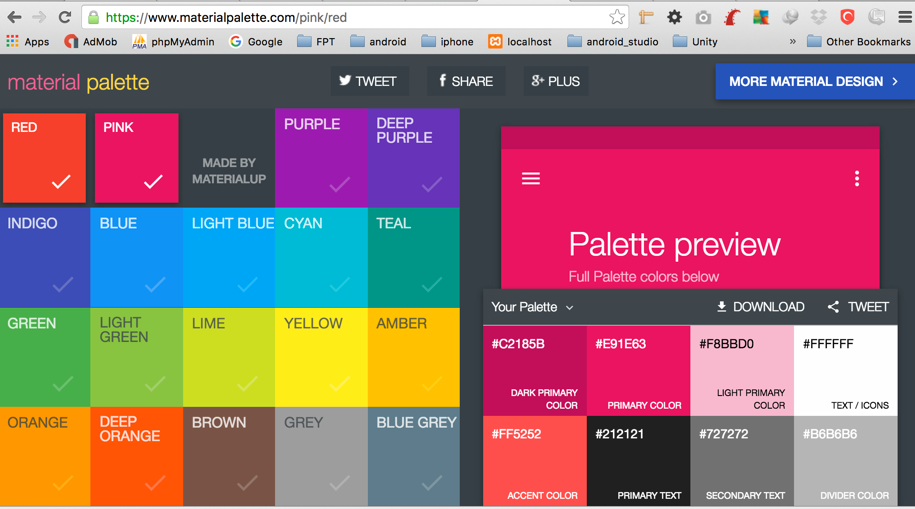
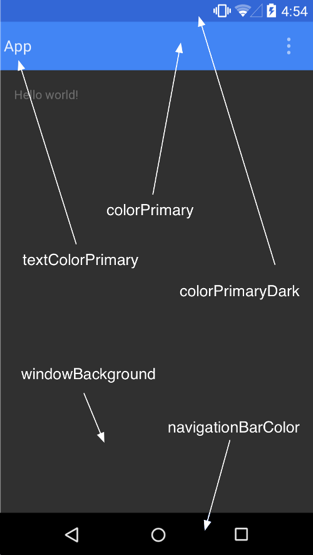
Tạo layout tổng hợp

# Chuẩn bị

1. Truy cập địa chỉ <https://www.materialpalette.com/> chọn cho mình 1 tông màu

1. Tạo một project mới trong android studio. Mở file color.xml lên thiết kế các màu (tham khảo tông màu ở bước 1)

*<?***xml version="1.0" encoding="utf-8"***?>*<**resources**>  
 <**color name="white"**>#FFFFFF</**color**>  
 <**color name="colorPrimary"**>#2196F3</**color**>  
 <**color name="colorPrimaryDark"**>#1976D2</**color**>  
 <**color name="colorPimaryight"**>#BBDEFB</**color**>  
 <**color name="colorAccent"**>#FF4081</**color**>  
 <**color name="textColorPrimary"**>#212121</**color**>  
 <**color name="textColorSecondary"**>#727272</**color**>  
 <**color name="colorIcons"**>#FFFFFF</**color**>  
 <**color name="colorDivider"**>#B6B6B6</**color**>  
</**resources**>

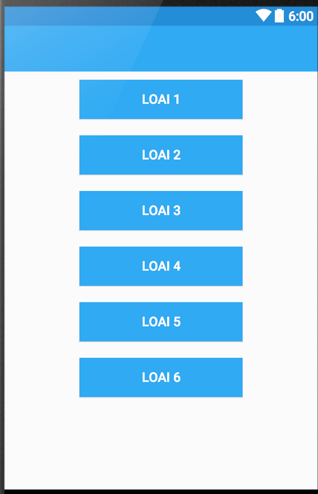
1. Mở file style.xml lên. Đầu tiên sửa style của AppTheme thành NoActionBar. Thêm 2 style cho button và toolbar

<**resources**>  
 *<!-- Base application theme. -->* <**style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar"**>  
 <**item name="android:statusBarColor"**>@color/colorPrimaryDark</**item**>  
 </**style**>  
 <**style name="TechniqueButton"**>  
 <**item name="android:background"**>@color/colorPrimary</**item**>  
 <**item name="android:textColor"**>#FFFFFF</**item**>  
 <**item name="android:layout\_width"**>200dp</**item**>  
 <**item name="android:layout\_height"**>wrap\_content</**item**>  
 <**item name="android:layout\_margin"**>10dp</**item**>  
 <**item name="android:padding"**>10dp</**item**>  
 <**item name="android:layout\_centerHorizontal"**>true</**item**>  
 <**item name="android:onClick"**>on\_click</**item**>  
 <**item name="android:textSize"**>16sp</**item**>  
 </**style**>  
 <**style name="TechniqueToolbar"**>  
 <**item name="android:layout\_width"**>match\_parent</**item**>  
 <**item name="android:layout\_height"**>?attr/actionBarSize</**item**>  
 <**item name="android:background"**>@color/colorPrimary</**item**>  
 <**item name="android:theme"**>@style/ThemeOverlay.AppCompat.Dark.ActionBar</**item**>  
 <**item name="android:popupTheme"**>@style/ThemeOverlay.AppCompat.Light</**item**>  
 <**item name="android:minHeight"**>?attr/actionBarSize</**item**>  
 </**style**>  
</**resources**>

1. Mở file activity\_main.xml. Trong RelativeLayout thêm 1 Toolbar và một button như sau. Bắt luôn sự kiện click trong giao diện luôn.

