

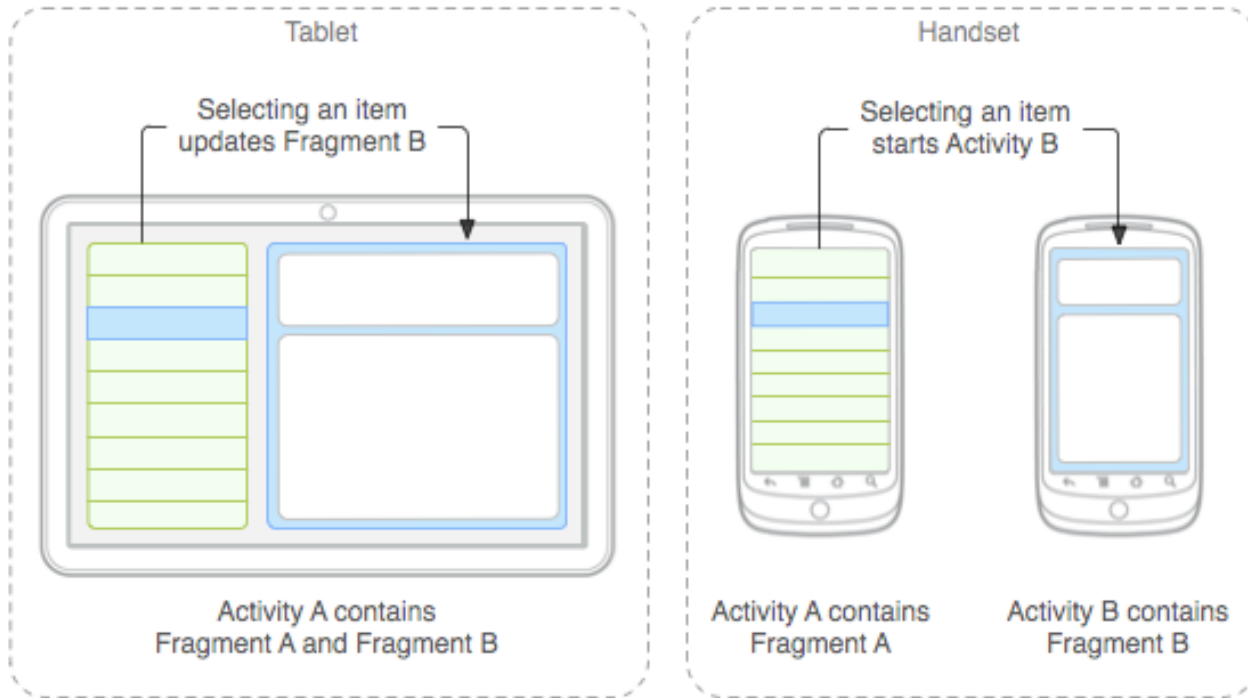
# ANDROID PROGRAMMING

## LESSON 6

# Agenda

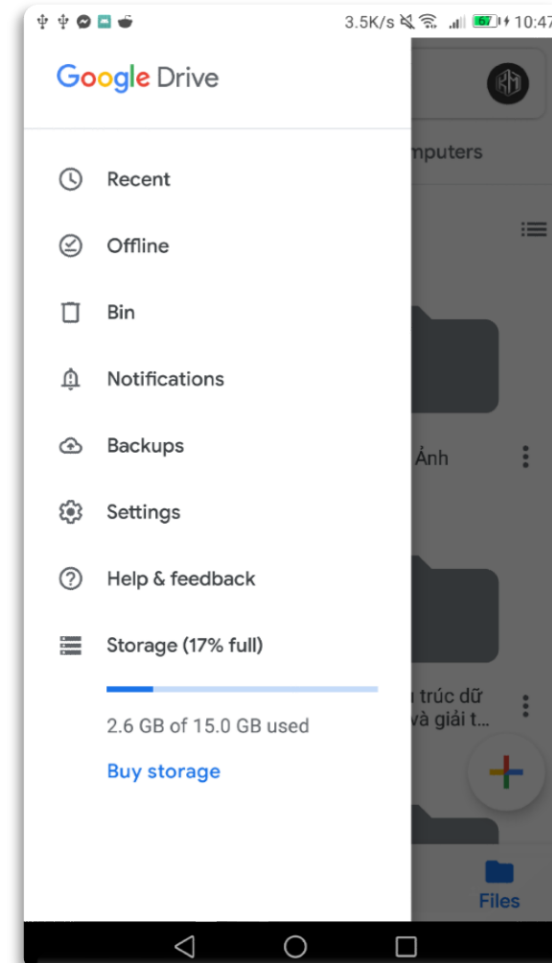
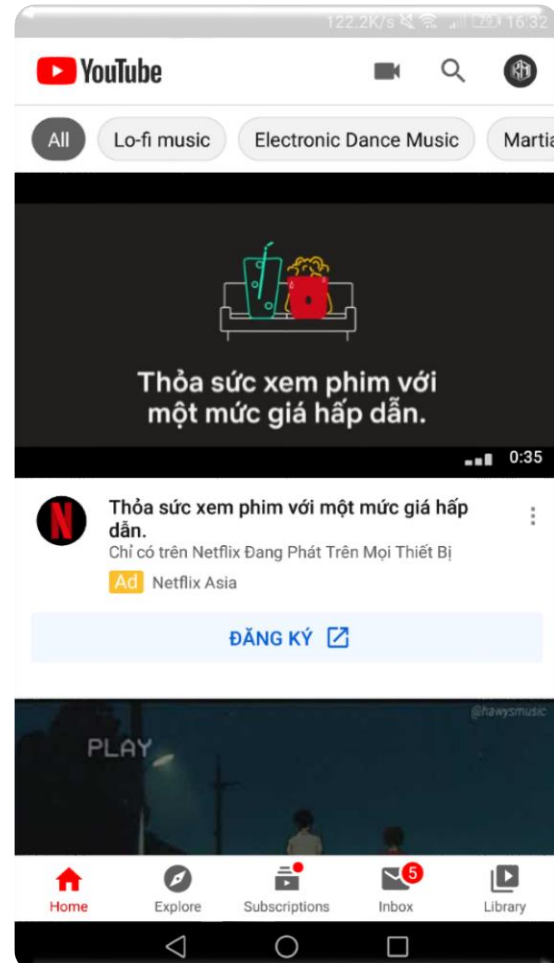
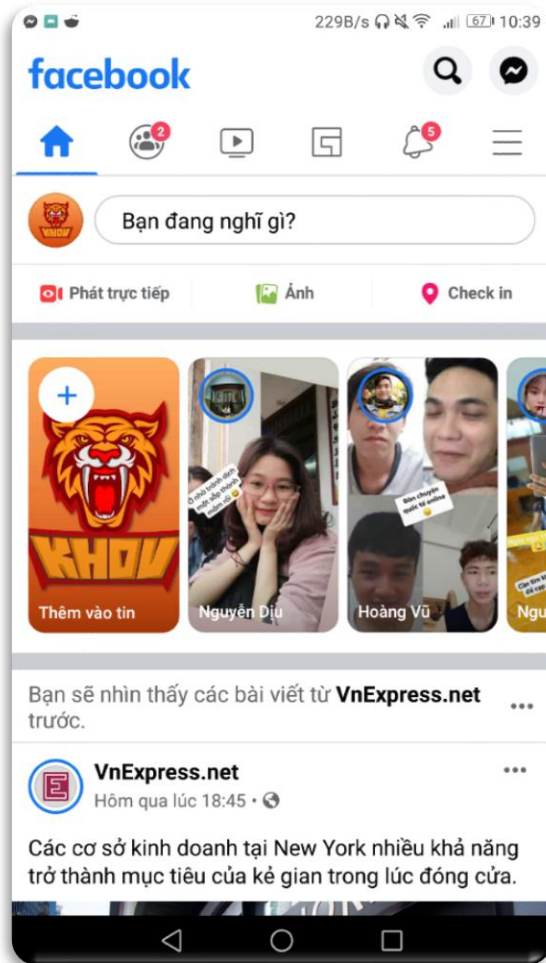
- Fragment in Android
- Static and Dynamic Fragment (01)
- Fragment from Backstack (02)
- ViewPager and FragmentPagerAdapter (03)
- Ví dụ kết hợp Fragment, TabLayout, ViewPager và RecyclerView (04)

