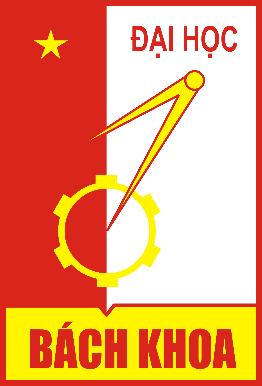
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**----- 🙡 🕮 🙣 -----**



**PROJECT II**

***Đề tài:* Phát triển ứng dụng mua hàng trên Android**

Giảng viên: **TS Nguyễn Đức Toàn**

Sinh viên thực hiện: Cao Văn Thiện- 20204853

Hà Nội, năm 2023

Mục lục

[**I.** **Tìm hiểu về lập trình Android** 3](#_Toc140799897)

[**1.** **Tìm hiểu về lập trình Android.** 3](#_Toc140799898)

[**2.** **Ôn tập lại một số kiến thức.** 3](#_Toc140799899)

[**3.** **Cài đặt môi trường.** 3](#_Toc140799900)

[**4.** **Học sử dụng Android Studio** 4](#_Toc140799901)

[**II.** **Phân tích chức năng và cơ sở dữ liệu cho ứng dụng.** 5](#_Toc140799902)

[**1.** **Chức năng bên phía người dùng.** 5](#_Toc140799903)

[**2.** **Phân tích cơ sở dữ liệu** 7](#_Toc140799904)

[**3.** **Thiết kế giao diện** 9](#_Toc140799905)

[**III.** **Sử dụng Android Studio lập trình ứng dụng** 19](#_Toc140799906)

[**1.** **Tạo Project trên ứng dụng Android studio** 19](#_Toc140799907)

[**2.** **Tạo Các Fragment chính của ứng dụng** 23](#_Toc140799908)

[**3.** **Chi tiết đơn hàng.** 31](#_Toc140799909)

[**4.** **Phần giỏ hàng** 33](#_Toc140799910)

[**5.** **Phần thanh toán cho sản phẩm** 37](#_Toc140799911)

[**6.** **Phần hồ sơ người dùng** 40](#_Toc140799912)

[**7.** **Phần thêm vào giỏ hàng** 49](#_Toc140799913)

[**8.** **Phần lịch sử mua hàng** 55](#_Toc140799914)

[**9.** **Xử lí đặt hàng** 62](#_Toc140799915)

[**IV.** **Tổng kết** 64](#_Toc140799916)

[**1.** **Mục tiêu.** 64](#_Toc140799917)

[**2.** **Tóm tắt ứng dụng** 64](#_Toc140799918)

[**3.** **Đường dẫn sản phẩm.** 64](#_Toc140799919)

# **Tìm hiểu về lập trình Android**

## **Tìm hiểu về lập trình Android.**

- Android là một hệ điều hành dựa trên nền tảng Linux và được phát triển bởi Google. Nó được sử dụng phổ biến trên các thiết bị di động như điện thoại thông minh và máy tính bảng. Android cung cấp một loạt các tính năng và công nghệ để phát triển ứng dụng di động, bao gồm Java và Kotlin để viết mã, XML để thiết kế giao diện người dùng, cùng với một loạt các thư viện hỗ trợ.

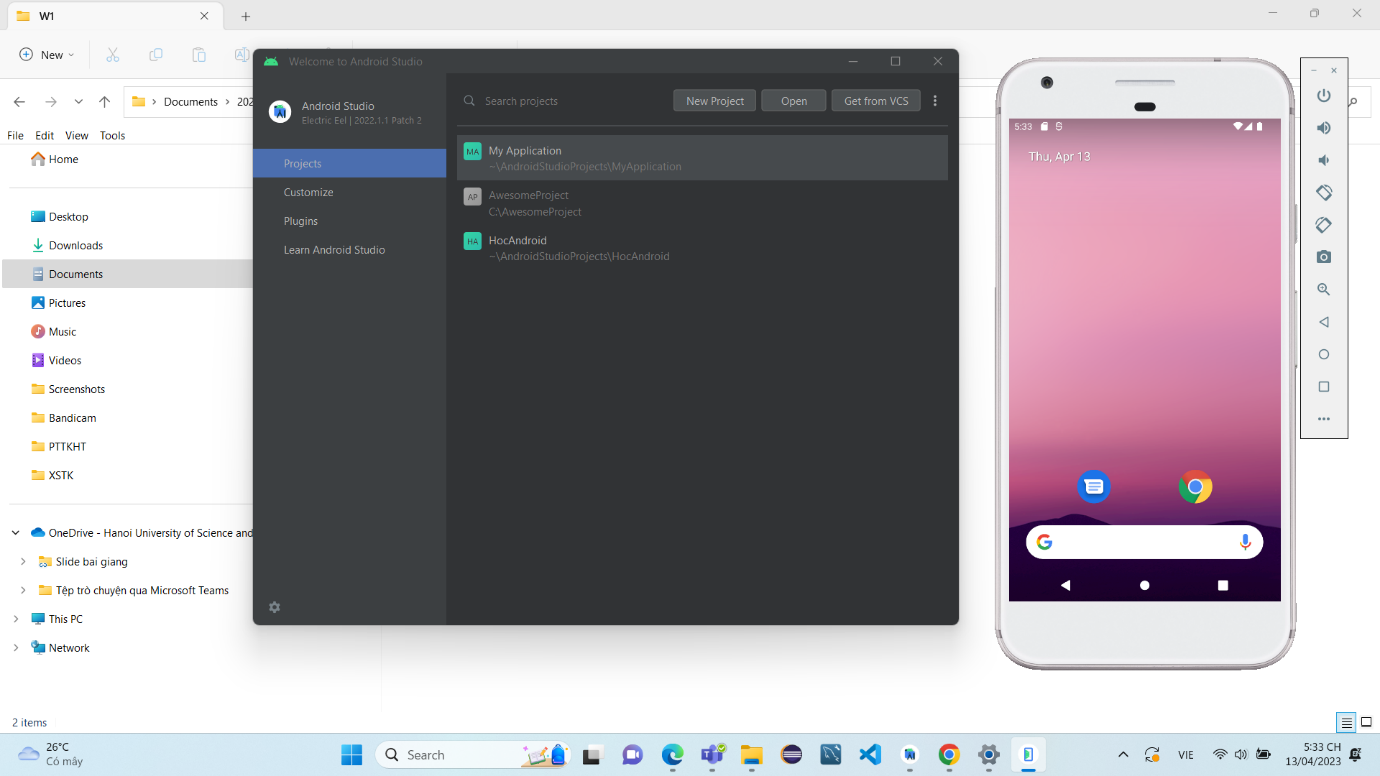
- Các ứng dụng Android được phát triển sử dụng Android Studio, một môi trường phát triển tích hợp (IDE) được cung cấp bởi Google.

