BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC GIA ĐỊNH**

**KHOA: CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**TIỂU LUẬN**

**ĐỀ TÀI: GIỚI THIỆU ỨNG DỤNG BÁN QUẦN ÁO TRỰC TUYẾN**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **KỸ THUẬT PHẦN MỀM**

Giảng viên hướng dẫn: **LÊ HUỲNH PHƯỚC**

Sinh viên thực hiện: **LÊ TUẤN**

MSSV: **2108110124**

Lớp: **K15DCPM05**

*TP. Hồ Chí Minh, tháng 4 năm 2024*

**Khoa/Viện: Công nghệ thông tin**

**NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN**

**TIỂU LUẬN MÔN: LẬP TRÌNH CHO THIẾT BỊ DI ĐỘNG NÂNG CAO**

1. **Họ và tên sinh viên: Lê Tuấn**

2. **Tên đề tài: GIỚI THIỆU ỨNG DỤNG BÁN QUẦN ÁO TRỰC TUYẾN**

3. **Nhận xét**:

***Những kết quả đạt được:***

***Những hạn chế:***

4. **Điểm đánh giá** *(theo thang điểm 10, làm tròn đến 0.5):*

Sinh viên: Lê Tuấn

Điểm số: ……….…… Điểm chữ: ………………………………………………

|  |  |
| --- | --- |
|  | *TP. HCM, ngày … tháng … năm 20……*  **Giảng viên chấm thi**  *(Ký và ghi rõ họ tên)* |

**MỤC LỤC**

[CHƯƠNG 1 : GIỚI THIỆU ĐỂ TÀI 1](#_Toc164079389)

[**1. Đặt vấn đề** 1](#_Toc164079390)

[**2. Mục đích nghiên cứu** 1](#_Toc164079391)

[**3. Phương pháp nghiên cứu** 1](#_Toc164079392)

[**4. Phạm vi nghiên cứu** 1](#_Toc164079393)

[**5. Các bước thực hiện để hoàn thành dự án:** 2](#_Toc164079394)

[CHƯƠNG 2 : MÔ TẢ HỆ THỐNG 3](#_Toc164079395)

[**1. Phía server** 3](#_Toc164079396)

[***1.1. Ưu nhược điểm của API Getway* 4**](#_Toc164079397)

[***1.2. RESTful API* 4**](#_Toc164079398)

[***1.3. Các file APL ( thêm hình ảnh )* 5**](#_Toc164079399)

[**2. Phía Client** 5](#_Toc164079400)

[***2.1 Về các Adapter.* 5**](#_Toc164079401)

[***2.2. Về giao diện.* 6**](#_Toc164079402)

[***2.3. Về các Activity.* 6**](#_Toc164079403)

[**3. Mô hình USECASE.** 7](#_Toc164079404)

[***Hình 1: Use Case tổng quát. 7***](#_Toc164079405)

[***Hình 2: Use Case tìm kiếm. 8***](#_Toc164079406)

[***Hình 3: Use Case mua hàng. 9***](#_Toc164079407)

[***Hình 4: Use Case thêm sản phẩm vào giỏ hàng 9***](#_Toc164079408)

[**4. Đặc tả Usecase** 10](#_Toc164079409)

[***4.1. Usecase Search* 10**](#_Toc164079410)

[***4.2. Usecase View Product Detail* 10**](#_Toc164079411)

[***4.3. Usecase Cart Management* 11**](#_Toc164079412)

[**4.4. Usecase Payment 11**](#_Toc164079413)

[5**. Sơ đồ hoạt động** 12](#_Toc164079414)

[***Hình 5. Activity diagram Search* 12**](#_Toc164079415)

[***Hình 6. Activity diagram giỏ hàng* 12**](#_Toc164079416)

[***Hình 7: Activity diagram thanh toán* 13**](#_Toc164079417)

[***Hình 8: Activity diagram chi tiết sản phẩm* 13**](#_Toc164079418)

[**6. Sơ đồ tuần tự** 14](#_Toc164079419)

[***Hình 9: Sequence diagram Search* 14**](#_Toc164079420)

[***Hình 10: Sequence diagram quản lý giỏ hàng* 14**](#_Toc164079421)

[***Hình 11: Sequence diagram thanh toán* 15**](#_Toc164079422)

[CHƯƠNG 3: MÔ TẢ CƠ SỞ DỮ LIỆU 16](#_Toc164079423)

[**1. Giới thiệu về hệ cơ sở dữ liệu SQLite.** 16](#_Toc164079424)

[**2. Mô tả về cơ sở dữ liệu.** 17](#_Toc164079425)

[CHƯƠNG 4: GIAO DIỆN ỨNG DỤNG 18](#_Toc164079426)

[CHƯƠNG 5: KẾT LUẬN 25](#_Toc164079427)

[TÀI LIỆU THAM KHẢO 26](#_Toc164079428)

**LỜI MỞ ĐẦU**

Trong thời đại ngày nay, với sự phát triển không ngừng của công nghệ thông tin, thiết bị di động đóng vai trò ngày càng quan trọng trong mọi lĩnh vực của cuộc sống và công việc. Trong số các hệ điều hành di động, Android nổi bật như một hệ sinh thái mạnh mẽ được sử dụng rộng rãi trên nhiều thiết bị khác nhau.

Android không chỉ là một hệ điều hành cho điện thoại di động, mà còn là một nền tảng mở cho các ứng dụng và dịch vụ trên thiết bị di động. Với sức mạnh của mình, Android đã tạo ra một cuộc cách mạng trong việc truyền tải thông tin và kết nối con người trên khắp thế giới.

Trong bối cảnh hiện nay, khi mỗi người đều có thể trở thành một nhà phát triển ứng dụng, việc sở hữu một ứng dụng quản lý thông tin hoặc tiếp thị trên nền tảng di động không còn là điều xa xỉ. Một ứng dụng bán quần áo trên Android không chỉ mang lại sự thuận tiện cho người tiêu dùng trong việc tìm kiếm và mua sắm sản phẩm, mà còn là cơ hội kinh doanh tiềm năng cho các doanh nghiệp.

Nhóm chúng em nhận thấy tầm quan trọng của việc phát triển một ứng dụng bán quần áo trên Android không chỉ là cơ hội kinh doanh mà còn là sự đáp ứng đúng đắn và linh hoạt với nhu cầu mua sắm ngày càng đa dạng của người tiêu dùng.

Chính vì những lý do trên, nhóm em đã quyết định xây dựng một ứng dụng bán quần áo trên nền tảng Android.

Nội dung đề tài bao gồm 5 phần như sau:

Chương 1: Giới thiệu đề tài.

Chương 2: Mô tả hệ thống.

Chương 3: Mô tả cơ sở dữ liệu.

Chương 4: Màn hình giao diện.

Kết luận.

# CHƯƠNG 1 : GIỚI THIỆU ĐỂ TÀI

## **1. Đặt vấn đề**

Trong thời buổi hiện nay, mua sắm trực tuyến trở nên phổ biến hơn bao giờ hết, đặc biệt là trong lĩnh vực thời trang. Tuy nhiên, việc tìm kiếm và mua quần áo trực tuyến vẫn gặp nhiều khó khăn, từ việc tìm sản phẩm phù hợp đến quy trình thanh toán và chất lượng sản phẩm.

