HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG CƠ SỞ THÀNH PHỐ HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

-------🙞🕮🙜-------



BÁO CÁO ĐỒ ÁN

Môn :

**PHÁT TRIỄN ỨNG DỤNG CHO CÁC THIẾT BỊ DI ĐỘNG**

Đề tài :

**ỨNG DỤNG HỆ THÔNG BÁN HÀNG ONLINE**

Giáo Viên Giảng Dạy: **TRƯƠNG BÁ THÁI**

**Họ Tên Sinh Viên Mã Sinh Viên**

**VÕ TÙNG NGHĨA N15DCCN136**

**HỒ QUANG VIỆT N15DCCN141**

**PHẠM DUY THƯƠNG N15DCCN187**

**NGUYÊN PHƯƠNG NAM N15DCCN181**

**ĐẬU ANH TUẤN N15DCCN188**

Thành Phố Hồ Chí Minh – 2019

**PHẦN I : GIỚI THIỆU**

1. **Giới thiệu**

Hiện nay, Thương Mại Điện Tử (TMĐT) đang được nhiều quốc gia quan tâm, coi là một trong những động lực phát triển chủ yếu của nền kinh tế. TMĐT đem lại những lợi ích tiềm tàng, giúp doanh nghiệp thu được thông tin phong phú về thị trường và đối tác, giảm chi phí tiếp thị và giao dịch, tạo dựng và củng cố quan hệ bạn hàng.

- Việc mua hàng qua mạng chỉ với thủ tục đăng ký mua sắm đơn giản nhưng đem lại nhiều lợi ích: tiết kiệm và chủ động về thời gian, tránh khỏi những phiền phức khó chịu. Vậy nên việc mua bán hàng qua mạng đang rất được mọi người quan tâm. Trên cơ sở các kiến thức được học trong nhà trường và quá trình tìm hiểu các website trong thực tế , em đã quyết định chọn đề tài “Xây dựng ứng dụng hệ thống bán hàng online”.

- Khách hàng chỉ cần các thao tác đơn giản trên điện thoại di động là có thể đặt hàng được ngay

1. **Ưu điểm**

- Dễ dàng quản lý

- Tăng năng suất làm việc

- Tiết kiệm thời gian

1. **Các tính năng**

**3.1 : Công nghệ sử dụng**

* Web service: Dùng RESTful API service

- Sử dụng LinQ to SQL Classes để kết nối CSDL.

* Client:

- Mobile app bán hàng được viết bằng Android.

- Website admin được viết bằng thư viện ReactJs.

**3.2 : Tính năng chính**

- Xem thông tin chi tiết các mặt hàng cần mua

- Đặt mua hàng một cách dễ dàng

- Có thể xem thông tin đơn hàng đã đặt, sản phẩm yêu thích

- Web admin dùng để quản lý việc thêm xóa sửa dữ liệu, thực hiện thông báo những thông tin khuyến mãi, thông tin mới nhất về sản phẩm cho người dùng thông qua Notification

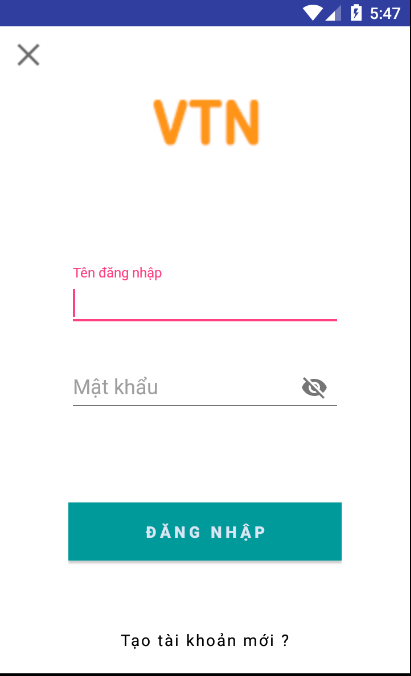
**PHẦN II : ANDROID**

1. **Giao diện**
   1. **Màn hình khởi động (Splash)**



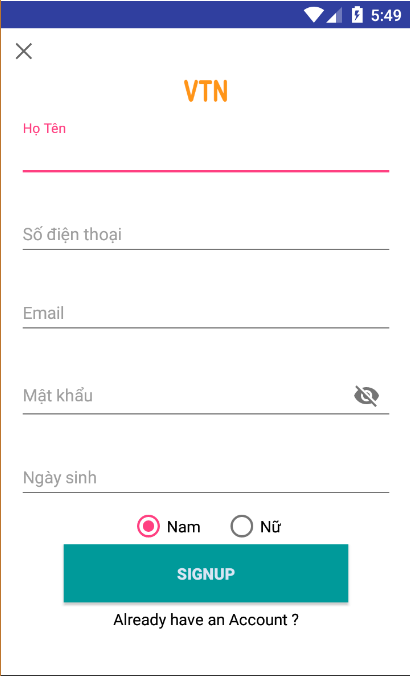
Màn hình khởi động (Splash) hiển thị khi người sử dụng click vào app. **(Võ Tùng Nghĩa)**

* 1. **Màn hình đăng nhập**



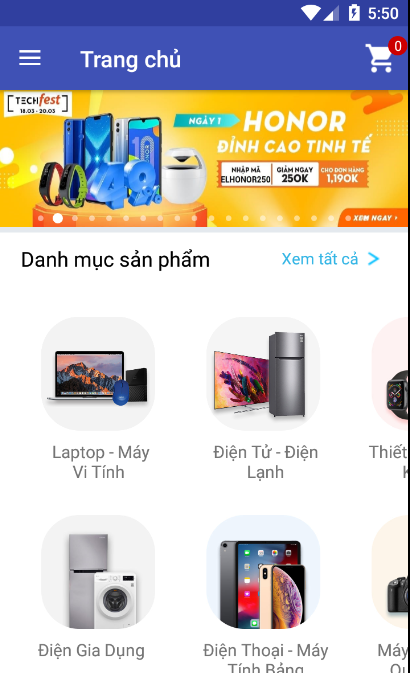
Màn hình đăng nhập khi cần mua hàng hay cập nhập thông tin cá nhân **(Hồ Quang Việt)**

* 1. **Màn hình đăng ký**



Màn hình đăng ký giúp khách hàng tạo tài khoản mới nếu muốn mua hàng **(Hồ Quang Việt)**

* 1. **Màn hình trang chủ**



Màn hình trang chủ bao gồm :

- Thanh toolbar chứa giỏ hàng và nút mở menu

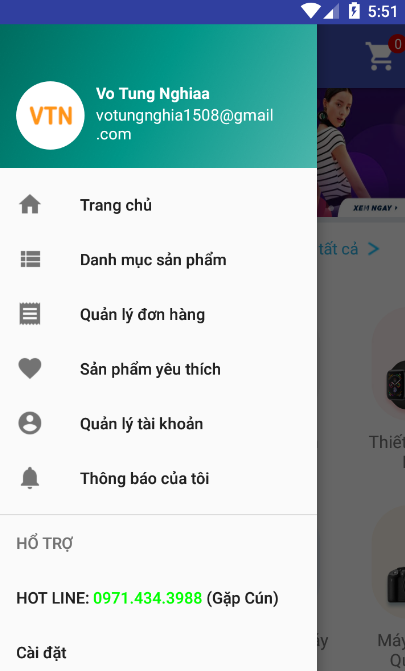
- Slide show dùng để quảng cáo các khuyến mãi

- List danh mục sản phẩm

- List danh sách sản phẩm

**(Võ Tùng Nghĩa)**

* 1. **Màn hình Menu**

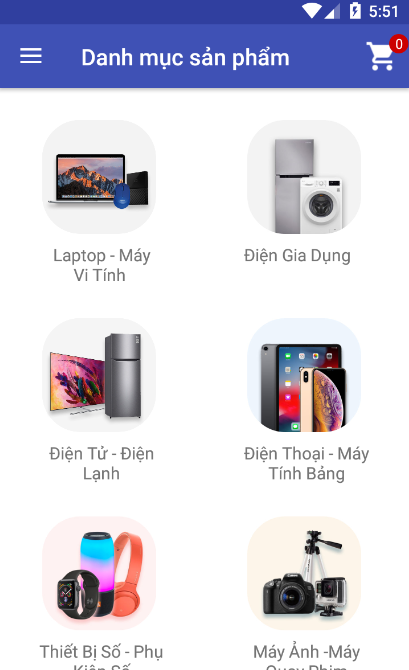


Màn hình menu hiện thị thông tin tài khoản đang đăng nhập.

Các danh mục, các màn hình khác .

**(Võ Tùng Nghĩa)**

* 1. **Màn hình danh mục sản phẩm**



Màn hình hiển thị tất cả các loại sản phẩm có.

**(Đậu Anh Tuấn)**

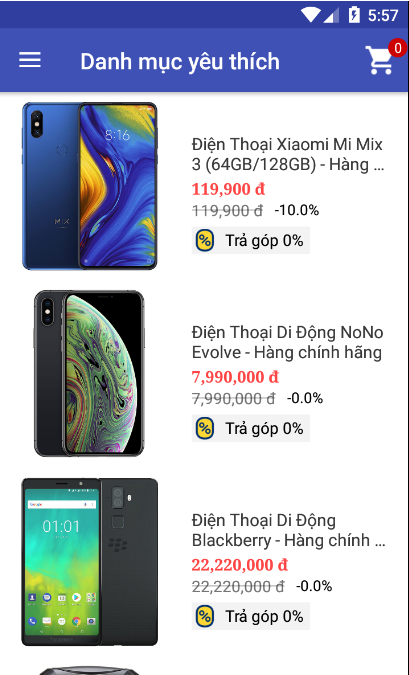
* 1. **Màn hình quản lý đơn hàng**

****

Màn hình hiển thị tất cả các đơn hàng đã đặt của khách hàng.

**(Võ Tùng Nghĩa)**

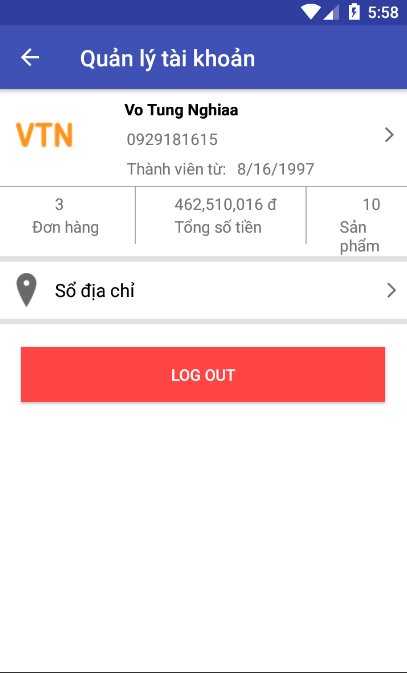
* 1. **Màn hình sản phẩm yêu thích**



Màn hình hiển thị tất cả các sản phẩm được thích bởi khách hàng.

**(Đậu Anh Tuấn)**

* 1. **Màn hình quản lý tài khoản**



Màn hình hiển thị tất cả các thông tin về tài khoản của khách hàng khách hàng.

**(Phạm Duy Thương)**

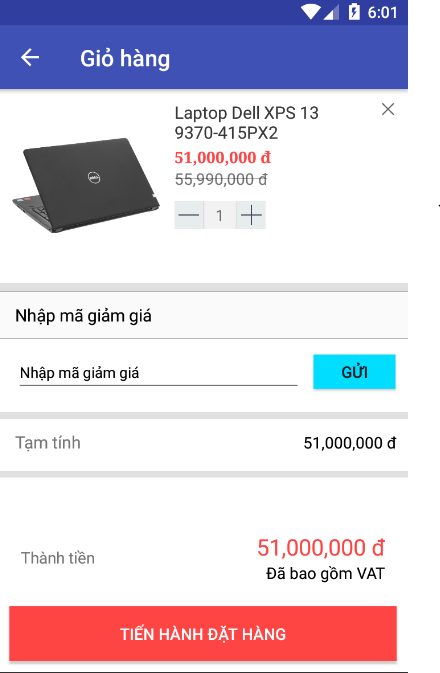
* 1. **Màn hình chi tiết sản phẩm**



Màn hình hiển thị chi tiết một sản phẩm có thể xử lý yêu thích và mua hàng trong này

**(Đậu Anh Tuấn)**

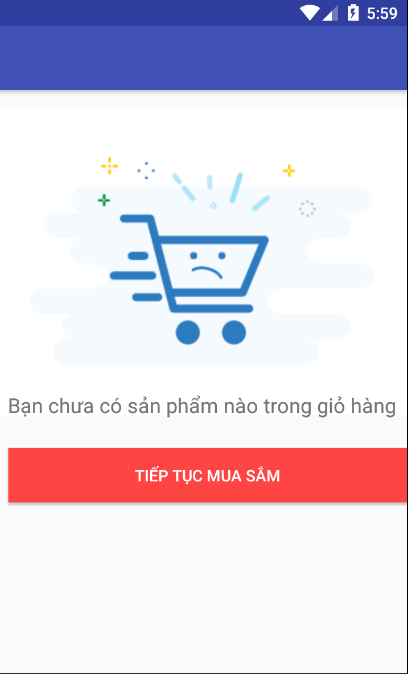
* 1. **Màn hình giỏ hàng**



Màn hình hiển thị tất cả các sản phẩm đã chọn mua

**(Nguyễn Phương Nam)**

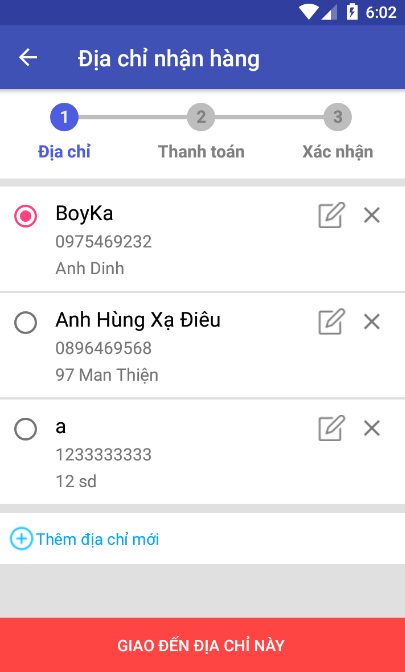
* 1. **Màn hình giỏ hàng trống**



Màn hình hiển thị thông báo là hiện tại không có sản phẩm nào được chọn mua

**(Nguyễn Phương Nam)**

* 1. **Màn hình thanh toán**



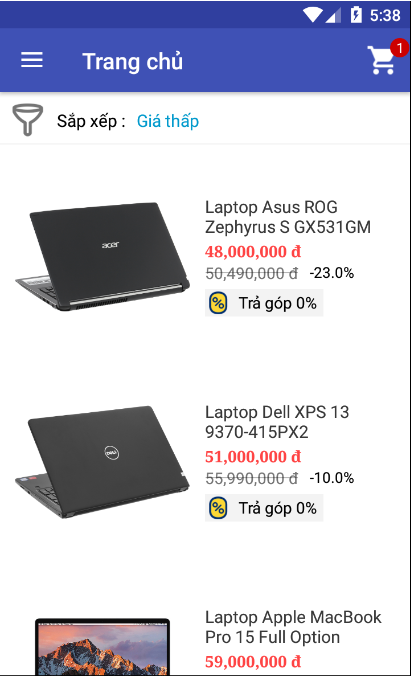
Màn hình hiển thị các bước thanh toán : địa chỉ, phương thức thanh toán và xác nhận.