- Lựa chọn ngôn ngữ Java để phát triển ứng dụng

## **Một số kiến thức cần thiết.**

- Ôn tập lại 1 số kiến thức Java như: cú pháp java, lập trình hướng đối tượng, exception handling, collection framework, I/O streams và Thread trong java

## **Cài đặt môi trường.**

**-** Cài đặt các môi trường như Android studio, máy ảo, xampp.

## **Học sử dụng Android Studio**

* Học sử dụng các Layout, button textview,.. Cơ bản và cách tạo giao diện cơ bản

**Ảnh có chứa văn bản, Điện thoại di động, ảnh chụp màn hình, đồ phụ tùng

Mô tả được tạo tự động**

# **Phân tích chức năng và cơ sở dữ liệu cho ứng dụng.**

## **Chức năng bên phía người dùng.**

**-** Các chức năng bên phía người dùng

* 1.Đăng nhập và đăng ký: cho phép người dùng đăng nhập vào hệ thống và tạo tài khoản mới.
* 2. Tìm kiếm sản phẩm: cho phép người dùng tìm kiếm sản phẩm theo tên, thương hiệu, danh mục hoặc từ khóa.
* 3. Danh mục sản phẩm: giúp người dùng tìm kiếm các sản phẩm theo danh mục cụ thể.
* 4. Giỏ hàng: cho phép người dùng thêm sản phẩm vào giỏ hàng để tiến hành mua sau.
* 5. Thanh toán: cho phép người dùng thanh toán online bằng các phương thức như thẻ tín dụng, ví điện tử hoặc COD.
* 6. Quản lý đơn hàng: cho phép người dùng theo dõi trạng thái đơn hàng của mình, bao gồm cả việc xem thông tin vận chuyển và thời gian giao hàng dự kiến.
* 7. Đánh giá sản phẩm: cho phép người dùng đánh giá sản phẩm sau khi mua hàng để giúp cho người khác có thể đưa ra quyết định mua hàng chính xác hơn.
* 8. Lịch sử mua hàng: cho phép người dùng xem lại lịch sử mua hàng của mình, các đơn hàng đã thanh toán và các sản phẩm đã mua.
* Có 2 loại người dùng:

+ Người dùng chưa đăng nhập.

+ Người dùng đã đăng nhập.

* Người dùng chưa đăng nhập sẽ bị hạn chế một số tính năng
* Biểu đồ chức năng:

Ảnh có chứa biểu đồ

Mô tả được tạo tự động

## **Phân tích cơ sở dữ liệu**

* Phân tích các thực thể gồm có 9 thực thể:

1. Người dùng: lưu trữ thông tin người dùng  
• ID người dùng  
• Tên đăng nhập  
• Mật khẩu  
• Họ tên  
• Email  
• Địa chỉ  
• Số điện thoại  
• Loại người dùng (Chủ cửa hàng/Khách hàng)

2. Sản phẩm: Lưu trữ thông tin sản phẩm

• Hình ảnh  
• ID sản phẩm  
• Tên sản phẩm  
• Mô tả  
• Giá sản phẩm  
• Thương hiệu  
• Danh mục

• Số lượng

• Đánh giá

• Đã bán

• Còn lại

3. Đơn hàng: lưu trữ thông tin đơn hàng.  
• ID đơn hàng  
• ID người dùng (Khách hàng)  
• Ngày đặt hàng  
• Giá  
• Ngày giao hàng (nếu có)  
• Trạng thái đơn hàng

• Phương thức thanh toán

4. Đánh giá sản phẩm: lưu trữ đánh giá sản phẩm của người dùng  
• ID đánh giá  
• Nội dung  
• Điểm đánh giá  
• Ngày đánh giá  
• ID người dùng (Khách hàng)  
• ID sản phẩm

5. Danh mục sản phẩm: lưu trữ danh mục của sản phẩm

• ID danh mục  
• Tên danh mục

6. Item đơn hàng: lưu trữ 1 sản phẩm trong đơn hàng của người dùng

• ID item đơn hàng

• ID đơn hàng

• ID sản phẩm

• ID thông số

• Số lượng sản phẩm

7.Item giỏ hàng: Lưu trữ 1 sản phẩm trong giỏ hàng của một người dùng.

• ID item giỏ hàng

• ID người dùng

• ID thông số

• Số lượng sản phẩm

• Trạng thái ( đã chọn hoặc không chọn )

8. Thông số: Lưu trữ các thông số của một sản phẩm

• ID sản phẩm

• ID thông số

• Kích thước

• Số lượng

• Còn lại

9. Thông báo: lưu trư các thông báo của người dùng.

• ID người dùng

• Tiêu đề

• Nội dung

• Ngày tạo

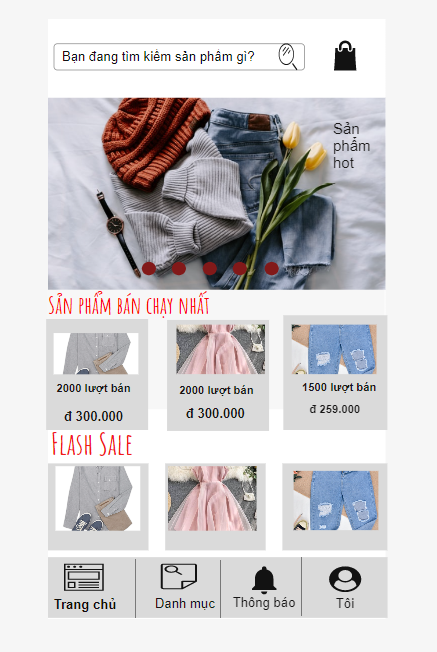
**Biểu đồ Database:**

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

Mô tả được tạo tự động**

## **Thiết kế giao diện**

* Khi khởi chạy ứng dụng, ứng dụng sẽ vào trang chủ:



* Ở trang chủ, khách hàng có thể tìm kiếm, xem sản phẩm bán chạy, sản phẩm flash sale,… có thể xem danh mục, thông báo và thông tin cá nhân
* Khi Chọn vào “danh mục” :

Ảnh có chứa bàn

Mô tả được tạo tự động

Ở danh mục có thể chọn đồ nam, nữ, trẻ em

* Khi chọn vào “áo”:

Ảnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự động

Ảnh và thông tin sản phẩm

* Khi chọn bộ lọc để lọc sản phẩm:

Ảnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự động

* Khi chọn vào 1 sản phẩm sẽ hiện ra chi tiết sản phẩm:

Ảnh có chứa văn bản, ảnh chụp màn hình, trang phục, thiết kế

Mô tả được tạo tự động

* Khi chọn vào “bài viết đánh giá”

Ảnh có chứa Hình chữ nhật

Mô tả được tạo tự động

Nội dung các bài viết

* khi chọn Thông báo, nếu chưa đăng nhập , hệ thống sẽ hiện màn hình đăng nhập hoặc đăng ký, nếu đã đăng nhập, sẽ hiện các thông báo.

Ảnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự độngẢnh có chứa văn bản

Mô tả được tạo tự động

Thông báo

* Khi người dùng chọn vào mục “ Tôi” , nếu chưa đăng nhập sẽ hiện ra màn hình thông báo:

Ảnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự động

* Nếu đăng nhập rồi, màn hình sẽ hiện ra:

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

Mô tả được tạo tự động

* Khi chọn thông tin cá nhân, người dùng có thể sửa

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

Mô tả được tạo tự động

Khi sửa xong có thể ấn

lưu

* Khi chọn quản lý đơn hàng:

Ảnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự động

Chọn vào để xem chi tiết

Hình ảnh sản phẩm đã đặt

* Khi xem chi tiết:

Ảnh có chứa biểu đồ

Mô tả được tạo tự động

* Chỉ hủy được khi trạng thái đơn hàng là “ Chờ xác nhận”
* Khi ấn vào giỏ hàng:

Ảnh có chứa hình vuông

Mô tả được tạo tự động

Chọn để đặt hàng

Muốn thanh toán tất cả thì chọn tất cả

Tích nếu muốn chọn thanh toán

* Khi chọn đặt hàng:

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

Mô tả được tạo tự động

* Nếu chưa có tài khoản, khi chọn đăng nhập:

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

Mô tả được tạo tự động

* Nếu chọn quên mật khẩu, chuyển sang trang lấy lại mật khẩu

Ảnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự động

* Xác nhận và đặt lại mật khẩu, khi đặt xong sẽ chuyển về trang đăng nhập

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Nếu đăng ký:

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

Mô tả được tạo tự động

# **Sử dụng Android Studio lập trình ứng dụng**

## **Tạo Project trên ứng dụng Android studio**

* Trong file activity\_main.xml, sử dụng 1 FrameLayout và Bottom Navigation View để chuyển đổi giữa các Fragment

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

* Sử dụng tạo Directory “menu” để cài các mục cho Bottom NavigationView:

Ảnh có chứa ảnh chụp màn hình, văn bản, Phần mềm đa phương tiện, phần mềm

Mô tả được tạo tự động

- Hình ảnh Bottom NavigationView sau khi set background:

Ảnh có chứa Thiết bị điện, đa phương tiện, đồ phụ tùng, ảnh chụp màn hình

Mô tả được tạo tự động-

- Thêm sự kiện cho các item khi click vào item trong bottomNavigationView:  
  
Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

* Tạo 4 Fragment cho 4 item: Trang chủ, Danh mục, Thông báo và hồ sơ.
* Cài đặt cho Fragment Trang chủ: Sử dụng LinearLayout để chia màn hình thành 4 phần theo kích thước tương đối ( sử dụng orientation và weightSum):
* Phần đầu cho thanh công cụ tìm kiếm.
* Phần 2 cho ViewFlipper để hiển thị quảng cáo.
* Phần 3 hiển thị listview sản phẩm bán chạy nhất ( số lượt bán cao nhất)
* Phần 4 hiện thi list view sản phẩm đang được sale

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

* Cài đặt ViewFlipper cho homepage để chạy quảng cáo:

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

Mô tả được tạo tự động

* Thêm thư viện vào build.gradle: implementation 'com.github.bumptech.glide:glide:4.15.1' để sử dụng glide.
* Tạo thêm directory “anim”: để lưu các file resource animotion cho flipperview:

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

Mô tả được tạo tự động

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

Mô tả được tạo tự động

Ảnh có chứa văn bản, đa phương tiện, đồ phụ tùng, Thiết bị điện

Mô tả được tạo tự động

## **Tạo Các Fragment chính của ứng dụng**

* Phần trang chủ được làm bằng 2 Recycer View và 1 View Flipper
* Các bước tạo: tạo Adapter cho sản phẩm với file product.xml

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

Mô tả được tạo tự động

AdapterProduct:

public class ProductAdapter extends RecyclerView.Adapter<ProductAdapter.ProductViewHolder> {  
 private List<Product> products;  
 public void setData(List<Product> products){  
 this.products = products;  
 notifyDataSetChanged();  
 }  
 @NonNull  
 @Override  
 public ProductViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*product*,parent,false);  
 return new ProductViewHolder(view);  
 }  
 @Override  
 public void onBindViewHolder(@NonNull ProductViewHolder holder, int position) {  
 Product product = products.get(position);  
 if(product == null){  
 return;  
 }  
 Picasso.*get*().load(product.getHinh\_anh()).into(holder.imageView);  
 holder.textView.setText(product.getTen\_san\_pham());  
 holder.textviewGia.setText("Giá: "+product.getGia\_san\_pham()+ " đ");  
 holder.textviewLuotban.setText("Lượt bán: "+product.getDa\_ban()+" lượt");  
 }  
  
 @Override  
 public int getItemCount() {  
 if(products != null) return products.size();  
 return 0;  
 }  
 public class ProductViewHolder extends RecyclerView.ViewHolder{  
 private ImageView imageView;  
 private TextView textView;  
 private TextView textviewGia;  
 private TextView textviewLuotban;  
 public ProductViewHolder(@NonNull View itemView) {  
 super(itemView);  
 imageView = itemView.findViewById(R.id.*anhsanpham*);  
 textView = itemView.findViewById(R.id.*tensanpham*);  
 textviewGia = itemView.findViewById(R.id.*giasanpham*);  
 textviewLuotban = itemView.findViewById(R.id.*luotban*);  
 }  
 }  
}

* Ở Homefragment, cài cho các RecyclerView theo chiều ngang

LinearLayoutManager linearLayoutManager = new LinearLayoutManager(this.getContext(), recyclerView1.*HORIZONTAL*, false);  
LinearLayoutManager linearLayoutManager2 = new LinearLayoutManager(this.getContext(), recyclerView2.*HORIZONTAL*, false);  
recyclerView1.setLayoutManager(linearLayoutManager);  
recyclerView2.setLayoutManager(linearLayoutManager2);

* Khởi tạo các hàm chứa danh sách các sản phẩm giảm giá và sản phẩm bán chạy nhất:

topSellingProducts = new ArrayList<>();  
saleProducts = new ArrayList<>();  
productAdapter1 = new ProductAdapter();  
productAdapter2 = new ProductAdapter();

* Tạo API để kết nối lấy dữ liệu từ máy chủ:

public interface API {  
  
 HttpLoggingInterceptor *log* = new HttpLoggingInterceptor().setLevel(HttpLoggingInterceptor.Level.*BODY*);  
  
