008/11/06/wrap-layout/>

Mã nguồn tham khảo từ: <https://github.com/anhnmt/java-coffee-shop-management/>