# Fragment in Android

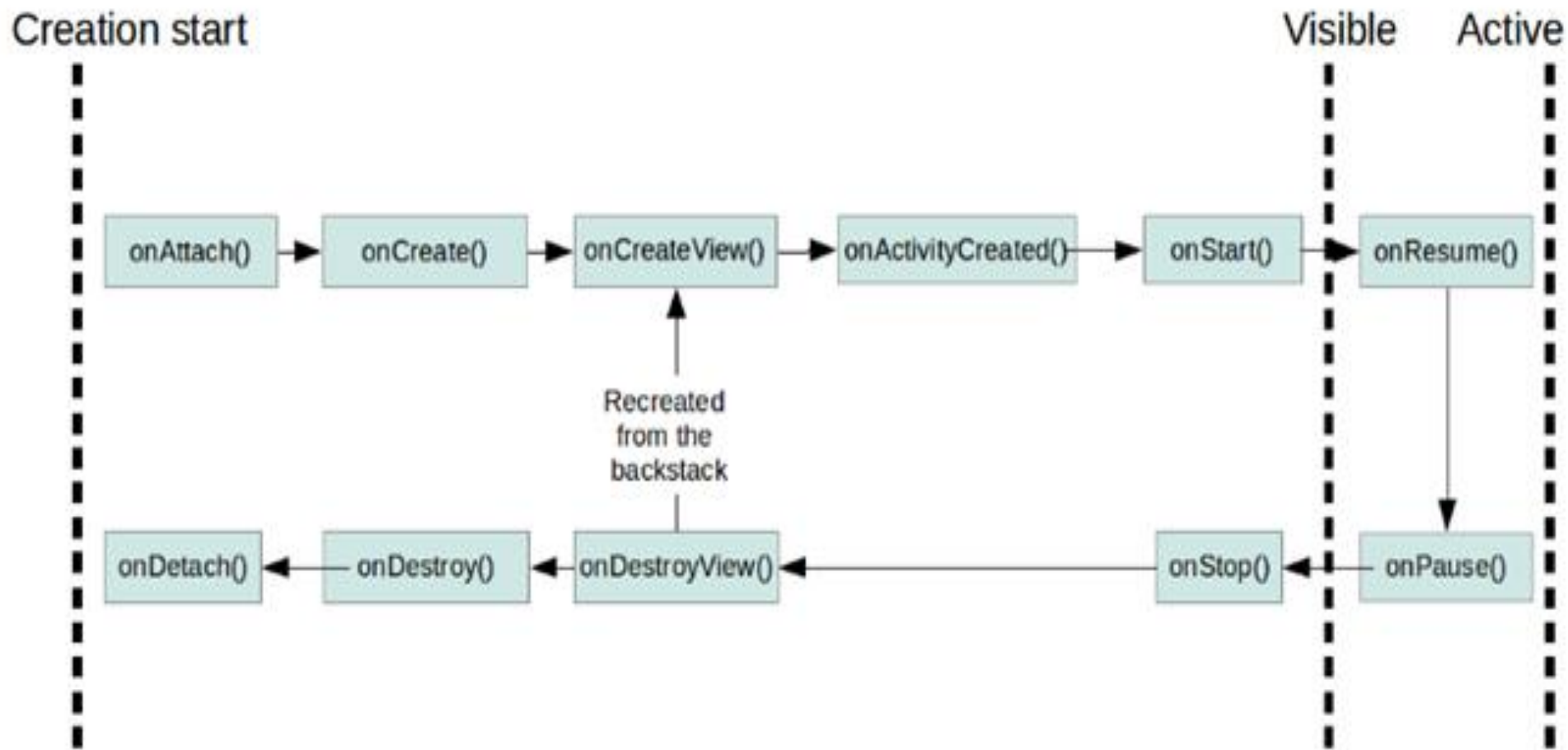


- Fragment chịu trách nhiệm quản lý một không gian màn hình
- Không gian màn hình đó của Fragment phải nằm trong một Activity nào đó
- Một Activity có thể có nhiều Fragment
- Một Fragment nào đó cũng có thể được khai báo và sử dụng bởi nhiều Activity khác nhau

# Fragment thường được sử dụng trong:



# Fragment Life Cycle



# Fragment Life Cycle

Khi Fragment được gắn vào Activity, các callback **onAttach()**, **onCreate()**, **onCreateView()**, **onActivityCreated()**, **onStart()**, **onResume()** lần lượt được gọi.

Sau khi các callback trên được gọi, fragment lúc đó mới chính thức được xem là đang chạy.

Sau đó, nếu người dùng bấm nút **Back** hay có bất kì thao tác gỡ/ thay thế fragment ra khỏi activity nào thì các callback **onPause()**, **onStop()**, **onDestroyView()**, **onDestroy()**, **onDetach()** sẽ được gọi. (Đây là trường hợp fragment chưa được thêm vào back stack. Ở phía dưới mình sẽ nói về trường hợp fragment được thêm vào back stack sau)

# Khai báo Fragment trong file giao diện (xml)

- **Statically** (Kiểu tĩnh) : cố định , không thay đổi bằng fragment khác được trong quá trình runtime

```
<fragment  
    android:id="@+id/list_fragment"  
    class="com.example.ExampleFragment "  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" />
```

- **Dynamically** (Kiểu động) : có thể được thay đổi bằng các fragment khác trong quá trình runtime

```
<FrameLayout  
    android:id="@+id/placeholder"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"/>
```

# fragment\_first.xml (01)

```
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFEB3B">
    <TextView
        android:id="@+id/textView"
        android:textSize="30dp"
        android:textColor="@color/colorPrimary"
        android:gravity="center"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Example for static fragment"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.188" />
</androidx.constraintlayout.widget.ConstraintLayout>
```