Để giải quyết vấn đề này, việc phát triển một ứng dụng bán quần áo trên nền tảng Android không chỉ mang lại trải nghiệm mua sắm thuận tiện hơn cho người dùng mà còn tạo ra cơ hội kinh doanh mới trong thị trường thương mại điện tử và thời trang trực tuyến.Top of Form

## **2. Mục đích nghiên cứu**

Xây dựng một ứng dụng bán quần áo trên nền tảng Android nhằm giải quyết nhu cầu mua sắm trực tuyến của người tiêu dùng, đặc biệt là trong bối cảnh cuộc sống hiện đại ngày càng bận rộn và diễn biến phức tạp.

Tạo ra một môi trường mua sắm trực tuyến tiện lợi và an toàn, giúp người dùng tiết kiệm thời gian và công sức khi mua sắm, đồng thời đảm bảo tính riêng tư và bảo mật thông tin của họ.

Nâng cao trải nghiệm người dùng thông qua việc cung cấp giao diện thân thiện, tính năng tìm kiếm thông minh, gợi ý sản phẩm phù hợp, và dịch vụ hỗ trợ khách hàng chuyên nghiệp.

Thúc đẩy phát triển thị trường thương mại điện tử trong lĩnh vực bán quần áo, đồng thời tạo ra cơ hội kinh doanh cho các nhà cung cấp và những người kinh doanh địa phương.

## **3. Phương pháp nghiên cứu**

Dựa trên những kiến thức về ngôn ngữ java lập trình hướng đối tượng đã học tại trường, xây dựng ứng dụng với công cụ hỗ trợ Android Studio mới nhất.

## **4. Phạm vi nghiên cứu**

Nghiên cứu tập trung vào phân tích nhu cầu và hành vi sử dụng ứng dụng bán quần áo trên nền tảng Android của người tiêu dùng. Nghiên cứu cũng xem xét các tính năng quan trọng của ứng dụng, hiệu quả kinh doanh, tính tương thích và đa dạng hóa trên các thiết bị và phiên bản Android khác nhau, cũng như khả năng tích hợp công nghệ mới để cải thiện trải nghiệm người dùng và tạo giá trị cho doanh nghiệp.

## **5. Các bước thực hiện để hoàn thành dự án:**

Phân tích yêu cầu: Xác định các chức năng và yêu cầu cụ thể của ứng dụng bán quần áo, bao gồm cả giao diện người dùng và các tính năng chức năng.

Thiết kế giao diện người dùng: Tạo các thiết kế giao diện người dùng sơ bộ dựa trên yêu cầu và nhu cầu của người dùng, bao gồm cả các màn hình chính và các phần tử giao diện như nút, hộp văn bản, danh sách và hình ảnh.

Phát triển chức năng: Phát triển các tính năng cơ bản của ứng dụng bao gồm tìm kiếm sản phẩm, xem chi tiết sản phẩm, thêm vào giỏ hàng, thanh toán.

Kết nối cơ sở dữ liệu: Tạo và quản lý cơ sở dữ liệu để lưu trữ thông tin về người dùng, sản phẩm, đơn hàng và các dữ liệu khác cần thiết cho ứng dụng.

Tối ưu hóa hiệu suất: Tối ưu hóa hiệu suất của ứng dụng bằng cách kiểm tra và điều chỉnh mã nguồn, tối ưu hóa truy vấn cơ sở dữ liệu và tối ưu hóa tốc độ tải trang.

Kiểm thử và sửa lỗi: Tiến hành kiểm thử tổng quát và kiểm tra các tính năng của ứng dụng để phát hiện và sửa lỗi, đảm bảo ứng dụng hoạt động một cách mượt mà và ổn định trước khi phát hành.

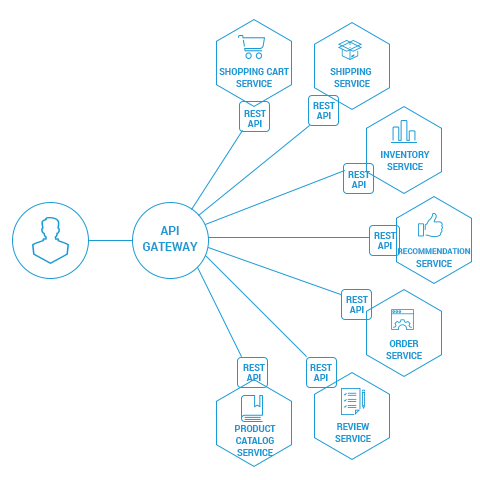
Hỗ trợ và cập nhật: Cung cấp hỗ trợ kỹ thuật và duy trì ứng dụng bằng cách cập nhật thường xuyên, sửa lỗi và cung cấp các tính năng mới để cải thiện trải nghiệm người dùng.

# CHƯƠNG 2 : MÔ TẢ HỆ THỐNG

## **1. Phía server**

API Gateway là một phương pháp hiệu quả trong việc tiếp cận kiến trúc microservices. Nó hoạt động như một điểm truy cập duy nhất vào hệ thống, tương tự như một lớp giao diện (Facade) trong thiết kế hướng đối tượng. API Gateway giúp che giấu chi tiết kiến trúc bên trong hệ thống và cung cấp các API tùy chỉnh cho mỗi client. Nó cũng có trách nhiệm về xác thực, giám sát, cân bằng tải, caching, định hình yêu cầu và quản lý thông tin, xử lý phản hồi tĩnh.

API Gateway thực hiện nhiệm vụ định tuyến, kết hợp và chuyển đổi các giao thức. Tất cả yêu cầu từ client đều phải đi qua API Gateway. Sau đó, API Gateway sẽ định tuyến các yêu cầu này tới các dịch vụ nhỏ phù hợp. Nó xử lý một yêu cầu bằng cách gọi đến một chuỗi các microservices và tổng hợp kết quả. API Gateway cũng có khả năng chuyển đổi giữa các giao thức web như HTTP, WebSocket và các giao thức nội bộ không thân thiện với web.



Hình 1: Sơ đồ hệ thống của API Getway.

### *1.1. Ưu nhược điểm của API Getway*

* Ưu điểm:
* Che dấu được cấu trúc của hệ thống microservices với bên ngoài.
* Phần code phía frontend sẽ gọn gàng hơn.
* Dễ dàng theo dõi và quản lý traffic.
* Requests caching và cân bằng tải.
* Thêm một lớp bảo mật nữa cho hệ thống.
* Thay thế authentication services.
* Nhược điểm:
* Tăng thời gian response.
* Thêm tác nhân gây lỗi.
* Có thể gây nghẽn cổ chai, quá tải, chậm hệ thống.
* Tốn thêm tiền để quản lý.

### *1.2.* *****RESTful API*****

RESTful API là một tiêu chuẩn được sử dụng trong thiết kế API cho các ứng dụng web, đặc biệt là trong việc quản lý các tài nguyên. Nó tập trung vào việc quản lý các tài nguyên hệ thống, bao gồm cả các loại tài nguyên như tệp văn bản, ảnh, âm thanh, video và dữ liệu động. RESTful API định dạng và truyền tải các trạng thái của các tài nguyên thông qua giao thức HTTP.



Hình 2: Quá trình hoạt động của Restfull API

REST (Representational State Transfer) là một kiểu kiến trúc được sử dụng để giao tiếp giữa các máy tính trên internet, bao gồm cả máy tính cá nhân và máy chủ của trang web, để quản lý các tài nguyên. REST sử dụng các phương thức biểu diễn khác nhau như text, JSON, XML, nhưng phổ biến nhất vẫn là JSON. Nó được sử dụng rộng rãi trong phát triển các dịch vụ Web sử dụng giao thức HTTP trên internet. Các ứng dụng phát triển theo kiểu RESTful thường tuân theo các nguyên tắc và quy tắc của REST.