<**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 style="@style/TechniqueToolbar"** />  
  
<**Button  
 android:id="@+id/bt\_1"  
 android:layout\_below="@+id/toolbar"  
 android:text="loai 1"  
 android:onClick="xuly\_onClick"  
 style="@style/TechniqueButton"**/>

Qua giao diện kéo vào thêm 5 button nữa nằm bên dưới Button1 (hoặc copy code) để có được như hình



1. Qua file MainActivity.java khai báo Toolbar toàn cục, trong onCreate ánh xạ và dùng lệnh setSupportActionBar() để gán toolbar thay cho actionbar

Toolbar **main\_toolbar**;  
@Override  
**protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **main\_toolbar** = (Toolbar) findViewById(R.id.***toolbar***);  
 setSupportActionBar(**main\_toolbar**);  
}

1. Viết thêm 1 hàm xử lý khi click vào sẽ mở các Activity như sau:

**public void** xuly\_onClick(View view) {  
 Intent intent;  
 **switch** (view.getId()) {  
 **case** R.id.***bt\_1***:  
 intent = **new** Intent(MainActivity.**this**, Loai1Activity.**class**);  
 **break**;  
 **case** R.id.***bt\_2***:  
 intent = **new** Intent(MainActivity.**this**, Loai2Activity.**class**);  
 **break**;  
 **case** R.id.***bt\_3***:  
 intent = **new** Intent(MainActivity.**this**, Loai3Activity.**class**);  
 **break**;  
 **case** R.id.***bt\_4***:  
 intent = **new** Intent(MainActivity.**this**, Loai4Activity.**class**);  
 **break**;  
 **case** R.id.***bt\_5***:  
 intent = **new** Intent(MainActivity.**this**, Loai5Activity.**class**);  
 **break**;  
 **default**:  
 intent = **new** Intent(MainActivity.**this**, Loai6Activity.**class**);  
 **break**;  
 }  
 startActivity(intent);  
}

1. Mở file string.xml lên và thêm string array để làm nguồn dữ liệu đổ lên listview

<**string-array name="days\_names"**>  
 <**item**>Thu Hai</**item**>  
 <**item**>Thu Ba</**item**>  
 <**item**>Thu Tu</**item**>  
 <**item**>Thu Nam</**item**>  
 <**item**>Thu Sau</**item**>  
 <**item**>Thu Bay</**item**>  
 <**item**>Chu Nhat</**item**>  
</**string-array**>

Đến đây ta đã chuẩn bị xong, mỗi activity ta sẽ thiết kế một loại khác nhau để phân biệt.

# Loại 1

Loại này ta sẽ dùng recycleview. RecycleView là nâng cao của dạng listview gridview. Thích hợp với danh sách lớn, nó sẽ giúp cho ta tối ưu bộ nhớ khi scroll với cách viết đơn giản hơn cách viết của getview trong custom adapter

1. Mở file build.gradle và thêm dòng compile như sau:

compile **'com.android.support:design:23.4.0'**compile **'com.android.support:recyclerview-v7:23.4.0'**

1. Mở file activity\_loai1.xml và thêm như sau:

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent" android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context="luffy.khoa.anh.layouttonghop.Loai1Activity"**>  
 <**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 style="@style/TechniqueToolbar"**/>  
 <**android.support.v7.widget.RecyclerView  
 android:id="@+id/days\_list"  
 android:layout\_below="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** />  
</**LinearLayout**>

1. Tạo một file layout mới tên “layout\_one\_item.xml” như sau:

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:padding="10dp"  
 android:layout\_height="wrap\_content"**>  
 <**TextView  
 android:id="@+id/planet\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="40dp"  
 android:textSize="20sp"  
 android:text="abc"  
 android:layout\_gravity="center\_horizontal"  
 android:background="@color/colorSecondaryText"  
 android:textColor="#FFFFFF"**/>  
</**LinearLayout**>

1. Mở file Loai1Activity.java và khai báo biến toàn cục toolbar và recyclerView, trong onCreate ánh xạ rồi set actionbar thành toolbar

Toolbar **toolbar**;  
RecyclerView **days\_list**;  
@Override  
**protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_loai1***);  
 **toolbar** = (Toolbar) findViewById(R.id.***toolbar***);  
 **days\_list** = (RecyclerView) findViewById(R.id.***days\_list***);  
 setSupportActionBar(**toolbar**);  
 **toolbar**.setTitle(**"loai 1"**);

}

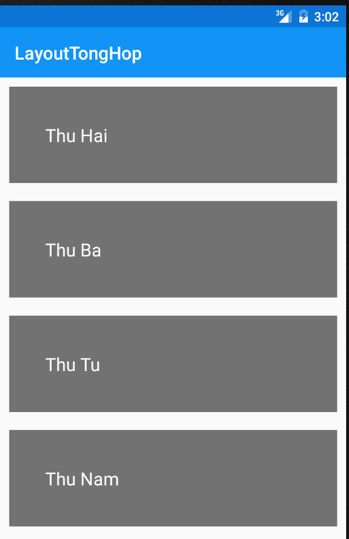
1. Tiếp theo ta sẽ đổ dữ liệu lên recyclerView. Tạo một file java mới tên RecycleViewAdapter rồi viết:

**public class** RecyclerViewAdapter **extends** RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder> {  
  
 String data[];  
  
 **public** RecyclerViewAdapter(String data[]){  
 **this**.data = data;  
 }  
  
 @Override  
 **public** RecyclerViewAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, **int** viewType) {  
  
 View v = LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.***layout\_one\_item***, parent, **false**);  
  
 ViewHolder vh = **new** ViewHolder(v);  
  
 **return** vh;  
 }  
  
 @Override  
 **public void** onBindViewHolder(RecyclerViewAdapter.ViewHolder holder, **int** position) {  
 holder.**text\_view**.setText(data[position]);  
 }  
  
 @Override  
 **public int** getItemCount() {  
 **return** data.**length**;  
 }  
  
 **public static class** ViewHolder **extends** RecyclerView.ViewHolder {  
 *// each data item is just a string in this case* **public** TextView **text\_view**;  
 **public** ViewHolder(View v) {  
 **super**(v);  
 **text\_view** = (TextView) v.findViewById(R.id.***planet\_name***);  
 }  
 }  
}

1. Quay về Loai1Activity.java viết thêm trong onCreate như sau:

**days\_list**.setLayoutManager(**new** LinearLayoutManager(**this**));  
**days\_list**.setAdapter(**new** RecyclerViewAdapter(  
 getResources().getStringArray(R.array.***days\_names***)  
));

1. Chạy chương trình, nhấn vào nút “Loai 1” và xem kết quả



## CardView

Cardview kế thừa từ FrameLayout cho phép bạn show thông tin trong một card (một khung) có border, shadow và corner.

