**Trường Đại học Thăng Long**

**Logo

Description automatically generated**

**BÁO CÁO BÀI TẬP LỚN**

**ỨNG DỤNG QUẢN LÝ BÁN GIÀY THỂ THAO**

**Sinh viên thực hiện: A35025 Nguyễn Công Hoàn**

**Giảng viên: Nguyễn Hùng Cường**

**HÀ NỘI – 2024**

**MỤC LỤC**

[PHẦN 1. cơ sở lý thuyết 1](#_Toc177759631)

[1.1. Java: 1](#_Toc177759632)

[1.2. MySQL 1](#_Toc177759633)

[1.3. MariaDB 2](#_Toc177759634)

[1.4. NetBeans IDE 2](#_Toc177759635)

[PHẦN 2. Phân tích và đặc tả chức năng 3](#_Toc177759636)

[2.1. Mô tả bài toán 3](#_Toc177759637)

[2.2. Xác định tác nhân 3](#_Toc177759638)

[2.3. Mô tả Use Case 3](#_Toc177759639)

[2.3.1. Sơ đồ Use Case 3](#_Toc177759640)

[2.3.2. Danh sách Use Case 4](#_Toc177759641)

[2.3.3. Biểu đồ Use Case phân rã 4](#_Toc177759642)

[2.4. Đặc tả Use Case 6](#_Toc177759643)

[2.4.1. Đăng ký 6](#_Toc177759644)

[2.4.2. Đăng nhập 7](#_Toc177759645)

[2.4.3. Tìm kiếm 8](#_Toc177759646)

[2.4.4. Thông tin cá nhân 8](#_Toc177759647)

[2.4.5. Thanh toán 9](#_Toc177759648)

[2.4.6. Giỏ hàng 9](#_Toc177759649)

[2.4.7. Thông tin đơn hàng 10](#_Toc177759650)

[2.4.8. Sản phẩm 11](#_Toc177759651)

[PHẦN 3. FrontEnd 12](#_Toc177759652)

[3.1. Tạo dự án 12](#_Toc177759653)

[3.2. Cài đặt thư viện JDBC kết nối tới Database 13](#_Toc177759654)

[3.3. Tạo form frmLogin 15](#_Toc177759655)

[3.4. Tạo form frmAddItem 16](#_Toc177759656)

[3.5. Tạo frm frmAllProductAdmin 16](#_Toc177759657)

[3.6. Tạo frmAllProductClient 17](#_Toc177759658)

[3.7. Tạo form frmCart 17](#_Toc177759659)

[3.8. Tạo form frmItemClient 18](#_Toc177759660)

[3.9. Tạo form frmProfile 18](#_Toc177759661)

[3.10. Tạo form frmRegister 19](#_Toc177759662)

[3.11. Tạo form frmSearch 19](#_Toc177759663)

[3.12. Tạo form frmUpdateInfo 20](#_Toc177759664)

[3.13. Tạo form frmUpdateItemInfoAdmin 20](#_Toc177759665)

[PHẦN 4. Backend 22](#_Toc177759666)

[4.1. Thiết kế cơ sở dữ liệu (Database) 22](#_Toc177759667)

[4.2. Công nghệ sử dụng 22](#_Toc177759668)

[4.3. Cấu trúc thư mực 23](#_Toc177759669)

[4.3.1. Package DataBaseConnection 23](#_Toc177759670)

[4.3.2. Package form 23](#_Toc177759671)

[4.3.3. Package image 23](#_Toc177759672)

[4.4. Triển khai code 23](#_Toc177759673)

[4.4.1. Form frmLogin 23](#_Toc177759674)

[4.4.2. Form frmItemClient 25](#_Toc177759675)

[4.4.3. Form frmSearch 31](#_Toc177759676)

[4.4.4. Form frmCart 35](#_Toc177759677)

[4.4.5. Form frmAllProductClient 42](#_Toc177759678)

[4.4.6. Form frmProfile 46](#_Toc177759679)

[4.4.7. Form frmUpdateInfo 48](#_Toc177759680)

[4.4.8. Phương thức 49](#_Toc177759681)

[4.4.9. Form frmRegister 50](#_Toc177759682)

[4.4.10. Form frmAllProductAdmin 51](#_Toc177759683)

[4.4.11. Form frmAddItem 54](#_Toc177759684)

[PHẦN 5. Build App 57](#_Toc177759685)

**PHỤ LỤC**

[Sơ đồ 2.3.1. Sơ đồ Use Case tổng quát 3](#_Toc177756518)

[Sơ đồ 2.3.2. Phân rã Use Case Đăng ký 4](#_Toc177756519)

[Sơ đồ 2.3.3. Phân rã Use Case Đăng nhập 4](#_Toc177756520)

[Sơ đồ 2.3.4. Phân rã Use Case Tìm kiếm 5](#_Toc177756521)

[Sơ đồ 2.3.5. Phân rã Use Case Thanh toán 5](#_Toc177756522)

[Sơ đồ 2.3.6. Phân rã Use Case Thông tin cá nhân 5](#_Toc177756523)

[Sơ đồ 2.3.7. Phân rã Use Case Giỏ hàng 6](#_Toc177756524)

[Sơ đồ 2.3.8. Phân rã Use Case Thông tin đơn hàng 6](#_Toc177756525)

[Sơ đồ 2.3.9. Phân rã Use Case Sản phẩm 6](#_Toc177756526)

[Sơ đồ 4.1.1. Mô hình UML 22](#_Toc177756527)

[Bảng 2.1. Danh sách tác nhân 3](#_Toc177756529)

[Danh sách 2.3.1. Danh sách Use Case 4](#_Toc177756541)

[Hình 1.1. Java 1](#_Toc177756551)

[Hình 1.2. MySQL 1](#_Toc177756552)

[Hình 1.3. MariaDB 2](#_Toc177756553)

[Hình 1.4. NetBeans IDE 2](#_Toc177756554)

# cơ sở lý thuyết

Những công nghệ mà em sử dụng để làm ra ứng dụng Quản lý bán giày thể thao bao gồm: Java , MySQL, MariaDB, NetBeans IDE.

## Java:



Hình 1.1. Java

Java là một ngôn ngữ lập trình hướng đối tượng, dựa trên lớp được thiết kế để có càng ít phụ thuộc thực thi càng tốt. Nó là ngôn ngữ lập trình có mục đích chung cho phép các nhà phát triển ứng dụng viết một lần, chạy ở mọi nơi (**WORA**), nghĩa là mã Java đã biên dịch có thể chạy trên tất cả các nền tảng hỗ trợ Java mà không cần biên dịch lại. Các ứng dụng Java thường được biên dịch thành bytecode có thể chạy trên bất kỳ máy ảo Java (**JVM**) nào bất kể kiến trúc máy tính bên dưới. Cú pháp của Java tương tự như C và C++, nhưng có ít cơ sở cấp thấp hơn các ngôn ngữ trên. Java Runtime cung cấp các khả năng động (chẳng hạn như phản ánh và sửa đổi mã thời gian chạy) thường không có sẵn trong các ngôn ngữ biên dịch truyền thống.

