Note làm app lazada

: lồng frame layout và linearlayout cho bg xám lại

nestedScrollView:sử dụng khi nhu caauf chế độ xem cuộn cuộn bên trong chế độ xem khác,

sử dụng khi có nhu cầu cho chế độ xem cuộn bên trong chế độ xem cuộn khác. Đặt được thanh cuộn trong cái này

centercrop : nằm ở chính giữa và mở rộng ra, chắc chắn ảnh sẽ bị cắt

**android:layout\_width="match\_parent"  
android:layout\_height="wrap\_content"  
android:layout\_weight="1"**

chia đều 2 cái vs nhau

Sư dụng drawer layout để làm menu đa cấp : đè lên cả appbar và gd tổng => nó là viewgroup lớn nhất

ExpandListView: dùng để add fragment động

Add toolbar vào DrawerLayout : đặt id rồi xử lí

Dấu 3 --- ngang : actionbarDrawerToggle

ActionbarDrawerToggle =new ActionbarDrawerToggle (activity, drawerlayout, toolbar, chuỗi đóng, chuỗi mở)

**drawerToggle**=**new** ActionBarDrawerToggle(**this**,**drawerLayout**,R.string.***open***, R.string.***close***);

sau đó gán drawtoggle và drawlayout

**drawerLayout**.addDrawerListener(**drawerToggle**);

getSupportActionBar().setHomeButtonEnabled(**true**); : cho phep mow drawertoggle

getSupportActionBar().setDisplayHomeUpEnabled(true) :

drawerToggle.syncState() đồng bộ lại tất cả dữ liệu của DrawerLayout

bản chất của drawerToggle là 1 menuItem= > chỉ cần click , bắt sự kiện trong onOptionItemSelected;

<**style name="changecolordrawertoogle" parent="Widget.AppCompat.DrawerArrowToggle"**>  
 <**item name="color"**>@color/colorWhite</**item**>

<**item name="drawerArrowStyle"**>@style/changecolordrawertoogle</**item**> : thuoc tinh nay cho vao menuitemcolor vi ban chat drawlayout la 1 menuitem

: custom mau cuar drawerToggle , vì drawerToggle nằm trên toolbar

Chuyển HashMap => Map.Entry (set) để lấy key và value

Map.Entry<String,String> mp=attrs.get(i).entrySet();

ExpanableListView: giống listView, có adapter

- Lớp đầu tiên đc gọi là Group

- Lớp thứ 2 đc gọi là Child

-getViewGroup : getView ccủa adapter, tạo giao diện cho group cha

- getChildGroup: tạo gd cho groupcon

- tạo 1 expandAdapter extends BaseExpendableListAdapter

LayoutInflater : dùng để khowrit ạo gd

Cú pháp tạo layout

LayoutInflater inflater= (LayoutInflater) **context**.getSystemService(Context.***LAYOUT\_INFLATER\_SERVICE***);  
View viewGroupcha=inflater.inflate(R.layout.***custom\_layout\_group\_cha***,parent,**false**);

Muốn làm lớp thuws3 thì lớp thứ 2( hay là lớp con cái trc ) phải chứa 1 expenddableListView

+ muốn làm lớp 3 thì phải có SecondAdapter;

onMeasure : kế thừa chiều rộng và chiều cao

Trong expandablelist view trong xml đặt

Android:choiceMode=”singlechoice” : chỉ click đc 1 thằng thôii

Android:groupIndicator=”” cái dẫu mũi tên nhỏ nhỏ, đặt bằng @nulll thì sẽ bỏ nó đi

Muốn thay đổi img trong expandable listview thì sử dụng biến isexpanded để xử lí

**android:screenOrientation="portrait"  
android:label="Đăng nhập/Đăng ký"**

\* trong activity : android:sreenOrientation : cho phép hiển thị đứng hay ngang

- Sử dụng textinputLayout : chứa edittext, cho phép khi nhập thì hint của editext sẽ hiện lên tittle

Textinputlayout : bao tron ca edit text, khi nhap edit text thi hint se ban len title, va khi nhap sai se xuat hien cai loi

Khai báo text input layout là 1 viewgroup bao trọn cái edttext=> size của nó là size mình thiết kế edtText, bên trong sẽ chứa 1 editext hoặc 1 custom editext, vẫn phawir đặt size cho edtext, id như bình thường