1. Mở file build.gradle thêm dòng rồi nhấn syn

compile **'com.android.support:cardview-v7:23.4.0'**

1. Mở file layout\_one\_item.xml lên, bọc textview vào một viewcard như codecode

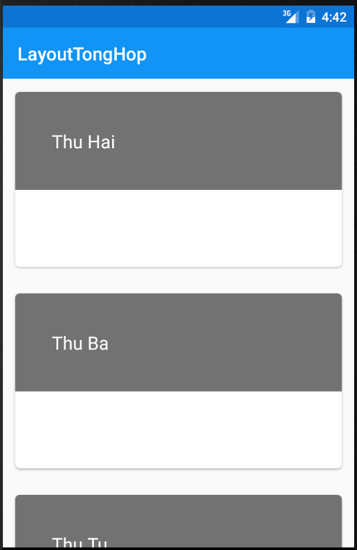
<**android.support.v7.widget.CardView  
 xmlns:card\_view="http://schemas.android.com/apk/res-auto"  
 android:id="@+id/card\_view"  
 android:layout\_gravity="center"  
 android:layout\_width="match\_parent"  
 android:layout\_height="200dp"  
 android:elevation="5dp"  
 card\_view:cardCornerRadius="5dp"  
 card\_view:cardUseCompatPadding="true"** >  
<**TextView  
 android:id="@+id/planet\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="40dp"  
 android:textSize="20sp"  
 android:text="abc"  
 android:layout\_gravity="center\_horizontal"  
 android:background="@color/colorSecondaryText"  
 android:textColor="#FFFFFF"**/>  
</**android.support.v7.widget.CardView**>

Để ý các thuộc tính của CardView như:

* card\_view:cardCornerRadius : bo góc
* android:elevation: là đổ bóng
* card\_view:cardBackgroundColor: màu nền cardview
* card\_view:cardUseCompatPadding: bằng true để hiện shadow trên android 5

<https://developer.android.com/training/material/shadows-clipping.html>

Chạy ct để xem kết quả



## Bắt sự kiện cho recycleview – bắt các view con trong viewItem.

Recycleview không có sự kiện setOnItemClickListener như ListView hay GridView, muốn bắt sự kiện cho nó ta phải bắt sự kiện trong Adapter

1. Mở file RecyclerViewAdapter.java. Trong hàm onBindViewHolder ta viết như sau:

holder.**text\_view**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Log.*d*(**"abc"**,position+**""**);  
 }  
  
});

1. Cách thứ 2 là (chặn cách trên lại) viết code như sau (cũng trong onBindViewHolder).

holder.**text\_view**.setTag(position);  
holder.**text\_view**.setOnClickListener(**click**);

1. Viết bên ngoài hàm onBindViewHolder

View.OnClickListener **click**=**new** View.OnClickListener(){  
  
 @Override  
 **public void** onClick(View v) {  
  
 Log.*d*(**"abc"**,(Integer)v.getTag()+**""**);  
 }  
};

1. Chạy ct, xem kết quả khi click vào. Cách bắt này tuy hơi rối 1 chút nhưng nó sẽ giúp cho từng mục con trong recycleview có thể có nhiều cách sử lý khác nhau, ví dụ như hình bên dưới, trong một mục con của recycleview ta có thể viết click vào nút share xử lý khác còn nhấn vào nút learn more thì xử lý khác (thật ra kiểu listview bình thường cũng làm được việc này)



## Bắt sự kiện cho recycleview – bắt khi click trên viewItem

1. Trong Loai1Activity.java viết thêm class và interface nội như code

**public interface** ClickListener {  
 **void** onClick(View view, **int** position);  
  
 **void** onLongClick(View view, **int** position);  
}  
  
**public static class** RecyclerTouchListener **implements** RecyclerView.OnItemTouchListener {  
  
 **private** GestureDetector **gestureDetector**;  
 **private** Loai1Activity.ClickListener **clickListener**;  
  
 **public** RecyclerTouchListener(Context context, **final** RecyclerView recyclerView, **final** Loai1Activity.ClickListener clickListener) {  
 **this**.**clickListener** = clickListener;  
 **gestureDetector** = **new** GestureDetector(context, **new** GestureDetector.SimpleOnGestureListener() {  
 @Override  
 **public boolean** onSingleTapUp(MotionEvent e) {  
 **return true**;  
 }  
  
 @Override  
 **public void** onLongPress(MotionEvent e) {  
 View child = recyclerView.findChildViewUnder(e.getX(), e.getY());  
 **if** (child != **null** && clickListener != **null**) {  
 clickListener.onLongClick(child, recyclerView.getChildPosition(child));  
 }  
 }  
 });  
 }  
  
 @Override  
 **public boolean** onInterceptTouchEvent(RecyclerView rv, MotionEvent e) {  
  
 View child = rv.findChildViewUnder(e.getX(), e.getY());  
 **if** (child != **null** && **clickListener** != **null** && **gestureDetector**.onTouchEvent(e)) {  
 **clickListener**.onClick(child, rv.getChildPosition(child));  
 }  
 **return false**;  
 }  
  
 @Override  
 **public void** onTouchEvent(RecyclerView rv, MotionEvent e) {  
 }  
  
 @Override  
 **public void** onRequestDisallowInterceptTouchEvent(**boolean** disallowIntercept) {  
  
 }  
}

1. Trong onCreate bắt sự kiện click lên recycleview như sau

**days\_list**.addOnItemTouchListener(**new** RecyclerTouchListener(getApplicationContext(), **days\_list**, **new** ClickListener() {  
 @Override  
 **public void** onClick(View view, **int** position) {  
  