**(Nguyễn Phương Nam)**

* 1. **Màn hình danh sách sản phẩm theo danh mục tương ứng**

Màn hình hiển thị danh sách các sản phẩm theo danh mục tương ứng.

**(Võ Tùng Nghĩa)**



1. **Thành phần, chức năng và hoạt động của từng màn hình**
   1. **Màn hình khởi động (Splash)**



Màn hình khởi động (Splash) hiển thị khi người sử dụng click vào app.

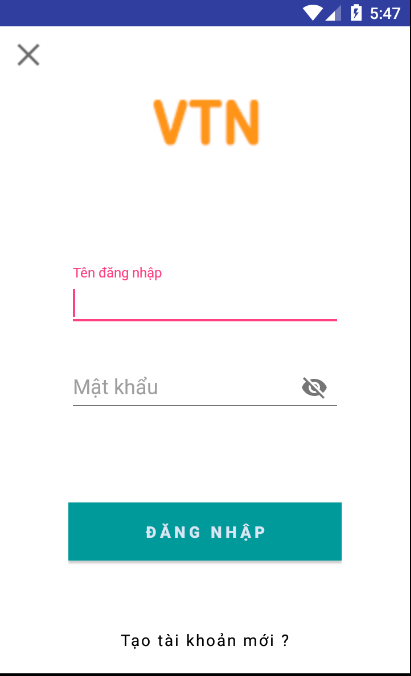
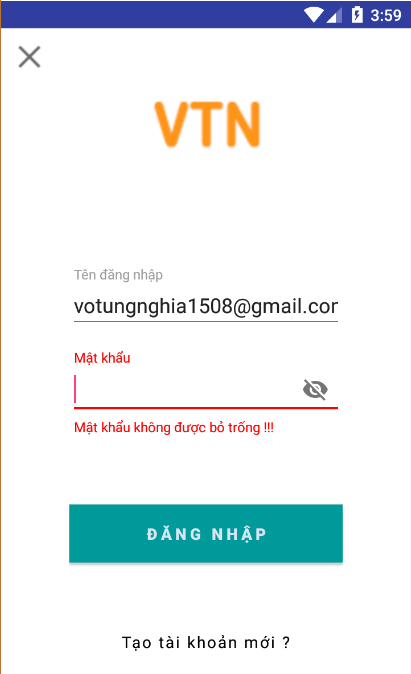
- Đầu tiên màn hình sẽ khởi động trong vòng 3s

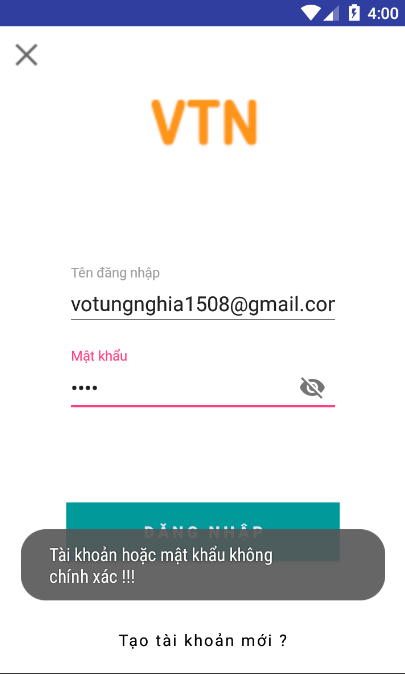
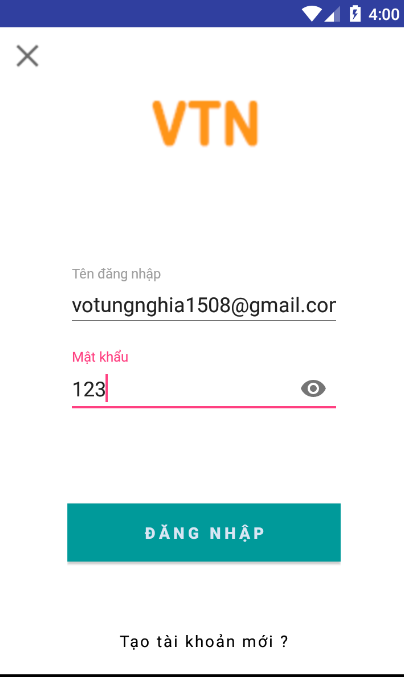
- Trong vòng 3s thì sẽ kiểm tra xem thiết bị của khác hàng có đang được kết nối mạng hay không.

- Nếu không có kết nối thì sẽ hiện ra một thông báo là “ Không có kết nối mạng “ SnackBar ở dưới cuối màn hình và ngược lại.

- Người dùng có thể kiểm tra lại kết nối và bấm vào “ THỬ LẠI ” để kiểm tra một lần nữa.

* 1. **Màn hình đăng nhập**

Màn hình đăng nhập khi cần mua hàng hay cập nhập thông tin cá nhân.

- Gồm : 2 EditText, 1 Button, 1 TextView, 1 ImageView

- Người dùng sẽ nhập “ Tên đăng nhập “ và “ Mật khẩu “.

- Khi click hình con mắt gạch xéo thì người dùng có thể nhìn thấy Mật khẩu mình nhập có đúng hay không.

- Khi Click vào Button “ Đăng Nhập “ sẽ kiểm tra xem có ô nào người dùng chưa nhập không nếu có sẽ báo lỗi.

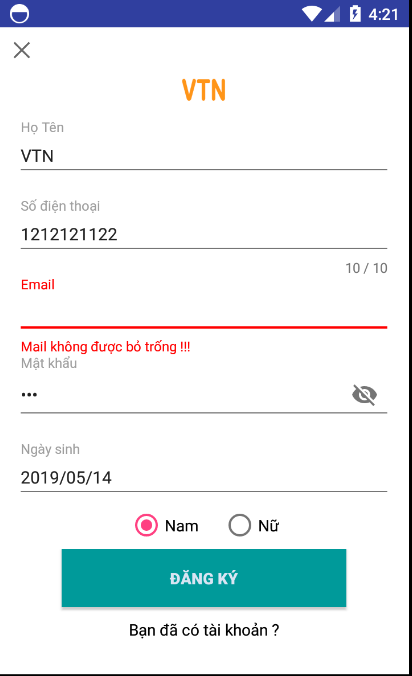
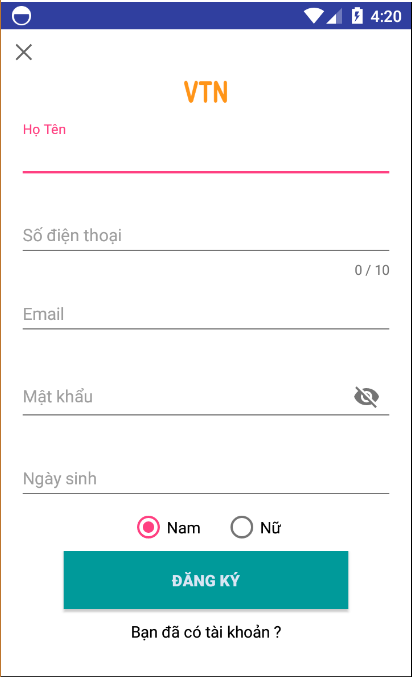
- Nếu người dùng đã nhập đầy đủ thì tiến hành gửi thông tin người dùng vừa nhập về Database bằng cách dùng Request API để kiểm tra xem thông tin người dùng nhập vào là đúng hay sai.

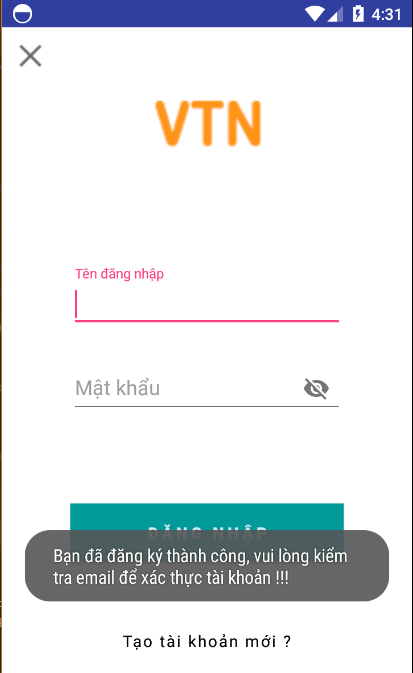
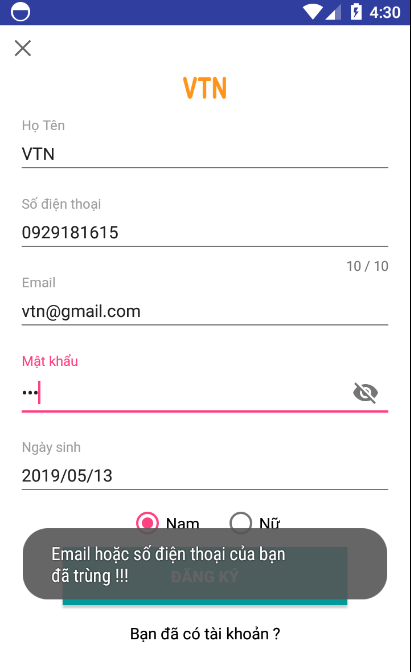
- Nếu ĐÚNG thì sẽ vào ứng dụng và bắt đầu thực hiện mua hàng. Nếu SAI sẽ hiện ra 1 thông báo bằng TOAST là " Tài khoản hoặc mật khẩu không chính xác !!!"

- Nếu tài khoản vừa đăng ký mà chưa được XÁC NHẬN bằng MAIL thì sẽ TOAST là "Tài khoản chưa được xác nhận !!!"

- Khi click vào TextView “ Tạo tài khoản mới ? “ sẽ chuyển sang màn hình Đăng Ký

* 1. **Màn hình đăng ký**





Màn hình đăng ký giúp khách hàng tạo tài khoản mới nếu muốn mua hàng.

- Gồm : 2 ImageView, 5 EditText, 2 RadioButton, 1 Button, 1 TextView

- Người dùng tiến hành nhập tất cả thông tin cần thiết vào các ô EditText.

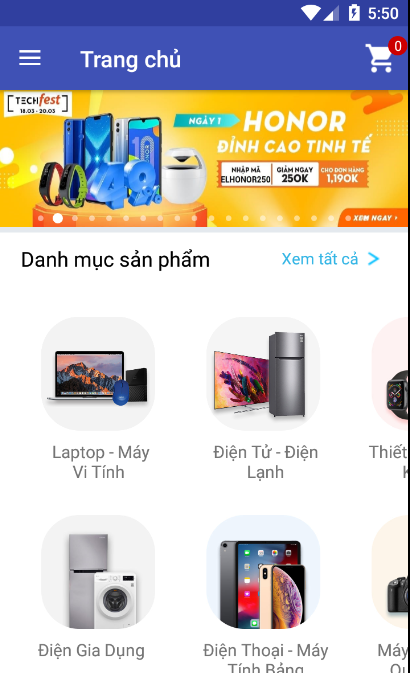
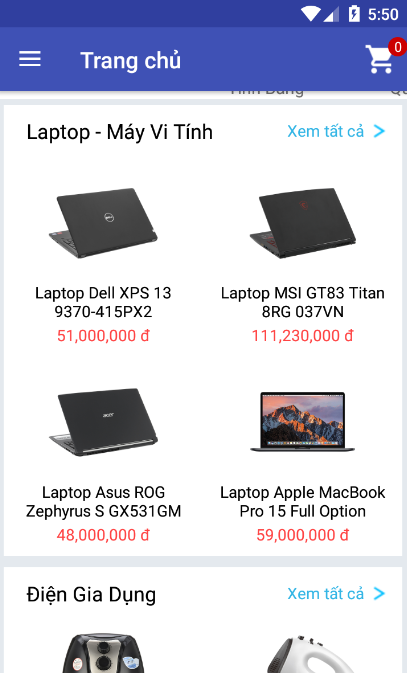
- Khi Click vào Button “ Đăng ký “ sẽ kiểm tra xem có ô nào người dùng chưa nhập không, nếu có sẽ báo lỗi.

- Nếu người dùng đã nhập đầy đủ thì tiến hành gửi thông tin người dùng vừa nhập về Database bằng cách dùng Request API để kiểm tra xem thông tin người dùng nhập vào có trùng lặp không.

- Nếu KHÔNG TRÙNG thì sẽ tự động chuyển sang màn hình ĐĂNG NHẬP, và yêu cầu XÁC NHẬN sẽ được gửi vào MAIL . Nếu TRÙNG sẽ hiện ra 1 thông báo bằng TOAST là "Email hoặc số điện thoại của bạn đã trùng !!!"

- Khi click vào “Bạn đã có tài khoản ? “ thì chuyển sang màn hình ĐĂNG NHẬP.

* 1. **Màn hình trang chủ**

Màn hình trang chủ là màn hình đầu tiên sẽ xuất hiện khi mở app.