-Trong tablayout : **app:tabTextColor="@color/colorGray"**

**app:tabTextAppearance="@style/capfale"**

<**style name="capfale" parent="TextAppearance.Design.Tab"**>  
<**item name="textAllCaps"**>false</**item**> cho cái text all caps nó = false

Android:drawableLeft=”” định dạng đc drawable bên trái của button

Custom EditText <**declare-styleable name="PassWordEditText"**>  
 <**attr name="useStrike" format="boolean"**></**attr**>  
</**declare-styleable**>

\*Cách thêm UseStrike vào android: tức là hình con mắt ý

Thêm thuộc tính vào style

<decaler-stylable name=” teen class( PasswordEditext>

<attr name=”useStrike”> format =”boolean”/>

Sau do vao xml thay cai editext trong textinputlayout = PasswordEditText

Sau do trong passwordEdittext dat

<apps: useStrike=”true”

\* Trong class PasswordEditText có 2 Drawable

Drawable eye, eyeStrike

Bản thân Activity sử dụng AppcompatActivity nên sử dụng contextCompat

Eye= ContextCompat.getDrawable(getContext,R.drawbale.ic).muate();

eyeStrike= ..............................................................R.ic.drawble).muate();

Lấy usetrike từ xml bằng cách

If(attribute!=null){// lay attr ra

TypeArray array=getContext.getTheme().obtainStyleAttributes(attrs,R.styleable.passwordEditext,0,0);

This.usetrike=array.getBoolean(R.styleable.passwwrodEditext\_usetrike,false)

}

Drawable[] drawables=getCompoundDrawables();

Trả về drawable của edttexxt

**public boolean** onTouchEvent(MotionEvent event) {*// kieem tra ng dung nhan vao man hinh  
 //vi actionDown chong vs su kien click vao edtText len phai de actionUP* **if**(event.getAction()==MotionEvent.***ACTION\_UP***&& event.getX() >= (getRight()-**drawable**.getBounds().width())){  
 **visible**=!**visible**;*// gan cho no la ngc lai* caidat();  
 invalidate(); *// kiem tra su kien click cho man hinh* }; *// lon hon hoac bang chieu rong edt-cr drawable* **return super**.onTouchEvent(event);*// su kien ke thua cua onTouchEvent*}

**nonecrossx**=ContextCompat.*getDrawable*(getContext(),android.R.drawable.***screen\_background\_dark\_transparent***);*// bgtrong suot cua android*

@Override  
**protected void** onTextChanged(CharSequence text, **int** start, **int** lengthBefore, **int** lengthAfter) {  
 **super**.onTextChanged(text, start, lengthBefore, lengthAfter);  
 **if**(lengthAfter==0&& start==0){ *// neu do dai edt ==0 hoac no dang bat dau thi k cho hien thi* **visible**=**false**;  
 cauhinh();  
 }  
 **else**{  
 **visible**=**true**;  
 cauhinh();  
 }  
}

**private void** caidat(){*// set syle, visible = false la khong hien thi password* setInputType(InputType.***TYPE\_CLASS\_TEXT*** |(**visible**? InputType.***TYPE\_TEXT\_VARIATION\_VISIBLE\_PASSWORD*** : InputType.***TYPE\_TEXT\_VARIATION\_PASSWORD***));  
 Drawable[] drawables = getCompoundDrawables();*// lay drwaable cedittext* **drawable** = **useStrike** && !**visible**? **eyeStrike**: **eye**; *// neu nhu su dung usetrike(nhap thuoc tinh usetrike) va khong hien thi password* **drawable**.setAlpha(**Alpha**);  
 setCompoundDrawablesWithIntrinsicBounds(drawables[0],drawables[1],**drawable**,drawables[3]);  
}

Sau đó gõ tên package+ folder chứa custom

Có 2 trạng thái : mắt gạch và mắt bình thường => định thuộc tính xml useStrike cho edtText

B1 : vào style khai báo declare- styleable name =”tên file java mà custom>

<attr name=”useStrike” format =”boolean”

Sau ddo vao file xml cua edttext go : app:useStrike=”true”

Có 2 trạng thái => phải cso 1 biến lưu trạng thái

Vif extend appcompatactivity => ContextCompat

**eye**= ContextCompat.*getDrawable*(getContext(), R.drawable.***ic\_visibility\_black\_24dp***).mutate();

TypeArray : lấy attribute từ ng dùng truyên vào

TypeArray aray=getContext.getTheme.obtainSyle(atribute, R.styleable.passwrodcontedxt,0,0):

setInputType(InputType.***TYPE\_CLASS\_TEXT***|(**visible**? InputType.***TYPE\_TEXT\_VARIATION\_PASSWORD*** : InputType.***TYPE\_TEXT\_VARIATION\_VISIBLE\_PASSWORD***));

visible ? => visible là true

Drawable[] drawables=getCompoundDrawables();