Một đặc điểm quan trọng của dịch vụ Web RESTful là sử dụng các phương thức HTTP một cách rõ ràng theo các tiêu chuẩn được định nghĩa trong RFC 2616. Ví dụ, phương thức HTTP GET được sử dụng để truy xuất tài nguyên từ máy chủ, hoặc thực hiện một truy vấn và nhận phản hồi từ máy chủ.

REST đặt ra các yêu cầu cho nhà phát triển sử dụng phương thức HTTP một cách rõ ràng và tuân thủ các giao thức tiêu chuẩn. Nguyên tắc cơ bản của REST thiết lập một ánh xạ 1-1 giữa các hoạt động tạo, đọc, cập nhật và xóa (CRUD) và các phương thức HTTP tương ứng:

- Để truy xuất một tài nguyên, sử dụng phương thức GET.

- Để tạo một tài nguyên mới trên máy chủ, sử dụng phương thức POST.

- Để cập nhật trạng thái của một tài nguyên hoặc cập nhật nó, sử dụng phương thức PUT.

- Để xoá hoặc huỷ bỏ một tài nguyên, sử dụng phương thức DELETE.

### *1.3. Các file APL ( thêm hình ảnh )*

## **2. Phía Client**

### *2.1 Về các Adapter.*

Ứng với từng adapter sẽ đảm nhiệm các nhiệm vụ khác nhau:

|  |  |  |
| --- | --- | --- |
| STT | Util | Chức năng |
| 1 | SanphamAdapter | Lấy ra dữ liệu các đối tượng quần áo hiển thị danh sách quần áo ra Adapter View tương ứng. |
| 2 | GiohangAdapter | Gồm những chức năng có trong 1 giỏ hàng như tăng/giảm số sản phẩm muốn mua, cập nhật số lượng, tiền hàng,… |
| 3 | ShowDialog | Hiển thị các hộp thoại modal cho các tác vụ. |

### *2.2. Về* ***giao*** *diện.*

Mỗi layout sẽ hiển thị giao diện phù họp với mục đích khởi tạo của nó.

|  |  |  |
| --- | --- | --- |
| STT | Giao diện | Chức năng |
| 1 | activity\_detail | Hiển thị chi tiết về một sản phẩm cụ thể |
| 2 | activity\_giohang | Hiển thị nội dung của giỏ hàng, bao gồm các sản phẩm đã thêm vào. |
| 3 | activity\_main | Layout file chính được sử dụng để định nghĩa giao diện của màn hình chính (activity chính) |
| 4 | one\_giohang | Giao diện của trang giỏ hàng |
| 6 | one\_item | Giao diện của một mục sản phẩm trong giỏ hàng |
| 7 | payment | Giao diện cho trang thanh toán trong ứng dụng |
| 8 | dialog | Giao diện thông báo |

Bảng 1: Chức năng của các Layouts

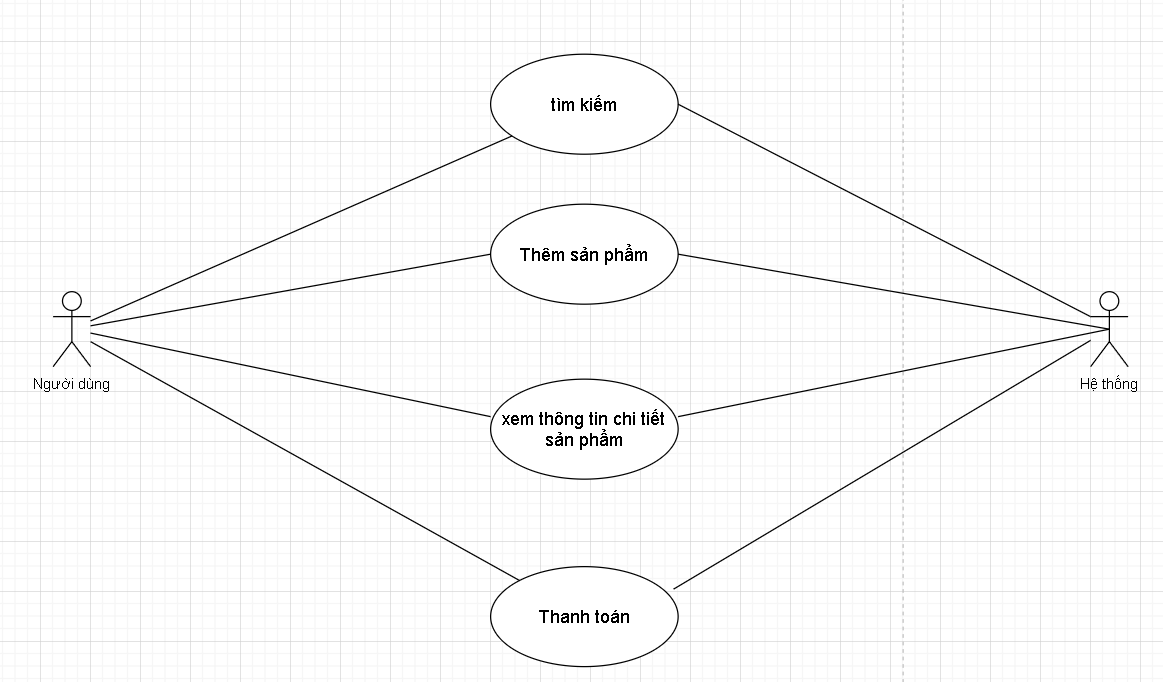
### *2.3. Về các Activity.*

Mỗi Activity sẽ là cầu nối để lấy dữ liệu từ cơ sở dữ liệu truyền vào phần giao diện.

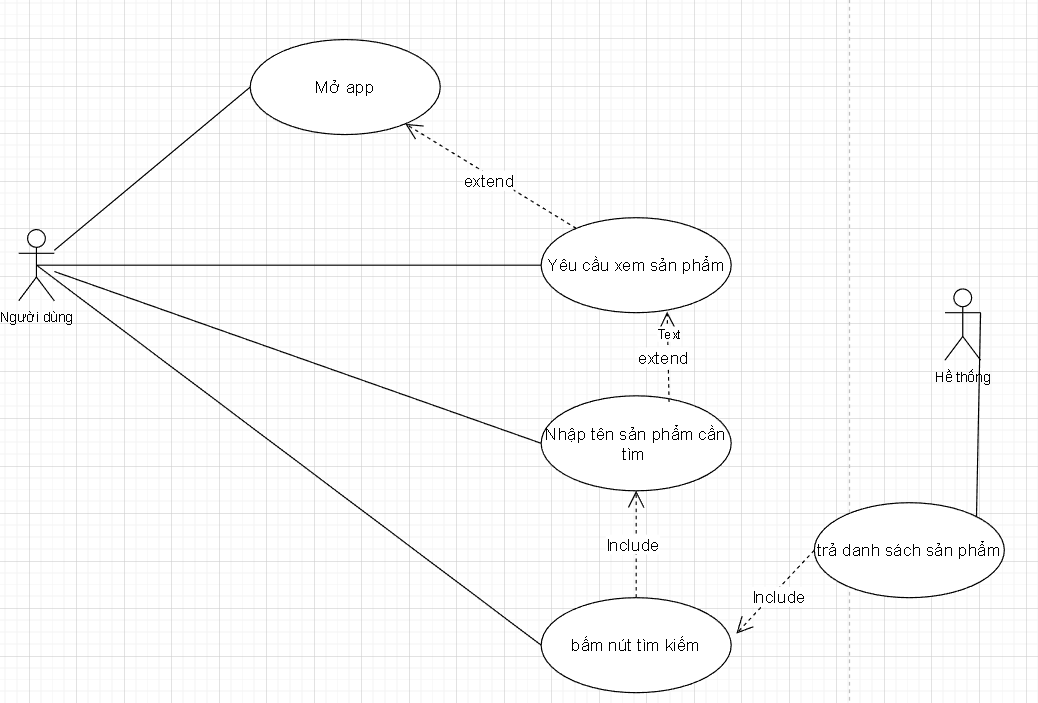
|  |  |  |
| --- | --- | --- |
| STT | Chức năng | Mục đích |
| 1 | Detail Activity | Xử lý chức năng hiện thị thông tin chi tiết về điện thoại |
| 2 | GioHangActivity | Xử lý chức năng trong giỏ hàng và thanh toán |
| 3 | MainActivity | Xử lí các sự kiện chính ở trang chủ |

Bảng 2: Chức năng của các Activitys

## **3.** **Mô hình USECASE.**



Hình 1: Use Case tổng quát.