## MySQL

A logo with a dolphin

Description automatically generated

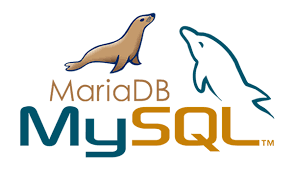
Hình 1.2. MySQL

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS, ...

MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL)

MySQL được sử dụng cho việc bổ trợ Node.js, PHP, Perl, và nhiều ngôn ngữ khác, là nơi lưu trữ những thông tin trên các trang web viết bằng NodeJs, PHP hay Perl,...

## MariaDB

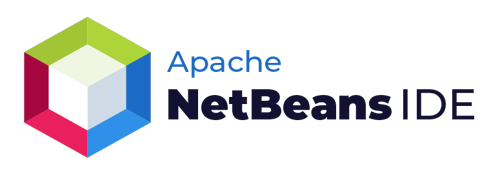


Hình 1.3. MariaDB

MariaDB là một sản phẩm mã nguồn mở tách ra từ mã mở do cộng đồng phát triển của hệ quản trị cơ sở dữ liệu quan hệ MySQL nhằm theo hướng không phải trả phí với GNU GPL.

MariaDB được định hướng để duy trì khả năng tương thích cao với MySQL, để đảm bảo khả năng hỗ trợ về thư viện đồng thời kết hợp một cách tốt nhất với các API và câu lệnh của MySQL. MariaDB đã có công cụ hỗ lưu trữ XtraDB thay cho InnoDB, cũng như một công cụ lưu trữ mới, Aria, dự định sẽ là một công cụ giao dịch và phi giao dịch thậm chí có thể bao gồm trong các phiên bản MySQL trong tương lai.

## NetBeans IDE



Hình 1.4. NetBeans IDE

NetBeans là một môi trường phát triển tích hợp (IDE) cho Java. NetBeans cho phép các ứng dụng được phát triển từ một tập hợp các thành phần phần mềm được gọi là modules. NetBeans chạy trên Windows, macOS, Linux và Solaris. Ngoài việc phát triển Java, nó còn có các phần mở rộng cho các ngôn ngữ khác như PHP, C, C++, HTML5, và JavaScript. Các ứng dụng dựa trên NetBeans, bao gồm NetBeans IDE, có thể được mở rộng bởi các nhà phát triển bên thứ ba.

# Phân tích và đặc tả chức năng

## Mô tả bài toán

Ứng dụng Quản lý bán giày thể thao giúp người dùng dễ dàng tìm kiếm và lựa chọn những đôi giày thể thao phù hợp với nhu cầu. Giúp admin dễ dàng quản lý thông tin của tất cả sản phẩm.

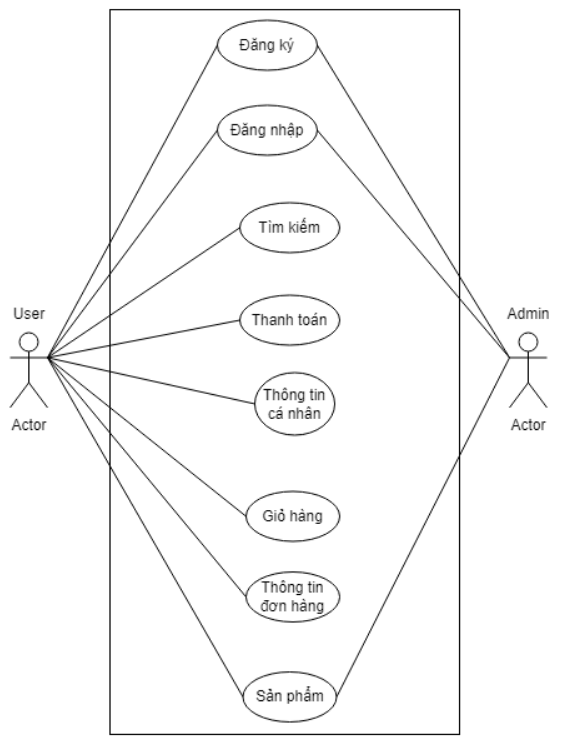
## Xác định tác nhân

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên tác nhân** | **Mô tả** |
| 1 | Người dùng | Người dùng có thể đăng kí, đăng nhập, tìm kiếm giày thể thao, kiểm tra giỏ hàng, thay đổi số lượng sản phẩm trong giỏ hàng ,thanh toán đơn hàng và kiểm tra thông tin cá nhân. |
| 2 | Admin | Admin quản lý thông tin của tất cả sản phẩm. |

Bảng .. Danh sách tác nhân

## Mô tả Use Case

### Sơ đồ Use Case



Sơ đồ 2.3.1. Sơ đồ Use Case tổng quát

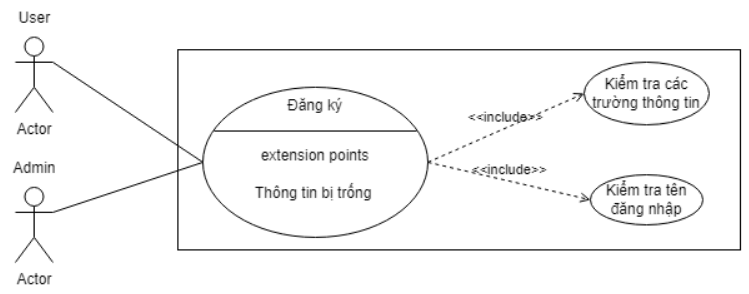
### Danh sách Use Case

|  |  |  |
| --- | --- | --- |
| **Mã Use Case** | **Tên Use Case** | **Tác nhân** |
| 1 | Đăng ký | Người dùng, admin |
| 2 | Đăng nhập | Người dùng, admin |
| 3 | Tìm kiếm | Người dùng |
| 4 | Thông tin cá nhân | Người dùng |
| 5 | Thanh toán | Người dùng |
| 6 | Giỏ hàng | Người dùng |
| 7 | Thông tin đơn hàng | Người dùng |
| 8 | Sản phẩm | Người dùng, admin |