- Gồm : 1 ViewPager, 1 RecyclerView, 1 Nested RecyclerView

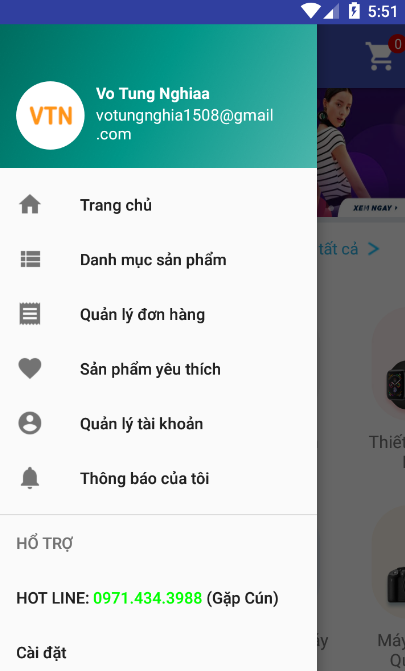
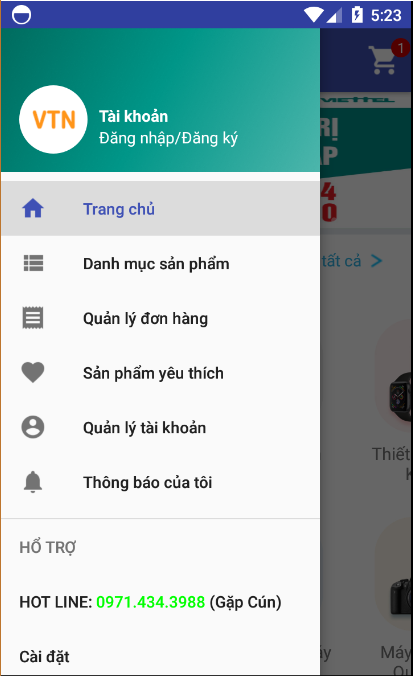
- Thanh toolbar chứa giỏ hàng và nút mở menu

- Slide show dùng ViewPager để trình chiếu quảng cáo các khuyến mãi tự động

- List danh mục sản phẩm : Dùng RecyclerView hiện thị các loại sản phẩm đầu tiên trong danh sách, nếu khách hàng muốn xem tất cả thì Click vào “ Xem tất cả > “

- List danh sách sản phẩm : Tương tự như trên.

* 1. **Màn hình Menu**

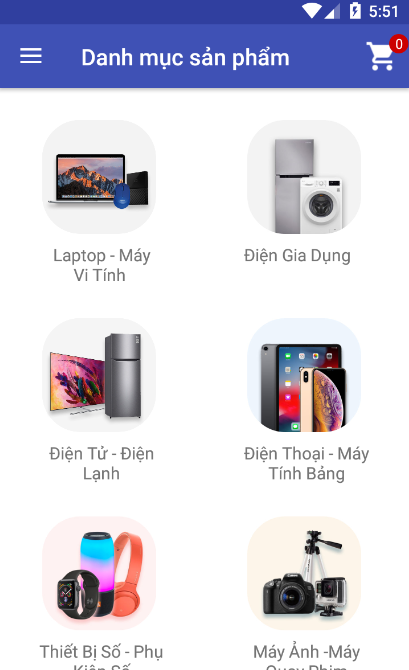
Màn hình MENU hiển thị thông tin và các màn hình để chọn

- Gồm : Header, Menu

- Phần Header chứa thông tin đăng nhập của khách hàng nếu khách hàng đã đăng nhập. Khi click vào nếu đã đăng nhập thì sẽ chuyển vào màn hình thông tin tài khoản của người dùng.Nếu chưa đăng nhập thì sẽ chuyển tới màn hình đăng nhập hoặc đăng ký

- Phần Menu chứa các Item tương ứng với các màn hình, khi click vào thì sẽ chuyển tới màn hình tương ứng.

* 1. **Màn hình danh mục sản phẩm**



Màn hình hiển thị tất cả các loại sản phẩm có.

- Gồm : 1 RecyclerView

- Dùng RecyclerView để hiển thị danh sách tất cả các loại sản phẩm hiện có theo 2 cột.

- Khi click vào 1 loại danh mục thì sẽ chuyển tới màn hình danh sách tất cả các sản phẩm tương ứng với danh mục đó.

* 1. **Màn hình quản lý đơn hàng**

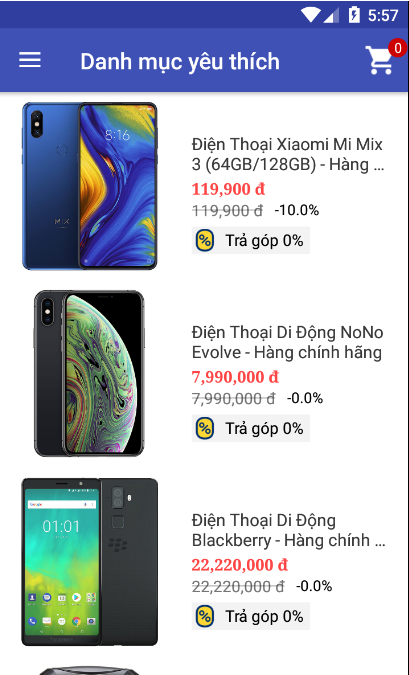


Màn hình hiển thị tất cả các đơn hàng đã đặt của khách hàng.

- Gồm : 1 RecyclerView

- Khi click vào 1 đơn hàng nào thì màn hình sẽ chuyển sang màn hình chứa thông tin chi tiết của đơn hàng.

* 1. **Màn hình sản phẩm yêu thích**

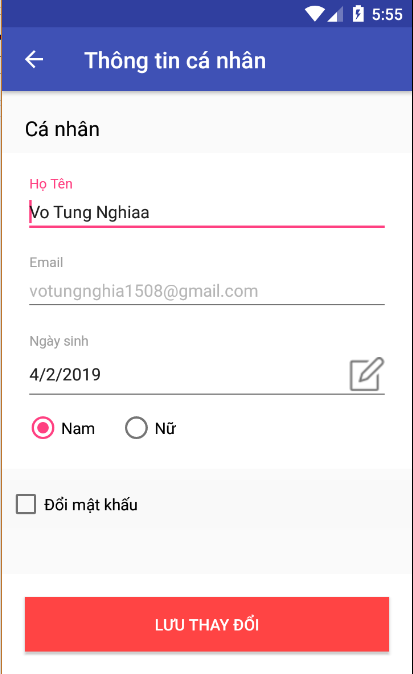
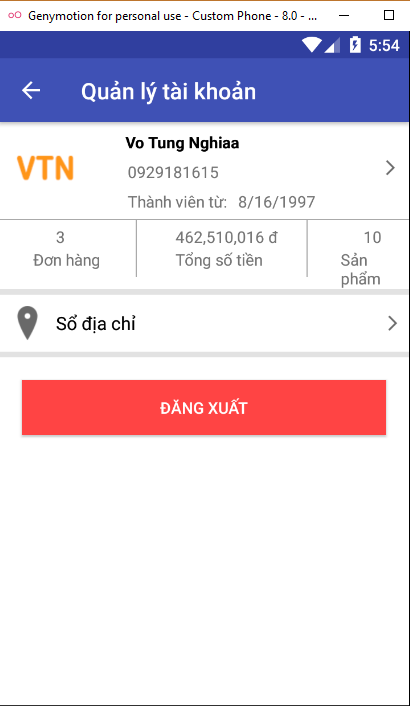


Màn hình hiển thị tất cả các sản phẩm được thích bởi khách hàng.

- Gồm : 1 RecyclerView

- Khi click vào 1 sản phẩm bất kì sẽ chuyển tới màn hình chi tiết sản phẩm đó chứa thông tin chi tiết về sản phẩm đó.

* 1. **Màn hình quản lý tài khoản**



Màn hình hiển thị tất cả các thông tin về tài khoản của khách hàng khách hàng.

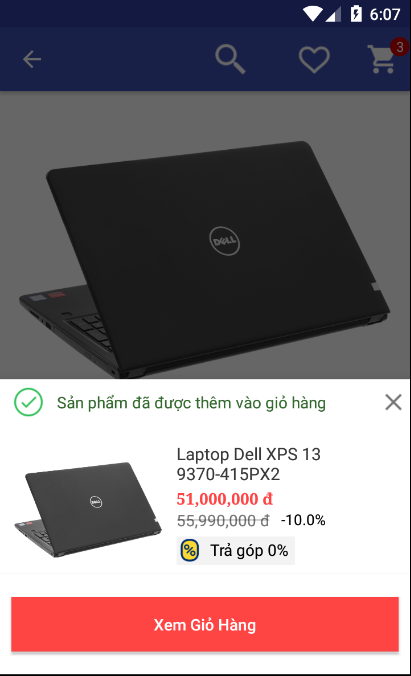
- Gồm : 2 ImageView, 1 Button

- Khi ta click vào ImageView “ > “ ở phía trên sẽ chuyển tới màn hình cập nhập tài khoản người dùng.

- Khi ta click vào ImageView “ > “ ở phía dưới “ Sổ địa chỉ “ sẽ chuyển tới màn hình chứa danh sách tất cả địa chỉ của người dùng hiện có.

- Khi ta click vào Button “ Đăng xuất “ thì chúng ta sẽ đăng xuất khỏi tài khoản mua hàng hiện tại và chuyển về màn hình chính.

* 1. **Màn hình chi tiết sản phẩm**



Màn hình hiển thị chi tiết một sản phẩm có thể xử lý yêu thích và mua hàng trong này.

- Gồm 2 Imageview, 2 Button, 1 WebView, TextViews

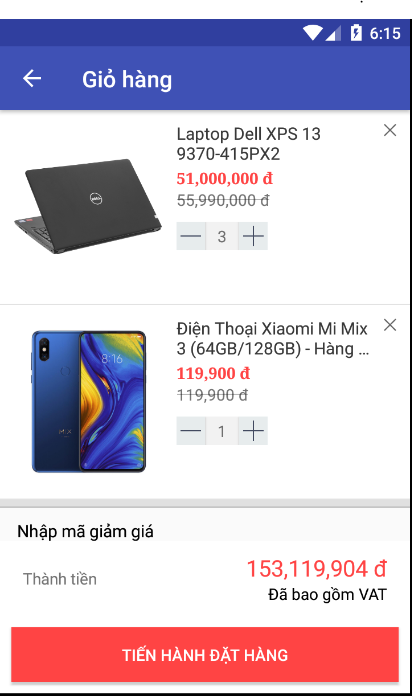
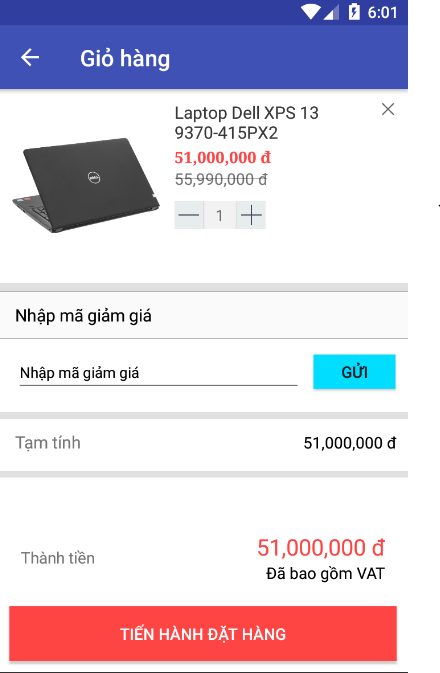
- Khi ta click ImageView hình trái tim trên thanh Toolbar nếu người dùng chưa yêu thích sản phẩm thì sản phẩm sẽ được thêm vào Danh mục sản phẩm yêu thích. Nếu sản phẩm đã được yêu thích rồi thì sẽ xóa sản phẩm đó khỏi danh mục yêu thích.

- Khi ta click vào Button “ Thêm vào giỏ hàng “ thì sản phẩm sẽ chỉ thêm vào giỏ hàng chứ không chuyển vào màn hình giỏ hàng luôn. Khi click sẽ có hiệu ứng (Animation) là di chuyển (Translate), làm mờ (Alpha) và thu nhỏ (Scale)

- Khi ta click vào Button “ Chọn mua “ thì sẽ hiển thị ra một trang/tờ (Bottom Sheet) từ dưới đáy của màn hình lên cho phép ta lựa chọn mua tiếp hay vào màn hình giỏ hàng

- WebView dùng để hiển thị thông tin chi tiết của sản phẩm thông qua HTML

* 1. **Màn hình giỏ hàng**



Màn hình hiển thị tất cả các sản phẩm đã chọn mua.

- Gồm : 1 RecyclerView, 1 Button, TextViews

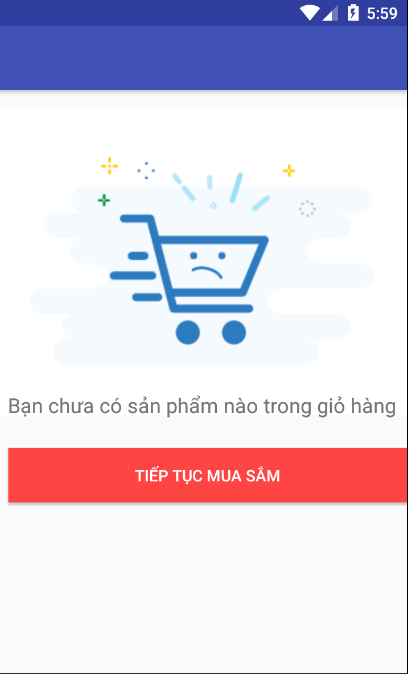
- Dùng RecyclerView để hiển thị danh sách các sản phẩm người dùng đã chọn để mua.

- Người dùng có thể lựa chọn thêm sản phẩm bằng cách bấm dấu “ + “ hoặc giảm bớt sản phẩm nếu cần.

- Người dùng có thể loại bỏ sản phẩm đã chọn bằng cách click nút “ X “ bên cạnh mỗi sản phẩm tương ứng.

- Khi click vào Button “ Tiến hành đặt hàng “ sẽ chuyển tới màn hình thanh toán.

* 1. **Màn hình giỏ hàng trống**



Màn hình hiển thị thông báo là hiện tại không có sản phẩm nào được chọn mua.

- Gồm: 1 ImageView, 1 Button

- Khi khách hàng xóa hết tất cả sản phẩm hoặc click vào icon giỏ hàng mà chưa chọn sản phẩm nào sẽ hiện thị ra màn hình như v.

- Khi click vào button “ Tiếp tục mua sắm “ sẽ chuyển về màn hình trang chủ.

* 1. **Màn hình thanh toán**
     1. **Màn hình địa chỉ**

Màn hình hiển thị các bước thanh toán : địa chỉ, phương thức thanh toán và xác nhận.

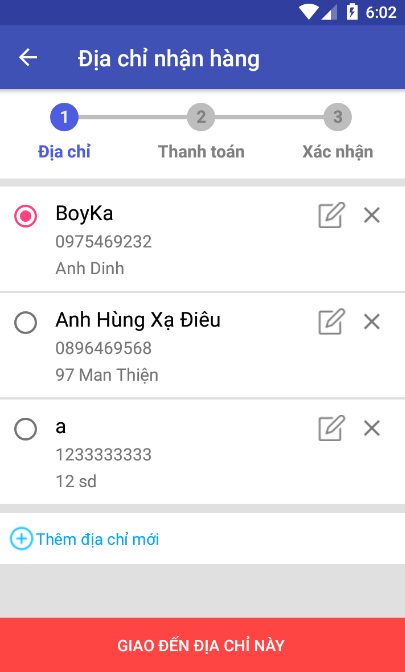
- Gồm : 1 StateProgressBar, 1 RecyclerView, 1 ImageView, 1 Button

- Thanh StateProgressBar dùng để hiện thị các giai đoạn cần thao tác, và hiển thị cho người dùng biết mình đang ở giai đoạn nào.