Trả drawable trái trên phải dưới

Event.getX() : tra ve vi tri dang cham vao man hinh theo X

Event. getY

getRight : tra ve chieu rong cua view dang chua

getDrawable() tra ve drawable

| : hoac

**drawable**.getBounds().width() tra ve chieu rong cua drawable tuong ung

android.R.drawable.***screen\_background\_dark\_transparent***

bg trong suot cua android

*//su kien click vao edt;*@Override  
**protected void** onTextChanged(CharSequence text, **int** start, **int** lengthBefore, **int** lengthAfter) {  
 **super**.onTextChanged(text, start, lengthBefore, lengthAfter);  
}

\* Cách sử dụng sdk android trreen developer/facebook

- thêm mavenCentral() vào repositories

Thêm implementation 'com.facebook.android:facebook-android-sdk: + phiên bản ms nhất

Vào developerfacebook, đăng nhập android, kéo xuống cuối cùng vào khắc phục sự cố , copy code packageinffo, thay đổi tên package trong đó và run, lấy cái haskey

- tạo 1 project ms

- sau đó vào project + đăng nhập android

- làm các bước theo yêu cầu

+viết đường dẫn package và activity đăng nhập

+ cop cái mã hashkey vào

+ copy string facebook\_id vào values string

+ add metadata vào application

- Khai báo facebook sdk : nằm dưới view trong fragemnt và trên setContentView trong activity

+fragment: FacebookSdk.sdkInitialize(getContext.getApplicationContext)

+activity: FaceBookSdk.sdkInitialize(getAppplicationContext)

- findView như bình thường

LoginButton : Sử dụng button có sẵn của fb

LoginManager : customButton và quản lí button đó

- callbackManager=CallBackManager.Factory.create();

- hàm sset sự kiện đăng nhập thành công hay thất bại

-LoginManager.getInstance().registerCallBack(callbackManager..) neu dangnhap thanh cong chya vao onSucess

Public profile : id name, first name, last name, age;

- trong sự kiện của button

Loginmanager.getInstance.LoginwithPermission(FragmentDangNhap,Array.asList(“**public\_profile**

”,”email”))

AccessToken : lưu trữ thông tin ng dùng và lưu trong file cache

GraphApi : tương tác đến webservice của fb, lấy username , pw, age

- Sử dụng graprequest :

Bất cứ cái gidf liên quan đến fb đều phải khai báo faceBookSdk

Vào graph đồ thị để lấy dữ liệu accestoken từ fb

\* đăng nhập bằng google

Đọc tài liệu google : có những mô tả ngắn

<https://developers.google.com/identity/sign-in/android/start-integrating?hl=vi>

b1 : cài đặt google play service trên sdk manager

b2 :

implementation **'com.google.android.gms:play-services-auth:15.0.1'**

**vào console developer để quản lí project của gg**

**tạo 1 project gg map để lấy mã sha-1**

**b3 tạo 1 incretial**

**tạo 1 android**

**copy sha1 vào**

**- tạo 1 google\_plus\_client, copy client id**

**tương tự facebook, google sẽ có googleSignInOptions**

**- tạo 1 gso**

GoogleSignInOptions gso = **new** GoogleSignInOptions.Builder(GoogleSignInOptions.***DEFAULT\_SIGN\_IN***)  
 .requestEmail()  
 .build();

**mGoogleSignInClient** = GoogleSignIn.*getClient*(**this**, gso);

GoogleSignInAccount acct = GoogleSignIn.*getLastSignedInAccount*(Main2Activity.**this**);// activity

**Builder() thì phải build**

**Fragment Activity bản chất là 1 activity**

Cách thay title của menuitem đăng nhập khi sin in : trong onCreateOptionMenu chọn

MenuItem ItDangnhap=menu.findItem(id)

menuItem itDangXuat=menu.findItem(id)

if(ac!=null){

laays code graph request cop vao

sửa trong hàm completed

tennguoidung=object.getString();

itDangNhap.settitle

elsse if(ggSignInAccount!=null){

tennguoidung=ggSignInAccount.getDisplayName();

itDangnhap.settitle(tennguoidung)

nếu 1 trong 2 khác null, it đãnguaaat.setVisible()