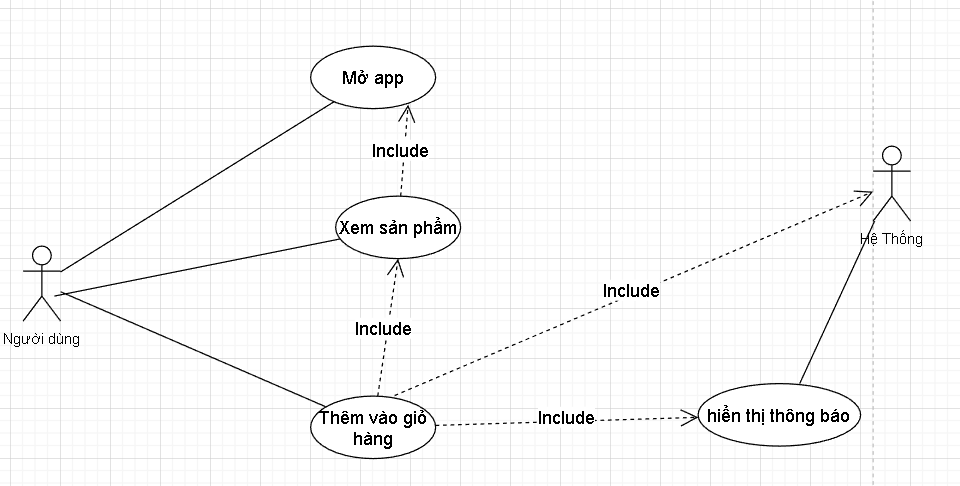
Hình 2: Use Case tìm kiếm.

A diagram of a diagram

Description automatically generated

Hình 3: Use Case mua hàng.

.



Hình 4: Use Case thêm sản phẩm vào giỏ hàng

## **4. Đặc tả Usecase**

### *4.1. Usecase Search*

Search:

- Tên use case: Tìm kiếm sản phẩm.

- Mô tả use case: Use case cho phép tìm kiếm và xem sản phẩm.

- Actor chính: User.

- Actor phụ: không có.

- Tiền điều kiện: User truy cập vào trang chủ.

- Hậu điều kiện: Xem được sản phẩm tìm kiếm.

- Luồng hoạt động:

1. Vào trang chủ.

2. Nhập thông tin và nhấn tìm kiếm.

3. Hệ thống xử lý.

4. Hiển thị kết quả tìm kiếm.

- Luồng thay thế: Không có.

- Luồng ngoại lệ: Ở bước 3, hệ thống kiểm tra không có sản phẩm, hiện thông báo chưa có sản phẩm đó.

### *4.2. Usecase View Product Detail*

Tên use case: Xem chi tiết sản phẩm.

- Mô tả use case: use case này mô tả hành động xem thông tin chi tiết của sản phẩm.

- Actor chính: user.

- Actor phụ: không có.

- Tiền điều kiện: user chọn sản phẩm cần xem.

- Hậu điều kiện: Hiển thị thông tin sản phẩm được chọn.

- Luồng hoạt động:

Chọn sản phẩm.

Hiển thị thông tin chi tiết sản phẩm.

- Luồng thay thế: Không có

- Luồng ngoại lệ: không có.

### *4.3. Usecase Cart Management*

Tên use case: Quản lý giỏ hàng.

- Mô tả use case: use case này mô tả quản lý giỏ hàng bằng cách thay đổi xóa sản phẩm.

- Actor chính: user.

- Actor phụ: không có.

- Tiền điều kiện: user đăng nhập bằng tài khoản của với quyền user.

- Hậu điều kiện: Giỏ hàng được cập nhật.

- Luồng hoạt động:

1. Chọn sản phẩm muốn thay đổi hoặc xóa.

2. Thực hiện thay đổi hoặc xóa.

3. Xác nhận thay đổi hoặc xóa sản phẩm.

4. Cập nhật lại giỏ hàng.

- Luồng thay thế: Không có.

- Luồng ngoại lệ: Ở bước 3, hệ thống kiểm tra yêu cầu, nếu không đồng ý giỏ hàng sẽ không có thay đổi.

### 4.4. Usecase Payment

Tên use case: Thanh toán.

- Mô tả use case: use case này mô tả hành động thanh toán sản phẩm của user.

- Actor chính: user.

- Actor phụ: không có.

- Tiền điều kiện: user đăng nhập bằng tài khoản của với quyền user.

- Hậu điều kiện: Thanh toán sản phẩm thành công.

- Luồng hoạt động:

1. Chọn giỏ hàng.

2. Chọn thanh toán.

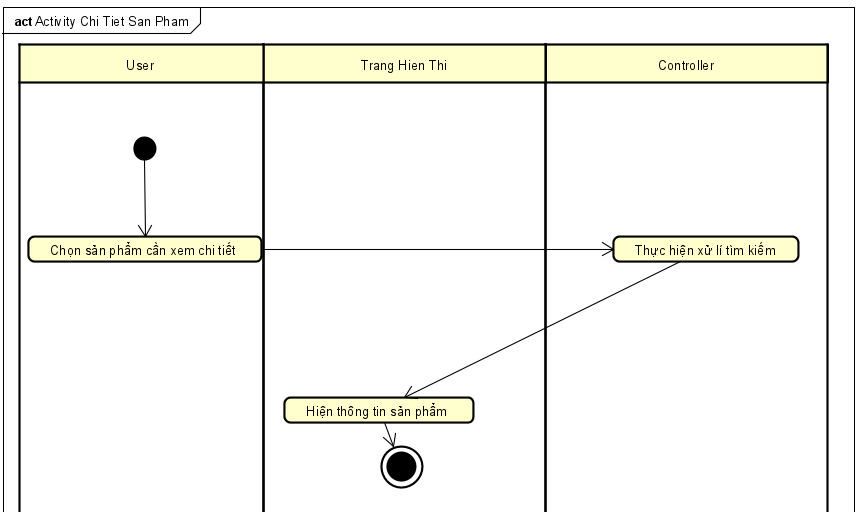
3. Xác nhận thanh toán.