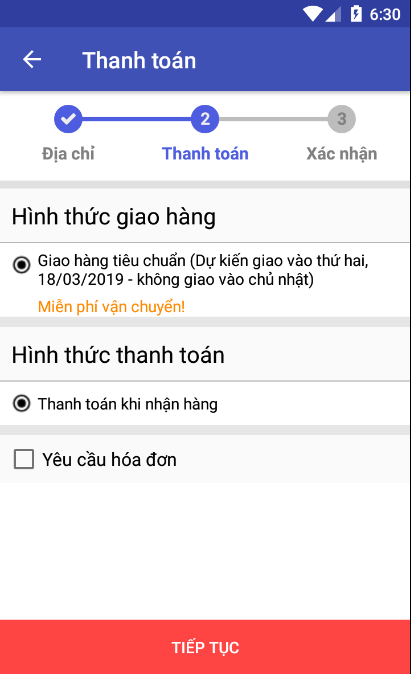
- Dùng RecyclerView để hiển thì danh sách địa chỉ cho người dùng chọn địa chỉ giao hàng.

- Khi click vào ImageView “ Thêm địa chỉ mới ” sẽ chuyển sang màn hình thêm địa chỉ.

- Khi click vào Button “ Giao đến địa chỉ này “ sẽ chuyển đến giai đoạn tiếp theo là Thanh Toán.



* + 1. **Màn hình hình thức thanh toán**



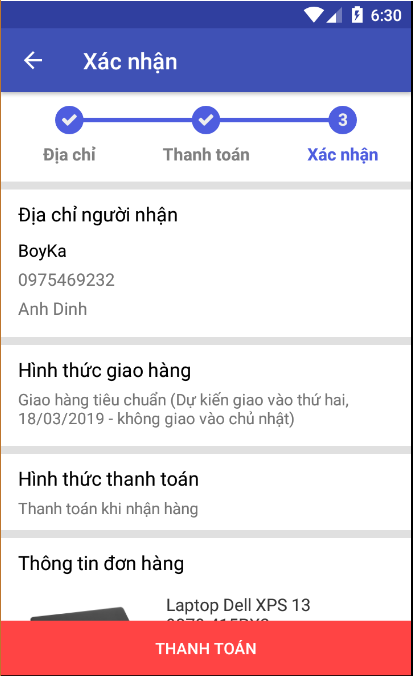
Màn hình này hiển thị thời gian giao hàng dự kiến và lựa chọn hình thức giao hàng.

- Gồm : 1 StateProgressBar, 1 Button

- Thanh StateProgressBar dùng để hiện thị các giai đoạn cần thao tác, và hiển thị cho người dùng biết mình đang ở giai đoạn nào.

- Khi click vào Button “ Tiếp tục “ sẽ chuyển đến giai đoạn tiếp theo là Xác nhận.

* + 1. **Màn hình xác nhận**



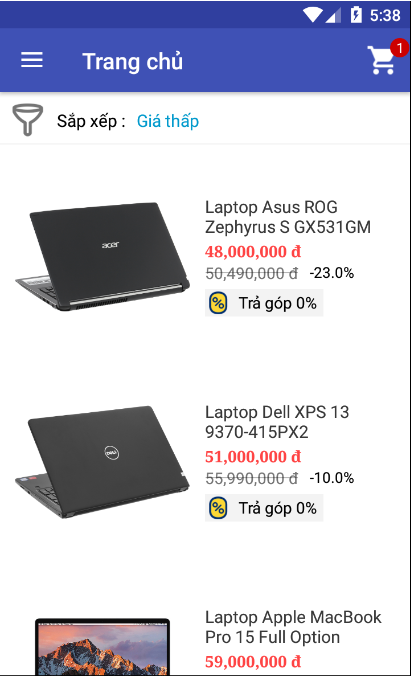
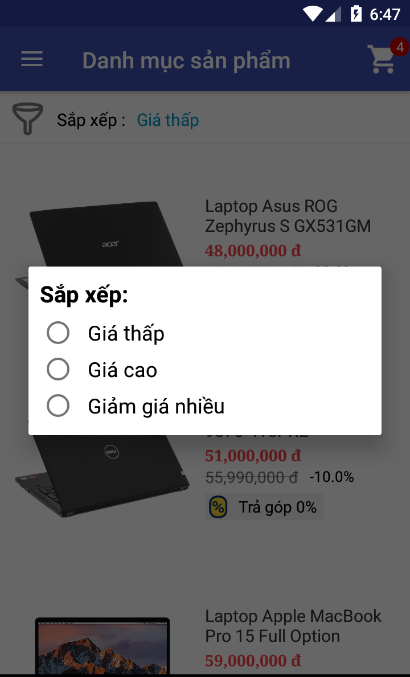
Màn hình này hiển thị chi tiết về địa chỉ người nhận, hình thức giao hàng, hình thức thanh toán.

- Gồm : 1 StateProgressBar, 1 Button

- Thanh StateProgressBar dùng để hiện thị các giai đoạn cần thao tác, và hiển thị cho người dùng biết mình đang ở giai đoạn nào.

- Khi click vào Button “ Thanh toán “ sẽ chuyển tất cả thông tin người dùng cung cấp cho đơn hàng về Database bằng cách gửi Request API

* 1. **Màn hình danh sách sản phẩm theo danh mục tương ứng**

Màn hình hiển thị danh sách các sản phẩm theo danh mục tương ứng.

- Gồm : 1 TextView, 1 RecyclerView

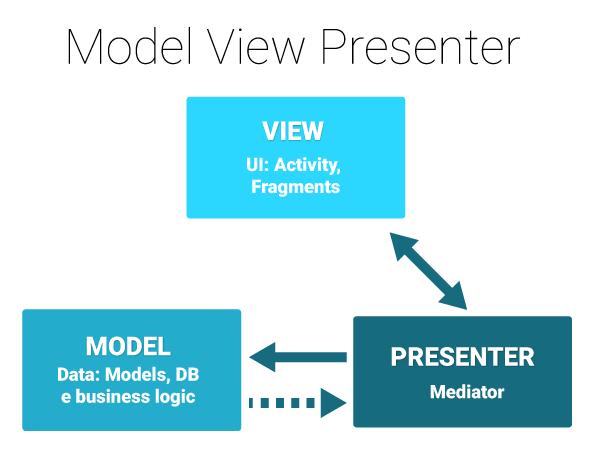
- Khi ta click vào 1 sản phẩm thì sẽ chuyển tới màn hình chi tiết về thông tin của sản phẩm đó.

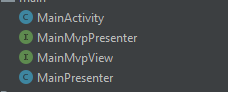
- Khi ta click vào TextView “ Sắp xếp “ thì sẽ hiện lên 1 Dialog cho phép ta lựa chọn sắp xếp các thứ tự sản phẩm như thế nào vd : Giá tăng dần, Giá giảm dần, Giá giảm nhiều nhất

1. **Cài đặt và kiểm thử** 
   1. **Cơ sở lý thuyết**
      1. **Mô hình MVP**

Công dụng : MVP là một mô hình kiến trúc hướng giao diện người dùng, được thiết kế để tạo thuận lợi cho việc kiểm thử đơn vị (unit testing) và tăng tính tách biệt giữa tầng dữ liệu và tầng hiển thị dữ liệu trong mô hình MVC.

Mô hình MVP cho phép tách tầng trình diễn (Presenter) ra khỏi tầng dữ liệu (Model), vì vậy tương tác với giao diện được tách biệt với cách chúng ta biểu diễn nó trên màn hình (View), hay nói cách khác, tất cả logic khi người dùng tương tác được tách ra và đưa vào tầng trình diễn. Thiết kế lý tưởng nhất là với cùng một logic được áp dụng cho nhiều View khác nhau và hoán đổi được cho nhau.



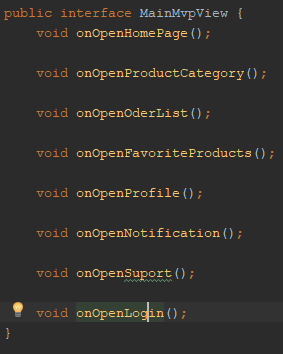
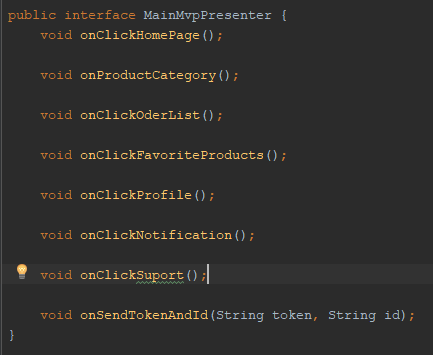
 Cách dùng :

- Cơ bản chúng ta sẽ chia các thành phần như bên gồm 2 interface và 2 class

- Ta tiếp tục thiết lập các phương thức cho 2 interface MainMvpPresenter và MainMvpView.

+ MainMvpView : Chúng ta dùng các phương thức này ở MainActivity để thay đổi giao diện .

+ MainMvpPresenter : Chúng ta dung các phương thức này ở MainPresenter để kích hoạt những phương thức của MainMvpView cần thiết.



- Cuối cùng chúng ta implements các interface tương ứng để xử lý các phương thức được định nghĩa sẵn.

C:\Users\votun\Desktop\New folder\Caspture.PNGC:\Users\votun\Desktop\New folder\a.PNG

* + 1. **Retrofit 2.0 Library**

- Việc phát triển thư viện HTTP type-safe để giao tiếp với Rest API có thể thực sự là một điều khó khăn vì bạn phải xử lí rất nhiều các chức năng chẳng hạn như : tạo kết nối, lưu trữ cache, thử lại các yêu cầu thất bại, phân tích luồng phản hồi, xử lí lỗi và nhiều hơn nữa. Nhưng mặt khác Retrofit được sinh ra để giải quyết một cách dễ dàng những công việc này và từ đó sẽ giúp bạn tiết kiệm thời gian hơn rất nhiều

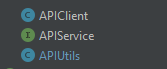
Công dụng : Retrofit là một thư viện vô cùng hữu ích cho việc kết nối internet và nhận dữ liệu từ server một cách dễ dàng và viết code theo mô hình chuẩn RESTFul Webservices.Giống như hầu hết các phần mềm mã nguồn mở khác, Retrofit được xây dựng dựa trên một số thư viện mạnh mẽ và công cụ khác. Retrofit không tích hợp bất kỳ một bộ chuyển đổi JSON nào để phân tích từ JSON thành các đối tượng Java. Thay vào đó nó đi kèm với các thư viện chuyển đổi JSON để xử lý điều đó



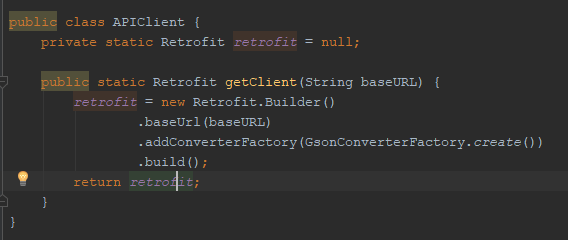
Cài đặt :

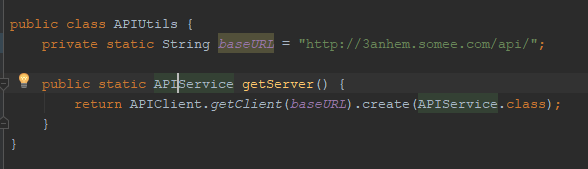
implementation 'com.squareup.retrofit2:retrofit:2.5.0'  
implementation 'com.squareup.retrofit2:converter-gson:2.4.0'  
implementation 'com.google.code.gson:gson:2.8.5'

Cách dùng :

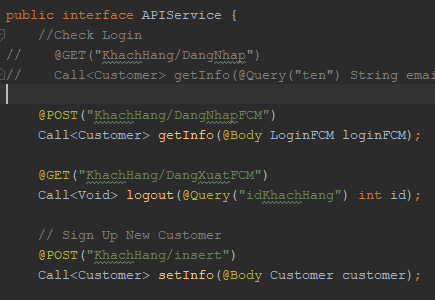
- Đầu tiên chúng ta sẽ cài đặt trước ta dùng 2 class và 1 interface như hình dưới :

- Tiếp theo chúng ta sẽ setup cho 2 class trên





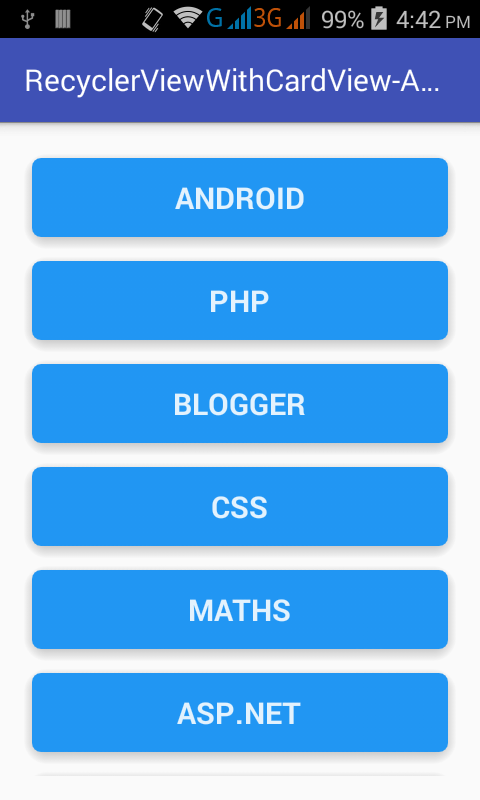
- Về phí Interface ApiService chúng ta sẽ chứa những phương thức Get, Post,Delete để thực hiện request lên server



* + 1. **RecyclerView Library**

- Công dụng: RecyclerView là một ViewGroup mới, nó là sự kế thừa và nâng cao của ListView và GridView được hỗ trợ trong support-v7 version. Một trong những ưu điểm nổi trội của Recycler là nó có khả năng mở rộng tốt hơn, nó có thể hiển thị dữ liệu theo cả chiều ngang và chiều dọc.