 Toast.*makeText*(getApplicationContext(),position + **" is selected!"**, Toast.***LENGTH\_SHORT***).show();  
 }  
  
 @Override  
 **public void** onLongClick(View view, **int** position) {  
  
 }  
}));

1. Chạy ct, click trên mục con của recycleview

# Loại 2 - Trược mất toolbar

Trên “Loại 1” ta thấy rằng layout của nó có cấu trúc

<RelativeLayout>

<Toolbar />

<RecycleView />

</RelativeLayout>

1. Mở file activity\_loai2.xml lên sữa như sau:

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.design.widget.CoordinatorLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="luffy.khoa.anh.layouttonghop.Loai2Activity"**>  
  
 <**android.support.design.widget.AppBarLayout  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:id="@+id/my\_appbar\_container"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"** >  
  
 <**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 style="@style/TechniqueToolbar"  
 app:layout\_scrollFlags="scroll|enterAlways"**>  
 *<!--  
 scroll  
 enterAlways  
 enterAlwaysCollapsed  
 exitUntilCollapsed  
 -->* </**android.support.v7.widget.Toolbar**>  
  
 </**android.support.design.widget.AppBarLayout**>  
  
 <**android.support.v7.widget.RecyclerView  
 android:id="@+id/days\_list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"**/>  
  
</**android.support.design.widget.CoordinatorLayout**>

Chú ý so với “loai1”, “loai2” khác như sau:

* RelativeLayout được thay bằng CoordinatorLayout
* Toolbar được bỏ trong AppBarLayout
* Toolbar và RecyclerView thêm 1 vài thuộc tính

CoordinatorLayout: là một container mới, nó gia tăng sức mạnh từ FrameLayout. Nó thêm tính năng điều khiển touch event giữa các view.

AppBarLayout: cũng là 1 container mới, được thiết kế đặc biệt để thực thi nhiều đặc điểm của thanh app bar trong Material Design, nhớ rằng nếu bỏ nó trong 1 viewgroup khác thì các tính năng của nó hầu như không làm việc được.

CoordinatorLayout là chìa khoá cho việc hiệu ứng khi scroll, thuộc tính app:layout\_behavior chỉ ra view nào phát sinh sự kiện trong Toolbar. Trong trường hợp này là recycleview.

app:layout\_behavior=“@string/appbar\_scrolling\_view\_behavior"

Toolbar có 1 thuộc tính là app:layout\_scrollFlags app:layout\_scrollFlags=“scroll|enterAlways"

Cờ này có 4 giá trị

Scroll: cờ này cần gán cho tất cả các view cần ẩn khi cuộn, nếu view không có cờ này sẽ được đính vào top của màn hình

enterAlways: cờ này gán để khi vừa cuộn xuống là hiện ra liền.

enterAlwaysCollapsed: cờ này gán để khi cuộn lên, cuộn lên hết mới hiện ra.

exitUntilCollapsed:

1. Mở file Loai2Activity.java lên, nội dung của file này tương tự như loại 1 đơn giản (không có phần sự kiện). Nó chỉ cần gắn toolbar vào và đổ dữ liệu lên adapter thôi.

**public class** Loai2Activity **extends** AppCompatActivity {  
  
 Toolbar **toolbar**;  
 RecyclerView **days\_list**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_loai2***);  
 **toolbar** = (Toolbar) findViewById(R.id.***toolbar***);  
 **days\_list** = (RecyclerView) findViewById(R.id.***days\_list***);

setSupportActionBar(**toolbar**);  
 **toolbar**.setTitle(**"loai 2"**);  
  
 **days\_list**.setLayoutManager(**new** LinearLayoutManager(**this**));  
 **days\_list**.setAdapter(**new** RecyclerViewAdapter(  
 getResources().getStringArray(R.array.***days\_names***)  
 ));  
 }  
}

1. Chạy ct, chọn nút “loại 2” và xem kết quả khi cuộn lên xuống

|  |  |
| --- | --- |
| njnj  Figure 1 : bình thường | Figure 2 cuộn xuống toolbar ẩn, cuộn lên toolbar hiện lại |

# Loại 3 – Thêm tablayout sau toolbar

Loại này sẽ thêm 1 tablayout

1. Copy source từ “activity\_loai2.xml” paste qua cho file “activity\_loai3.xml”
2. Trong file “activity\_loai3.xml” thêm 1 tablayout vào sau thẻ đóng của toolbar

<**android.support.design.widget.TabLayout  
 android:id="@+id/tabs"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@color/colorPrimary"** />

1. Mở file Loai3Activity.java lên, khai báo biến toàn cục

Toolbar **toolbar**;  
RecyclerView **planets\_list**;  
RecyclerViewAdapter **adapter**;  
TabLayout **tab\_layout**;

1. Trong onCreate ánh xạ giao diện

**toolbar** = (Toolbar) findViewById(R.id.***toolbar***);  
**planets\_list** = (RecyclerView) findViewById(R.id.***days\_list***);  
**tab\_layout** = (TabLayout) findViewById(R.id.***tabs***);

1. Cũng trong onCreate viết tiếp

*//gan toolbar*setSupportActionBar(**toolbar**);  
getSupportActionBar().setTitle(**"Loai 3"**);  
  
*//layout cho recyclerView*RecyclerView.LayoutManager layout\_manager = **new** LinearLayoutManager(**this**);  
**planets\_list**.setLayoutManager(layout\_manager);  
  
*//tao adapter***adapter** = **new** RecyclerViewAdapter(getResources().getStringArray(R.array.***days\_names***));  
**planets\_list**.setAdapter(**adapter**);  
  
*//tao tab***tab\_layout**.setTabMode(TabLayout.***MODE\_FIXED***);  
**tab\_layout**.addTab(**tab\_layout**.newTab().setText(**"Tab 1"**));  
**tab\_layout**.addTab(**tab\_layout**.newTab().setText(**"Tab 2"**));  
**tab\_layout**.addTab(**tab\_layout**.newTab().setText(**"Tab 3"**));