4. Thông báo thanh toán thành công.

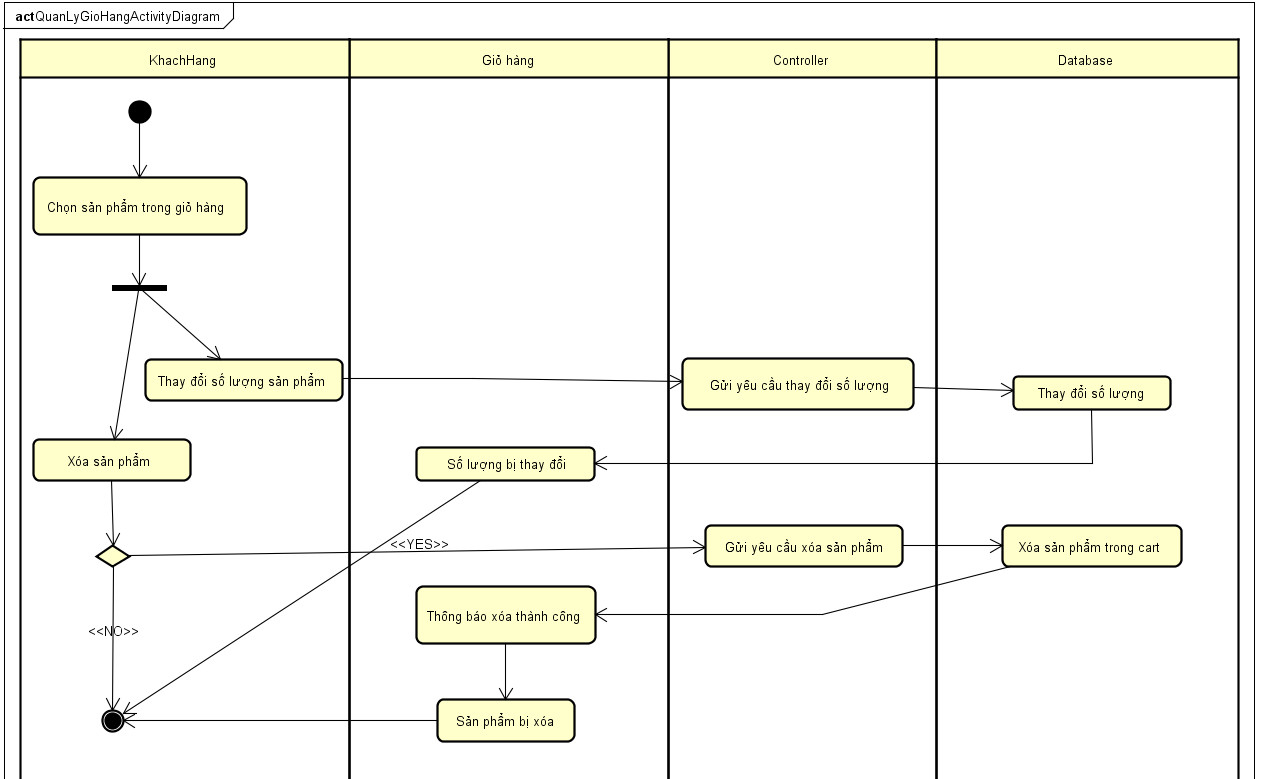
- Luồng thay thế: Không có.

- Luồng ngoại lệ: Ở bước 3, hệ thống xác nhận thanh toán, nếu không đồng ý thì sản phẩm trong giỏ không được thanh toán.