**\*\* So sánh khác biệt cơ bản với ListView \*\***

* \*Required ViewHolder in Adapters \* ListView không yêu cầu bắt buộc sử dụng ViewHolder để cải thiện hiệu suất. Ngược lại khi bạn sử dụng \* RecyclerView thì điều đó là bắt buộc.
* \* Customizable Item Layouts \* Để hiển thị dữ liệu theo chiều ngang thì bạn không thể sử dụng ListView mà bạn cần đến HorizontalScrollView. Nhưng với Recycler thì chúng ta có tùy chọn RecyclerView.LayoutManager để chỉ định cách hiển thị dữ liệu.
* \* Manual Data Source \* ListView có 2 Adapter để bạn có thể tùy chỉnh điều hướng dữ liệu là \*ArrayAdapter \* và CursorAdapter, còn với RecyclerView thì bạn bắt buộc phải sử dụng RecyclerView.Adapter. 

- Cài dặt :

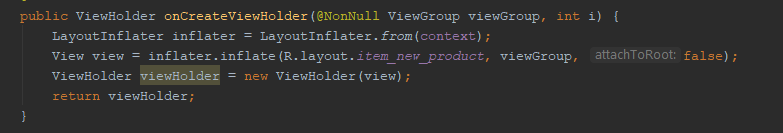
* implementation 'com.android.support:recyclerview-v7:28.0.0'

- Cách dùng :

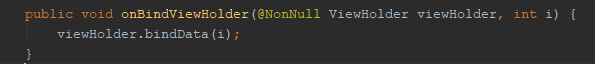
+ Đầu tiên bạn cần setup 1 adapter cho 1 recyclerview

C:\Users\votun\Desktop\New folder\Captưure.PNG

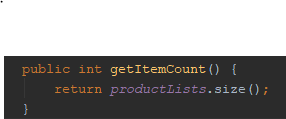
Dùng để gán giao diện cho một phần tử của RecyclerView



Dùng để gán dữ liệu từ listData vào viewHolder

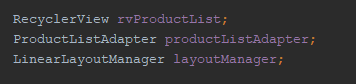


Trả về số lượng phần tử

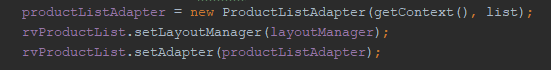


- Ở Activity ta cần

Khởi tạo các phần tử cần thiết



Khi ta đã có dữ liệu ta DB ta sẽ truyền 1 Context, 1 listData vào adapter, set layout cho recycelerview cuối cùng là set adapter cho nó



* + 1. **Glide Library**

- Công dụng : Glide là một thư viện open source hỗ trợ load ảnh trên Android. Dùng Glide sẽ đơn giản hóa các công việc mà bạn cần làm khi sử dụng một bức ảnh trong Android đi rất nhiều. Chúng ta không cần quan tâm đến việc decoding, memory and disk caching mà thay vào đó chỉ cần sử dụng interface rất đơn giản từ Glide.

- Glide hỗ trợ fetching, decoding và hiển thị cả ảnh tĩnh hoặc ảnh động đó là điểm nổi bật của nó so với các thư viện load ảnh khác cho android. Glide có thể được dễ dàng đưa vào bất kỳ mô hình mạng nào. Mặc định Glide sử dụng mô hình custom của HttpUrlConnection tuy nhiên chúng ta có thể dễ dàng cho nó hoạt động với Volley hoặc OkHttp

- Glide tính đến hai khía cạnh chính của hiệu suất tải hình ảnh trên Android.

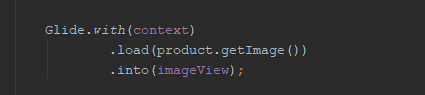
* Tốc độ hình ảnh có thể được decoded
* Số lượng jank phát sinh khi trong khi decoding hình ảnh. Để người dùng có trải nghiệm tuyệt vời với ứng dụng, hình ảnh không chỉ xuất hiện nhanh mà còn phải gây ra nhiều jank và lặp từ Main Thread hoặc gây ra nhiều rác quá mức. Glide thực hiện một số bước để đảm bảo tải hình ảnh nhanh và mượt mà nhất có thể trên Android:
* Tự động downsampling và giảm thiểu chi phí lưu trữ cộng với thời gian decode.
* Tích cực tái sử dụng resource như mảng byte và bitmap giảm thiểu garbage collections và phân mảnh heap.
* Tích hợp sâu vào vòng đời và đảm bảo chỉ yêu cầu đối với Activity và Fragment được ưu tiên, và các ứng dụng giải phóng tài nguyên khi cần thiết



- Cài đặt :

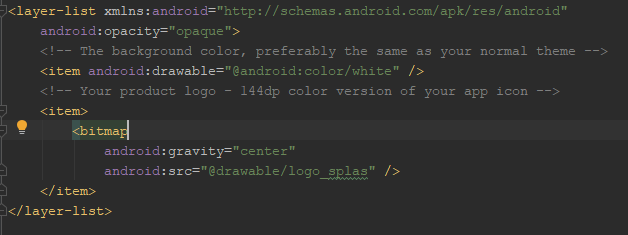
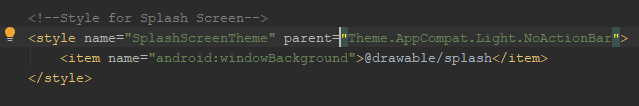
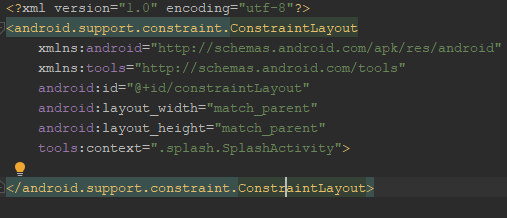
implementation 'com.github.bumptech.glide:glide:4.8.0'  
annotationProcessor 'com.github.bumptech.glide:compiler:4.8.0'

- Cách dùng :



* 1. **Cài đặt màn hình** 
     1. **Màn hình Splash**

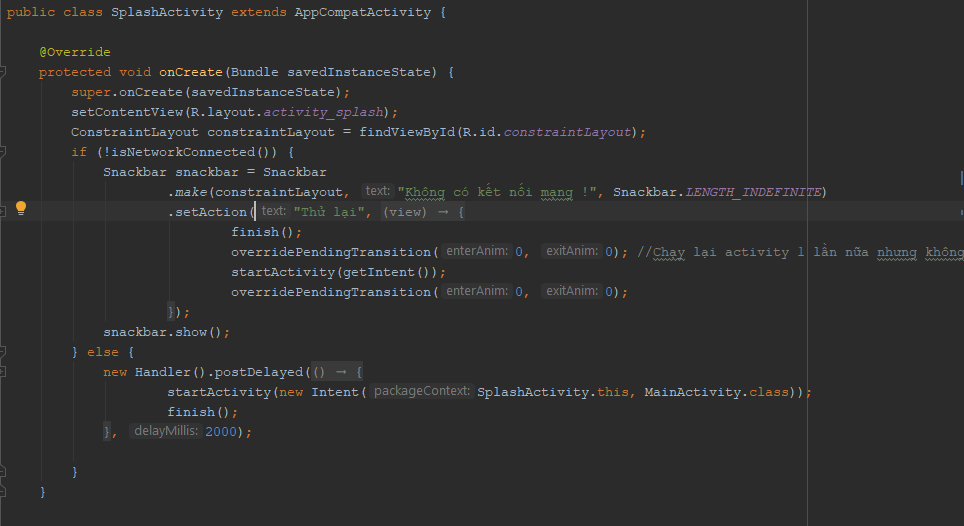
1. **Giao diện**



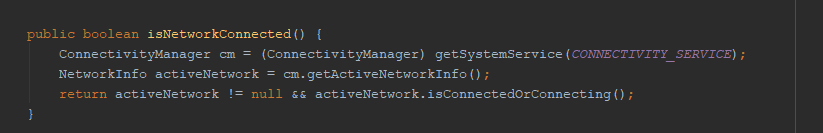
1. **Xử lí**

- Kiểm tra nếu không có kết nối mạng thì sẽ show ra 1 cái Snackbar thông báo ở cuối màn hình

- Người dùng bấm “ Thử lại “ để kiểm tra kết nối mạng 1 lần nữa

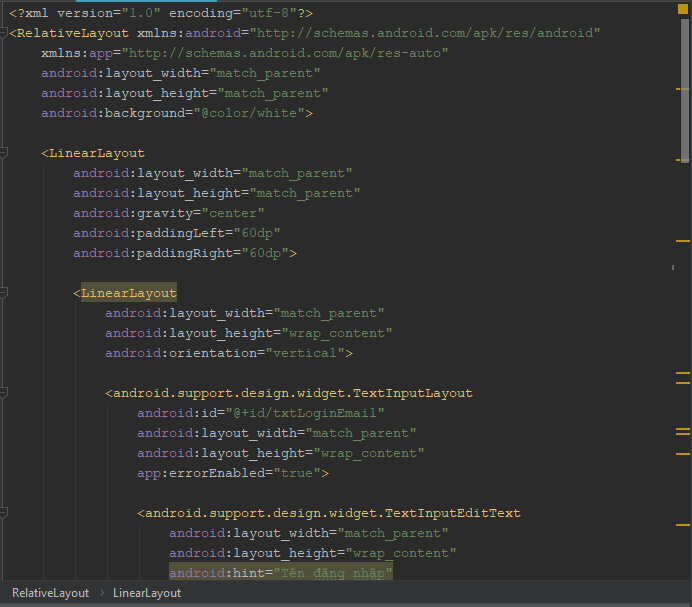


- Hàm kiểm trả thiết bị có kết nối mạng không



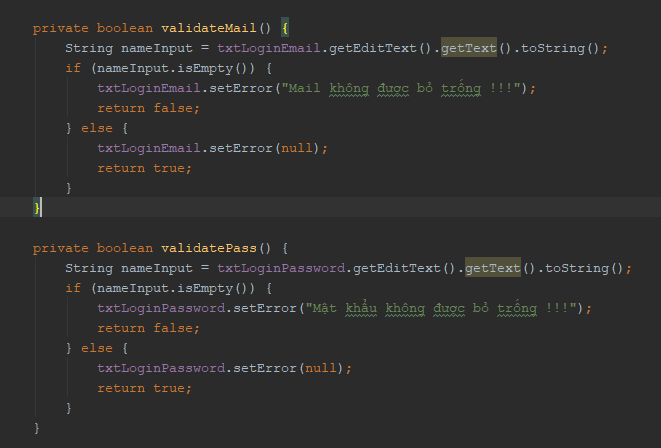
* + 1. **Màn hình đăng nhập**

1. **Giao diện**

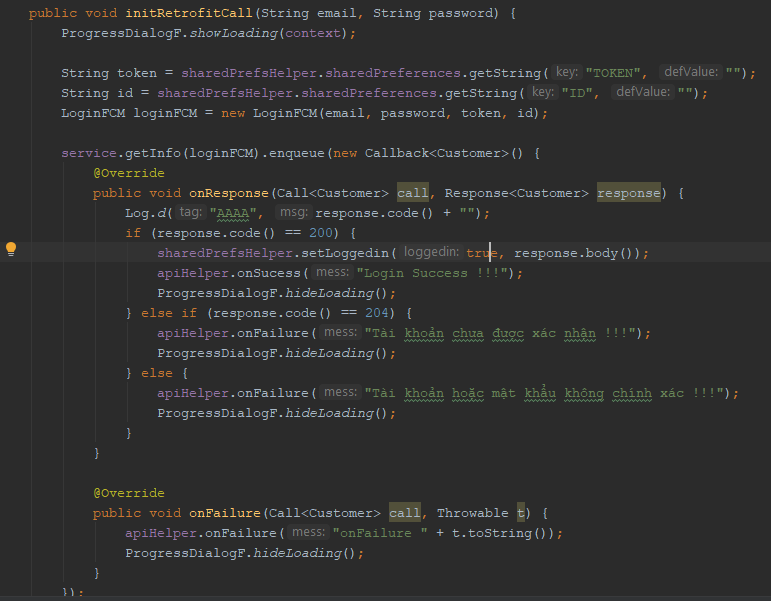


1. **Xử lý**

- Hàm kiểm tra người dùng có nhập thiếu dữ liệu ở Tên Đăng Nhập hay Mật Khẩu không.

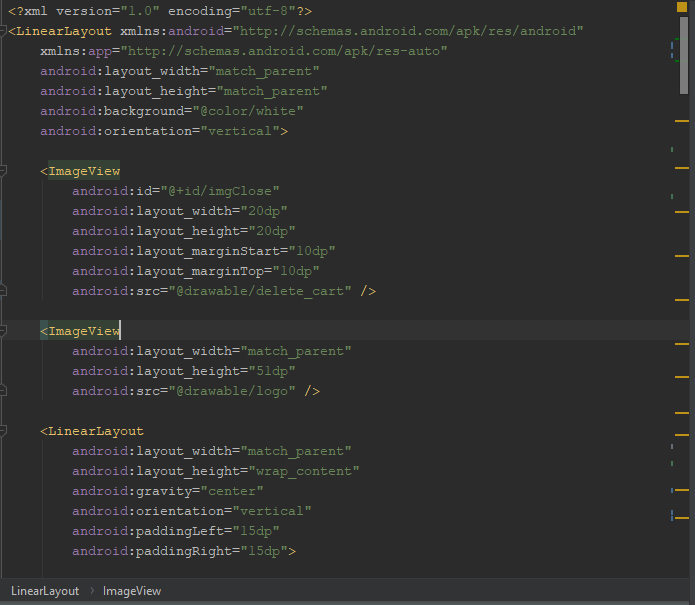


- Hàm gửi yêu cầu (Request) API về Database để kiểm tra thông tin đăng nhập



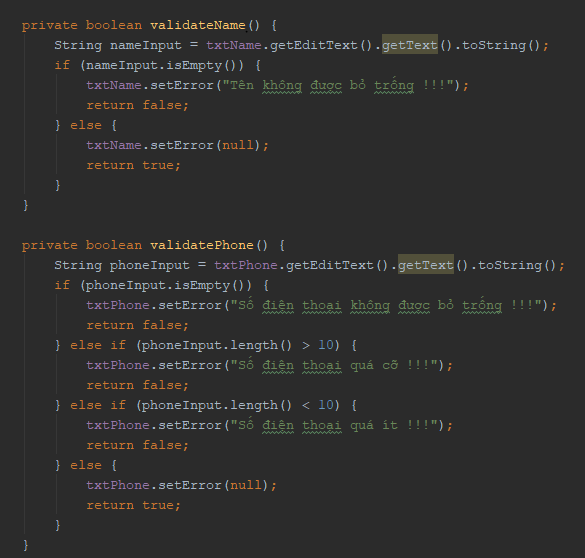
* + 1. **Màn hình đăng ký**

1. **Giao diện**

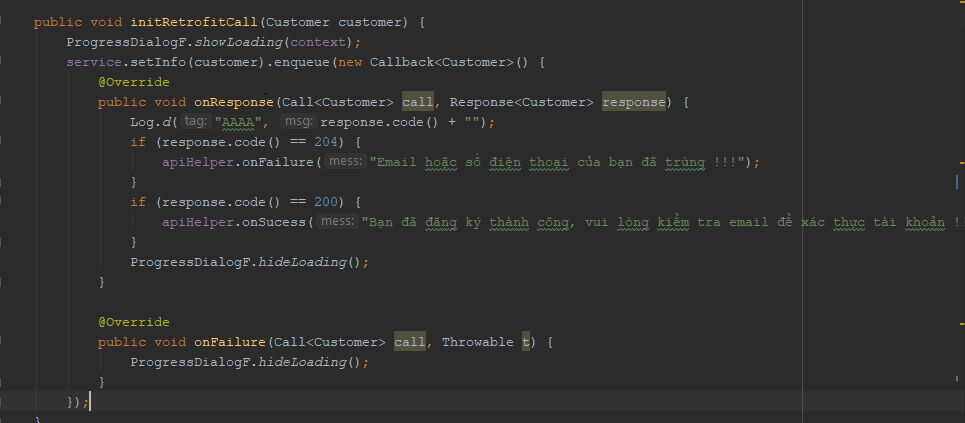


1. **Xử lý**

- Hàm kiểm tra người dùng có nhập thiếu dữ liệu ở các ô nhập liệu không.