# StaticFragment.java

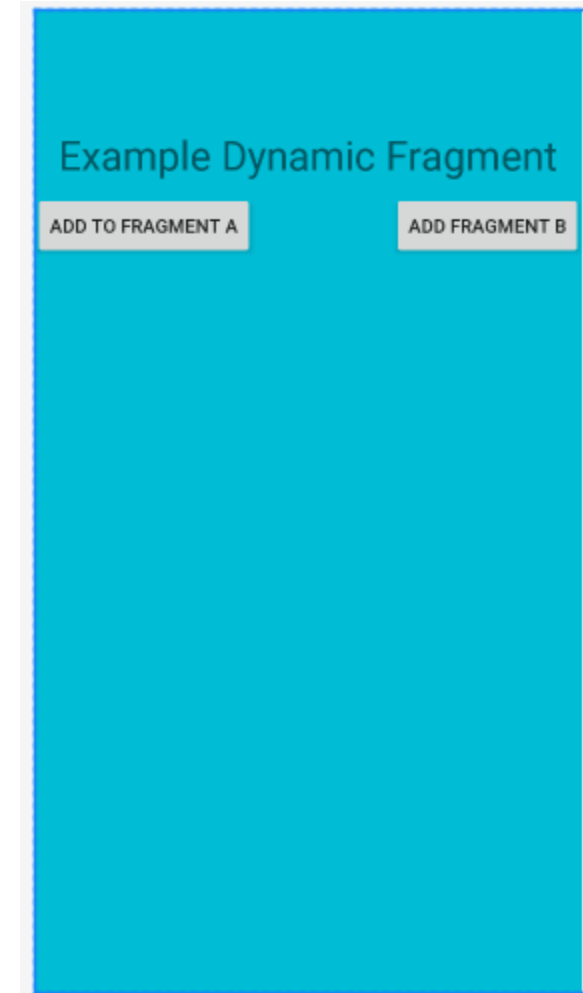
```
public class StaticFragment extends Fragment {  
    @Nullable  
    @Override  
    public View onCreateView(@NonNull LayoutInflater inflater,  
                             @Nullable ViewGroup container,  
                             @Nullable Bundle savedInstanceState) {  
        View view=inflater.inflate(R.layout.fragment_first,container,false);  
        return view;  
    }  
}
```

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
<TextView
    android:id="@+id/tv"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="This is a Activity"
    android:textSize="28dp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.128" />
<fragment
    android:id="@+id/fragA"
    app:layout_constraintTop_toBottomOf="@id/tv"
    android:layout_marginTop="50dp"
    android:name="com.example.a06_01_fragmentdemo.StaticFragment"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

## Static fragment

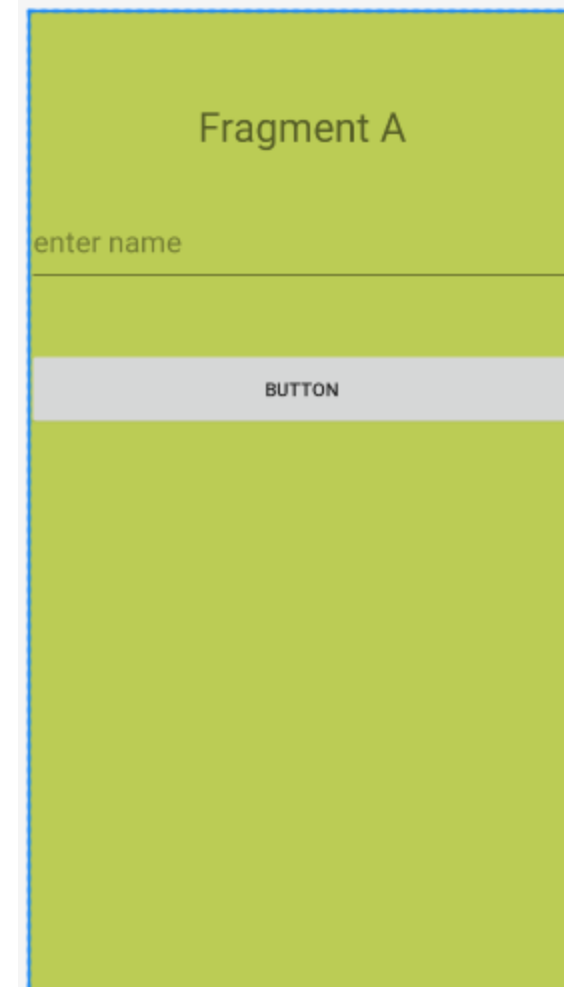
# main\_activity.xml – Dynamic Fragment

```
<TextView
    android:id="@+id/textView2"
    android:gravity="center"
    android:textSize="30dp"
    android:layout_width="match_parent"
    android:layout_height="58dp"
    android:text="Example Dynamic Fragment"/>
<Button
    android:id="@+id/addA"
    android:text="ADD to Fragment A"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
<Button
    android:id="@+id/addB"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Add fragment B"/>
<FrameLayout
    android:id="@+id/layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
```



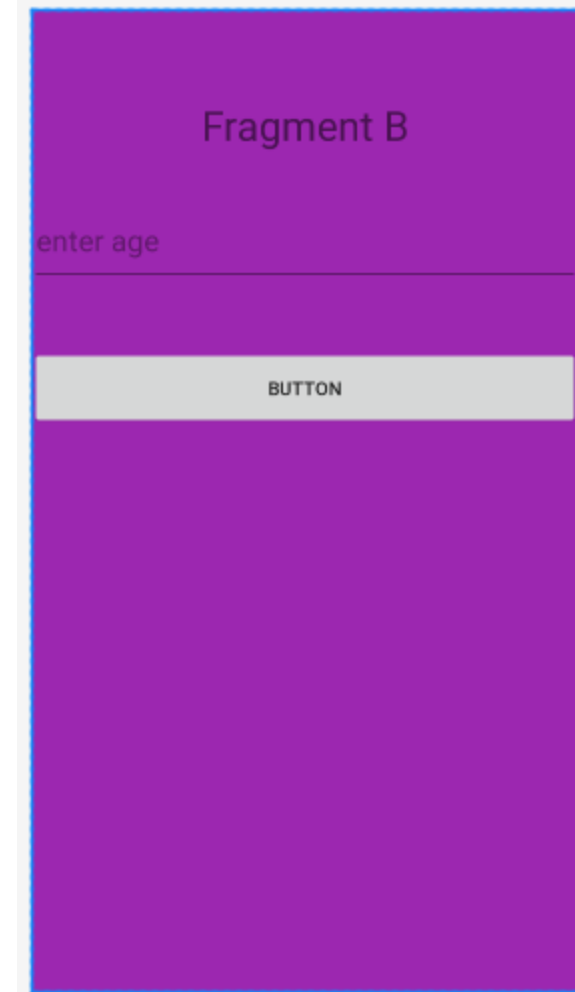
# Fragment\_a.xml

```
<TextView
    android:id="@+id/tv"
    android:layout_width="match_parent"
    android:layout_height="49dp"
    android:gravity="center"
    android:text="Fragment A"
    android:textSize="30dp"/>
<EditText
    android:id="@+id/name"
    android:layout_width="match_parent"
    android:layout_height="67dp"
    android:layout_marginTop="28dp"
    android:hint="enter name"
    android:textSize="22dp"/>
<Button
    android:id="@+id/bt"
    android:layout_width="match_parent"
    android:layout_height="59dp"
    android:layout_marginBottom="308dp"
    android:text="Button"/>
```



# Fragment\_b.xml

```
<TextView
    android:id="@+id/tv"
    android:layout_width="match_parent"
    android:layout_height="49dp"
    android:gravity="center"
    android:text="Fragment B"
    android:textSize="30dp"/>
<EditText
    android:id="@+id/name"
    android:layout_width="match_parent"
    android:layout_height="67dp"
    android:hint="enter age"
    android:textSize="22dp"/>
<Button
    android:id="@+id/bt"
    android:layout_width="match_parent"
    android:layout_height="59dp"
    android:text="Button"/>
```



```

public class FragmentA extends Fragment {
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,
@Nullable Bundle savedInstanceState) {
        View v=inflater.inflate(R.layout.fragment_a,container,false);
        return v;
    }
    @Override
    public void onViewCreated(@NonNull View view,
                                @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        final EditText txtName=view.findViewById(R.id.name);
        Button bt=view.findViewById(R.id.bt);
        bt.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String name=txtName.getText().toString();
                txtName.setTextSize(30);
                txtName.setTextColor(Color.BLUE);
                txtName.setText("Hello "+name+"!");
            }
        });
    }
}