1. Chạy chương trình xem kết quả

|  |  |
| --- | --- |
| Figure Bình thường, có thêm tab | Figure Khi cuộn lên ẩn toolbar nhưng vẫn hiện tab |

1. Thử sửa dòng setTabMode từ MODE\_FIXED thành MODE\_SCROLLABLE ta sẽ tạo ra được tab có thể cuộn ngang được (giống tab của google play)

**tab\_layout**.setTabMode(TabLayout.***MODE\_SCROLLABLE***);

# Loại 4 – CollapsingToolbarLayout

1. Mở file activityloai4.xml và sửa như sau:

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.design.widget.CoordinatorLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="luffy.khoa.anh.layouttonghop.Loai2Activity"**>  
  
 <**android.support.design.widget.AppBarLayout  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:id="@+id/my\_appbar\_container"  
 android:layout\_width="match\_parent"  
 android:layout\_height="192dp"  
 android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"  
 android:background="@color/colorPrimary"** >  
  
 <**android.support.design.widget.CollapsingToolbarLayout  
 android:id="@+id/collapsing\_container"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_scrollFlags="scroll|exitUntilCollapsed"  
 app:contentScrim="@color/colorPrimary"  
 app:expandedTitleMarginStart="48dp"  
 app:expandedTitleMarginEnd="64dp"**>  
  
 <**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 style="@style/TechniqueToolbar"  
 app:layout\_collapseMode="pin"** />  
  
 </**android.support.design.widget.CollapsingToolbarLayout**>  
  
  
 </**android.support.design.widget.AppBarLayout**>  
  
  
 <**android.support.v7.widget.RecyclerView  
 android:id="@+id/days\_list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"**/>  
  
 <**android.support.design.widget.FloatingActionButton  
 android:id="@+id/fab"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="16dp"  
 android:src="@drawable/plus"  
 app:borderWidth="0dp"  
 app:layout\_anchor="@id/my\_appbar\_container"  
 app:layout\_anchorGravity="bottom|right|end"** />  
  
</**android.support.design.widget.CoordinatorLayout**>

Thật ra cũng như loại 3 nhưng sửa như sau:

* Toolbar được bỏ trong CollapsingToolBarLayout (và là con của AppBarLayout)
* App:layout\_scrollFlags di chuyển từ Toolbar sang CollapsingToolBarLayout.
* Thuộc tính mới app:layout\_collaseMode được thêm vào Toolbar, thuộc tính này đảm bảo toolbar được gắn vào đầu của màn hình.
* AppBarLayout được fix chiều cao là 192dp
* Bỏ TabLayout (nếu có).
* FloatingActionButton được thêm vào cuối layout, bên dưới RecyclerView.

Những class mới mà ta cần quan tâm:

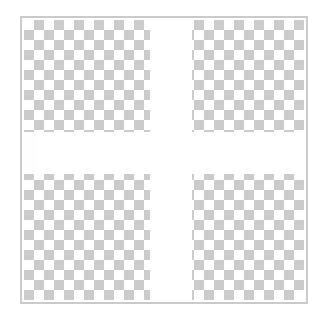
CollapsingToolBarLayout: Một view mới được thiết kế đặt biệt để bọc Toolbar và thực thi việc thu gọn appbar khi cuộn. Khi dùng nó phải để ý thuộc tính app:contentScrim, thuộc tính này chỉ ra màu hiển thị khi đã thu xong.

app:expandedTitleMarginStart/app:expandedTitleMarginEnd : thuộc tính này chỉ ra margin khi title được phóng to. Nó dung khi ta sử dụng setDisplayHomeAsUpEnabled trong activity và fill khoảng trống được tạo quanh title.

FloatingActionButton: là một action quan trong trong Material Design. Dùng app:fabSize để chọn 2 chuẩn kích thước là standard (56dp) và mini (40dp). Hiệu ứng biến mất tự động của nó dựa vào AppBarLayout bằng cách dung app:layout\_anchor. Có thể chỉnh vị trí chon ó bằng thuộc tính app:layout\_anchorGravity.

Chú ý: floatactionbutton bản thân nó đã có hình tròn do đó hình của nó sẽ như sau:

Và màu của nó là android:backgroundTint, có thể thêm eleelevation để nổi lên và clickable=truetrue



1. Trong file Loai4Activity.java chỉ cần tạo các view toàn cục, trong onCreate ánh xạ nó và viết thêm như sau:

**toolbar** = (Toolbar) findViewById(R.id.***toolbar***);  
**days\_list** = (RecyclerView) findViewById(R.id.***days\_list***);  
  
setSupportActionBar(**toolbar**);  
getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
  
**toolbar**.setTitle(**"loai 4"**);  
  
  
**days\_list**.setLayoutManager(**new** LinearLayoutManager(**this**));  
**days\_list**.setAdapter(**new** RecyclerViewAdapter(  
 getResources().getStringArray(R.array.***days\_names***)  
));

1. Thêm nút back trên toolbar để quay về bằng cách

getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);

và trong AndroidManifest.xml thêm meta-data vào activity để chỉ ra quay về đâu.

<**activity android:name=".Loai4Activity"**>  
 <**meta-data android:name="android.support.PARENT\_ACTIVITY"  
 android:value=".MainActivity"** />  
 </**activity**>

1. Muốn đổi title thì không dùng toolbar.settitle nữa mà dùng CollapsingToolbarLayout

**collapsing\_container**.setTitle(**"loai 4"**);

1. Chạy ct, cuộn để xem kết quả

|  |  |
| --- | --- |
| Figure Khi chưa cuộn | Figure Khi cuộn |

# Loại 5 – Hình trên toolbar

1. Mở file activity\_loai5.xml và sửa như sau:

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.design.widget.CoordinatorLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="luffy.khoa.anh.layouttonghop.Loai2Activity"**>  
  
 <**android.support.design.widget.AppBarLayout  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:id="@+id/my\_appbar\_container"  
 android:layout\_width="match\_parent"  
 android:layout\_height="192dp"  
 android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"** >  
  