 OkHttpClient.Builder *okBuilder* = new OkHttpClient.Builder()  
 .readTimeout(30, TimeUnit.*SECONDS*)  
 .connectTimeout(30,TimeUnit.*SECONDS*)  
 .retryOnConnectionFailure(true)  
 .addInterceptor(*log*);  
// Gson gson = new GsonBuilder().setDateFormat("yyyy-MM-dd HH-mm-ss").create();  
 API *api* = new Retrofit.Builder().baseUrl("http://192.168.3.102/")  
 .addConverterFactory(GsonConverterFactory.*create*())  
 .addCallAdapterFactory(RxJava3CallAdapterFactory.*create*())  
 .client(*okBuilder*.build())  
 .build().create(API.class);  
 @GET("server/getTopSellingProducts.php")  
 Observable<List<Product>> getTopSellingProducts();  
 @GET("server/getSaleProducts.php")  
 Observable<List<Product>> getSaleProducts();  
 @GET("server/getsearchProducts.php")  
 Observable<List<Product>> getSearchProduct();  
  
  
}

* Gọi hàm để lấy dữ liệu JSON từ máy chủ xuống và xử lý về object và xét adapter cho Recyclerview:

private void getTopSellingProducts() {  
 API.*api*.getTopSellingProducts().subscribeOn(Schedulers.*io*())  
 .observeOn(AndroidSchedulers.*mainThread*())  
 .subscribe(new Observer<List<Product>>() {  
 @Override  
 public void onSubscribe(@NonNull Disposable d) {  
 disposable = d;  
 }  
  
 @Override  
 public void onNext(@NonNull List<Product> products) {  
 topSellingProducts = products;  
 String name;  
 for(int i=0; i< topSellingProducts.size();i++){  
 name = topSellingProducts.get(i).getTen\_san\_pham();  
 nameProducts.add(name);  
 }  
 }  
  
 @Override  
 public void onError(@NonNull Throwable e) {  
 Log.*d*(*TAG*, "onError: " + e.getMessage());  
  
 }  
  
 @Override  
 public void onComplete() {  
 productAdapter1.setData(topSellingProducts);  
 recyclerView1.setAdapter(productAdapter1);  
 recommend = new ArrayAdapter(getContext(), android.R.layout.*simple\_dropdown\_item\_1line*,nameProducts);  
 autoCompleteTextView.setAdapter(recommend);  
 autoCompleteTextView.setThreshold(1);  
 }  
 });  
}  
private void getSaleProducts() {  
 API.*api*.getSaleProducts().subscribeOn(Schedulers.*io*())  
 .observeOn(AndroidSchedulers.*mainThread*())  
 .subscribe(new Observer<List<Product>>() {  
 @Override  
 public void onSubscribe(@NonNull Disposable d) {  
 disposable = d;  
 }  
  
 @Override  
 public void onNext(@NonNull List<Product> products) {  
 saleProducts = products;  
 }  
  
 @Override  
 public void onError(@NonNull Throwable e) {  
 Log.*d*(*TAG*, "onError: " + e.getMessage());  
  
 }  
  
 @Override  
 public void onComplete() {  
 productAdapter2.setData(saleProducts);  
 recyclerView2.setAdapter(productAdapter2);  
 }

* Các file PHP để lấy dữ liệu từ máy chủ với connect.php là hàm kết nối.
* getTopSellingProduct để lấy sản phẩm bán chạy nhất từ máy chủ

<?php

    // Kết nối tới cơ sở dữ liệu

    include 'connect.php';

    // Truy vấn danh sách sản phẩm

    $query = "SELECT \* FROM san\_pham ORDER BY da\_ban DESC";

    $result = mysqli\_query($conn, $query);

    // Chuyển đổi kết quả thành dạng json

    $san\_pham = array();

    while ($row = mysqli\_fetch\_assoc($result)) {

        $san\_pham[] = $row;

    }

    $json = json\_encode($san\_pham);

    // Trả về kết quả dưới dạng json

    header('Content-Type: application/json');

    echo $json;

    // Đóng kết nối

    mysqli\_close($conn);

?>

+/getSaleProducts.php để lấy sản phẩm đang được giảm giá:

<?php

    // Kết nối tới cơ sở dữ liệu

    include 'connect.php';

    // Truy vấn danh sách sản phẩm

    $query = "SELECT \* FROM san\_pham WHERE giam\_gia != 0 ORDER BY giam\_gia DESC;

    Giam ";

    $result = mysqli\_query($conn, $query);

    // Chuyển đổi kết quả thành dạng json

    $san\_pham = array();

    while ($row = mysqli\_fetch\_assoc($result)) {

        $san\_pham[] = $row;

    }

    $json = json\_encode($san\_pham);

    // Trả về kết quả dưới dạng json

    header('Content-Type: application/json');

    echo $json;

    // Đóng kết nối

    mysqli\_close($conn);

?>

* Hàm cài đặt cho ViewFlipper với view là danh sách các đường link hình ảnh quảng cáo lấy trên máy chủ:

private void ActionViewFlipper() {  
 for (int i = 0; i < view.size(); i++) {  
 ImageView imageView = new ImageView(requireContext());  
 Glide.*with*(requireContext()).load(view.get(i)).into(imageView);  
 imageView.setScaleType(ImageView.ScaleType.*FIT\_XY*);  
 viewFlipper.addView(imageView);  
 }  
 viewFlipper.setFlipInterval(3000);  
 viewFlipper.setAutoStart(true);  
 Animation slide\_in = AnimationUtils.*loadAnimation*(requireContext(), R.anim.*slide\_in\_right*);  
 Animation slide\_out = AnimationUtils.*loadAnimation*(requireContext(), R.anim.*slide\_out\_right*);  
 viewFlipper.setInAnimation(slide\_in);  
 viewFlipper.setOutAnimation(slide\_out);  
}

* Phần danh mục của ứng dụng:

Ảnh có chứa ảnh chụp màn hình, đa phương tiện, Thiết bị liên lạc, văn bản

Mô tả được tạo tự động

* Tạo danh mục với TableLayout và 4 fragment khác nhau.
* Sau khi click vào 1 danh mục hoặc dùng thanh công cụ ở Home thì Fragment sẽ chuyển:

Ảnh có chứa văn bản, ảnh chụp màn hình, trang phục, Website

Mô tả được tạo tự động

* Cách thực hiện:
* Thiết lập Adapter khác cho sản phẩm với product2.xml, lấy dữ liệu xử lí và đổ ra Recylerview theo chiều dọc ( Như ở trang chủ)

public void init(){  
 recyclerView = view.findViewById(R.id.*recyclerview3*);  
 LinearLayoutManager linearLayoutManager = new LinearLayoutManager(this.getContext(), recyclerView.*VERTICAL*, false);  
 recyclerView.setLayoutManager(linearLayoutManager);  
 editText = view.findViewById(R.id.*namesearch*);  
 productAdapter2 = new ProductAdapter2();  
 products1 = new ArrayList<>();  
 getSearchProduct();  
  