```

```

public class FragmentB extends Fragment {
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,
@Nullable Bundle savedInstanceState) {
        View v=inflater.inflate(R.layout.fragment_b,container,false);
        return v;
    }
    @Override
    public void onViewCreated(@NonNull View view,
                             @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        final EditText txtName=view.findViewById(R.id.name);
        Button bt=view.findViewById(R.id.bt);
        bt.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String age=txtName.getText().toString();
                txtName.setTextSize(30);
                txtName.setTextColor(Color.RED);
                txtName.setText("Age: "+age+" to old!");
            }
        });
    }
}

```

## Dynamic Fragment

### FragmentManager

Đây là một lớp dùng để quản lý các Fragment, lớp này được tích hợp vào trong mỗi activity để giúp các activity có thể dễ dàng để thêm (add), xóa (remove) hoặc thay thế (replace) các fragment ra khỏi một vùng không gian một cách linh động

### FragmentTransaction

add() - Khi FrameLayout còn rỗng, tức chưa chứa đựng bất kỳ fragment nào, thì bạn có thể dùng phương thức này để add fragment vào cho FrameLayout đó. Nếu bạn tiếp tục thêm fragment khi đã tồn tại thì các fragment sẽ chạy song song nhưng fragment sau sẽ che mất view của fragment trước

replace() - Khi bạn muốn thay thế một fragment đang có sẵn ở FrameLayout bằng một fragment nào đó khác.

remove() - Khi bạn muốn gỡ bỏ fragment ra khỏi một FrameLayout.

commit() - Để FragmentTransaction biết sẽ bắt đầu thực hiện các transaction mà bạn đã ra lệnh đó.



# MainActivity.java

```
public class MainActivity extends AppCompatActivity{
    private FragmentManager manager;
    private FragmentTransaction transaction;
    Fragment fragment=null;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main1);

        manager=getSupportFragmentManager();
        transaction=manager.beginTransaction();
        FragmentA fragment=new FragmentA();
        transaction.add(R.id.layout,fragment);
        transaction.commit();
    }
}
```

# MainActivity.java

```
public class MainActivity extends AppCompatActivity
    implements View.OnClickListener{
    private FragmentManager manager;
    private FragmentTransaction transaction;
    private Button btA, btB;
    Fragment fragment=null;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main1);
        btA=findViewById(R.id.addA);
        btB=findViewById(R.id.addB);
        btA.setOnClickListener(this);
        btB.setOnClickListener(this);
    }
```

@Override

```
public void onClick(View v) {  
    manager=getSupportFragmentManager();  
    transaction=manager.beginTransaction();  
    if(v==btA){  
        fragment=new FragmentA();  
        transaction.replace(R.id.Layout,fragment);  
    }  
    if(v==btB){  
        fragment=new FragmentB();  
        transaction.replace(R.id.Layout,fragment);  
    }  
    transaction.commit();  
}
```

```
}
```

## BackStack

Khi bạn quản lý fragment bởi các phương thức `replace()` hay `remove()` trên đây, bạn có thể sử dụng thêm phương thức `addToBackStack()` này. Nếu bạn gọi đến phương thức này trước khi gọi `commit()` sẽ được nói ở dưới, thì hệ thống sẽ đưa fragment ở transaction hiện tại vào Back Stack. Điều này có ý nghĩa là, fragment bị thay thế hay bị gỡ ra khỏi `FrameLayout` ở transaction này sẽ không bị xóa khỏi hệ thống mà vẫn còn được quản lý bên trong Back Stack. Và do fragment không bị hủy khỏi back stack, nên nếu user nhấn nút back sau đó, họ hoàn toàn có thể quay trở lại với fragment trước đó đã bị gỡ ra.

# Cơ chế Backstack

- Giả sử trong trường hợp chỉ có một khung Fragment, và muốn hoán đổi nhiều Fragment khác lần lượt vào cái khung đó?
- Nút **Back** trên thiết bị Android mặc định sẽ chỉ có tác dụng với Activity hiện thời chứ không phải Fragment. Mà ở đây nút **Back** để quay lại Fragment trước? Ta sẽ cần sử dụng Fragment Backstack: Đưa các Fragment vào một ngăn xếp (giống như khi ta duyệt trang web vậy, nhấn **back** thì trình duyệt sẽ quay lại trang trước đó).
- Ta sử dụng phương thức `addToBackStack()` để quay lại fragment trước đó

# main\_activity.xml (02)

```
<LinearLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
xmlns:android="http://schemas.android.com/apk/res/android">
..... • •
    <Button
        android:id="@+id/addA"
        android:onClick="addA"
        android:text="Add A"
        android:layout_weight="1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
..... • ..... • •
    </LinearLayout>
    <FrameLayout
        android:layout_marginTop="50dp"
        android:id="@+id/frame"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
</LinearLayout>
```

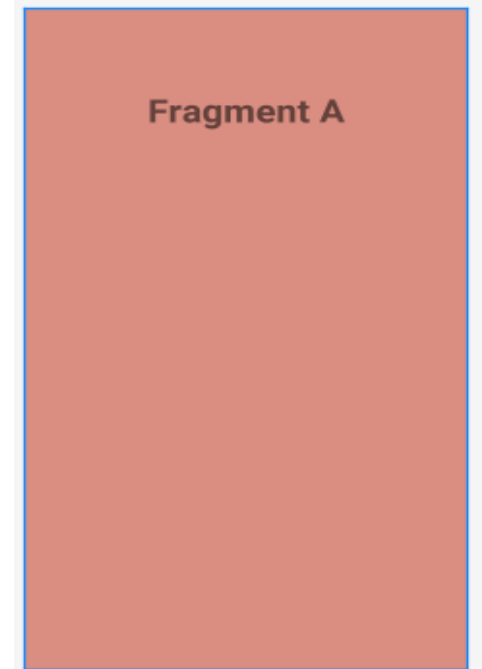


# fragment\_a.xml and FragmentA.java

<TextView

```
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Fragment A"
    android:textSize="35dp"
    android:textStyle="bold"
/>
```

```
public class FragmentA extends Fragment {
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable
ViewGroup container, @Nullable Bundle savedInstanceState) {
        View view=inflater.inflate(R.layout.fragment_a,container,false);
        return view;
    }
}
```



# fragment\_b.xml and FragmentB.java

<TextView

```
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Fragment B"
    android:textSize="35dp"
    android:textStyle="bold"
/>
```

```
public class FragmentB extends Fragment {
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable
ViewGroup container, @Nullable Bundle savedInstanceState) {
        View view=inflater.inflate(R.layout.fragment_b,container,false);
        return view;
    }
}
```





# fragment\_c.xml and FragmentC.java

<TextView

```
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Fragment C"
    android:textSize="35dp"
    android:textStyle="bold"
/>
```

```
public class FragmentC extends Fragment {
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable
ViewGroup container, @Nullable Bundle savedInstanceState) {
        View view=inflater.inflate(R.layout.fragment_c,container,false);
        return view;
    }
}
```



## addA & removeA

```
public void addA(View v){  
    FragmentTransaction transaction=manager.beginTransaction();  
    FragmentA fragment=new FragmentA();  
    transaction.add(R.id.frame,fragment,"fragA");  
    transaction.addToBackStack("fa");  
    transaction.commit();  
}
```

```
public void removeA(View v){  
    FragmentTransaction transaction=manager.beginTransaction();  
    FragmentA fragment=(FragmentA)  
getSupportFragmentManager().findFragmentByTag("fragA");  
    transaction.remove(fragment);  
    transaction.commit();  
}
```