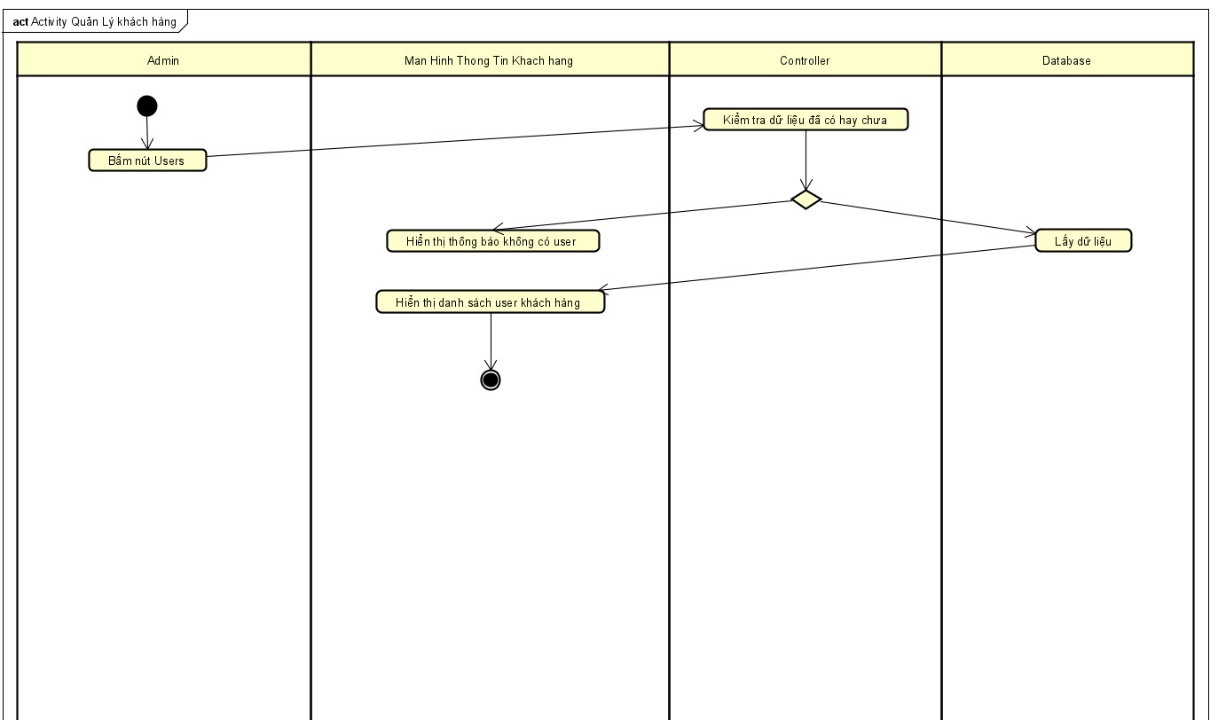
## 5**. Sơ đồ hoạt động**



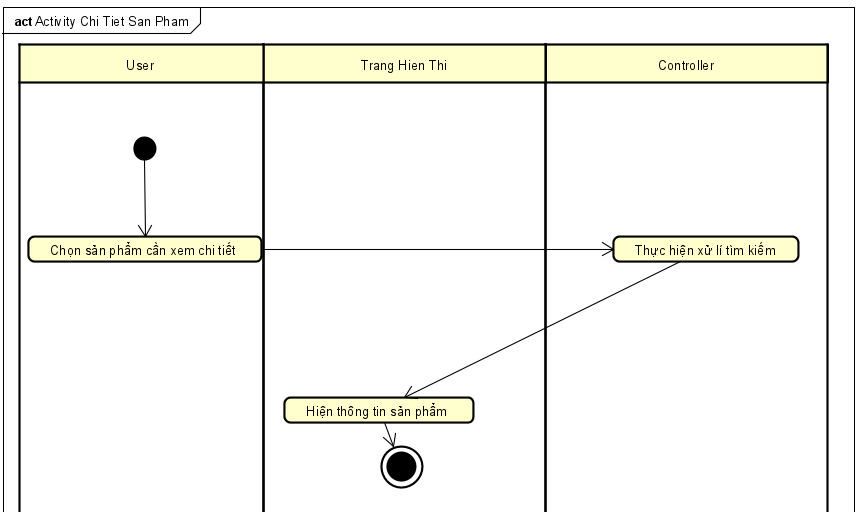
### *Hình 5. Activity diagram Search*



### *Hình 6. Activity diagram giỏ hàng*

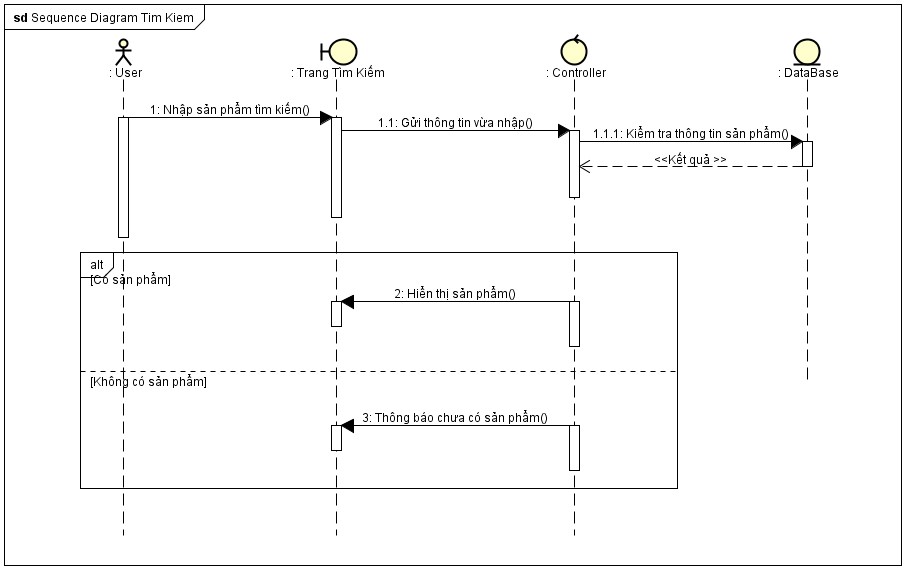


### *Hình 7: Activity diagram thanh toán*

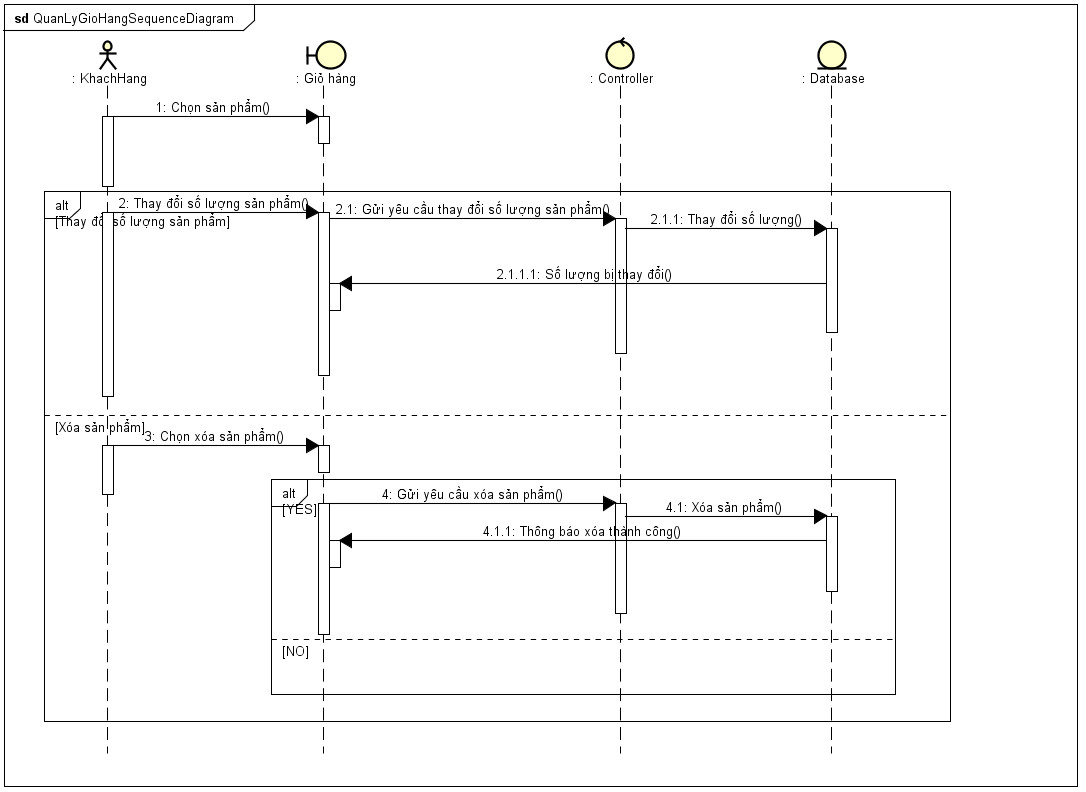


### *Hình 8: Activity diagram chi tiết sản phẩm*

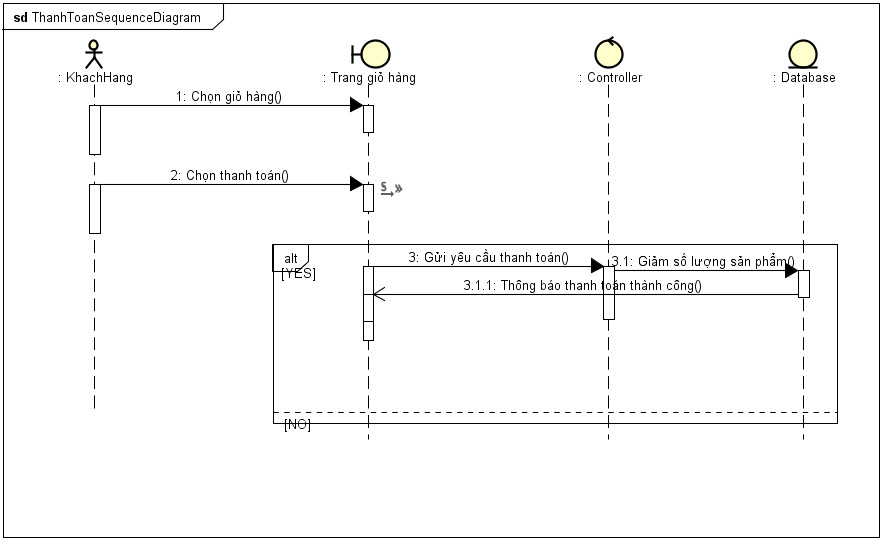
## **6. Sơ đồ tuần tự**

****

### *Hình 9: Sequence diagram Search*



### *Hình 10: Sequence diagram quản lý giỏ hàng*



### *Hình 11: Sequence diagram thanh toán*

# CHƯƠNG 3: MÔ TẢ CƠ SỞ DỮ LIỆU

## **1. Giới thiệu về hệ cơ sở dữ liệu SQLite.**

SQLite là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở nhẹ và linh hoạt, được phát triển bởi Dwayne Richard Hipp vào năm 2000. Đặc điểm nổi bật của SQLite là nó không yêu cầu một máy chủ riêng biệt, mà dữ liệu được lưu trữ trong một tệp duy nhất, thường là một tệp với đuôi ".sqlite" hoặc ".db". Điều này làm cho SQLite trở thành một lựa chọn phổ biến cho các ứng dụng nhỏ, ứng dụng di động và các trình phát triển phần mềm cần tích hợp một cơ sở dữ liệu nhẹ vào ứng dụng của họ mà không cần triển khai một máy chủ cơ sở dữ liệu riêng lẻ.

Một số tính năng chung thường được sử dụng trênSQLite:

- Hỗ trợ SQL chuẩn: SQLite hỗ trợ một loạt các câu lệnh SQL chuẩn, bao gồm SELECT, INSERT, UPDATE, DELETE, và các loại câu lệnh khác như JOIN và GROUP BY.