 <**android.support.design.widget.CollapsingToolbarLayout  
 android:id="@+id/collapsing\_container"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_scrollFlags="scroll|exitUntilCollapsed"  
 app:contentScrim="@color/colorPrimary"** >  
  
 <**ImageView  
 android:id="@+id/imgToolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:scaleType="centerCrop"  
 android:src="@drawable/small"  
 app:layout\_collapseMode="parallax"** />  
  
 <**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_height="?attr/actionBarSize"  
 android:layout\_width="match\_parent"  
 app:layout\_collapseMode="pin"** />  
  
 </**android.support.design.widget.CollapsingToolbarLayout**>  
  
  
 </**android.support.design.widget.AppBarLayout**>  
  
  
 <**android.support.v7.widget.RecyclerView  
 android:id="@+id/days\_list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"**/>  
  
 <**android.support.design.widget.FloatingActionButton  
 android:id="@+id/fab"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="16dp"  
 android:src="@drawable/plus"  
 app:borderWidth="0dp"  
 app:layout\_anchor="@id/my\_appbar\_container"  
 android:backgroundTint="#ff0000"  
 android:elevation="8dp"  
 android:clickable="true"  
 app:layout\_anchorGravity="bottom|right|end"** />  
  
</**android.support.design.widget.CoordinatorLayout**>

Thật ra chính là loai\_4 nhưng sửa như sau:

* Thuộc tính android:background của AppBarLayout được bỏ (vì imageview sẽ ở đó nên không cần background).
* App:expandedTitleMarginStart và app:expandedTitleMargineEnd được bỏ vì không dùng setDisplayHomeAsUpEnabled trong activity nữa.
* Toolbar bỏ thuộc tính style để mất màu đi
* Một Imageview được thêm vào trước Toolbar. Trong ImageView có thuộc tính app:layout\_collapseMode được gán bằng parallax để thực hiện hiệu ứng parallax khi cuộn. có thể thêm app:layout\_collapseParallaxMultiplier để gắn multiplier khi cuộn.

1. File Loai5Activity.java thì giống y như Loai4Activity.java
2. Chạy ct để xem kết quả:

|  |  |
| --- | --- |
| Figure chưa cuộn | Figure Cuộn |

# Loại 6

1. Mở file activity\_loai6.xml và sửa như sau:

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.design.widget.CoordinatorLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="luffy.khoa.anh.layouttonghop.Loai2Activity"**>  
  
 <**android.support.design.widget.AppBarLayout  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:id="@+id/my\_appbar\_container"  
 android:layout\_width="match\_parent"  
 android:layout\_height="192dp"  
 android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"** >  
  
 <**android.support.design.widget.CollapsingToolbarLayout  
 android:id="@+id/collapsing\_container"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_scrollFlags="scroll|exitUntilCollapsed"  
 android:background="@color/colorPrimary"** >  
  
 <**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_height="?attr/actionBarSize"  
 android:layout\_width="match\_parent"  
 app:layout\_collapseMode="pin"  
 android:background="@color/colorPrimary"** />  
  
 </**android.support.design.widget.CollapsingToolbarLayout**>  
  
  
 </**android.support.design.widget.AppBarLayout**>  
  
  
 <**android.support.v7.widget.RecyclerView  
 android:id="@+id/days\_list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 app:behavior\_overlapTop="64dp"** />  
  
  
</**android.support.design.widget.CoordinatorLayout**>

Khác loại 4 ở chổ

ImageView và FloatingActionButton bên trong CollapsingToolbarLayout đã được bỏ.

Trong CollapsingToolbarLayout, thuộc tính app:contentScrim được thay bằng android:background.

Android:background được thêm vào ToolbarToolbar

App:behavior\_overlapTop được thêm vào RecyclerView.

Trên giao diện cũng không có phần text của toolbar và nó bị dè lên thì vào manifrest tìm đến activiy6 và gán android:label=””

1. File Loai6Activity.java cũng tương tự loại 5 tuy nhiên để thêm text cho toolbar viết thêm như sau:

TextView text = **new** TextView(**this**);  
text.setText(**"loai 6"**);  
text.setTextAppearance(**this**, android.R.style.***TextAppearance\_Material\_Widget\_ActionBar\_Title\_Inverse***);  
 **toolbar**.addView(text);

1. Chạy ct và xem kết quả

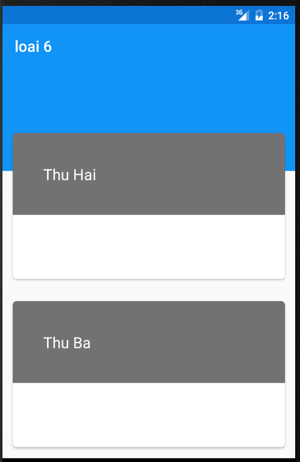


Figure Một phần Recyclerview đè lên toolbar

# Nói thêm về LayoutManager của RecyclerView

## Loại LinearLayout và GridLayout

LinearLayout là từng hang (giống như listview) loại GridLayout có cột có hang (giống GridView)

1. Mở Loai6Activity.java, sửa lại phần LayoutManager như sau:

**days\_list**.setLayoutManager(**new** LinearLayoutManager(**this**,LinearLayoutManager.***HORIZONTAL***,**false**));

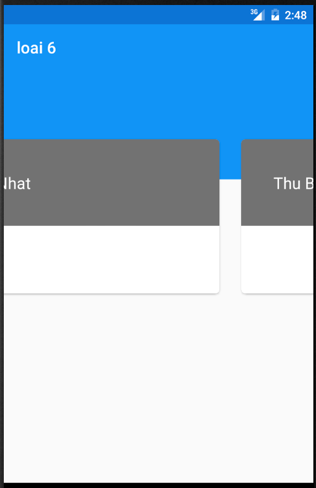


Figure RecyclerView cuộn ngang

1. Sửa lại như sau:

GridLayoutManager layoutManager=**new** GridLayoutManager(**this**,  
 2,//so o tren 1 hang  
 GridLayoutManager.***VERTICAL***,  
 **false** );  
 **days\_list**.setLayoutManager(layoutManager);