# Back and popA

```
public void back(View v){
    getSupportFragmentManager().popBackStack();
}
@Override
public void onBackPressed() {
    if(getSupportFragmentManager().getBackStackEntryCount()>0)
        getSupportFragmentManager().popBackStack();
    else
        super.onBackPressed();
}
public void popA(View v){
    getSupportFragmentManager().popBackStack("fa",0);
}
```

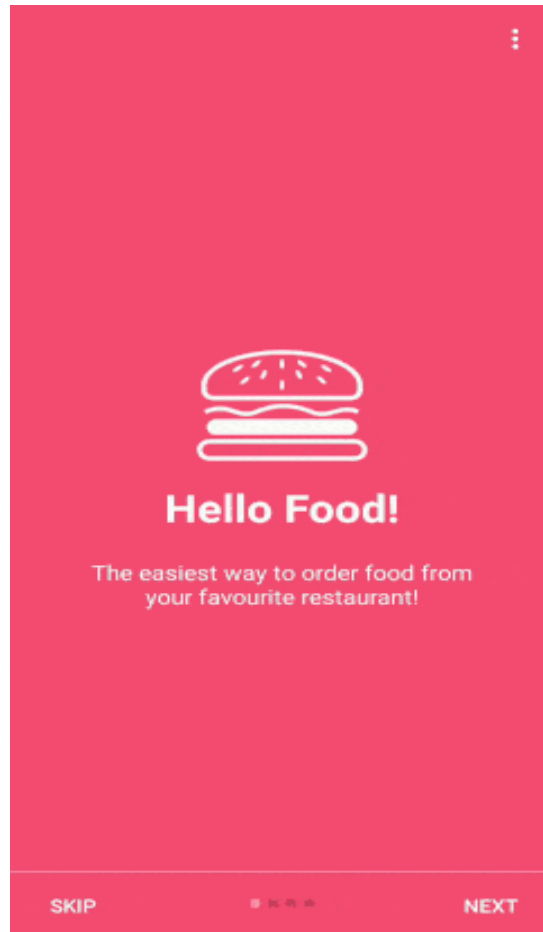
# ViewPager

- ViewPager là một trình quản lý Layout cho phép người dùng vuốt tay qua trái hay qua bên phải để chuyển sang nội dung của trang khác một cách đồng thời. Điểm khác nhau của ViewPager và cách sử dụng Gesture đó là, gesture sẽ chỉ thực hiện khi người dùng thực hiện xong thao tác vuốt tay, trong khi ViewPager thực hiện chuyển màn hình đồng thời với cử chỉ tay người dùng
- ViewPager có thể trượt chuyển đổi giữa các giao diện một cách nhẹ nhàng và khá mượt, thay vì chuyển đổi màn hình qua một sự kiện chớp đèn như trên tivi. Màn hình hiển thị trước nó hoặc sau nó sẽ được hiển thị ra ngay tức thì liên với nó.
- ViewPager hỗ trợ từ Android API 13 trở lên

# Adapter

- ViewPager cung cấp cho chúng ta các Adapter để dễ dàng trong việc quản lý các Fragment. Có 2 lớp là **FragmentPagerAdapter** và **FragmentStatePagerAdapter** là 2 Adapter thường được sử dụng để quản lý các Fragment của ViewPager
- **FragmentPagerAdapter**: Adapter này thích hợp với ViewPager có số lượng ít các Fragment, vì trong trường hợp này các Fragment không còn hiển thị sẽ được lưu trữ lại trong bộ nhớ của hệ thống và chỉ hủy đi các View. Lúc quay trở lại Fragment cũ thì nó chỉ khởi tạo lại View cho Fragment đó trong hàm `onCreateView()` chứ không khởi tạo lại Fragment đó.
- **FragmentStatePagerAdapter**: Adapter này thích hợp với ViewPager có số lượng nhiều các Fragment, vì trong trường hợp này các Fragment không dùng đến sẽ bị hủy khỏi hệ thống và chỉ lưu trữ lại trạng thái của nó. Khi quay trở lại nó sẽ khởi tạo lại cả Fragment đó, việc cần làm trong này chính là khôi phục lại trạng thái của Fragment bằng hàm **`onSaveInstanceState()`**.

## Ví dụ (03)



Thêm icon vào file drawable :

ic\_food.png

ic\_movie.png

ic\_travel.png

Thêm màu vào file colors.xml :

```
<color name="colorPink">#f64c73</color>
```

```
<color name="colorBlue">#3395ff</color>
```

# TabLayout

```
implementation 'com.android.support.design:29.1.0'
```

```
<com.google.android.material.tabs.TabLayout  
    android:id="@+id/tabLayout"  
    app:tabMode="scrollable"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintRight_toRightOf="parent"/>
```

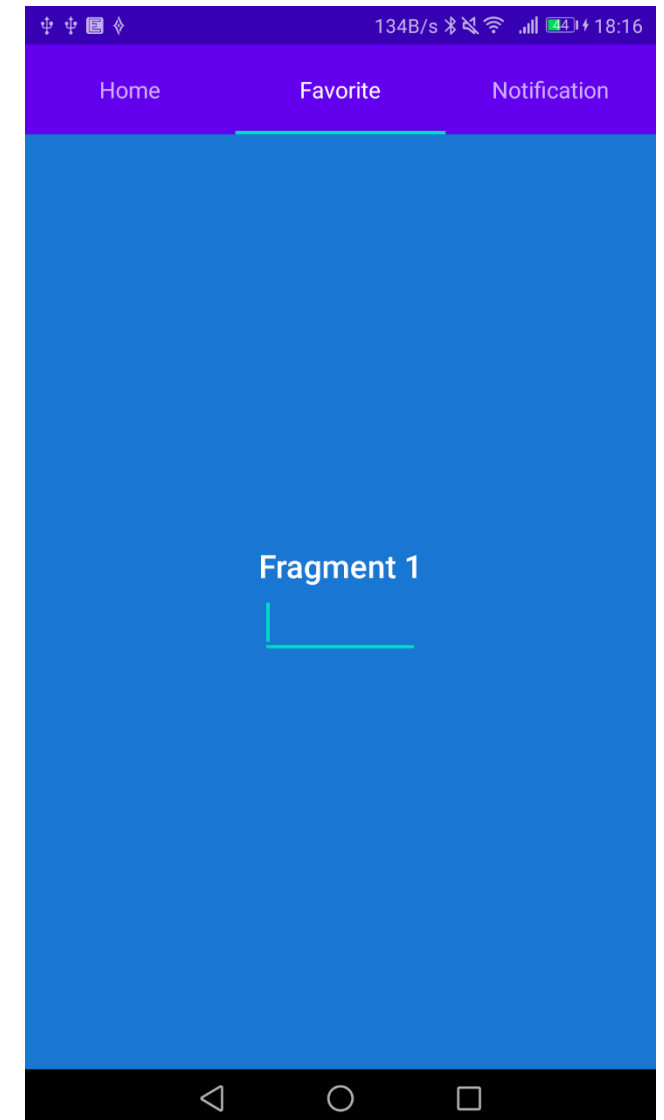
# TabLayout với ViewPager

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".DemoViewPagerActivity">

    <com.google.android.material.tabs.TabLayout
        android:id="@+id/tab_layout"
        android:layout_width="match_parent"
        android:layout_height="?actionBarSize"
        app:tabBackground="@color/colorPrimary"
        app:tabIconTint="#fff"
        app:tabInlineLabel="true"
        app:tabTextAppearance="@style/TextAppearance.AppCompat.Body1"
        app:tabSelectedTextColor="#fff"
        app:tabTextColor="#B2FFFFFF" />