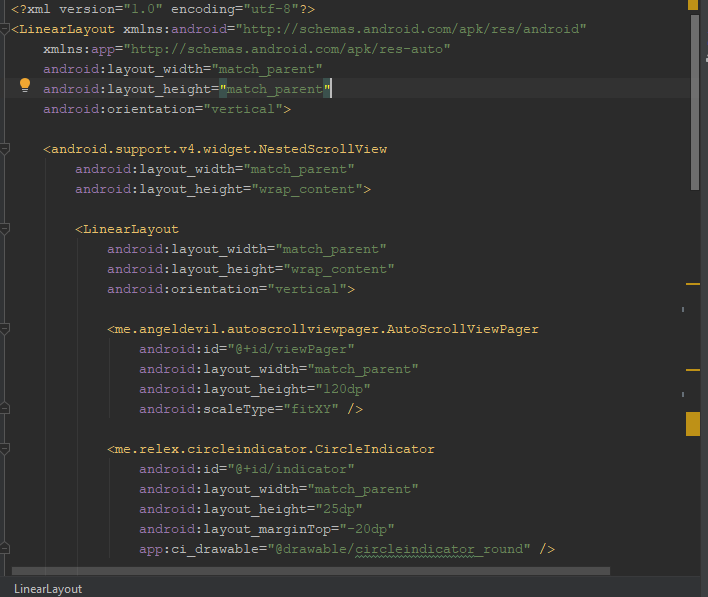


- Hàm gửi yêu cầu (Request) API về Database để kiểm tra thông tin đăng ký.



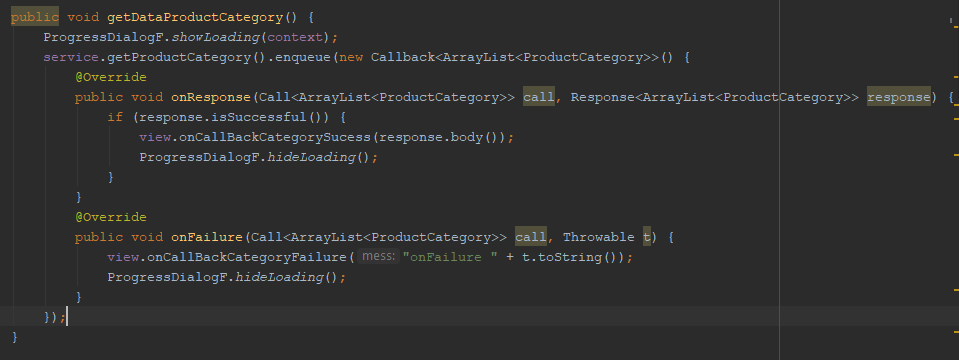
* + 1. **Màn hình trang chủ**

1. **Giao diện**

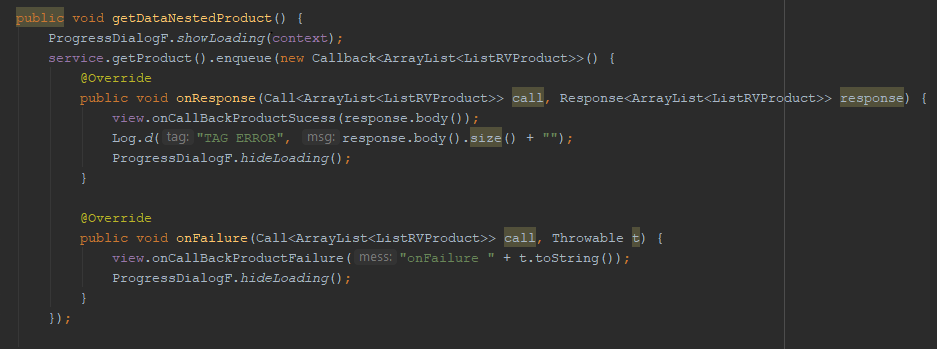


1. **Xử lý**

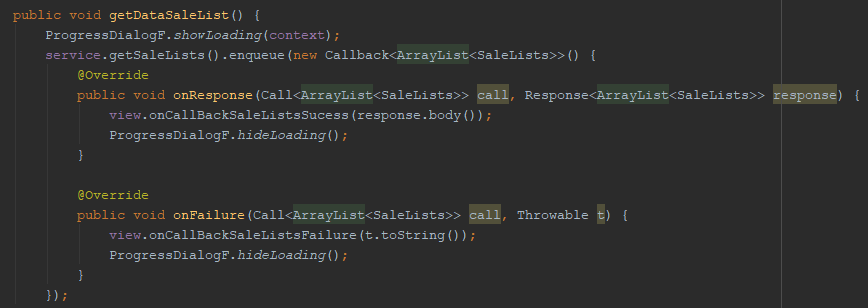
- Hàm gửi yêu cầu (Request) API về Database để lấy danh sách tất cả loại sản phẩm



- Hàm gửi yêu cầu (Request) API về Database để Danh sách sản phẩm theo danh mục

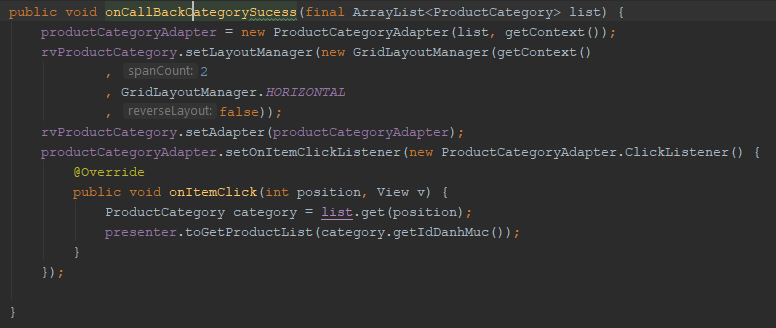


- Hàm gửi yêu cầu (Request) API về Database để Danh sách khuyến mãi



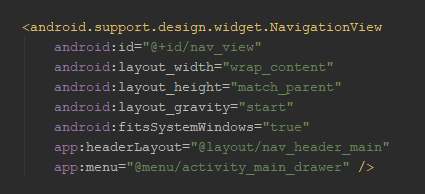
- Hàm này dùng để thiết lập các yêu cầu của RecyclerView để show dữ liệu lên

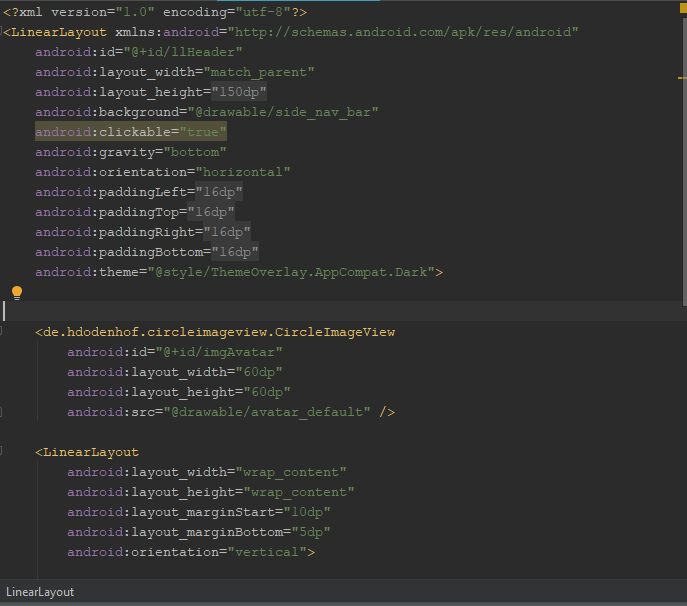
Vd : Thiết lập LayoutManager, set Adapter



* + 1. **Màn hình Menu**

1. **Giao diện**

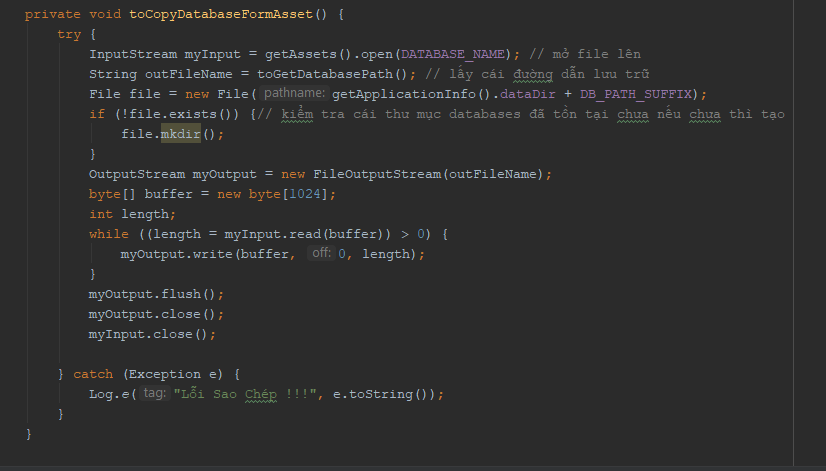




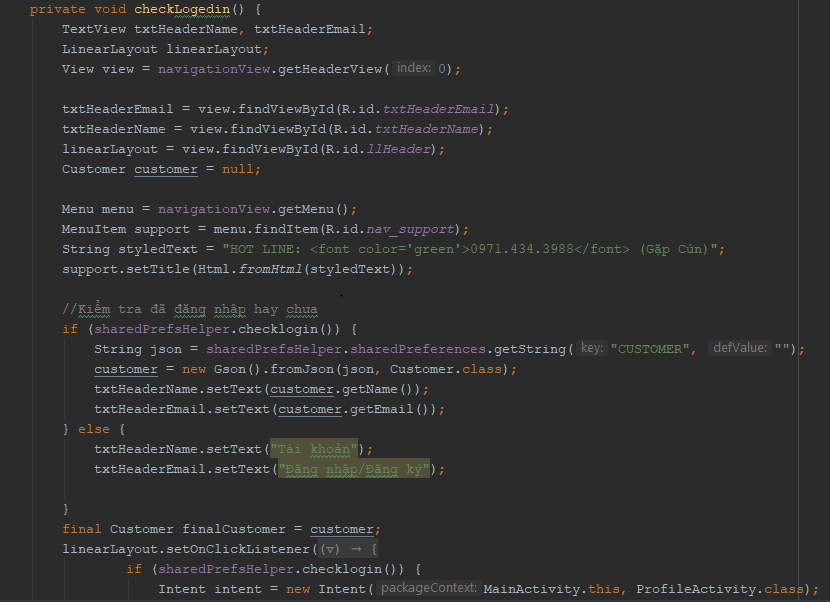


1. **Xử lý**

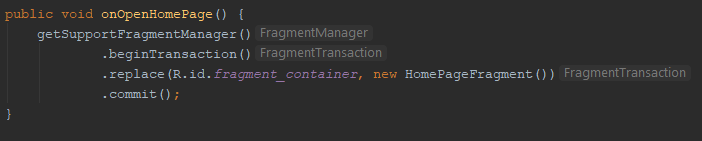
- Hàm này sẽ Copy file Database ở thư mục Assets vào bộ nhớ của điện thoại người dùng nếu người dùng tải app lần đầu tiên.



- Hàm này dùng để kiểm tra xem có tài khoản nào đã đăng nhập rồi thì không cần đăng nhập nữa (Lưu lại thông tin đăng nhập cho lần sau)

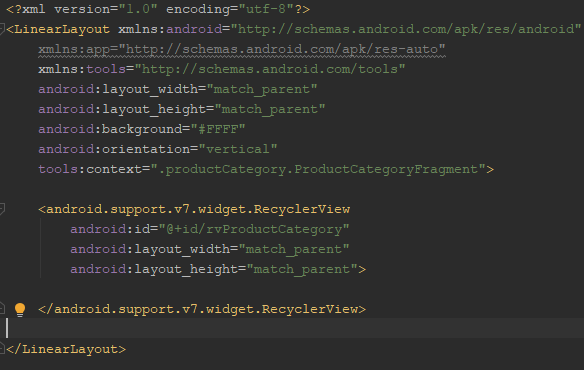


- Đây là hàm khi click vào 1 item trên menu sẽ chuyển tới màn hình đó.