\* xử lí sự kiện của dăng nhập khi có tk

Nếu cái kia \*gg Acc và accestoken ==null

Start Activity

Neu gg signinresult!=null{  
**this**.**menu**.clear();  
**this**.onCreateOptionsMenu(**this**.**menu**);

Dang xuat

LoginManager.*getInstance*().logOut();  
**mGoogleSignInClient**.signOut()  
 .addOnCompleteListener(**this**, **new** OnCompleteListener<Void>() {  
 @Override  
 **public void** onComplete(@NonNull Task<Void> task) {  
  
 **tennguoidung**=**null**;  
  
 }  
 });  
**this**.**menu**.clear();  
**this**.onCreateOptionsMenu(**menu**);

Cách cuộn xuống mất thanh toolbar: đó là matterio design

Lên gg search [android handle ***scroll*** coordinator](https://www.google.com/search?rlz=1C1CHBF_enVN851VN851&q=android+handle+scroll+coordinator&spell=1&sa=X&ved=0ahUKEwiU_rOw6dvjAhWH-2EKHVeqBdEQkeECCCwoAA) , bbanr thay layout này là frame layout

Coordinator layout: android:fitsSystemWindows="true" : khi để là true khi scroll thì nó k che cái thanh statusbar

Trong toolbar: khai báo layout\_scrollFlag: cho phép mở tính năng hiệu ứng scrool khi ng dùng scroll lên

enterAlways : scroll lên mất toolbar

app:layout\_behavior : chỉ áp dụng trong các thuộc tính support , điều khiển hành vi của layout, chỉ cần nằm trong coordinator layout

**"@string/appbar\_scrolling\_view\_behavior" nó sẽ nằm dưới app bar**

**app:layout\_scrollFlags="scroll|enterAlways"**

trong appbar khai báo thuộc tính này : ảnh hưởng của scrool và nó sẽ mất đi

nếu tablayout muốn mất thì cũng định thuộc tính này vào đó

appbarlayout bản chất là linearLayout dạng vertical

Vì mình muốn ẩn cả linearlayout : Button, View ,ImageButton , nên muốn ẩn nó như app bar trong condinator layout phải sử dụng thuộc tính

<**android.support.design.widget.CollapsingToolbarLayout**

**: Chứa toolbar và đặt hiệu ứng chồng cho toolbar, bản chất là framelayout, toolbar k có hiệu ứng thì toàn bộ không có hiệu ứng**

**Muốn toolbar ở lại : layout\_collapsMode =”pin”**

**Scroll|exitUnitCollapse: tạo hiệu ứng khi cuộn . ngưng collap này khi nó collap**

App:layout\_collapsMode:”palalax” cho nó ẩn đi

Snap : khi kéo đến phân nữa sẽ mất đi : quá 50% sẽ mất đi, dưới 50% k mất đi

Sự kiện click : implement OnOffsetChangedListener

\* bắt sự kiện lỗi nhập mật khẩu

- thêm 1 attr vào declare pw editext là useValidate> format= “boolean”

- vào edttext và khai báo app:usevalidate=”true”

Khai báo biến useBoolean ở bên initData() pwEDT

- Sự dụng matcher\_patern để ss edt vs 1 format

- khai báo

String **MATCHER\_PATERN**=**"(?=.\*\\d)" kiểm tra người dùng cos nhập vào số k**

**(?=.\*[A-Z]) (?=.\*[a-z]){?=.\*#$%}.{6,20}**

**Cho phep chu hoa cho phep chu thuong dieu kien 6-20 ki tu**

**. la dieu kien**

**() la mo ra 1 dieu kien**

**[] la dieu kien k bat buoc**

**{} la dieu kien bat buoc**

**(A)+(B) : dấu cộng là bắt buộc, thằng đầu tiên là A, tiếp theo là B**

**- khai báo pattern : pattern=Pattern.Compile(MATCHER\_PATTERN)**

**- kahi báo mattcher**

**If(useValidate==true){**

**setOnFocusLítenter(new onFO....**

**mattcher=pattern.matcher(chuoi);**

**if(!matcher.matchers()**

**TextInputLayout tl = v.getParent.getParent();**

**Tl.setERROREnable(true);**

**Tl.seterror(“:....)**

**Else**

**Tl.setERROREnable(false);**

**Tl.seterror()**