}  
private void getSearchProduct() {  
 API.*api*.getTopSellingProducts().subscribeOn(Schedulers.*io*())  
 .observeOn(AndroidSchedulers.*mainThread*())  
 .subscribe(new Observer<List<Product>>() {  
 @Override  
 public void onSubscribe(@NonNull Disposable d) {  
 disposable = d;  
 }  
  
 @Override  
 public void onNext(@NonNull List<Product> products) {  
 products1 = products;  
 Toast.*makeText*(getContext(), products1.get(1).getTen\_san\_pham(), Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void onError(@NonNull Throwable e) {  
 Log.*d*(*TAG*, "onError: " + e.getMessage());  
  
 }  
  
 @Override  
 public void onComplete() {  
 productAdapter2.setData(products1);  
 recyclerView.setAdapter(productAdapter2);  
 }  
 });  
}

## **Chi tiết đơn hàng.**

* Khi người dùng chọn vào một item trong các sản phẩm hiện trên trang đầu hoặc trong danh mục thì sẽ hiện lên chi tiết.

Ảnh có chứa văn bản, ảnh chụp màn hình, đa phương tiện, Máy tính bảng

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, trang phục, đàn ông

Mô tả được tạo tự động

* Sau khi click vào một item thì sẽ hiện ra chi tiết sản phẩm :

Ảnh có chứa văn bản, ảnh chụp màn hình, trang phục

Mô tả được tạo tự động

* Cách thực hiện: bắt sự kiện cho các Adapter Product -> sau đó dùng intern để thực hiện gửi Object sang activity mới -> ở Activity mới, lấy Object truy xuất dữ liệu ra màn hình.

## **Phần giỏ hàng**

* Khi người dùng click vào giỏ hàng của mình, sẽ hiện ra những sản phẩm hiện có trong giỏ hàng
* Phần giỏ hàng có các button tăng, giảm số lượng, các checkbox để tính toán tống tiền
* Khi người dùng chọn vào một vào giỏ hàng,thông báo hoặc profile mà chưa đăng nhập thì màn hình sẽ hiển thị

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

Mô tả được tạo tự động

* Cách tạo: Tạo 1 lớp CheckLogin chứa các biến static:

package com.example.appbanhang.service;  
  
public class CheckLogin {  
 public static int *UserID*;  
 public static boolean *Login*;  
}

* Khi chưa đăng nhập thì UserID = null và Login = false;
* Tạo hàm check xem đã đăng nhập chưa:

if (CheckLogin.*Login* == false) {  
 com.example.appbanhang.fragment.CheckLogin fragment = new com.example.appbanhang.fragment.CheckLogin();  
 Bundle bundle = new Bundle();  
 fragment.setArguments(bundle); // Truyền bundle cho fragment  
 getActivity().getSupportFragmentManager()  
 .beginTransaction()  
 .replace(R.id.*framelayout*, fragment)  
 .addToBackStack(null)  
 .commit();  
} else {  
 Intent intent = new Intent(getActivity(), CartActivity.class);  
 startActivity(intent);  
}

* Nếu chưa đăng nhập thì sẽ trả về màn đăng nhập, còn nếu đăng nhập rồi sẽ trả về màn hình giỏ hàng,…  
  Ảnh có chứa văn bản, ảnh chụp màn hình, Website, Trang web

  Mô tả được tạo tự động
* Khi chọn đăng nhập, sẽ hiển thị màn hình đăng nhập:

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

Mô tả được tạo tự động Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động Ảnh có chứa đồ điện tử, văn bản, ảnh chụp màn hình, số

Mô tả được tạo tự động

* Khi nhập sai tài khoản mật khẩu hoặc ký tự đặc biệt không phù hợp thì sẽ hiển thị thông báo:
* Hàm check kí tự đặc biệt hoặc trống

public boolean isValidCredentials(String username, String password) {  
 // Kiểm tra xem tên người dùng và mật khẩu có rỗng hay không  
 if (username.isEmpty() || password.isEmpty()) {  
 return false;  
 }  
  
 // Kiểm tra xem tên người dùng và mật khẩu có chứa các ký tự không hợp lệ hay không  
 String regex = "^[a-zA-Z0-9]+$"; // Chỉ cho phép ký tự chữ và số  
 if (!username.matches(regex) || !password.matches(regex)) {  
 return false;  
 }  
  
 return true;  
}

* Hàm thực hiện đăng nhập:

API.*api*.getUser(name,pass).subscribeOn(Schedulers.*io*())  
 .observeOn(AndroidSchedulers.*mainThread*())  
 .subscribe(new Observer<List<User>>() {  
 @Override  
 public void onSubscribe(@NonNull Disposable d) {  
 disposable = d;  
 }  
  
 @Override  
 public void onNext(@NonNull List<User> user1) {  
 user = user1;  
 }  
  
 @Override  
 public void onError(@NonNull Throwable e) {  
 Log.*d*(*TAG*, "onError: " + e.getMessage());  
 }  
  
 @Override  
 public void onComplete() {  
 if (!user.isEmpty()){  
 CheckLogin.*Login* = true;  
 CheckLogin.*UserID* = user.get(0).getId\_nguoi\_dung();  
 error.setText("Đăng nhập thành công");  
 error.setTextColor(Color.*GREEN*);  
 error.setVisibility(View.*VISIBLE*);  
 usernameerror.setVisibility(View.*GONE*);  
 Handler handler = new Handler();  
 handler.postDelayed(new Runnable() {  
 @Override  
 public void run() {  
 error.setVisibility(View.*GONE*);  
 }  
 }, 5000);  
 Intent intent = new Intent(getApplicationContext(), MainActivity.class);  
 intent.addFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK* | Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(intent);  
 finish();  
 }  
 else{  
 error.setVisibility(View.*VISIBLE*);  
 usernameerror.setVisibility(View.*GONE*);  
 Handler handler = new Handler();  
 handler.postDelayed(new Runnable() {  
 @Override  
 public void run() {  
 error.setVisibility(View.*GONE*);  
 }  
 }, 5000);  
 }  
 }  
 });

* Kiểm tra nếu trống hoặc sai hoặc kí tự đặc biệt thì hiện thông báo:

error.setVisibility(View.*VISIBLE*);  
usernameerror.setVisibility(View.*GONE*);  
Handler handler = new Handler();  
handler.postDelayed(new Runnable() {  
 @Override  
 public void run() {  
 error.setVisibility(View.*GONE*);  
 }  
}, 5000);

* Nếu người dùng đăng nhập thành công thì sẽ trả về màn hình chính.