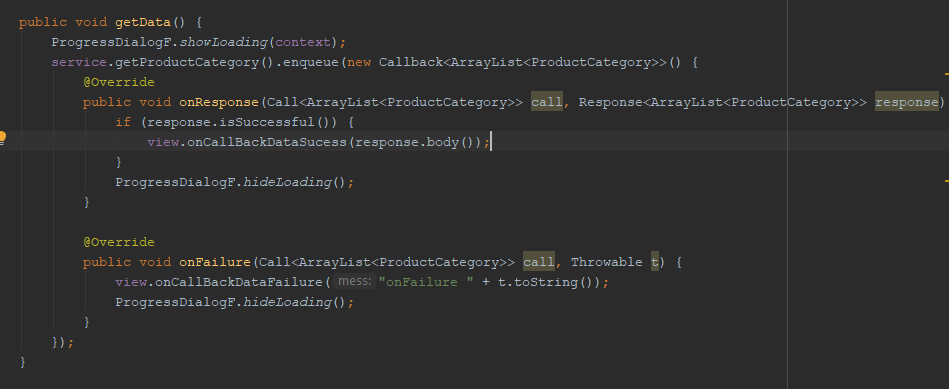
* + 1. **Màn hình danh mục sản phẩm**

1. **Giao diện**

****

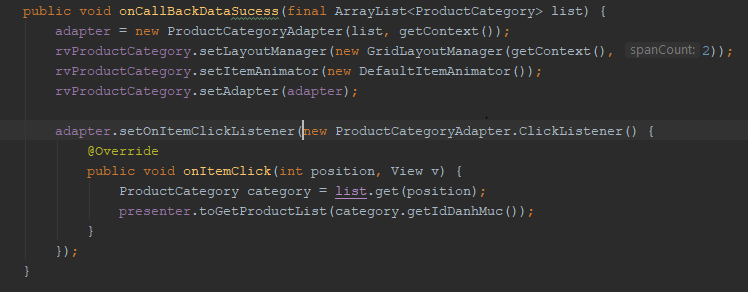
1. **Xử lý**

- Hàm gửi yêu cầu (Request) API về Database để lấy danh sách cái loại sản phẩm

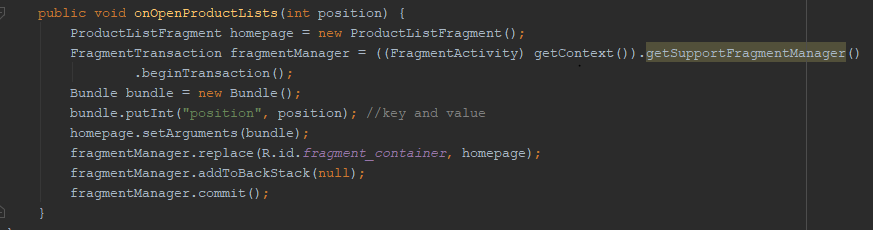


- Hàm này dùng để thiết lập các yêu cầu của RecyclerView để show dữ liệu lên

Vd : Thiết lập LayoutManager, set Adapter

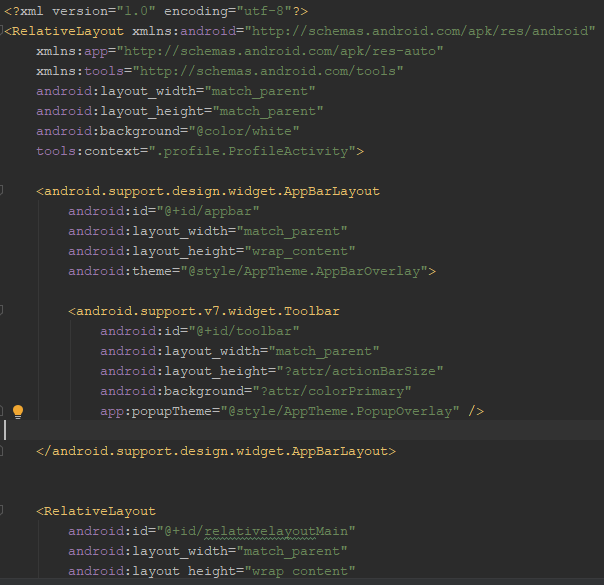


- Hàm này chạy khi người dùng click vào 1 loại sản phẩm sẽ chuyển tới danh sách các sản phẩm tương ứng



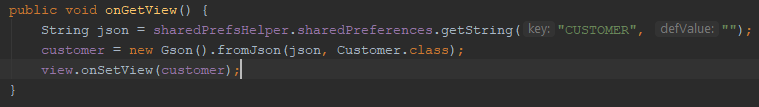
* + 1. **Màn hình quản lý tài khoản**

1. **Giao diện**

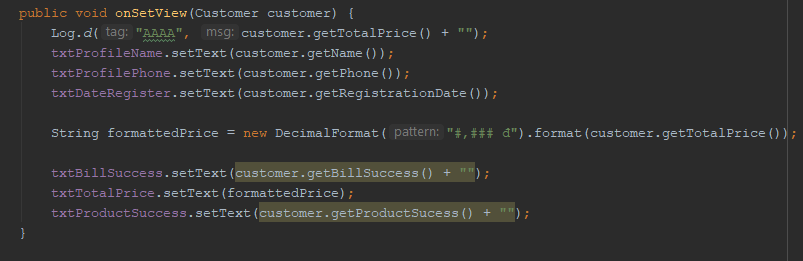


1. **Xử lý**

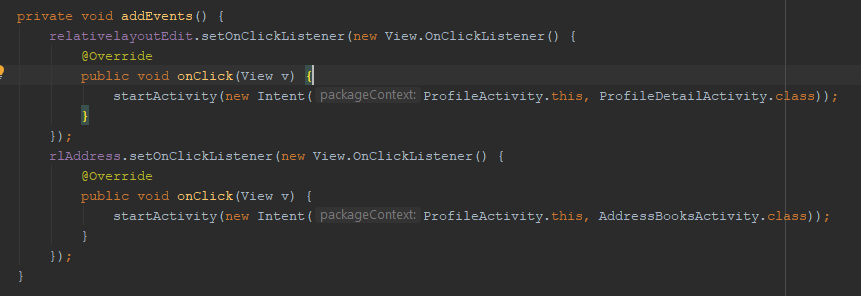
- Lấy dữ liệu đã được lưu trong SharePreferences khi đăng nhập



- Set dữ liệu lấy được từ SharePreferences lên View

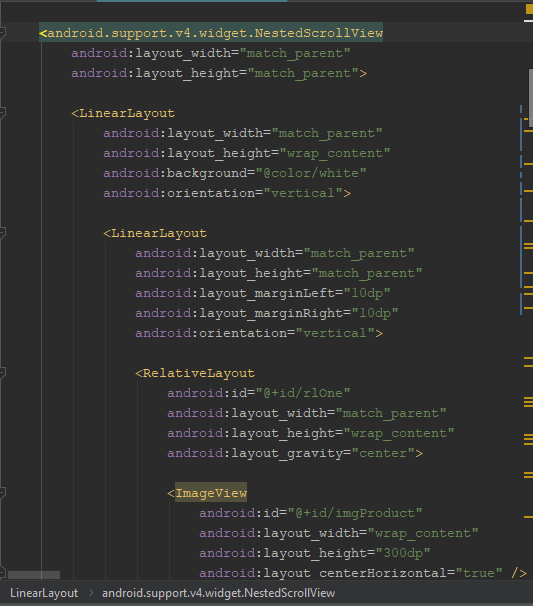


- Xử lý event khi click vào layout cập nhập thông tin và sổ địa chỉ



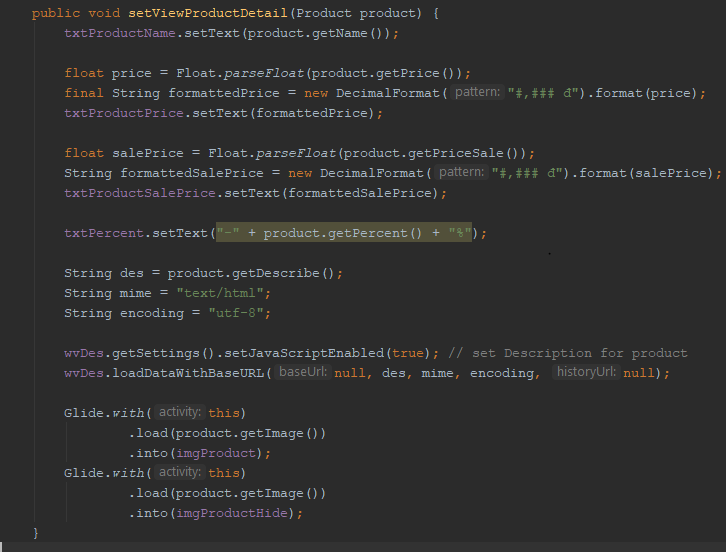
* + 1. **Màn hình chi tiết sản phẩm**

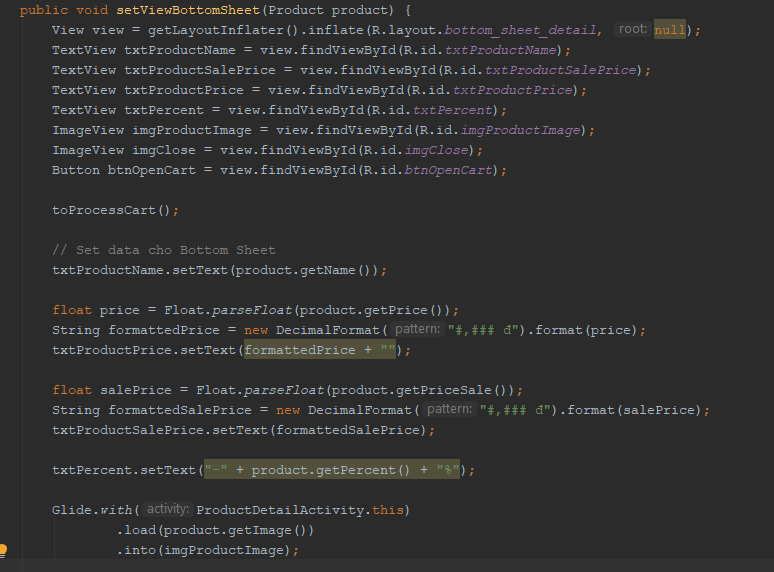
1. **Giao diện**

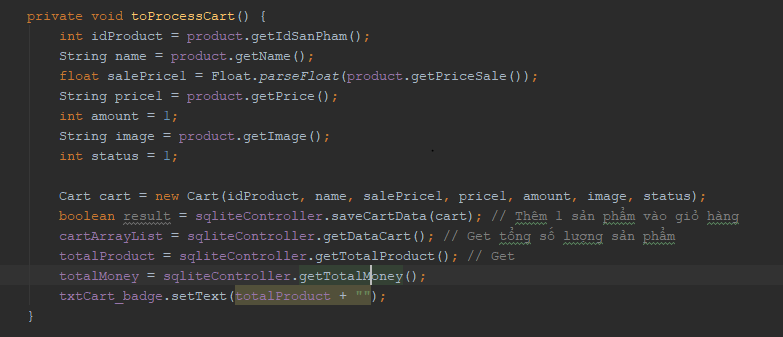
****

1. **Xử lý**

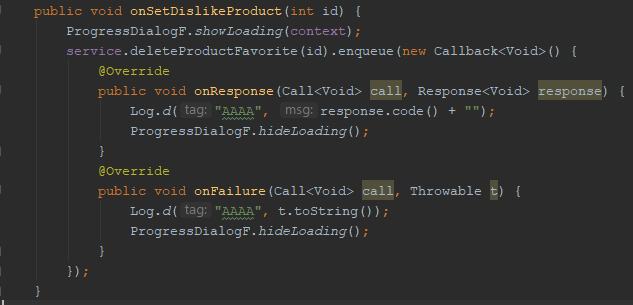
- Nhận dữ liệu từ Màn hình danh sách sản phẩm và set lên view tương ứng.

****

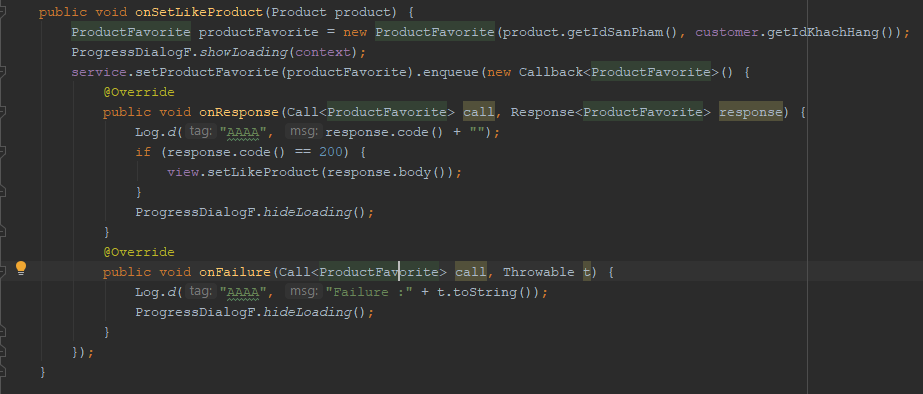
**-** Set thông tin sản phẩm vừa chọn mua vào BottomSheet****

- Khi click vào button Mua Hàng hoặc Thêm vào giỏ hàng để xử lý 

- Hàm gửi yêu cầu (Request) API về Database để set Dislike cho sản phẩm

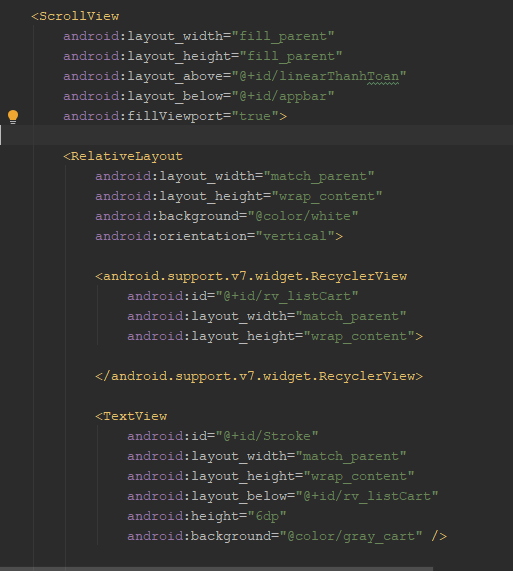


- Hàm gửi yêu cầu (Request) API về Database để set Dislike cho sản phẩm



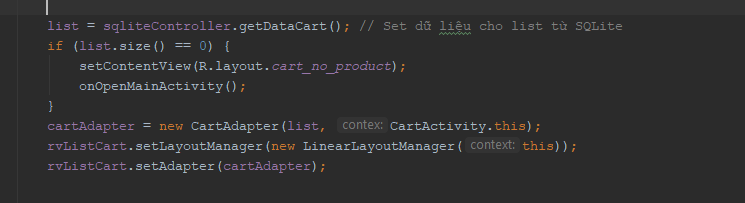
* + 1. **Màn hình giỏ hàng**

1. **Giao diện**

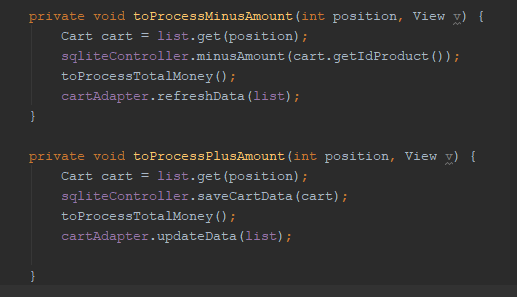
****

1. **Xử lý**

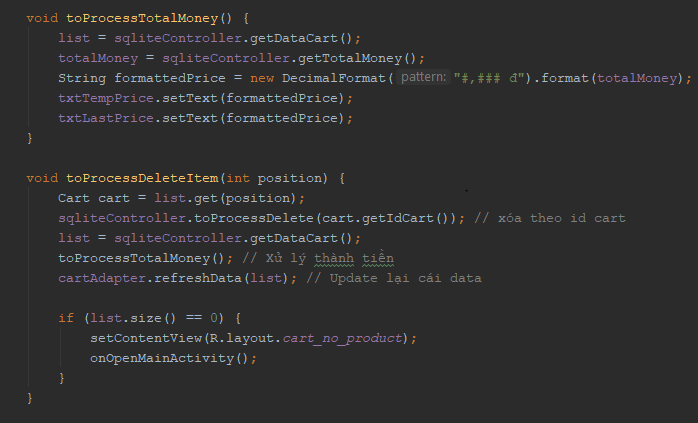
- Lấy list sản phẩm được chọn mua từ Database bằng SQLite cập nhập lên RecyclerView