Danh sách 2.3.1. Danh sách Use Case

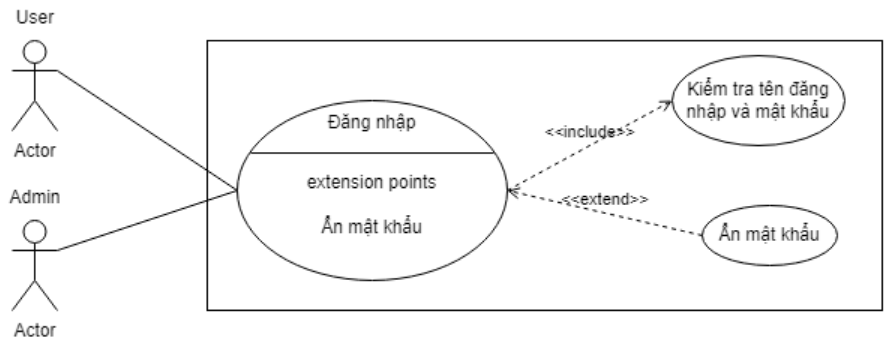
### Biểu đồ Use Case phân rã

#### Phân rã Use Case Đăng ký



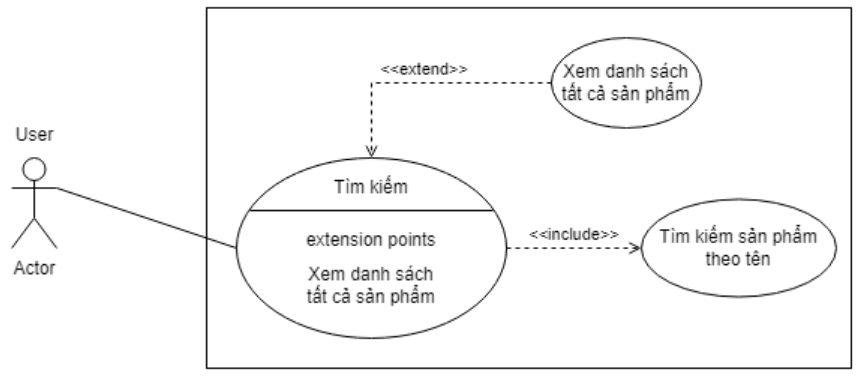
Sơ đồ 2.3.2. Phân rã Use Case Đăng ký

#### Phân rã Use Case Đăng nhập



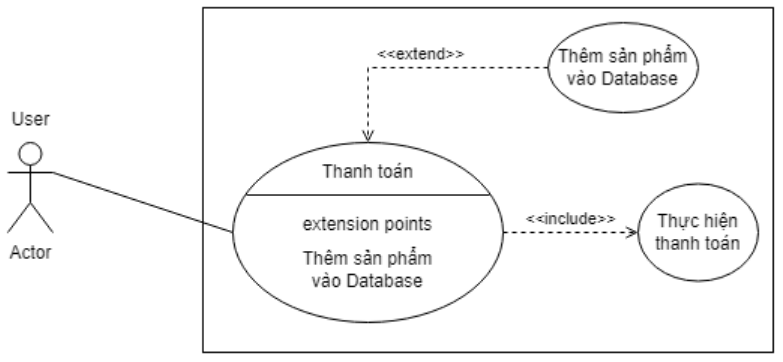
Sơ đồ .. Phân rã Use Case Đăng nhập

#### Phân rã Use Case Tìm kiếm



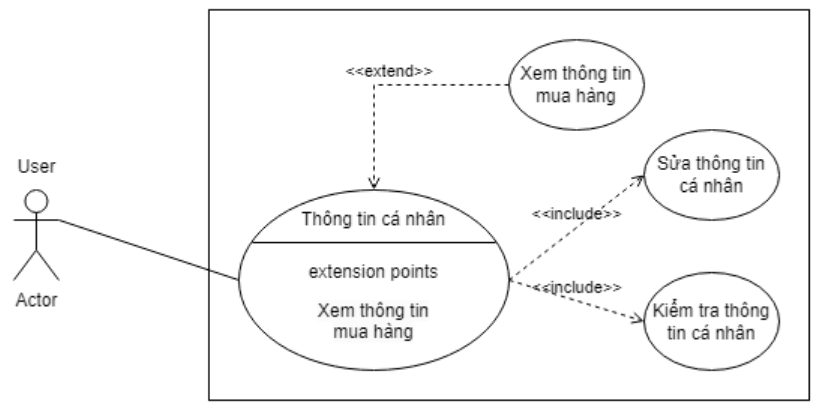
Sơ đồ 2.3.4. Phân rã Use Case Tìm kiếm

#### Phân rã Use Case Thanh toán



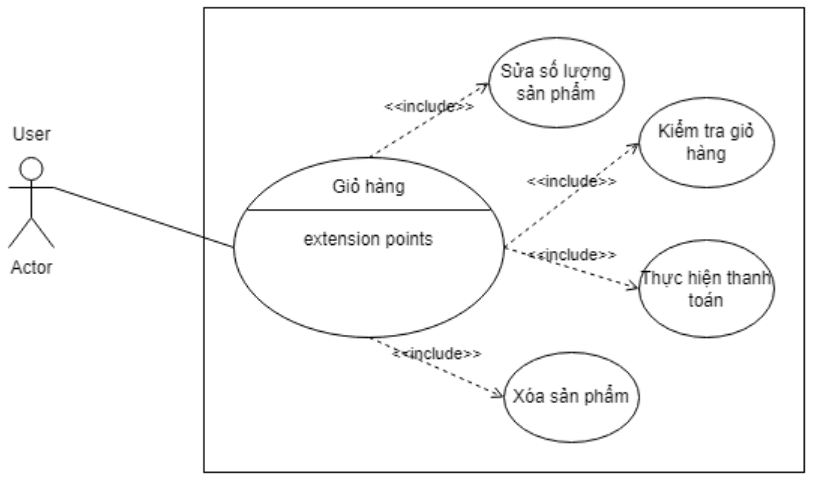
Sơ đồ 2.3.5. Phân rã Use Case Thanh toán

#### Phân rã Use Case Thông tin cá nhân



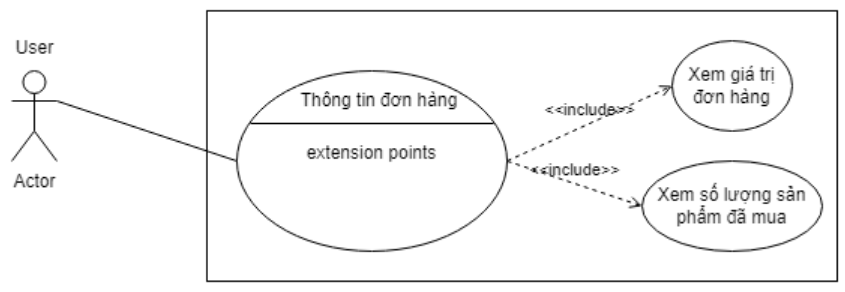
Sơ đồ 2.3.6. Phân rã Use Case Thông tin cá nhân