    <androidx.viewpager.widget.ViewPager
        android:id="@+id/view_pager"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

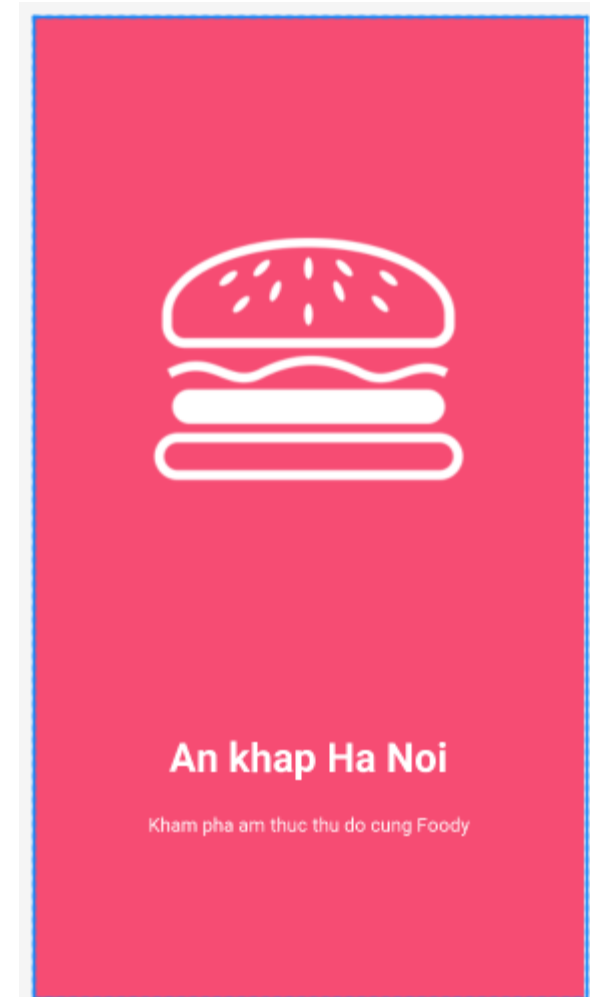
</LinearLayout>
```





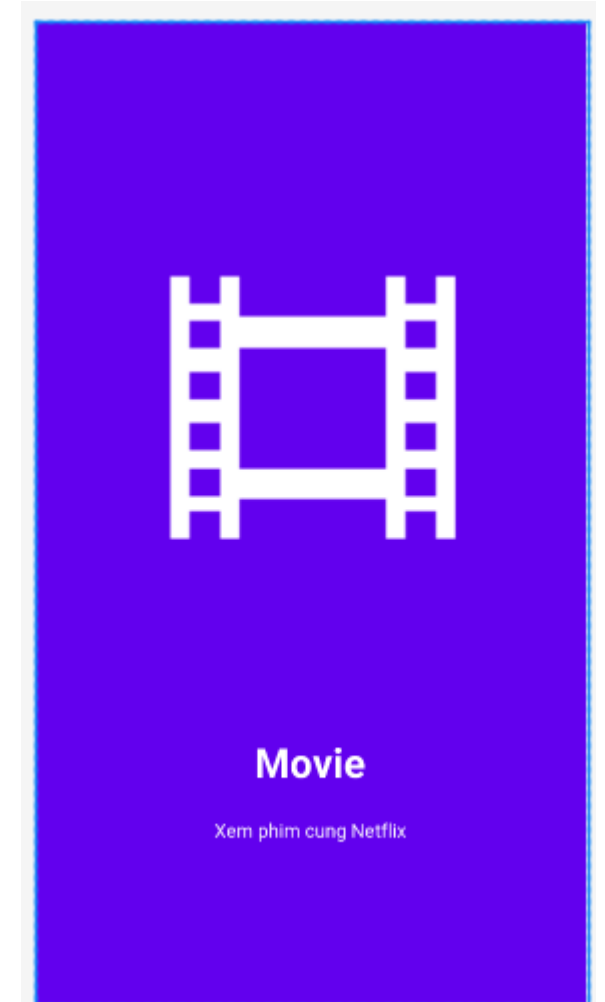
```
android:background="@color/colorPink"
<ImageView
    android:id="@+id/img1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_food"/>
<TextView
    android:id="@+id/text1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="An khap Ha Noi"
    android:textColor="#ffffff"
    android:textSize="30sp"
    android:textStyle="bold"/>
<TextView
    android:id="@+id/description1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Kham pha am thuc cung Foody"
    android:textColor="#ffffff"/>
```

# fragment\_food.xml



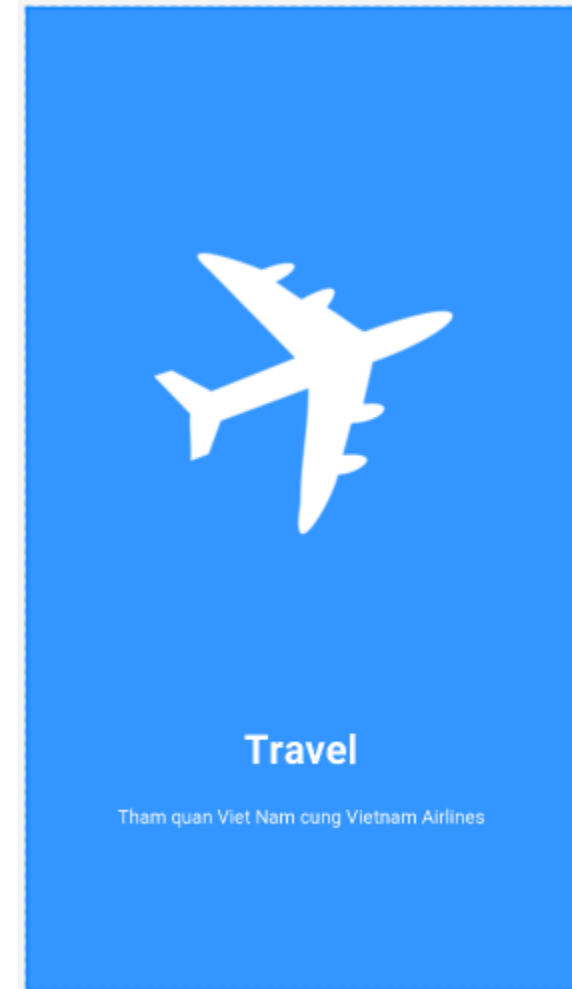
# fragment\_movie.xml

```
android:background="@color/colorPrimary">
<ImageView
    android:id="@+id/img3"
    android:src="@drawable/ic_movie"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
<TextView
    android:id="@+id/text3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Movie"
    android:textColor="#ffffff"
    android:textSize="30sp"
    android:textStyle="bold"/>
<TextView
    android:id="@+id/description3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Xem phim cung Netflix"
    android:textColor="#ffffff"/>
```



# fragment\_travel.xml

```
android:background="@color/colorBlue">
<ImageView
    android:id="@+id/img2"
    android:src="@drawable/ic_travel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
<TextView
    android:id="@+id/text2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Travel"
    android:textColor="#ffffff"
    android:textSize="30sp"
    android:textStyle="bold"/>
<TextView
    android:id="@+id/description2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tham quan cung Vietnam Airlines"
    android:textColor="#ffffff"/>
```



# FragmentFood.java