****

**-** Xử lý thêm hoặc giảm bớt số lượng sản phẩm dùng câu lệnh của SQLite

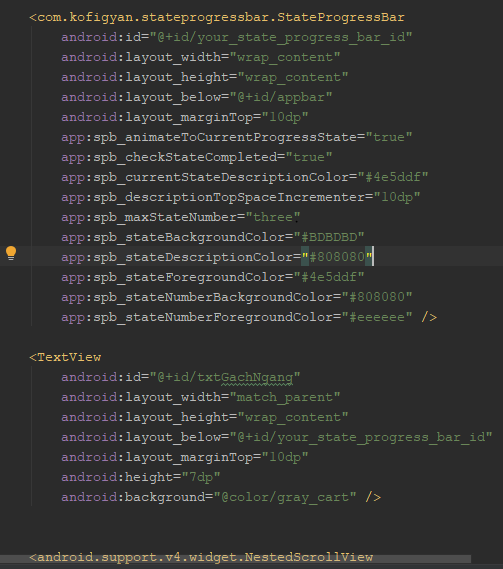
****

**-** Xử lý xóa sản phẩm có sẵn trong giỏ hàng , và cập nhập lại tổng số tiền phải trả

****

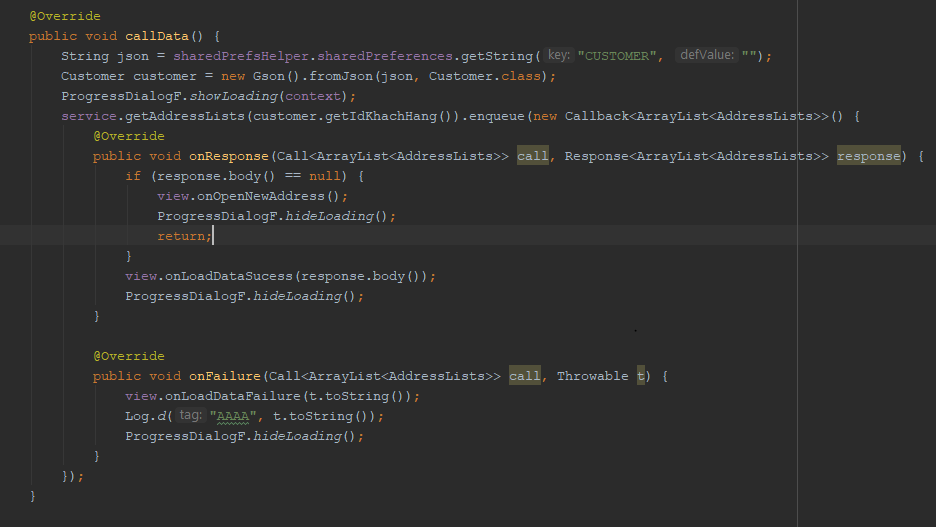
* + 1. **Màn hình thanh toán**

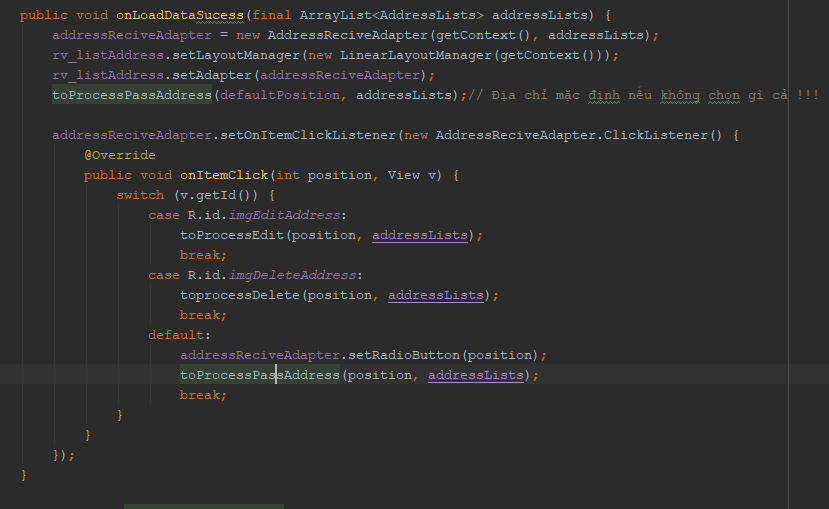
1. **Giao diện**

****

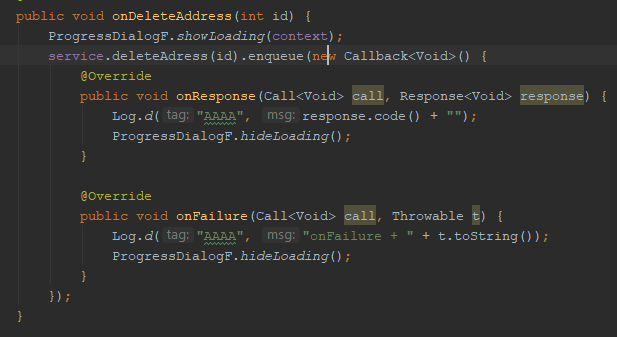
1. **Xử lý**

- Hàm gửi yêu cầu (Request) API về Database để lấy danh sach địa chỉ của người dùng

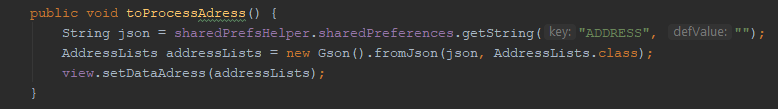
****

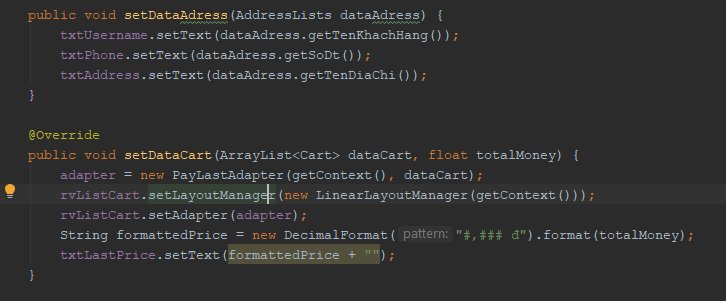
- Set dữ liệu lên RecyclerView , xử lý cập nhập, xóa, chọn địa chỉ

- Hàm gửi yêu cầu (Request) API về Database để xóa địa chỉ cần thiết

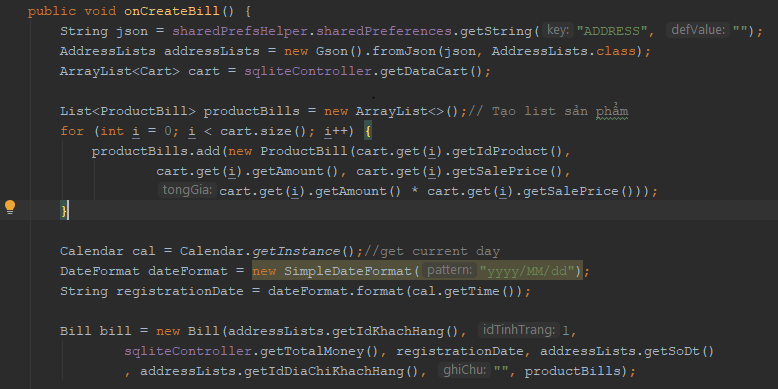


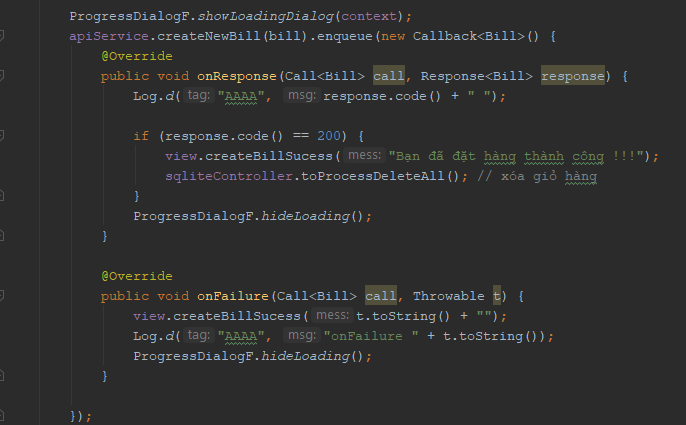
- Set dữ liệu đã chọn như Địa chỉ, Phương thức thanh toán, Danh sách các sản phẩm còn trong giỏ sẽ được thanh toán





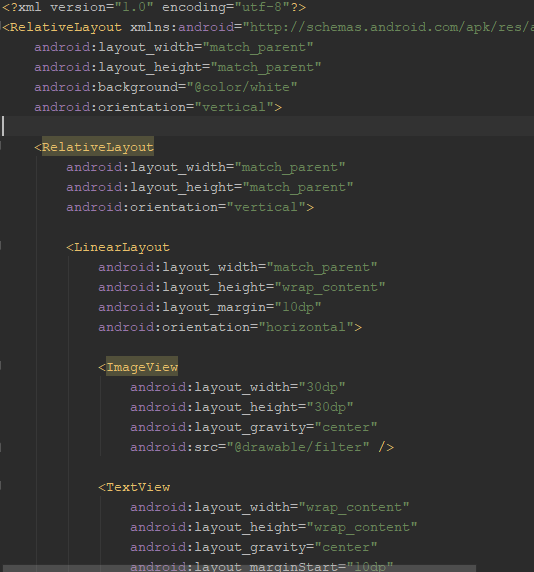
- Xử lý bước cuối cùng là tạo phiếu mua hàng gửi yêu cầu (Request) API về Database



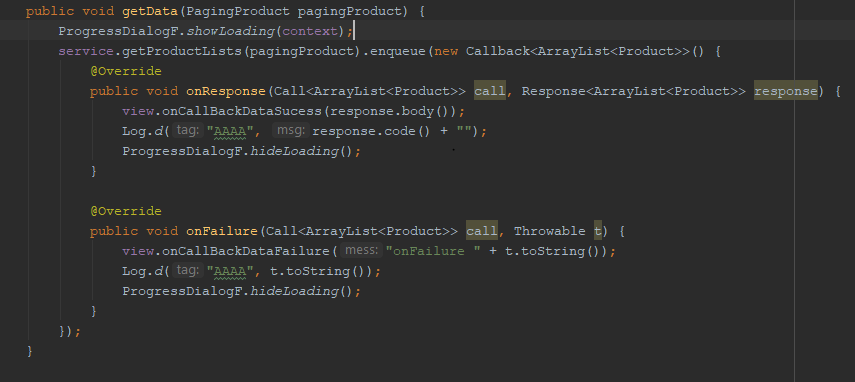


* + 1. **Màn hình danh sách sản phẩm theo danh mục tương ứng**

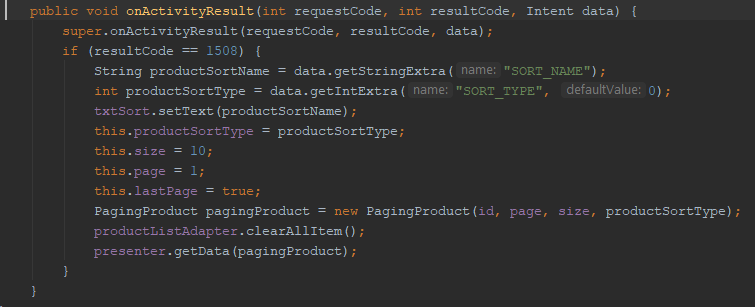
1. **Giao diện**

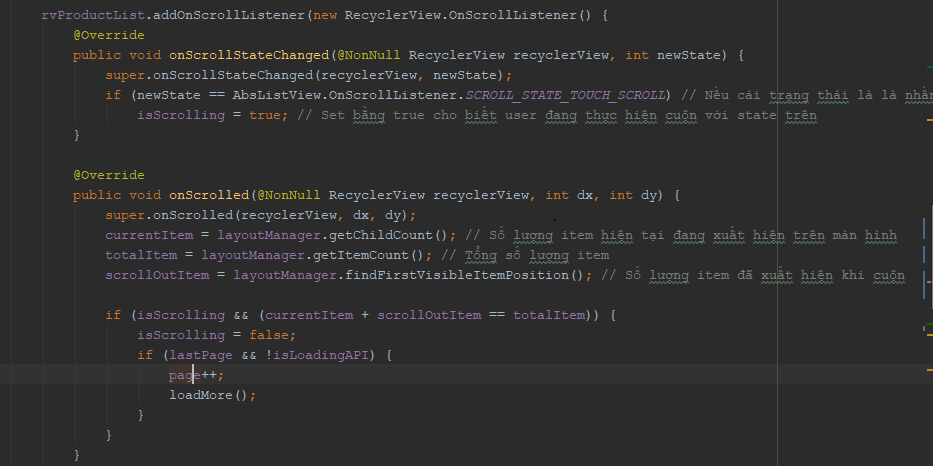
****

1. **Xử lý**

- Gửi yêu cầu (Request) API về Database để get danh sách sản phẩm 

- Xử lý mở Dialog sắp xếp, và gửi thông tin ngược lại về Màn hình sản phẩm

- Sau khi nhận được tùy chọn sắp xếp, thực hiện Request API 

- Thực hiện phân trang danh sách sản phẩm 

**PHẦN III : Kết Luận**

1. **Kết quả đạt được**

- Hiểu thêm về cách một ứng dụng android được tạo ra như thế nào.

- Hiểu về các thành phần, các view trong khi thiết kế layout .

- Biết được cách kết nối giữa client và web service.

- Tìm hiểu được nhiều thư viện cần cho nhu cầu thiết kế ứng dụng.

1. **Hạn chế**

- Ứng dụng còn đơn giản, thô sơ, hạn chế về các chức năng.

- Code trong phần XML chưa được tối ưu .

- Code phần xử lý (Java) chưa được rõ ràng (Clean).

- Đang tìm hiểu và chưa áp dụng được công nghệ mới vào ứng dụng.