#### Phân rã Use Case Giỏ hàng



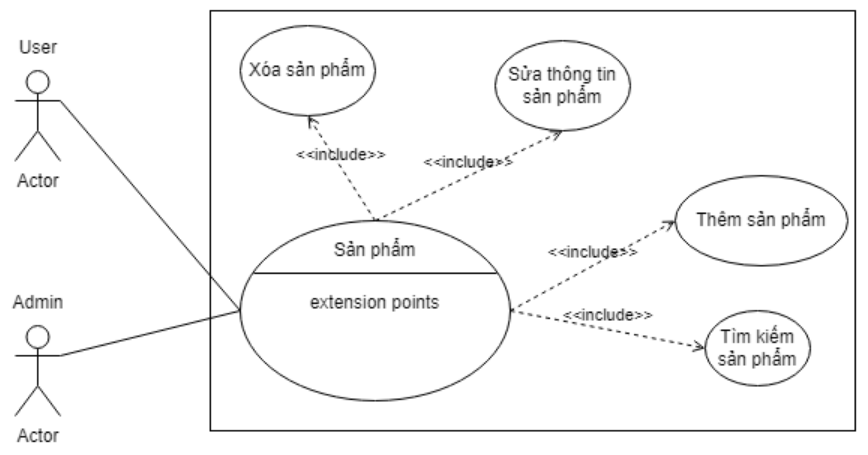
Sơ đồ 2.3.7. Phân rã Use Case Giỏ hàng

#### Phân rã Use Case Thông tin đơn hàng



Sơ đồ 2.3.8. Phân rã Use Case Thông tin đơn hàng

#### Phân rã Use Case Sản phẩm



Sơ đồ 2.3.9. Phân rã Use Case Sản phẩm

## Đặc tả Use Case

### Đăng ký

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#01** | **Đăng ký** | | **Độ phức tạp: Trung bình** |
| **Mô tả** | Cho phép người dùng, admin tạo tài khoản, để đăng nhập vào ứng dụng | | |
| **Tác nhân** | Người dùng, admin | | |
| **Tiền điều kiện** | - Tài khoản người dùng chưa tồn tại trong hệ thống | | |
| **Hậu diều kiện** | **Thành công** | Hệ thống đưa ra thông báo: Đăng ký thành công | |
| **Lỗi** | Hệ thống đưa ra thông báo: Tên đăng nhập đã tồn tại | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/ kịch bản chính** | | | |
| 1. Người dùng, admin truy cập vào ứng dụng  - Chọn đăng ký  - Người dùng, admin nhập tên đăng nhập (Username), mật khẩu (Password), họ tên, địa chỉ và số điện thoại  - Sau nó bấm nút đăng ký  2. Hệ thống kiểm tra thông tin vừa nhập  - Nếu tên đăng nhập (Username) chưa tồn tại, hệ thống đưa ra thông báo: Đăng ký thành công.  - Nếu tên đăng nhập (Username) đã tồn tại, hệ thống đưa ra thông báo: Tên đăng nhập đã tồn tại.  - Nếu có trường thông tin bị bỏ trống, hệ thống đưa ra thông báo: Hãy nhập thông tin cho tất cả các trường. | | | |
| **Luồng sự kiện phát sinh/ kịch bản phát sinh** | | | |

### Đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#02** | **Đăng nhập** | | **Độ phức tạp: Trung bình** |
| **Mô tả** | Cho phép người dùng, admin đăng nhập vào ứng dụng | | |
| **Tác nhân** | Người dùng, admin | | |
| **Tiền điều kiện** | Người dùng, admin có tài khoản | | |
| **Hậu diều kiện** | **Thành công** | Đăng nhập ứng dụng thành công | |
| **Lỗi** | Đăng nhập không thành công | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/ kịch bản chính** | | | |
| 1. Người dùng, admin truy cập vào ứng dụng  2. Nhập thông tin tên đăng nhập, mật khẩu  3. Sau nó bấm nút đăng nhập để truy cập vào ứng dụng  4. Hệ thống xác thực thông tin đăng nhập và cho phép người dùng truy cập ứng dụng. | | | |
| **Luồng sự kiện phát sinh/ kịch bản phát sinh** | | | |
| - Nếu tên đăng nhập hoặc mật khẩu không trùng khớp, hệ thống đưa ra thông báo: Tên đăng nhập hoặc mật khẩu không đúng | | | |

### Tìm kiếm

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#03** | **Tìm kiếm** | | **Độ phức tạp: Trung bình** |
| **Mô tả** | Cho phép người dùng tìm kiếm sản phẩm theo tên sản phẩm | | |
| **Tác nhân** | Người dùng | | |
| **Tiền điều kiện** | Người dùng đã đang ký tài khoản | | |
| **Hậu diều kiện** | **Thành công** | Hiển thị danh sách sản phẩm đúng với tên tìm kiếm | |
| **Lỗi** | Hệ thống thông báo: Không tìm thấy sản phẩm | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/ kịch bản chính** | | | |
| 1. Người dùng truy cập vào ứng dụng  2. Nhập tên đăng nhập, mật khẩu  3. Nhập tên sản phẩm  4. Click biểu tượng tìm kiếm  5. Hệ thống trả về danh sách sản phẩm theo tên tìm kiếm | | | |
| **Luồng sự kiện phát sinh/ kịch bản phát sinh** | | | |
| 1. Sản phẩm không tồn tại trong hệ thống | | | |

### Thông tin cá nhân

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#04** | **Thông tin cá nhân** | | **Độ phức tạp: Trung bình** |
| **Mô tả** | Cho phép người dùng kiểm tra và thay đổi thông tin cá nhân | | |
| **Tác nhân** | Người dùng | | |
| **Tiền điều kiện** | - Người dùng đã đang ký tài khoản  - Tài khoản đã được phân quyền | | |
| **Hậu diều kiện** | **Thành công** | Hiển thị thông tin cá nhân và thông tin mua hàng | |
| **Lỗi** |  | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/ kịch bản chính** | | | |
| 1. Người dùng truy cập vào ứng dụng  2. Nhập tên đăng nhập, mật khẩu  3. Click nút Thông tin cá nhân  4. Hệ thống hiển thị cửa sổ thông tin cá nhân | | | |
| **Luồng sự kiện phát sinh/ kịch bản phát sinh** | | | |
| Người dùng có thể sửa thông tin cá nhân | | | |

### Thanh toán

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#05** | **Thanh toán** | | **Độ phức tạp: Trung bình** |
| **Mô tả** | Cho phép người dùng thanh đoán đơn hàng | | |
| **Tác nhân** | Người dùng | | |
| **Tiền điều kiện** | - Người dùng đã đang ký tài khoản  - Có sản phẩm trong giỏ hàng | | |
| **Hậu diều kiện** | **Thành công** | Hệ thống hiển thị thông báo thanh toán thành công | |
| **Lỗi** | Không thể thanh toán | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/ kịch bản chính** | | | |
| 1. Người dùng truy cập vào ứng dụng  2. Nhập tên đăng nhập, mật khẩu  3. Thêm sản phẩm vào giỏ hàng  4. Click biểu tượng giỏ hàng  5. Click nút thanh toán  6. Hệ thống thực hiện thanh toán và lưu thông tin vào database | | | |
| **Luồng sự kiện phát sinh/ kịch bản phát sinh** | | | |
| Xảy ra lỗi và không thể thanh toán | | | |

### Giỏ hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#06** | **Giỏ hàng** | | **Độ phức tạp: Trung bình** |
| **Mô tả** | Cho phép người dùng kiểm tra những sản phẩm đã thêm vào giỏ hàng | | |
| **Tác nhân** | Người dùng | | |
| **Tiền điều kiện** | Người dùng đã đang ký tài khoản | | |
| **Hậu diều kiện** | **Thành công** | Thông tin sản phẩm trong giỏ hàng được hiển thị | |
| **Lỗi** | Thông tin sản phẩm không được hiển thị | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/ kịch bản chính** | | | |
| 1. Người dùng truy cập vào ứng dụng  2. Nhập tên đăng nhập, mật khẩu  3. Thêm sản phẩm vào giỏ hàng  4. Click biểu tượng giỏ hàng | | | |
| **Luồng sự kiện phát sinh/ kịch bản phát sinh** | | | |
| 1. Người dùng có thể thay đổi số lượng của từng sản phẩm hoặc xóa sản phẩm đó khỏi giỏ hàng  2. Người dùng có thể thực hiện thanh toán đơn hàng | | | |