```
public class FragmentFood extends Fragment {  
    @Nullable  
    @Override  
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,  
@Nullable Bundle savedInstanceState) {  
        View v=inflater.inflate(R.layout.fragment_food,container,false);  
        return v;  
    }  
}
```

# FragmentMovie.java

```
public class FragmentMovie extends Fragment {  
    @Nullable  
    @Override  
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable  
Bundle savedInstanceState) {  
        View v=inflater.inflate(R.layout.fragment_movie,container,false);  
        return v;  
    }  
}
```

# FragmentTravel.java

```
public class FragmentTravel extends Fragment {  
    @Nullable  
    @Override  
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable  
Bundle savedInstanceState) {  
        View v=inflater.inflate(R.layout.fragment_travel,container,false);  
        return v;  
    }  
}
```

# ViewPager Adapter

- Tạo lớp adapter cho viewpager

```
public class DemoViewPagerAdapter extends FragmentStatePagerAdapter {  
  
    public DemoViewPagerAdapter(@NonNull FragmentManager fm) {  
        super(fm);  
    }  
  
    @NonNull  
    @Override  
    public Fragment getItem(int position) {  
        // trả về fragment tùy vào position  
    }  
  
    @Override  
    public int getCount() {  
        return 0; // trả về số lượng fragment  
    }  
}
```

- Set adapter cho Viewpager

```
viewPager.setAdapter(new DemoViewPagerAdapter(getSupportFragmentManager()));
```

# Khai báo TabLayout sử dụng ViewPager

- Thêm hàm getPageTitle() vào trong class Adapter của ViewPager

```
@Nullable
@Override
public CharSequence getPageTitle(int position) {
    return "";
}
```

- Trong file Activity

```
ViewPager viewPager = findViewById(R.id.view_pager);
TabLayout tabLayout = findViewById(R.id.tab_layout);

viewPager.setAdapter(new DemoViewPagerAdapter(getSupportFragmentManager()));
tabLayout.setupWithViewPager(viewPager);
```



# MainActivity.java

```
public class MainActivity extends AppCompatActivity
    implements View.OnClickListener{
    private ViewPager viewPager;
    private TabLayout tabLayout;
    private Button mBtnPre, mBtnNext;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mBtnPre = (Button) findViewById(R.id.btnPre);
        mBtnNext = (Button) findViewById(R.id.btnNext);

        mBtnPre.setOnClickListener(this);
        mBtnNext.setOnClickListener(this);

        tabLayout = (TabLayout) findViewById(R.id.tabLayout);
        viewPager = (ViewPager) findViewById(R.id.viewPager);
```

```
public void onBackPressed() {  
    if(viewPager.getCurrentItem() == 0) {  
        super.onBackPressed();  
    } else {  
        viewPager.setCurrentItem(viewPager.getCurrentItem() - 1);  
    }  
}
```

```
public void setTabLayoutTitleColor() {  
    switch (viewPager.getCurrentItem()) {  
        case 0:  
            tabLayout.setTabTextColors(Color.BLACK,  
getResources().getColor(R.color.colorPink));  
            break;  
        case 1:  
            tabLayout.setTabTextColors(Color.BLACK,  
getResources().getColor(R.color.colorPrimary));  
            break;  
        case 2:  
            tabLayout.setTabTextColors(Color.BLACK,  
getResources().getColor(R.color.colorBlue));  
            break;  
        default:  
    }  
}
```

```
@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.btnPre:
            if(viewPager.getCurrentItem() == 0) {
                return;
            } else {
                viewPager.setCurrentItem(viewPager.getCurrentItem() - 1);
                setTabLayoutTitleColor();
            }
            break;
        case R.id.btnNext:
            if(viewPager.getCurrentItem() == 2) {
                return;
            } else {
                viewPager.setCurrentItem(viewPager.getCurrentItem() + 1);
                setTabLayoutTitleColor();
            }
    }
}
```

```

public class HorizontalFlipTransformation implements
    ViewPager.PageTransformer {
    @Override
    public void transformPage(@NonNull View page, float position) {
        page.setTranslationX(-position * page.getWidth());
        page.setCameraDistance(12000);
        if (position < 0.5 && position > -0.5) {
            page.setVisibility(View.VISIBLE);
        } else {
            page.setVisibility(View.INVISIBLE);
        }
        if (position < -1) { // [-Infinity, -1)
            page.setAlpha(0);
        } else if (position <= 0) { // [-1, 0]
            page.setAlpha(1);
            page.setRotationY(180 * (1 - Math.abs(position) + 1));
        } else if (position <= 1) { // (0, 1]
            page.setAlpha(1);
            page.setRotationY(-180 * (1 - Math.abs(position) + 1));
        } else {
            page.setAlpha(0);
        }
    }
}

```

# Fragment với ViewPager

- Lắng nghe sự kiện khi thay đổi trang

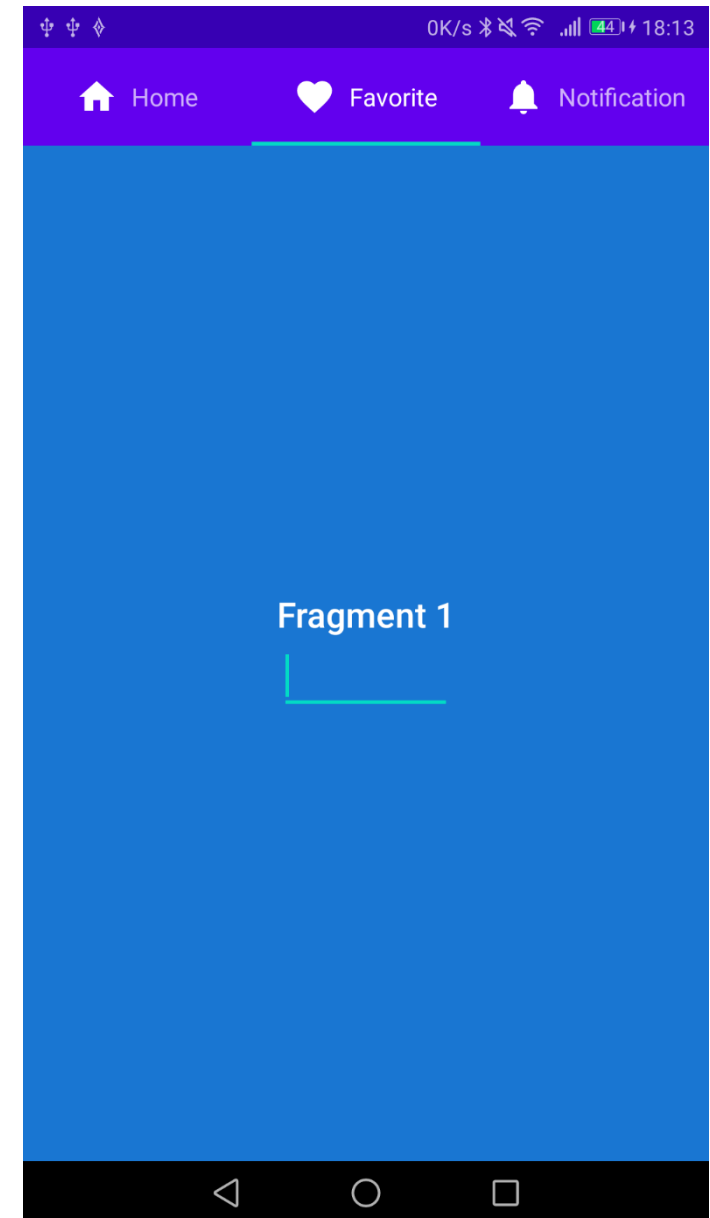
```
viewPager.addOnPageChangeListener(new ViewPager.OnPageChangeListener() {  
    @Override  
    public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {  
  