## **Phần thanh toán cho sản phẩm**

* Khi người dùng chọn vào một vào giỏ hàng, sẽ có nút Thanh Toán cho giỏ hàng

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

Mô tả được tạo tự động

* Khi người dùng click vào thanh toán, ngay lập tức chuyển sang màn hình thanh toán:

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

Mô tả được tạo tự động

* Khi người dùng chọn đặt hàng thì ứng dụng sẽ thêm đơn hàng vào cơ sở dữ liệu bảng đơn hàng.
* Cách tạo: tạo giao diện cho thanh toán có sử dụng recyclerview

: Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Hệ điều hành

Mô tả được tạo tự động

* Sau đó tạo Adapter đơn hàng:

public void onBindViewHolder(@NonNull PaymetAdapter.PaymentViewHolder holder, int position) {

item\_cart itemCart = item\_carts.get(position);

Product product = itemCart.getProduct();

Parameter parameter = itemCart.getParameter();

int j = position;

Picasso.*get*().load(product.getHinh\_anh()).into(holder.imageView);

holder.textViewname.setText(product.getTen\_san\_pham());

holder.textviewGia.setText("Thành tiền " + decimalFormat.format(product.getGia\_san\_pham() \* itemCart.getSo\_luong\_san\_pham()) + "đ ");

holder.soluong.setText(itemCart.getSo\_luong\_san\_pham() + "");

}

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

Mô tả được tạo tự động

* Sau khi tạo Adapter xong thì lấy dữ liệu gửi từ giỏ hàng và đổ dữ liệu cho Recyclerview cho đơn hàng
* Hàm tính tổng tiền của các sản phẩm trong đơn hàng:

@SuppressLint("SuspiciousIndentation")

public void Sum() {

double sum = 0;

for (item\_cart x :

CartAdapter.*item\_carts*) {

if (x.getTrang\_thai() == 1)

sum += x.getSo\_luong\_san\_pham() \* x.getProduct().getGia\_san\_pham();

}

tvtongtien.setText(decimalFormat.format(sum) + " đ.");

}

+ Với Trạng thái nghĩa là (1 là được chọn qua đơn hàng, 0 là không được chọn)

## **Phần hồ sơ người dùng**

- Khi người dùng chọn hồ sơ, có 2 trường hợp: Nếu chưa đăng nhập, sẽ trả về màn hình đăng nhập hoặc đăng ký, nếu đăng nhâp rồi thì trả về màn hình hồ sơ.

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

Mô tả được tạo tự động

* Khi người dùng chọn vào quản lí hồ sơ, sẽ hiện ra màn hình. Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

  Mô tả được tạo tự động
* Khi khách hàng sửa thông tin, sẽ có thông báo:

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

Mô tả được tạo tự động

* Nếu ấn có, sẽ hiện ra thanh điền vào mật khẩu:
* Ảnh có chứa đồ điện tử, văn bản, ảnh chụp màn hình, số

  Mô tả được tạo tự động
* Nếu mật khẩu đúng, dư liệu sẽ được cập nhật lên Database
* Cách tạo:   
    
  Dùng hàm để lấy về thông tin người dùng với id đăng nhập đã có:

API.*api*.getUser(CheckLogin.*UserID*)

.subscribeOn(Schedulers.*io*())

.observeOn(AndroidSchedulers.*mainThread*())

.subscribe(new Observer<User>() {

@Override

public void onSubscribe(@io.reactivex.rxjava3.annotations.NonNull Disposable d) {

// Xử lý khi đăng ký

}

@Override

public void onNext(@io.reactivex.rxjava3.annotations.NonNull User user2) {

// Xử lý khi nhận được dữ liệu phản hồi

user = user2;

}

@Override

public void onError(@io.reactivex.rxjava3.annotations.NonNull Throwable e) {

// Xử lý khi xảy ra lỗi

}

@Override

public void onComplete() {

// Xử lý khi hoàn thành

name.setText(user.getHo\_ten());

}

});

- Khi người dùng click vào phím tắt, chuyển màn hình qua thông tin người dùng có chuyển kèm cho 1 đối tượng là người dùng:

Intent intent = new Intent(getActivity(), Profile.class);

Bundle bundle = new Bundle();

bundle.putSerializable("user", user);

intent.putExtras(bundle);

startActivity(intent);

* Hàm xử lí lựa chọn:

public void onClick(View view) {

AlertDialog.Builder builder = new AlertDialog.Builder(Profile.this);

builder.setMessage("Bạn có muốn lưu thay đổi?")

.setPositiveButton("Có", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) {

showInputDialog();

}

})

.setNegativeButton("Không", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) {

// Hủy bỏ lưu thay đổi ở đây

}

});

AlertDialog dialog = builder.create();

dialog.show();

}

* Hàm xử lí mật khẩu:

private void showInputDialog() {

AlertDialog.Builder builder = new AlertDialog.Builder(Profile.this);

builder.setTitle("Nhập vào mật khẩu của bạn");

// Tạo trường nhập liệu

final EditText input = new EditText(Profile.this);

input.setInputType(InputType.*TYPE\_CLASS\_TEXT* | InputType.*TYPE\_TEXT\_VARIATION\_PASSWORD*);

builder.setView(input);

// Đặt nút "Lưu" cho hộp thoại nhập liệu

builder.setPositiveButton("Xác nhận", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

String inputValue = input.getText().toString();

if(inputValue.equals(user.getMat\_khau())){

Toast.*makeText*(Profile.this, "lưu thành công", Toast.*LENGTH\_SHORT*).show();

setUser(user.getId\_nguoi\_dung(),name.getText().toString().trim(),email.getText().toString().trim(),phone.getText().toString().trim(),address.getText().toString());

}

else {

Toast.*makeText*(Profile.this, "Mật khẩu sai", Toast.*LENGTH\_SHORT*).show();

}

// Xử lý giá trị nhập vào

}

});

// Đặt nút "Hủy" cho hộp thoại nhập liệu

builder.setNegativeButton("Hủy", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

dialog.cancel();

}

});

// Hiển thị hộp thoại nhập liệu

AlertDialog dialog = builder.create();

dialog.show();

}

* Khi nhập đúng mật khẩu sẽ sửa dữ liệu trên database, hàm set user mới là:
* public String setUser(int id,String name, String email, String phone, String address){
* API.*api*.setInfo(id,name,email,phone,address)
* .subscribeOn(Schedulers.*io*())
* .observeOn(AndroidSchedulers.*mainThread*())
* .subscribe(new Observer<ResponseData>() {
* @Override
* public void onSubscribe(@io.reactivex.rxjava3.annotations.NonNull Disposable d) {
* // Xử lý khi đăng ký
* }
* @Override
* public void onNext(@io.reactivex.rxjava3.annotations.NonNull ResponseData ResponseData) {
* // Xử lý khi nhận được dữ liệu phản hồi
* res = ResponseData.getMessage().toString();
* }
* @Override
* public void onError(@io.reactivex.rxjava3.annotations.NonNull Throwable e) {
* // Xử lý khi xảy ra lỗi
* }
* @Override
* public void onComplete() {
* // Xử lý khi hoàn thành
* Toast.*makeText*(Profile.this, "Đã lưu thay đổi", Toast.*LENGTH\_SHORT*).show();
* }
* });
* return res;
* }