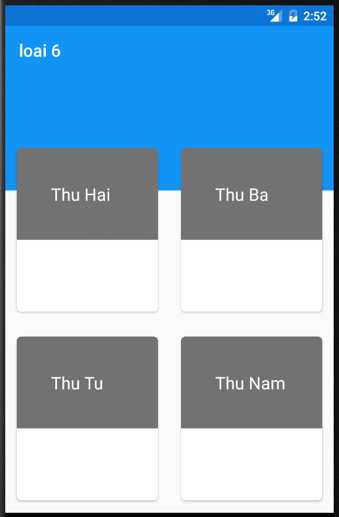


Figure hiện kiểu lưới

1. Sửa lại như code (thêm phần setSpanSizeLookup)

GridLayoutManager layoutManager=**new** GridLayoutManager(**this**,  
 2,  
 GridLayoutManager.***VERTICAL***,  
 **false**);  
layoutManager.setSpanSizeLookup(  
 **new** GridLayoutManager.SpanSizeLookup() {  
 @Override  
 **public int** getSpanSize(**int** position) {  
 **return** (position % 3 == 0 ? 2 : 1 );  
 }  
 }  
);  
**days\_list**.setLayoutManager(layoutManager);

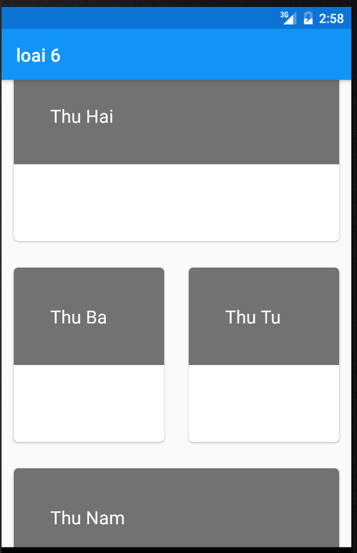
 

Figure một hàng 1, 1 hàng 2

## Loại StaggeredGridLayout

Loại StaggeredGridLayout phức tạp hơn, nó có thể tạo ra các hàng cột lệch nhau



1. Tạo một file layout mới tên “layout\_one\_item\_loai6.xml” và code

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.v7.widget.CardView  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:card\_view="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 card\_view:cardCornerRadius="5dp"  
 android:layout\_margin="5dp"**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"** >  
  
 <**ImageView  
 android:layout\_margin="7dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/img\_row"** />  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/txv\_row"  
 android:text="Hello How Are you"  
 android:layout\_marginLeft="10dp"  
 android:layout\_gravity="center\_horizontal"**/>  
  
 </**LinearLayout**>  
</**android.support.v7.widget.CardView**>

Cũng không có gì, chỉ là CardView có 1 hình và một TextView

1. Copy hình từ thư mục “bo\_hinh\_bai10” vào thư mục drawable
2. Tạo một file mới tên Information.java và code

**public class** Information {  
 **public int imageId**;  
 **public** String **title**;  
}

1. Tạo một file Tên Data.java và copy code (file này chỉ là làm ra data để sử dụng)

**public class** Data {  
  
 **public static** ArrayList<Information> getData() {  
  
 ArrayList<Information> data = **new** ArrayList<>();  
  
 **int**[] images = {  
 R.drawable.***ani\_cat\_one***,  
 R.drawable.***ani\_cat\_two***,  
 R.drawable.***ani\_cat\_three***,  
 R.drawable.***ani\_cat\_four***,  
 R.drawable.***ani\_cat\_five***,  
 R.drawable.***ani\_cat\_six***,  
 R.drawable.***ani\_cat\_seven***,  
  
 R.drawable.***ani\_dog\_one***,  
 R.drawable.***ani\_dog\_two***,  
 R.drawable.***ani\_dog\_three***,  
 R.drawable.***ani\_dog\_four***,  
 R.drawable.***ani\_dog\_five***,  
  
 R.drawable.***ani\_deer\_one***,  
 R.drawable.***ani\_deer\_two***,  
 R.drawable.***ani\_deer\_three***,  
 R.drawable.***ani\_deer\_four***,  
  
 R.drawable.***bird\_parrot\_one***,  
 R.drawable.***bird\_parrot\_two***,  
 R.drawable.***bird\_parrot\_three***,  
 R.drawable.***bird\_parrot\_four***,  
 R.drawable.***bird\_parrot\_five***,  
 R.drawable.***bird\_parrot\_six***,  
 R.drawable.bird\_parrot\_seven,  
 R.drawable.***bird\_parrot\_eight*** };  
  
 String[] Categories = {**"Cat 1"**, **"Cat 2"**, **"Cat 3"**, **"Cat 4"** ,**"Cat 5"** ,**"Cat 6"**,**"Cat 7"**,  
 **"Dog 1"**,**"Dog 2"**,**"Dog 3"**,**"Dog 4"**,**"Dog 5"**,  
 **"Deer 1"**,**"Deer 2"**,**"Deer 3"**,**"Deer 4"**,  
 **" Parrot 1"**,**" Parrot 2"**,**" Parrot 3"**,**" Parrot 4"**,**" Parrot 5"**,**" Parrot 6"**,**" Parrot 7"**,**" Parrot 8"**};  
  
 **for** (**int** i = 0; i < images.**length**; i++) {  
  
 Information current = **new** Information();  
 current.**title** = Categories[i];  
 current.**imageId** = images[i];  
  
 data.add(current);  
 }  
  
 **return** data;  
 }  
  
}

1. Tạo một file mới tên AnimationUtil.java copy code (file này chỉ là tạo animation, bỏ cũng được)

**public class** AnimationUtil {  
  
 **public static void** animate(RecyclerView.ViewHolder holder ,**boolean** goesDown){  
  
  
 AnimatorSet animatorSet = **new** AnimatorSet();  
  
 ObjectAnimator animatorTranslateY = ObjectAnimator.*ofFloat*(holder.**itemView**, **"translationY"**, goesDown==**true** ? 200 : -200, 0);  
 animatorTranslateY.setDuration(1000);  
  