### Thông tin đơn hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#07** | **Thông tin đơn hàng** | | **Độ phức tạp: Trung bình** |
| **Mô tả** | Cho phép người dùng kiểm tra thông tin đơn hàng của khách hàng | | |
| **Tác nhân** | người dùng | | |
| **Tiền điều kiện** | Người dùng đã có tài khoản | | |
| **Hậu diều kiện** | **Thành công** | Thông tin về số lượng sản phẩm và giá trị đơn hàng được hiển thị | |
| **Lỗi** |  | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/ kịch bản chính** | | | |
| 1. Người dùng truy cập vào ứng dụng  2. Nhập tên đăng nhập, mật khẩu  3. Click nút Thông tin cá nhân  4. Thông tin đơn hàng sẽ được hiển thị | | | |
| **Luồng sự kiện phát sinh/ kịch bản phát sinh** | | | |

### Sản phẩm

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#08** | **Sản phẩm** | | **Độ phức tạp: Trung bình** |
| **Mô tả** | - Người dùng có thể xem thông tin của tất cả sản phẩm và thêm sản phẩm vào giỏ hàng  - Admin có thể thêm mới sản phẩm, xóa sản phẩm hoặc sửa thông tin sản phẩm | | |
| **Tác nhân** | Người dùng, admin | | |
| **Tiền điều kiện** | - Người dùng đã có tài khoản  - Admin đã có tài khoản | | |
| **Hậu diều kiện** | **Thành công** | Thông tin của sản phẩm được hiển thị | |
| **Lỗi** |  | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/ kịch bản chính** | | | |
| 1. Người dùng/ admin truy cập vào ứng dụng  2. Nhập tên đăng nhập, mật khẩu  - Cửa sổ Window All Item Admin sẽ được hiển thị nếu là admin  - Cửa sổ WindowTrending sẽ hiển thị nếu là người dùng  - Người dùng click nút Tất cả sản phẩm  - Danh sách sản phẩm sẽ được hiển thị | | | |
| **Luồng sự kiện phát sinh/ kịch bản phát sinh** | | | |
|  | | | |

# FrontEnd

## Tạo dự án

B1: Tại màn hình NetBeans, click File, chọn New Project

A screenshot of a computer

Description automatically generated

B2: Chọn Java with Ant, chọn Java Application, click Next

A screenshot of a project

Description automatically generated

B3: Đặt tên cho Project, click Finish

A screenshot of a computer

Description automatically generated

Hiển thị giao diện sau khi click Finish

A screenshot of a computer

Description automatically generated

## Cài đặt thư viện JDBC kết nối tới Database

B1: Click chuột phải vào tên Project, chọn Properties

A screenshot of a computer program

Description automatically generated

B2: Tại cửa sổ Properties, chọn Libraries, click biểu tượng “+” Classpath, chọn Add Jar/Folder, click Ok

A screenshot of a computer

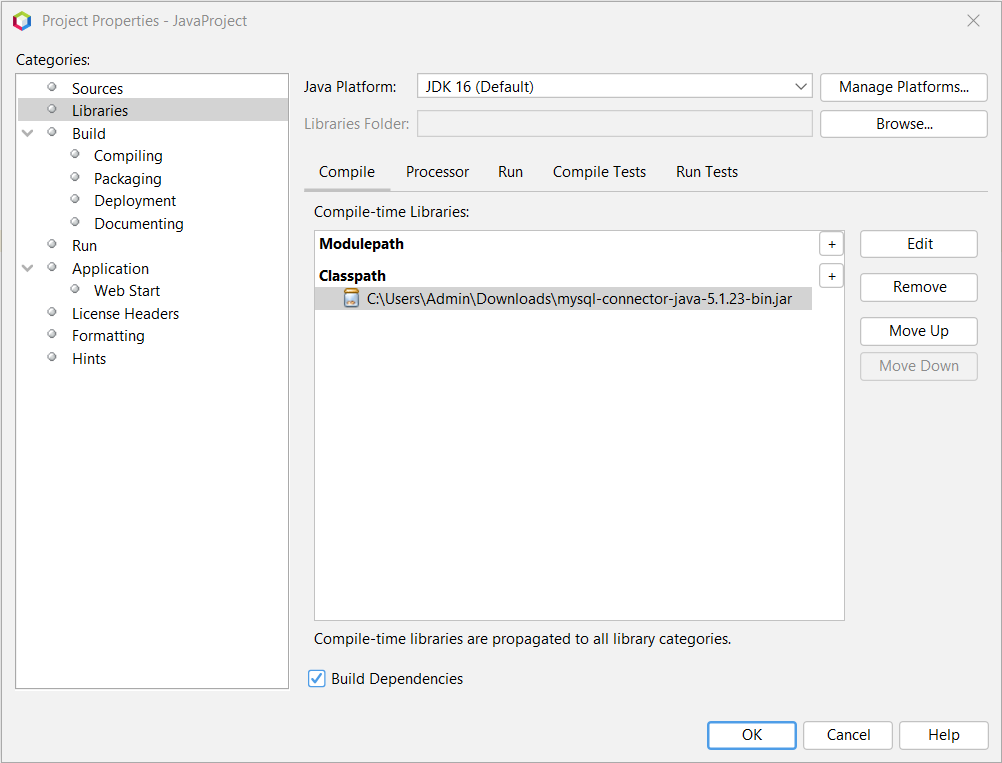
Description automatically generated

B3: Chọn file mysql-connector-java-5.1.23-bin.jar, click Open

A screenshot of a computer

Description automatically generated

File đã được thêm, click Ok



Thư viện JDBC đã được thêm vào Project

A screenshot of a computer

Description automatically generated

## Tạo form frmLogin

B1: Chuột phải vào tên Project, chọn New, chọn Java Package, đặt tên là form

B2: Chuột phải vào package form, chọn New, chọn Java Class, đặt tên là frmLogin.java

B3: Viết code cài đặt giao diện cho form Login

A screen shot of a computer code

Description automatically generated

Kết quả

A screenshot of a computer login window

Description automatically generated

## Tạo form frmAddItem

B1: Chuột phải vào package form, chọn New, chọn Jframe Form, đặt tên là frmAddItem.java

B2: Thiết kế giao diện cho form frmAddItem

Kết quả

A screenshot of a computer

Description automatically generated

## Tạo frm frmAllProductAdmin

B1: Chuột phải vào package form, chọn New, chọn Jframe Form, đặt tên là frmAllProductAdmin.java

B2: Thiết kế giao diện cho form frmAddItem

Kết quả

A screenshot of a computer

Description automatically generated

## Tạo frmAllProductClient

B1: Chuột phải vào package form, chọn New, chọn Jframe Form, đặt tên là frmAllProductClient.java

B2: Thiết kế giao diện cho form frmAllProductClient

Kết quả

A screenshot of a card

Description automatically generated

## Tạo form frmCart

B1: Chuột phải vào package form, chọn New, chọn Jframe Form, đặt tên là frmCart.java

B2: Thiết kế giao diện cho form frmCart

Kết quả

A screenshot of a computer

Description automatically generated

## Tạo form frmItemClient

B1: Chuột phải vào package form, chọn New, chọn Jframe Form, đặt tên là frmItemClient.java