File PHP set dữ iệu người dùng là:   
<?php

    // Kết nối tới cơ sở dữ liệu

    include 'connect.php';

    $id = $\_POST['id'];

    $name = $\_POST['name'];

    $email = $\_POST['email'];

    $phone = $\_POST['phone'];

    $address = $\_POST['address'];

    // Sửa đổi bảng item\_gio\_hang

    $query = "UPDATE `nguoi\_dung` SET `ho\_ten` = ".$name.",email = ".$email.",so\_dien\_thoai = ".$phone.",dia\_chi = ".$address." WHERE `nguoi\_dung`.`id\_nguoi\_dung` = ".$id."" ;

    $result = mysqli\_query($conn, $query);

    if ($result) {

        // Thành công, trả về thông báo

        $response = array('message' => 'Success');

    } else {

        // Lỗi, trả về thông báo lỗi

        $response = array('message' => 'Error');

    }

    // Chuyển đổi kết quả thành dạng json

    $json = json\_encode($response);

    // Trả về kết quả dưới dạng json

    header('Content-Type: application/json');

    echo $json;

    // Đóng kết nối

    mysqli\_close($conn);

?>

- Nếu tin nhắn trả về à Succes thì đã thay đổi, còn Error thì bị lỗi

## **Phần thêm vào giỏ hàng**

Khi người dùng chọn vào một vào chi tiết giỏ hàng, sẽ có nút thêm vào giỏ hàng

Ảnh có chứa đồ điện tử, văn bản, đa phương tiện, đồ phụ tùng

Mô tả được tạo tự động

+ Khi người dùng chưa click vào chọn size thì nút tăng, giảm và mua sẽ ở chế độ Enable(false)

Ảnh có chứa văn bản, ảnh chụp màn hình, đa phương tiện, Thiết bị điện

Mô tả được tạo tự động

+Khi người dùng click vào chọn size thì 3 button sẽ có thể click được, khi ta click vào button mua thì nó sẽ thêm vào giỏ hàng.

* Cách thực hiện:
* Đầu tiên cần tạo file Layout cho Dialog   
  Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Hình chữ nhật

  Mô tả được tạo tự động

ReycyclerView sẽ để chứa các Adapter size, tạo Adapter size:

Ảnh có chứa ảnh chụp màn hình, Hình chữ nhật, thiết kế, lỗ cắm

Mô tả được tạo tự động

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

Mô tả được tạo tự động

+ Đăng ký sự kiện khi click vào size ở Activity

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

Mô tả được tạo tự động

+ Sự kiện khi điền số lượng, nếu điền quá số lượng sẽ cài về số lượng lớn nhất

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

Mô tả được tạo tự động

+Sự kiện cho nút tăng, giảm, không vượt quá số lượng còn lại hoặc dưới 0:

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

Mô tả được tạo tự động

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

Mô tả được tạo tự động

+Khi người dùng click vào nút thêm vào giỏ thì sản phẩm sẽ được thêm vào giỏ

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

Mô tả được tạo tự động

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

Mô tả được tạo tự động

+ File php: nếu sản phẩm đã tồn tại trong giỏ hàng thì sẽ tăng số lượng cho sản phẩm trong giỏ hàng, nếu chưa có thì sẽ thêm sản phẩm vào

<?php

    // Kết nối tới cơ sở dữ liệu

    include 'connect.php';

    $id = $\_POST['id'];

    $id\_product = $\_POST['id\_product'];

    $id\_parameter = $\_POST['id\_parameter'];

    $quantity = $\_POST['quantity'];

    // Kiểm tra xem mục đã tồn tại hay chưa

    $checkQuery = "SELECT \* FROM item\_gio\_hang WHERE id\_san\_pham = $id\_product AND id\_thong\_so = $id\_parameter AND id\_nguoi\_dung = $id ";

    $checkResult = mysqli\_query($conn, $checkQuery);

    if (mysqli\_num\_rows($checkResult) > 0) {

        // Mục đã tồn tại, thực hiện cập nhật quantity

        $updateQuery = "UPDATE item\_gio\_hang SET so\_luong\_san\_pham = so\_luong\_san\_pham + $quantity WHERE id\_san\_pham = $id\_product AND id\_thong\_so = $id\_parameter AND id\_nguoi\_dung = $id";

        $result = mysqli\_query($conn, $updateQuery);

    } else {

        // Mục chưa tồn tại, thực hiện thêm mới

        $insertQuery = "INSERT INTO `item\_gio\_hang`( `id\_nguoi\_dung`, `id\_san\_pham`, `id\_thong\_so`, `so\_luong\_san\_pham`, `trang\_thai`) VALUES ($id,$id\_product,$id\_parameter,$quantity,0)";

        $result = mysqli\_query($conn, $insertQuery);

    }

    if ($result) {

        // Thành công, trả về thông báo

        $response = array('message' => 'Success');

    } else {

        // Lỗi, trả về thông báo lỗi

        $response = array('message' => 'Error');

    }

    // Chuyển đổi kết quả thành dạng json

    $json = json\_encode($response);

    // Trả về kết quả dưới dạng json

    header('Content-Type: application/json');

    echo $json;

    // Đóng kết nối

    mysqli\_close($conn);

?>

## **Phần lịch sử mua hàng**

Ảnh có chứa văn bản, đa phương tiện, đồ phụ tùng, ảnh chụp màn hình

Mô tả được tạo tự động

+Nếu chưa có đơn hàng thì sẽ hiện:

Ảnh có chứa đồ phụ tùng, ảnh chụp màn hình, Thiết bị liên lạc, Thiết bị điện

Mô tả được tạo tự động

* Cách thực hiện:

+Đầu tiên tạo TableLayout cùng ViewPager2  
Tạo ViewPagerAdapter

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

Mô tả được tạo tự động

+ Sau đó tạo View

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

Mô tả được tạo tự động

+ Với mỗi case thì sẽ có dữ liệu gửi đi để lấy dữ liệu khác nhau:

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

Mô tả được tạo tự động

+Ở HistoryFragMent có hàm lấy dữ liệu:

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

Mô tả được tạo tự động

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

Mô tả được tạo tự động

Hàm getOrder sẽ lấy dữ liệu trên database về đơn hàng với trạng thái được gửi: file PHP