- Giao diện nhúng đơn giản: SQLite cung cấp một giao diện lập trình ứng dụng (API) đơn giản và dễ sử dụng cho nhiều ngôn ngữ lập trình khác nhau, bao gồm C/C++, Python, Java, và nhiều ngôn ngữ khác.

- Không cần máy chủ: Dữ liệu trong SQLite được lưu trữ trong một tệp duy nhất, không cần một máy chủ cơ sở dữ liệu riêng biệt. Điều này làm cho việc triển khai và quản lý dễ dàng hơn.

- Giao thức ACID: SQLite đảm bảo tính nhất quán dữ liệu bằng cách thực hiện các giao thức ACID (Atomicity, Consistency, Isolation, Durability), giúp đảm bảo rằng các thay đổi dữ liệu sẽ được thực hiện một cách an toàn và đáng tin cậy.

- Khóa và đồng bộ hóa: SQLite hỗ trợ các khóa và đồng bộ hóa để đảm bảo tính nhất quán của dữ liệu khi có nhiều kết nối cùng thao tác trên cơ sở dữ liệu.

- Chỉ số và ràng buộc: SQLite cho phép tạo chỉ mục (index) để cải thiện hiệu suất truy vấn, cũng như ràng buộc (constraints) để đảm bảo tính toàn vẹn dữ liệu.

- Backup và phục hồi dữ liệu: SQLite cung cấp các công cụ và tính năng cho việc sao lưu và phục hồi dữ liệu, giúp đảm bảo an toàn cho dữ liệu trong trường hợp sự cố xảy ra.

- Phiên bản hoạt động đa nền tảng: SQLite hoạt động trên nhiều nền tảng hệ điều hành khác nhau, bao gồm Windows, macOS, Linux, iOS, và Android

Tuy SQLite có nhiều ưu điểm, nhưng cũng có một số hạn chế, bao gồm hạn chế về đa người dùng và không phù hợp cho các ứng dụng có yêu cầu về tải cao hoặc dữ liệu lớn. Tuy nhiên, với các ứng dụng di động, máy tính cá nhân và các ứng dụng nhỏ, SQLite vẫn là một lựa chọn phổ biến và mạnh mẽ.

## **2. Mô tả về cơ sở dữ liệu.**

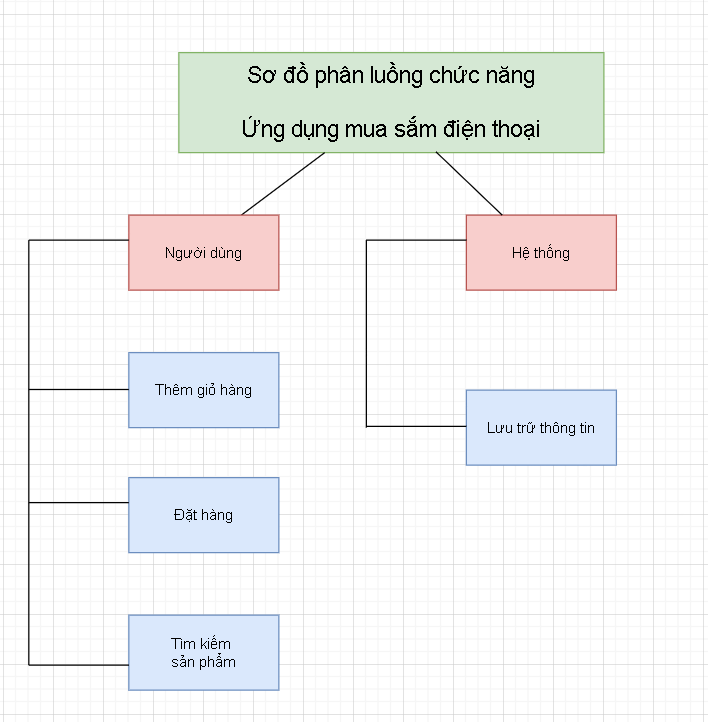
Ứng dụng mua sắm điện thoại của nhóm em được xây dựng trên hệ quản trị cơ sở dữ liệu MySQL với công cụ xử lý các tác vụ là phpMyAdmin. Hệ cơ sở dữ liệu của nhóm em được nhóm thiết kế và phân tầng theo từng đối tượng chịu sự quản lý có trong ứng dụng.

Mô tả:

|  |  |  |
| --- | --- | --- |
| STT | Bảng | Mục đích |
| 1 | giohang | Lưu trữ thông tin giỏ hàng |

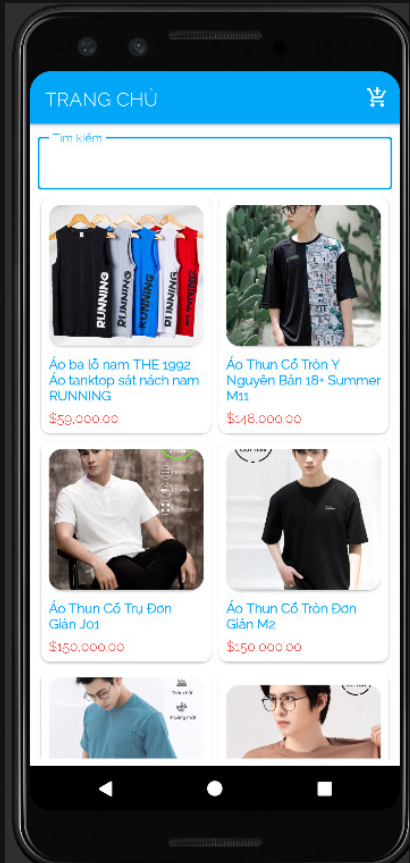
Bảng 3: Các bảng trong cơ sở dữ liệu

# CHƯƠNG 4: GIAO DIỆN ỨNG DỤNG



Hình 12: Sơ đồ phân lường chức năng ứng dụng.

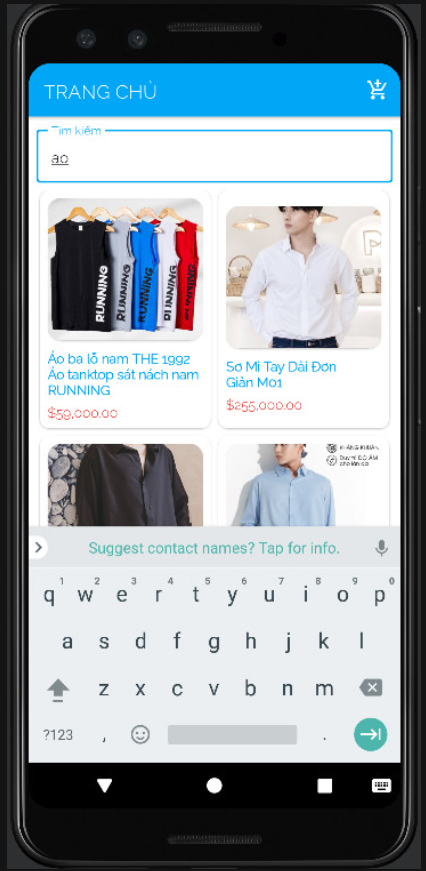
* Màn hình giao diện trang chủ.