B2: Thiết kế giao diện cho form frmItemClient

Kết quả

A screenshot of a computer

Description automatically generated

## Tạo form frmProfile

B1: Chuột phải vào package form, chọn New, chọn Jframe Form, đặt tên là frmProfile.java

B2: Thiết kế giao diện cho form frmProfile

Kết quả

A screenshot of a computer

Description automatically generated

## Tạo form frmRegister

B1: Chuột phải vào package form, chọn New, chọn Jframe Form, đặt tên là frmRegister.java

B2: Thiết kế giao diện cho form frmRegister

Kết quả

A screenshot of a login form

Description automatically generated

## Tạo form frmSearch

B1: Chuột phải vào package form, chọn New, chọn Jframe Form, đặt tên là frmSearch.java

B2: Thiết kế giao diện cho form frmSearch

Kết quả

A screenshot of a card

Description automatically generated

## Tạo form frmUpdateInfo

B1: Chuột phải vào package form, chọn New, chọn Jframe Form, đặt tên là frmUpdateInfo.java

B2: Thiết kế giao diện cho form frmUpdateInfo

Kết quả

A screenshot of a login form

Description automatically generated

## Tạo form frmUpdateItemInfoAdmin

B1: Chuột phải vào package form, chọn New, chọn Jframe Form, đặt tên là frmUpdateItemInfoAdmin.java

B2: Thiết kế giao diện cho form frmUpdateItemInfoAdmin

Kết quả

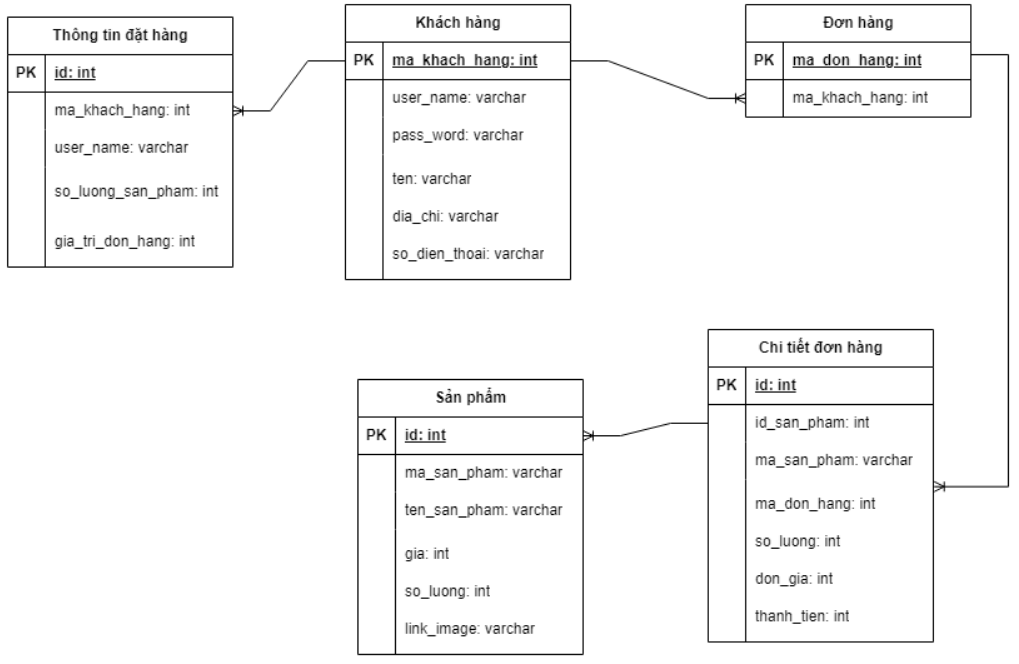
A screenshot of a computer

Description automatically generated

# Backend

## Thiết kế cơ sở dữ liệu (Database)

**Mô hình UML**



Sơ đồ 4.1.1. Mô hình UML

## Công nghệ sử dụng

1. MariaDB version 10.4.27

2. MySQL version 8.0.25

3. JDBC version 5.1.23

- Là API trong Java cho phép ứng dụng Java kết nối và tương tác với cơ sở dữ liệu.

4. NetBeans IDE version 16

5. Java version 16.0.2

## Cấu trúc thư mực

### Package DataBaseConnection

Package chứa file DataBaseConnection.java, file này chứa phương thức getConnection(), tạo kết nối tới cơ sở dữ liệu bằng JDBC.

A computer screen shot of text

Description automatically generated

### Package form

Chứa nhứng file .java - các form trong dự án.

### Package image

Chứa file file .png – các file ảnh sản phẩm được sử dụng trong dự án.

## Triển khai code

### Form frmLogin

#### Thư viện sử dụng:

A screen shot of a computer program

Description automatically generated

**A screenshot of a computer code

Description automatically generated**

A screenshot of a computer code

Description automatically generated



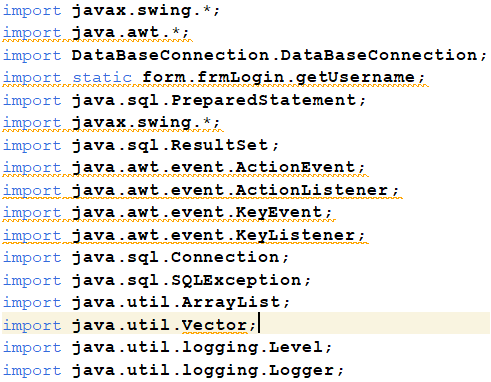
* static String username: Khởi tạo biến **usename**
* Từ khóa “static” giúp các form khác truy cập tới biến **username** mà không cần khai báo khai báo frmLogin
* Biến **username** lưu tên đăng nhập của người dùng
* String password: Khởi tạo biến **password** – lưu mật khẩu của người dùng

#### Phương thức

* **btnLogin**.**addActionListener()**: Kiểm tra thông tin ở TextField **tfUsername** và **ptPassword** khi người dùng click button Đăng nhập.
* Nếu **username** và **password** trùng khớp với dữ liệu trong database, hệ thống đưa ra thông báo: Đăng nhập thành công.
* Nếu **username** và **password** khôngtrùng khớp với dữ liệu trong database, hệ thống đưa ra thông báo: Tên đăng nhập hoặc mật khẩu không đúng.
* Nếu **username** là admin, hệ thống sẽ hiện thị form **frmAllProductAdmin**
* Nếu **username** không phải là admin, hệ thống sẽ hiển thị form **frmItemClient.**
* **btnLogin.addKeyListener()**: Thực hiện công việc tương tự phương thức **btnLogin**.**addActionListener()** khi người dùng nhấn phím Enter.