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

Mô tả được tạo tự động

<?php

    // Kết nối tới cơ sở dữ liệu

    include 'connect.php';

    $id = $\_POST['id'];

    $trang\_thai\_don\_hang = $\_POST['trang\_thai\_don\_hang'];

    // Truy vấn danh sách sản phẩm

    $query = "SELECT \* FROM don\_hang where id\_nguoi\_dung = $id and trang\_thai\_don\_hang = '".$trang\_thai\_don\_hang."'";

    $result = mysqli\_query($conn, $query);

    // Chuyển đổi kết quả thành dạng json

    $san\_pham = array();

    while ($row = mysqli\_fetch\_assoc($result)) {

        $san\_pham[] = $row;

    }

    $json = json\_encode($san\_pham);

    // Trả về kết quả dưới dạng json

    header('Content-Type: application/json');

    echo $json;

    // Đóng kết nối

    mysqli\_close($conn);

?>

+ Mỗi đơn hàng sẽ là 1 Adapter cho recyclerView của HistoryFragment

+Mỗi Item sẽ là 1 item cho 1Adapter đơn hàng:

Tạo adpater cho mỗi đơn hàng:

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

Mô tả được tạo tự động

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

Mô tả được tạo tự động

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

Mô tả được tạo tự động

File php lấy ra tất cả các item\_don\_hang của 1 đơn hàng:

<?php

// Kết nối tới cơ sở dữ liệu

include 'connect.php';

$id = $\_POST['id'];

// Truy vấn danh sách sản phẩm

$query = "SELECT \* FROM san\_pham,thong\_so,item\_don\_hang WHERE item\_don\_hang.id\_don\_hang = '".$id."' and item\_don\_hang.id\_thong\_so = thong\_so.id\_thong\_so and item\_don\_hang.id\_san\_pham = san\_pham.id\_san\_pham ";

$result = mysqli\_query($conn, $query);

// Kiểm tra số bản ghi trả về

if (mysqli\_num\_rows($result) > 0) {

    $item\_list = array();

    while ($row = mysqli\_fetch\_assoc($result)) {

        // Tạo đối tượng ShoppingCart

        $item = new stdClass();

        $item->id\_item\_don\_hang = $row['id\_item\_don\_hang'];

        $item->id\_don\_hang = $row['id\_don\_hang'];

        $item->so\_luong\_san\_pham = $row['so\_luong\_san\_pham'];

        // Tạo đối tượng Product

        $product = new stdClass();

        $product->id\_san\_pham = $row['id\_san\_pham'];

        $product->ten\_san\_pham = $row['ten\_san\_pham'];

        $product->mo\_ta = $row['mo\_ta'];

        $product->gia\_san\_pham = $row['gia\_san\_pham'];

        $product->thuong\_hieu = $row['thuong\_hieu'];

        $product->id\_danh\_muc = $row['id\_danh\_muc'];

        $product->da\_ban = $row['da\_ban'];

        $product->danh\_gia = $row['danh\_gia'];

        $product->hinh\_anh = $row['hinh\_anh'];

        $item->product = $product;

        $parameter = new stdClass();

        $parameter->id\_thong\_so = $row['id\_thong\_so'];

        $parameter->id\_san\_pham = $row['id\_san\_pham'];

        $parameter->so\_luong = $row['so\_luong'];

        $parameter->con\_lai = $row['con\_lai'];

        $parameter->kich\_thuoc = $row['kich\_thuoc'];

        // Gán đối tượng Product cho đối tượng ShoppingCart

        $item->parameter = $parameter;

        // Thêm ShoppingCart vào danh sách

        $item\_list[] = $item;

    }

    // Chuyển đổi danh sách thành JSON

    $json = json\_encode($item\_list);

    // Trả về kết quả dưới dạng JSON

    header('Content-Type: application/json');

    echo $json;

} else {

    // Trả về danh sách rỗng nếu không có bản ghi

    echo json\_encode(array());

}

// Đóng kết nối

mysqli\_close($conn);

?>

+Tạo Adapter cho Mỗi item của 1 đơn hàng:

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

Mô tả được tạo tự động

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

Mô tả được tạo tự động

## **Xử lí đặt hàng**

Ở phần giỏ hàng, khi khách hàng tăng, giảm số lượng, nếu có người khác mua mất thì sẽ cập nhật số lượng còn lại.

* Cài thêm rằng buộc giữa số lượng còn lại của sản phẩm luôn lớn hơn số lượng của sản phẩm mà khách hàng thêm vào giỏ hàng trên database, nếu khi Update lỗi sẽ trả về error.

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

Mô tả được tạo tự động

* Trên ứng dụng nếu nhận về dữ liệu phản hồi là error thì thông báo và cập nhật lại giỏ hàng trong Activity.

Ảnh có chứa văn bản, Phần mềm đa phương tiện, phần mềm, ảnh chụp màn hình

Mô tả được tạo tự động

Ở phần giỏ hàng, nếu khách hàng ấn chuyển sang màn hình thanh toán với các sản phẩm được chọn nhưng sản phẩm bị người khác đặt mất thì sẽ thông báo và cập nhật lại.

* Ở phần thanh toán, nếu sản phẩm đã bị đặt bởi người khác thì khách hàng không thể ấn đặt hàng được, sau khi đặt hàng thì sẽ check qua các sản phẩm đã đặt ở bảng item\_gio\_hang

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

Mô tả được tạo tự động

Ở phần thanh toán, nếu sản phẩm đã bị đặt bởi người khác thì khách hàng không thể ấn đặt hàng được

* Ở phần thanh toán sau khi đặt hàng thì sẽ check qua các sản phẩm đã đặt ở bảng item\_gio\_hang

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

Mô tả được tạo tự động

# **Tổng kết**

## **Mục tiêu đồ án.**

* Tạo ra một ứng dụng mua hàng cung cấp cho 1 cửa hàng để người dùng có thể đặt mua hàng trực tuyến trên ứng dụng, giúp chủ cửa hàng dễ dàng quản lý cửa hàng của mình
* Tìm hiểu thêm về cách thức hoạt động cũng như tạo ra một ứng dụng android thực tế

## **Tóm tắt ứng dụng**

* Ứng dụng cho phép người dùng có thể xem các sản phẩm của cửa hàng, xem chi tiết sản phẩm, tìm kiếm, thêm vào giỏ hàng và đặt mua sản phẩm. Người dùng có thể quản lý đơn hàng của mình, khi đơn hàng chưa xác nhận thì có thể hủy đơn. Người dùng có thể đăng ký tài khoản cho mình, chỉnh sửa thông tin khi đăng nhập và đổi mật khẩu.

## **Đường dẫn sản phẩm.**

<https://github.com/CVThien2k2/Sales-App.git>