    }  
  
    @Override  
    public void onPageSelected(int position) {  
  
    }  
  
    @Override  
    public void onPageScrollStateChanged(int state) {  
  
    }  
});
```

```
FragmentManager fragmentManager = getSupportFragmentManager();
FragmentAdapter fragmentAdapter = new FragmentAdapter(fragmentManager, 3);
viewPager.setPageTransformer(true, new HorizontalFlipTransformation());
viewPager.setAdapter(fragmentAdapter);

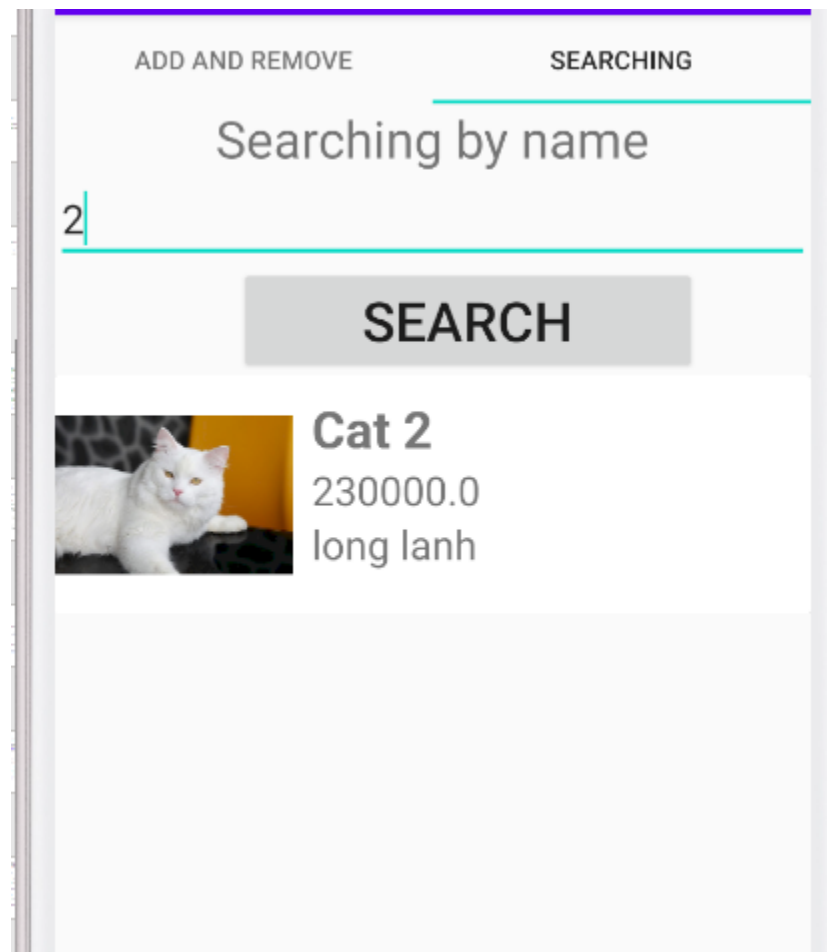
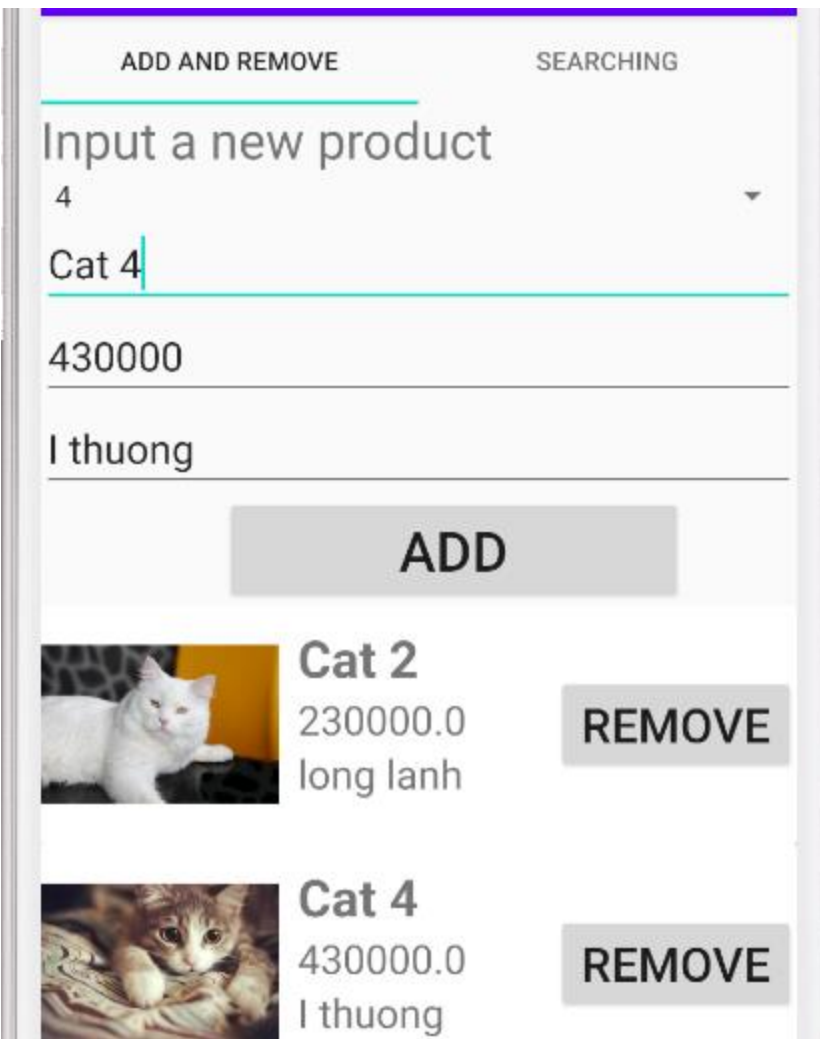
tabLayout.setupWithViewPager(viewPager);
setTabLayoutTitleColor();
tabLayout.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {
    @Override
    public void onTabSelected(TabLayout.Tab tab) {
        switch (tab.getPosition()) {
            case 0:
                tabLayout.setTabTextColors(Color.BLACK, getResources().getColor(R.color.colorPink));
                break;
            case 1:
                tabLayout.setTabTextColors(Color.BLACK,
getResources().getColor(R.color.colorPrimary));
                break;
            case 2:
                tabLayout.setTabTextColors(Color.BLACK, getResources().getColor(R.color.colorBlue));
                break;
            default:
        }
    }
}
```

# Set icon cho TabLayout

```
tabLayout.getTabAt(0).setIcon(R.drawable.ic_home);  
tabLayout.getTabAt(1).setIcon(R.drawable.ic_favorite);  
tabLayout.getTabAt(2).setIcon(R.drawable.ic_notification);
```



# Ví dụ kết hợp Fragment, TabLayout, ViewPager và RecyclerView (04)





## Bài tập về nhà

- Làm bài như ví dụ 3 (liên quan đến đồ án của mình)

- End of Lesson 6



Thank you!