### Form frmItemClient

#### Thư viện sử dụng:



#### Thuộc tính:

A screenshot of a computer code

Description automatically generated

* **windowSearch**: Đối tượng frmSearch
* **listItemName**: Danh sách tên của từng sản phẩm trong database
* **listPrice**: Danh sách giá của từng sản phẩm trong database
* **listAmount**: Danh sách số lượng của từng sản phẩm trong database
* **listItemNameLowerCase**: Danh sách tên sản phẩm ở dạng không viết hoa
* Vector **result**: Danh sách tên của từng sản phẩm sau khi click button Search
* **listTtemAdded**: Danh sách sản phẩm đã được thêm vào giỏ hàng
* **countItem1**: Lưu số lần nhấn button btnAdd1
* **countItem2**: Lưu số lần nhấn button btnAdd2
* **countItem3**: Lưu số lần nhấn button btnAdd3
* **countItem4**: Lưu số lần nhấn button btnAdd4
* **countItem5**: Lưu số lần nhấn button btnAdd5
* **countItem6**: Lưu số lần nhấn button btnAdd6

#### Phương thức

**private void formWindowOpened(java.awt.event.WindowEvent evt)**:

Set thông tin cho 6 sản phẩm:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**private void btnCartActionPerformed(java.awt.event.ActionEvent evt)**

* Nếu sản phẩm được thêm vào giỏ hàng, hiển thị thông báo: Đã thêm sản phẩm vào giỏ hàng.
* Nếu sản phẩm vượt quá số lượng trong database, hiển thị thông báo: Số lượng sản phẩm vượt quá số lượng trong kho.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated

**private void btnSearchActionPerformed(java.awt.event.ActionEvent evt)**

Kiểm tra sản phẩm mà người dùng muốn tìm có trong database hay không

* Nếu có sản phẩm có trong database, hiển thị thông báo: Đã tìm thấy sản phẩm
* Nếu sản phẩm không có trong database, hiển thị thông báo: Không tìm thấy sản phẩm
* Nếu trường nhập thôn tin tên sản phẩm bị trống, hiển thị thông báo: Hãy nhập tên sản phẩm.

A screenshot of a computer program

Description automatically generated

**private void btnAdd1ActionPerformed(java.awt.event.ActionEvent evt)**

Tăng countItem1 lên 1 đơn vị mỗi lần click button btnAdd1.

A white background with black and white text

Description automatically generated

**private void btnAdd2ActionPerformed(java.awt.event.ActionEvent evt)**

Tăng countItem2 lên 1 đơn vị mỗi lần click button btnAdd2.

A screen shot of a computer code

Description automatically generated

**private void btnAdd3ActionPerformed(java.awt.event.ActionEvent evt)**

Tăng countItem3 lên 1 đơn vị mỗi lần click button btnAdd3.

A white background with black and blue text

Description automatically generated

**private void** **btnAdd4ActionPerformed(java.awt.event.ActionEvent evt)**

Tăng countItem4 lên 1 đơn vị mỗi lần click button btnAdd4.

A white box with black text

Description automatically generated

**private void** **btnAdd5ActionPerformed(java.awt.event.ActionEvent evt)**

Tăng countItem5 lên 1 đơn vị mỗi lần click button btnAdd5.

A white background with black text

Description automatically generated

**private void** **btnAdd6ActionPerformed(java.awt.event.ActionEvent evt)**

Tăng countItem6 lên 1 đơn vị mỗi lần click button btnAdd6.

A white background with black text

Description automatically generated

### Form frmSearch

#### Thư viện sử dụng

A screenshot of a computer program

Description automatically generated

#### Thuộc tính:

A screenshot of a computer code

Description automatically generated

* **itemName**: Tên sản phẩm
* **itemPrice**: Giá sản phẩm
* **itemAmount**: Số lượng sản phẩm
* **itemLinkImage**: Đường dẫn tới hình ảnh sản phẩm
* **listItem**: Danh sánh tên sản phẩm
* **listItemAdded**: Danh sách sản phẩm đã được thêm vào giỏ hàng
* **listItemAmount**: Danh sách số lượng sản phẩm
* **selectedRow**: Dòng được chọn trong DefaultTableModel
* **itemInTable**: Số lượng phần tử của listItem
* **itemCount**: Đếm số lượng của từng sản phẩm khi click button btnAddToCart
* Vector **vListPriceItem**: Danh sách giá của từng sản phẩm

**displayDetails(int selectIndex)**

Hiển thị dữ liệu trên label **lbItemName.**

A close up of text

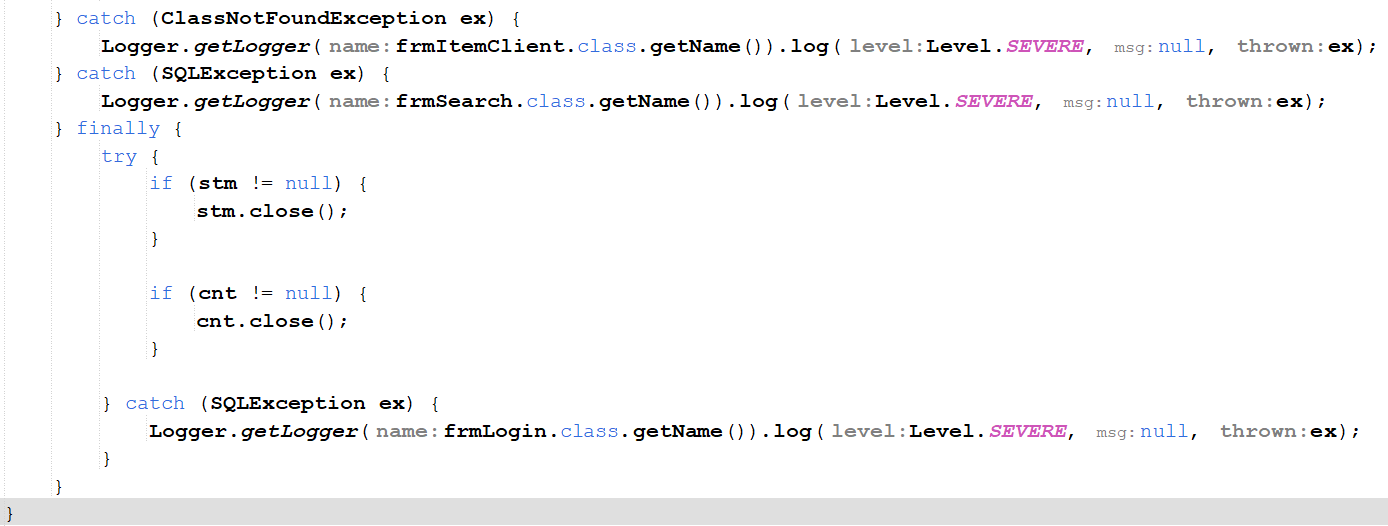
Description automatically generated

**private void formWindowOpened(java.awt.event.WindowEvent evt)**

Hiển thị dữ liệu trên **tbItem**, set giá trị cho **itemPrice** và thêm giá trị cho **vListPriceItem**, **listItemAmount** khi form được mở.

A screenshot of a computer code

Description automatically generated



**private void formWindowClosing(java.awt.event.WindowEvent evt)**

Thêm những sản phẩm đã được thêm vào giỏ hàng vào **listItemAdded** và set toàn bộ giá trị của **itemCount** bằng 0.