  
 ObjectAnimator animatorTranslateX = ObjectAnimator.*ofFloat*(holder.**itemView**,**"translationX"**,-50,50,-30,30,-20,20,-5,5,0);  
 animatorTranslateX.setDuration(1000);  
  
 animatorSet.playTogether(animatorTranslateX,animatorTranslateY);  
  
 *//animatorSet.playTogether(animatorTranslateY);* animatorSet.start();  
  
 }  
}

1. Tạo file mới tên AdapterLoai6.java (file này chính là apdater cho recyclerview)

**public class** AdapterLoai6 **extends** RecyclerView.Adapter<AdapterLoai6.MyViewHolder> {  
  
 **private** Context **context**;  
 **private** ArrayList<Information> **data**;  
 **private** LayoutInflater **inflater**;  
 **private int previousPosition** = 0;  
  
 **public** AdapterLoai6(Context context, ArrayList<Information> data) {  
  
 **this**.**context** = context;  
 **this**.**data** = data;  
 **inflater** = LayoutInflater.*from*(context);  
 }  
  
 @Override  
 **public** MyViewHolder onCreateViewHolder(ViewGroup parent, **int** position) {  
  
 View view = **inflater**.inflate(R.layout.***layout\_one\_item\_loai6***, parent, **false**);  
 MyViewHolder holder = **new** MyViewHolder(view);  
 **return** holder;  
 }  
  
 @Override  
 **public void** onBindViewHolder(MyViewHolder myViewHolder, **final int** position) {  
  
 myViewHolder.**textview**.setText(**data**.get(position).**title**);  
 myViewHolder.**imageView**.setImageResource(**data**.get(position).**imageId**);  
 **if**(position > **previousPosition**){ *// We are scrolling DOWN* AnimationUtil.*animate*(myViewHolder, **true**);  
 }**else**{ *// We are scrolling UP* AnimationUtil.*animate*(myViewHolder, **false**);  
 }  
 **previousPosition** = position;  
  
 **final int** currentPosition = position;  
 **final** Information infoData = **data**.get(position);  
 myViewHolder.**imageView**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
  
 Toast.*makeText*(**context**, **"OnClick Called at position "** + position, Toast.***LENGTH\_SHORT***).show();  
 addItem(currentPosition, infoData);  
 }  
 });  
  
 myViewHolder.**imageView**.setOnLongClickListener(**new** View.OnLongClickListener() {  
 @Override  
 **public boolean** onLongClick(View v) {  
 Toast.*makeText*(**context**, **"OnLongClick Called at position "** + position, Toast.***LENGTH\_SHORT***).show();  
 removeItem(infoData);  
 **return true**;  
 }  
 });  
 }  
  
 @Override  
 **public int** getItemCount() {  
 **return data**.size();  
 }  
  
 **class** MyViewHolder **extends** RecyclerView.ViewHolder{  
  
 TextView **textview**;  
 ImageView **imageView**;  
  
 **public** MyViewHolder(View itemView) {  
 **super**(itemView);  
 **textview** = (TextView) itemView.findViewById(R.id.***txv\_row***);  
 **imageView** = (ImageView) itemView.findViewById(R.id.***img\_row***);  
 }  
 }  
  
 *// This removes the data from our Dataset and Updates the Recycler View.* **private void** removeItem(Information infoData) {  
  
 **int** currPosition = **data**.indexOf(infoData);  
 **data**.remove(currPosition);  
 notifyItemRemoved(currPosition);  
 }  
  
 *// This method adds(duplicates) a Object (item ) to our Data set as well as Recycler View.* **private void** addItem(**int** position, Information infoData) {  
  
 **data**.add(position, infoData);  
 notifyItemInserted(position);  
 }  
}

1. Quay về file Loai6Activity.java, loại bỏ hết các lệnh về adapter cũ đi và thêm như sau:

AdapterLoai6 adapter = **new** AdapterLoai6(**this**, Data.*getData*());  
StaggeredGridLayoutManager layoutManager = **new** StaggeredGridLayoutManager(2, StaggeredGridLayoutManager.***VERTICAL***);  
**days\_list**.setLayoutManager(layoutManager);  
  
**days\_list**.setAdapter(adapter);

1. Chạy ct và xem kết quả:

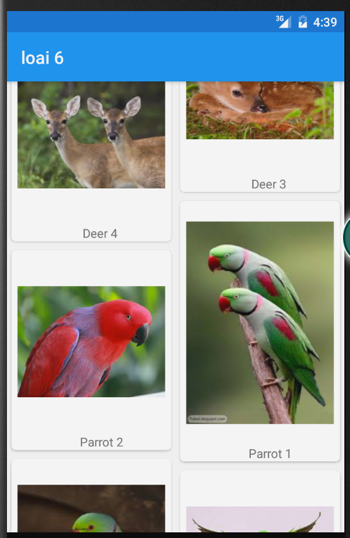


Figure Hình lệnh nhau và khi cuộn có hiệu ứng

* View ôm theo kích thước hình, nên hình đứng và hình ngang sen kẽ làm cho layout lệch đi. Hoặc có thể tuỳ theo vị trí mà set kích thước view cho nó to nhỏ để lệch, tham khảo <http://www.theappguruz.com/blog/learn-recyclerview-with-an-example-in-android>
* Link trên cũng có chứa cách dùng Picasso để load hình trên mạng trong hàm onBindViewHolder vd:

Picasso.with(activity)

.load(images.get(position))

.error(R.drawable.ic\_empty)

.placeholder(R.drawable.ic\_launcher)

.resize(screenWidth / 2, height)

.centerCrop()

.into((myHolder.images));

* Có thể sửa .VERTICAL thành .HORIZONTAL để chạy cuộn ngang
* Chú ý: click vào 1 item sẽ double item đó, do trong adapter viết vậy, muốn thì đọc adapter và bỏ chức năng đó đi.

Xong, mệt quá trời