Hình 1: Giao diện trang chủ

Màn hình giao diện trang chủ được sắp xếp theo bố cục hợp lý và hiện đại, Màn hình trang chủ như là một menu điều hướng người dùng tới tất cả những chức năng mà người dùng muốn thực hiện như: tìm kiếm, xem danh sách sản phẩm, xem giỏ hàng, …

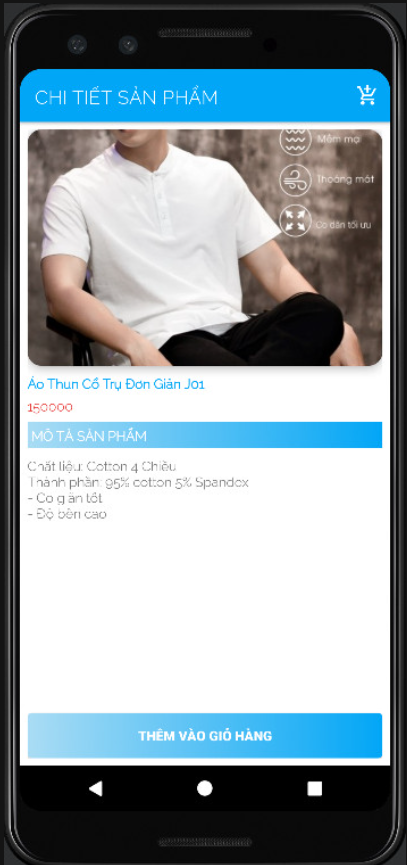
* Màn hình giao diện tìm kiếm và kết quả



Hình 2: Giao diện tìm kiếm sản phẩm.

Chức năng tìm kiếm được tích hợp ngay phía trên cùng của màn hình giao diện, để sử dụng chức năng tìm kiếm, người dùng chỉ cần nhập vào thông tin của sản phẩm muốn tìm và chọn tìm kiếm. Ngay lập tức, hệ thống sẽ trả về kết quả của sản phẩm người dùng muốn tìm và cả những sản phẩm có chứa từ khóa do người dùng nhập vào.

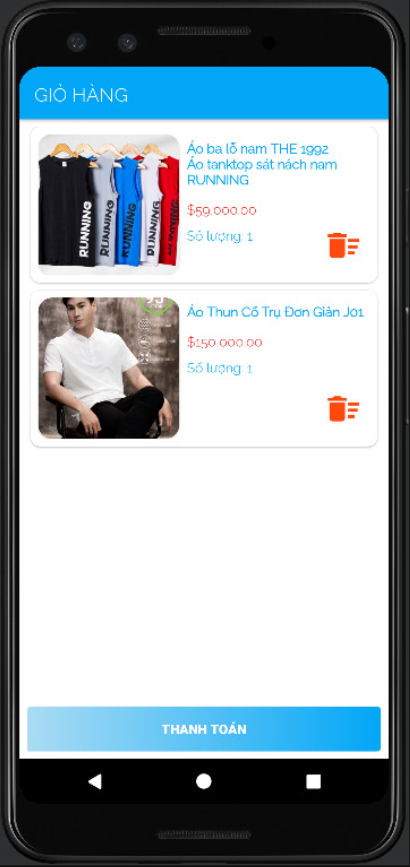
* Màn hình giao diên chi tiết sản phẩm.



Hình 3: Giao diện chi tiết sản phẩm

Mỗi sản phẩm đều có những thông tin thuộc tính khác nhau vì vậy để người dùng có cái nhìn tiệm cận nhất tới tất cả sản phẩm, thì người dùng chỉ cần bấm vào từng sản phẩm có trong danh sách sản phẩm. Với hệ thống được thiết kế kĩ lưỡng thì việc sản phẩm trả về đầy đủ và khớp vào những trường dữ liệu đã được thiết kế trước có mức độ chính xác gần như tuyệt đối. Trong chi tiết sản phẩm bao gồm thông tin chi tiết của sản phẩm kèm thoa nút thêm sản phẩm vào giỏ hàng và trở về danh sách sản phẩm.

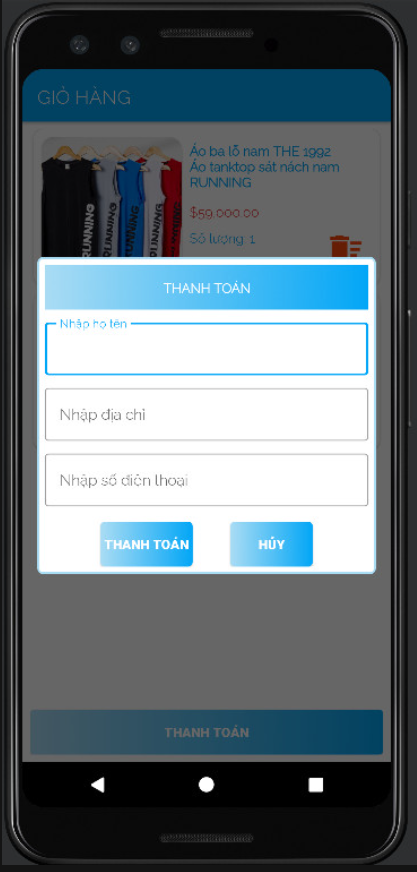
* Màn hình giao diện giỏ hàng.



Hình 4: Giao diện giỏ hàng

Phần lớn chức năng trong giỏ hàng đều ứng với từng sản phẩm, có thể kế đến như: tăng, giảm số lượng và xóa khỏi giỏ hàng.. Ngoài ra, còn có 1 số chức năng như mua hàng, trở về màn hình trang chủ và trở về danh sách sản phẩm.

* Màn hình giao diện thanh toán.



Hình 5: Giao diện thanh toán

Màn hình giao diện thanh toán, là màn hình kết thúc chuỗi quá trình mua hàng của người dùng. Tại đây người dùng bắt buộc phải nhập những thông tin liên hệ cũng như địa chỉ để đội ngũ giao hàng có thể thực hiện giao hàng tới tay người dùng. Sau khi điền đúng và đủ thông tin thì người dùng cần bấm vào nút thanh toán, sau một khoảng thời gian thì nhân viên của hệ thống sẽ gọi điện để xác nhận đơn hàng của người dùng. Ngoài ra còn có nút hủy bỏ nếu người dùng cảm thấy không thích sản phẩm cũng như cháy túi.

# CHƯƠNG 5: KẾT LUẬN

1. **Kết quả đạt được**

- Đáp ứng được nhu cầu sử dụng cơ bản của người dùng.

- Giao diện dễ sử dụng, bắt mắt người dùng.

- Áp dụng được các phương pháp thiết kế giao diện hiện đại, phối hợp các hiệu ứng động giúp ứng dụng bắt mắt và sinh động.

- Sử dụng thành thạo và hiệu quả cơ sở dữ liệu SQLite để lưu trữ dữ liệu cho ứng dụng.

**2. Hướng phát triển**

- Cho phép người dùng đăng nhập thông qua Google, Facebook, hoặc qua số điện thoại.

- Thêm các chức năng cho phép người dùng đánh giá sản phẩm, phản hồi thông tin về các sản phẩm cho nhà bán hàng.

# TÀI LIỆU THAM KHẢO

1. Android Studio Tutorial, [Learn Android Tutorial | Android Studio Tutorial - Javatpoint](https://www.javatpoint.com/android-tutorial)
2. Android Tutorial, [Android Tutorial (tutorialspoint.com)](https://www.tutorialspoint.com/android/index.htm)
3. Developer Guides, [Developer guides  |  Android Developers](https://developer.android.com/guide)
4. Android, <https://www.journaldev.com/android>
5. SQLite Tutorial With Example In Android Studio, [SQLite Tutorial With Example In Android Studio | Android Database Tutorial In Android Studio: Store Your Data (abhiandroid.com)](https://abhiandroid.com/database/sqlite#gsc.tab=0)
6. Android SQLite Database Tutorial, <https://material.io/components?platform=android>
7. Material Design Components, [Components – Material Design 3](https://m3.material.io/components)