A screenshot of a computer program

Description automatically generated

**private void tbItemMouseClicked(java.awt.event.MouseEvent evt)**

Hiển thị thông tin về giá, số lượng và ảnh của sản phẩm khi click 1 dòng trong **tbItem.**

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer code

Description automatically generated

**private** **void** btnAddToCartActionPerformed(java.awt.event.ActionEvent evt)

Thêm sản phẩm trong danh sách kết quả tìm kiếm vào giỏ hàng khi click button **btnAddToCart.**

A screenshot of a computer

Description automatically generated

### Form frmCart

#### Thư viện sử dụng:

A screen shot of a computer code

Description automatically generated

#### Thuộc tính:

A computer screen shot of a code

Description automatically generated

* **selectedRow**: Chỉ số dòng hiện tại trong tbListItem đang được chọn
* **totalAmount**: Tổng số sản phẩm trong giỏ hàng
* **totalMoney**: Tổng số tiền của đơn hàng
* **itemName**: Tên sản phẩm
* **amount**: Số lượng của từng sản phẩm đang có trong giỏ hàng
* **itemLinkImage**: Đường dẫn hình ảnh của sản phẩm
* **dataTable**: Số dòng trong tbListItem
* **listItem**: Danh sách sản phẩm đã được thêm vào giỏ hàng
* **listItemAmount**: Danh sách số lượng sản phẩm lấy từ database

#### Phương thức

**private void displayDetails(int selectIndex)**

Hiển thị dữ liệu trong bảng **tbListItem** lên **tfAmount** khi click chuột vào 1 dòng trong bảng.

A screenshot of a computer code

Description automatically generated

**private void formWindowOpened(java.awt.event.WindowEvent evt)**

Thêm dữ liệu vào bảng **tbListItem**, set dữ liệu cho **lbTotalAmount** và **lbTotalMoney**, thêm dữ liệu vào **listItemAmount** khi form được mở.

A screenshot of a computer program

Description automatically generated

A computer screen shot of a computer screen

Description automatically generated

**private void tbListItemMouseClicked(java.awt.event.MouseEvent evt)**

Hiển thị hình ảnh của sản phẩm khi click vào 1 dòng trong bảng **tbListItem.**

A screen shot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

**private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt)**

Xóa 1 dòng trong bảng **tbListItem** khi click button **btnDelete**, sau đó cập nhật lại **lbTotalAmount** và **lbTotalMoney.**

A screenshot of a computer program

Description automatically generated

**private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt)**

Cập nhật lại số lượng sản phẩm tại dòng đang được chọn khi click button **btnUpdate.**

A screenshot of a computer

Description automatically generated

**private void btnPaymentActionPerformed(java.awt.event.ActionEvent evt)**

Thanh toán đơn hàng, sau đó cập nhật số lượng sản phẩm vào database, đồng thời in dữ liệu đơn hàng vào database sau khi click button **btnPayment.**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

### Form frmAllProductClient

#### Thư viện sử dụng

A screen shot of a computer program

Description automatically generated

#### Thuộc tính

A white background with black text

Description automatically generated

#### Phương thức

**private void displayDetails(int selectIndex)**

Set dữ liệu cho lbItemName.

A close up of text

Description automatically generated

**private void formWindowOpened(java.awt.event.WindowEvent evt)**

Set dữ liệu cho bảng **tbItemClient** và thêm dữ liệu cho **vListPriceItem**, **listItemAmount.**

A screen shot of a computer

Description automatically generated

A screen shot of a computer code

Description automatically generated

**private void tbItemClientMouseClicked(java.awt.event.MouseEvent evt)**

Hiển thị ảnh, giá tiền, số lượng của từng sản phẩm khi click 1 dòng trong **tbItemClient.**

A screenshot of a computer screen

Description automatically generated

**private void btnAddToCardActionPerformed(java.awt.event.ActionEvent evt)**

Thêm sản phẩm vào giỏ hàng

A screen shot of a computer

Description automatically generated

**private void formWindowClosing(java.awt.event.WindowEvent evt)**

Thêm dữ liệu cho listItemAdded và set tất cả giá trị của itemCount bằng 0 trước khi đóng form.

A screenshot of a computer program

Description automatically generated

### Form frmProfile

#### Thư viện sử dụng

A screen shot of a computer code

Description automatically generated

#### Phương thức

**private void btnChangeInfoActionPerformed(java.awt.event.ActionEvent evt)**

Hiển thị form **frmUpdateInfo** khi click button **btnChangeInfo**

A close up of a text

Description automatically generated with medium confidence

**private void formWindowOpened(java.awt.event.WindowEvent evt)**

Hiển thị thông tin của người dùng và set dữ liệu cho **tbHistory**

A screenshot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

### Form frmUpdateInfo

#### Thư viện sử dụng

A screen shot of a computer program

Description automatically generated

### Phương thức

**private void btnChangeInfoActionPerformed(java.awt.event.ActionEvent evt)**

Thực hiện update thông tin của người dùng khi click button **btnChangeInfo.**

A screenshot of a computer program

Description automatically generated

A computer screen shot of a computer screen

Description automatically generated

### Form frmRegister

#### Thư viện sử dụng

A screen shot of a computer program

Description automatically generated

#### Phương thức

**private void btnResActionPerformed(java.awt.event.ActionEvent evt)**

* Nếu tên đăng nhập đã tồn tại trong database, hệ thống đưa ra thông báo: Tên đăng nhập đã tồn tại
* Nếu tên đăng nhập chưa tồn tại trong database, hệ thống đưa ra thông báo: Đăng ký thành công

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### Form frmAllProductAdmin

#### . Thư viện sử dụng

A screen shot of a computer program

Description automatically generated

#### . Thuộc tính

A screenshot of a computer code

Description automatically generated

#### . Phương thức

**private void displayDetails(int selectIndex)**

Hiển thị ảnh của sản phẩm được chọn trong bảng **tbItem**

A close up of text

Description automatically generated

**private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt)**

Xóa dữ liệu tại dòng được chọn sau khi click button **btnDelete**

A screenshot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

**private void btnAddActionPerformed(java.awt.event.ActionEvent evt)**

Hiển thị form frmAddItem

A close-up of a computer code

Description automatically generated

**private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt)**

Hiển thị form frmUpdateItemInfoAdmin

A screen shot of a computer code

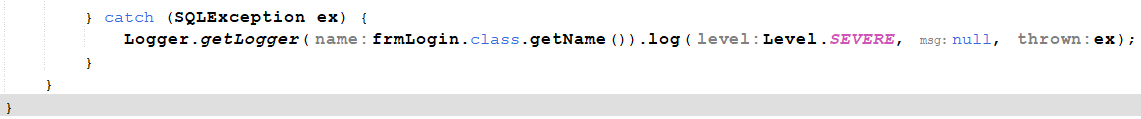
Description automatically generated

**private void formWindowOpened(java.awt.event.WindowEvent evt)**

Thêm dữ liệu vào bảng **tbItem**

A screenshot of a computer code

Description automatically generated



A screen shot of a computer code

Description automatically generated

### Form frmAddItem

#### . Thư viện sử dụng

A screen shot of a computer program

Description automatically generated

#### . Thuộc tính

A close up of text

Description automatically generated

#### . Phương thức

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

# Build App

B1: Click tên Project, click Clean and Build

A screenshot of a computer

Description automatically generated

B2: Theo đường dẫn sau khi build, ta sẽ có thể chạy ứng dụng



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated