Lập trình trên thiết bị di động

THIẾT KẾ GIAO DIỆN (PHẦN 2)

GV: Nguyễn Huy Cường
Email: nh.cuong@hutech.edu.vn

Nội dung

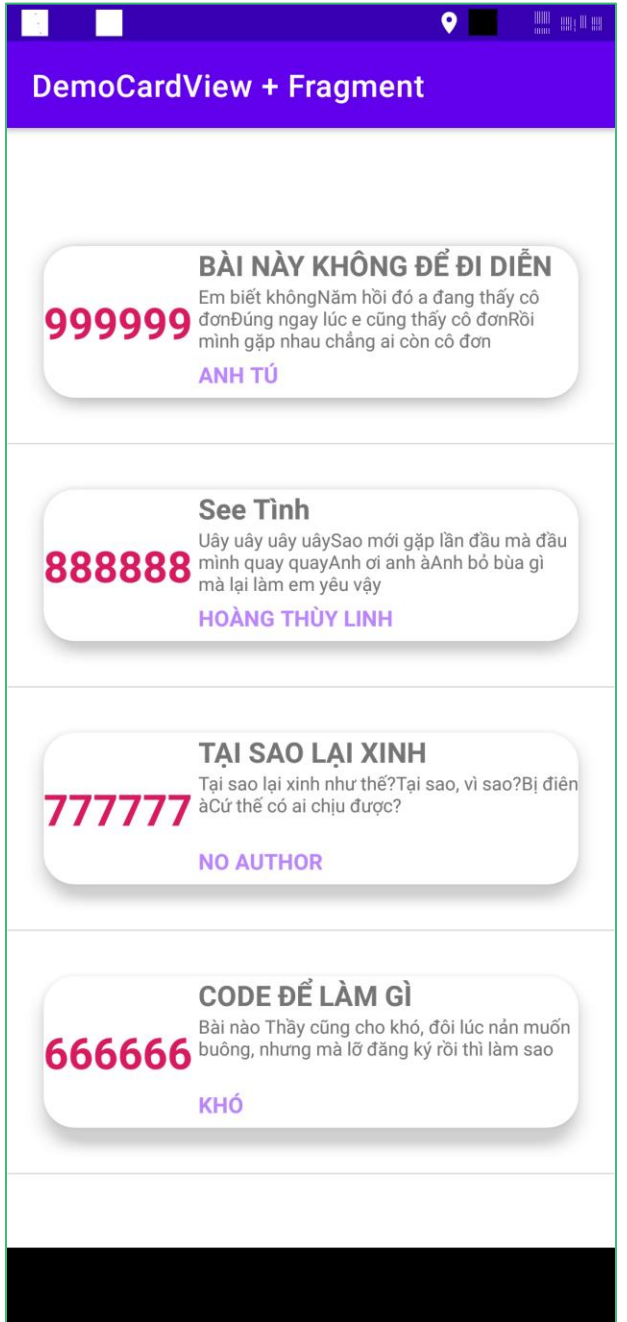
- CardView
- **Fragment** – Vòng đời của Fragment
- ViewPager, **ViewPager2**
- TabLayout
- NavigationView, NavigationDrawer, **BottomNavigationView**
- AlertDialog, Custom AlertDialog …

1. CardView

- CardView: giống như **FrameLayout**, có hỗ trợ bo tròn bốn góc và hiệu ứng đổ bóng.

CardView được sử dụng làm **container** cho các dòng trong **ListView, RecyclerView**.

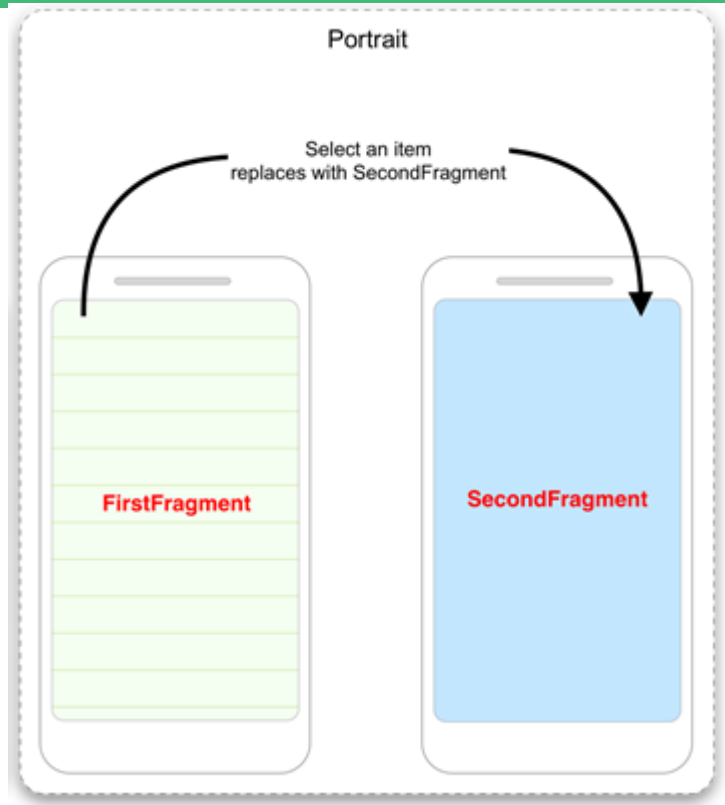
- Cách tạo:
- 1 số thuộc tính thường dùng với CardView
  - card_view:cardCornerRadius
  - card_view:cardBackgroundColor
  - card_view:cardElevation

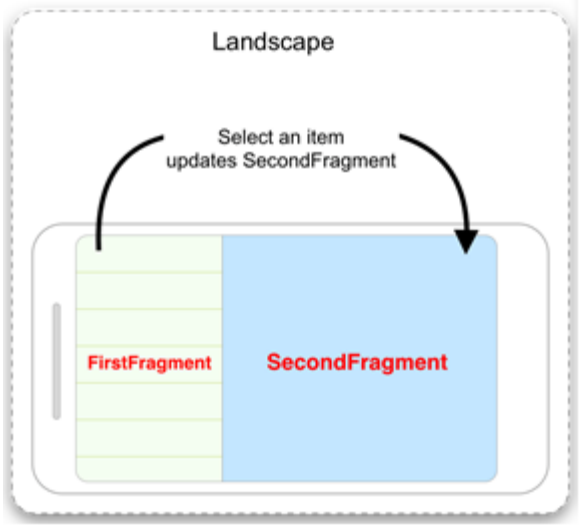


2. Fragment

- **Fragment** là một phần giao diện được nhúng vào một Activity, fragment không chạy độc lập với activity được.
  - ❑ Một fragment có thể chiếm một phần hoặc toàn bộ giao diện Activity. Một Activity có thể có nhiều fragment.
  - ❑ Các fragment là độc lập chứa: View, events, logic.
  - ❑ Fragment có thể cố định hoặc có thể được tạo, thêm, xóa,… trong quá trình Activity hoạt động.
- Cách tạo: New/ Fragment

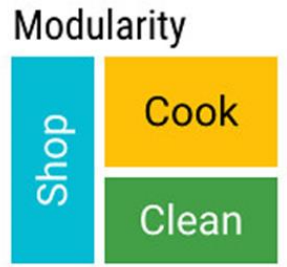
```
<fragment
 android:id="@+id/fragments“
 android:layout_width="match_parent"
 android:layout_height="match_parent" />
```





Ưu điểm của Fragment

- **Module hóa (Modularity):** Với các Activity phức tạp thì code sẽ implement ở các Fragment. Mỗi Fragment là một module độc lập. Điều này sẽ làm cho code dễ tổ chức và bảo trì tốt hơn.

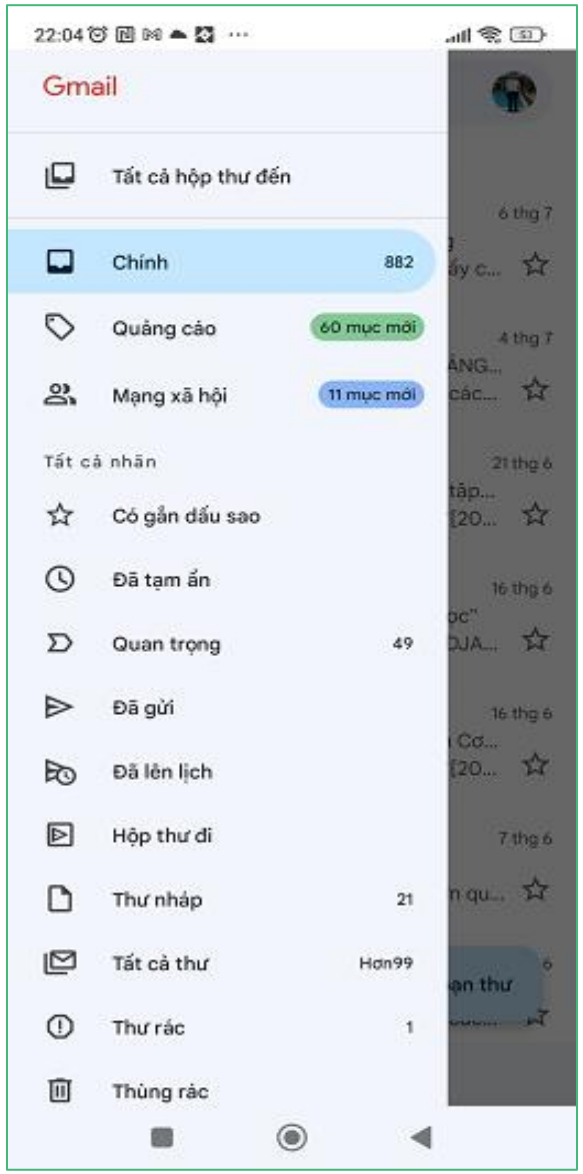


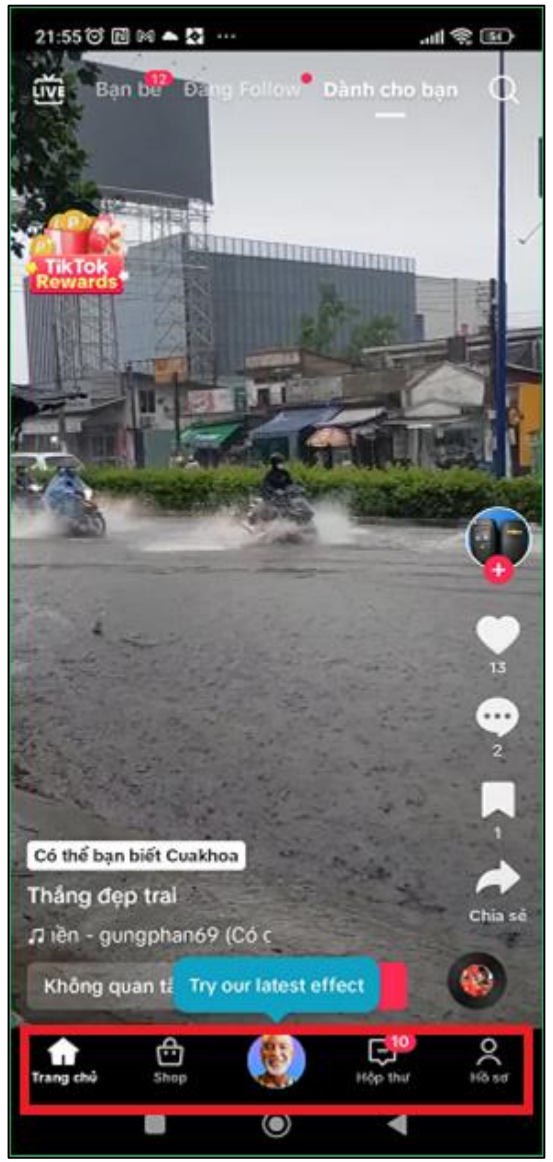
- **Tái sử dụng (Reusability):** Viết fragment để có thể chia sẻ chúng với các Activity khác.

- **Hỗ trợ đa màn hình (Adaptability):** Fragment cung cấp cách để trình bày giao diện người dùng (UI) phù hợp và tối ưu cho các loại thiết bị Android có kích thước màn hình và mật độ điểm ảnh khác nhau.

Fragment

Ứng dụng của fragment

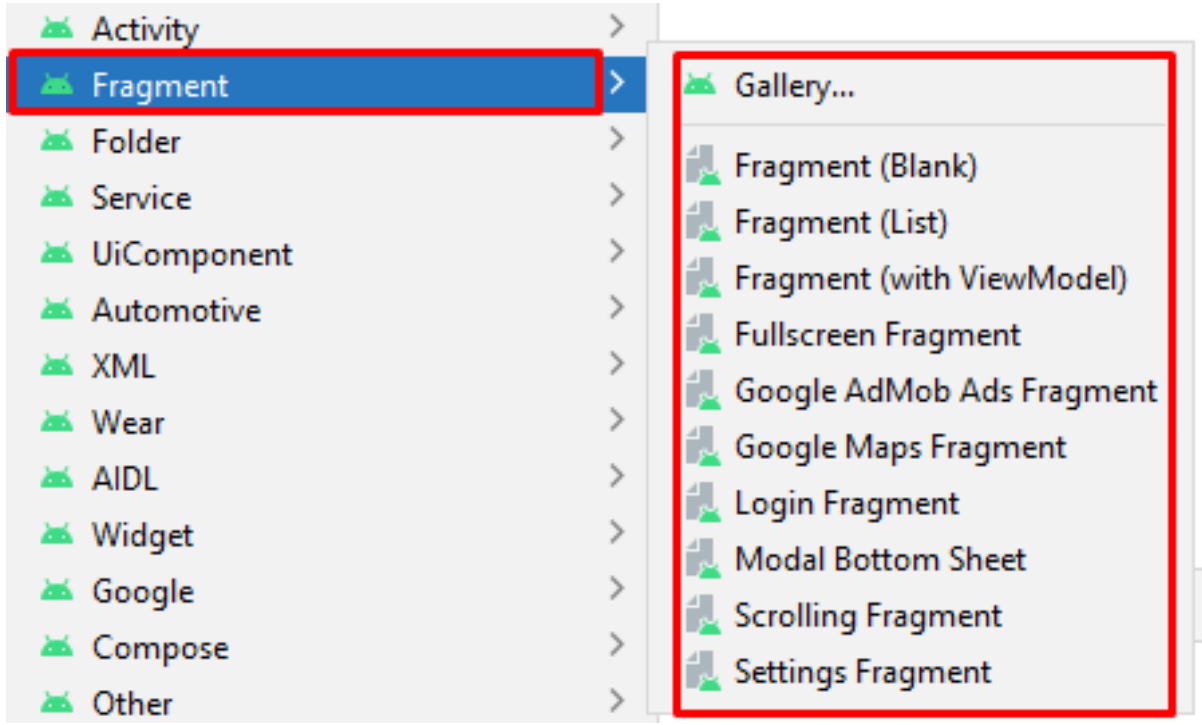




Fragment

1 số template fragment từ Android API

- Blank
- List: Hiển thị UI dạng ListView
- Fragment with ViewModel
- Fullscreen Fragment
- Login Fragment
- Settings Fragment
- …



Fragment

Vòng đời của Fragment

- Tương tự vòng đời Activity

❑ onCreateView():

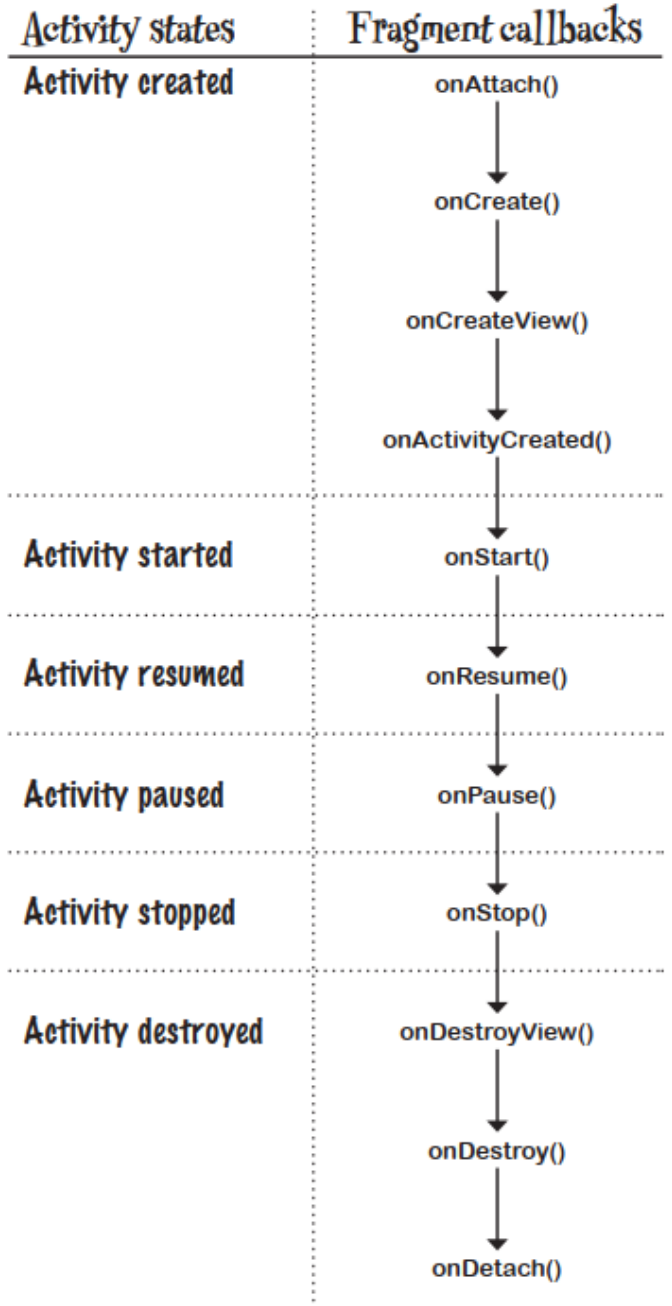
❑ onStart():

- Muốn lấy view:

View view = **getView**()

- Muốn ánh xạ view:

view.findViewById()



onAttach(Context)
This happens when the fragment is associated with a context, in this case an activity.

onCreate(Bundle)
This is very similar to the activity's `onCreate()` method. It can be used to do the initial setup of the fragment.

onCreateView(LayoutInflater, ViewGroup, Bundle)
Fragments use a layout inflater to create their view at this stage.

onActivityCreated(Bundle)
Called when the `onCreate()` method of the activity has completed.

onStart()
Called when the fragment is about to become visible.

onResume()
Called when the fragment is visible and actively running.

onPause()
Called when the fragment is no longer interacting with the user.

onStop()
Called when the fragment is no longer visible to the user.

onDestroyView()
Gives the fragment the chance to clear away any resources that were associated with its view.

onDestroy()
In this method, the fragment can clear away any other resources it created.

onDetach()
Called when the fragment finally loses contact with the activity.

Fragment trong Activity

- Static Fragment: được khai báo trực tiếp trong **Activity**
- Dynamics Fragment:
  - Lớp **FragmentManager** cho phép thêm, xóa, thay thế fragment trong layout của activity.
  - Sử dụng **getFragmentManager()** hoặc **getSupportFragmentManager()** để lấy ra một đối tượng **FragmentManager**
  - Việc sửa đổi phải được thực hiện trong một giao dịch thông qua lớp **FragmentTransaction**
  - Fragment cũng có thể truy cập tới Activity chứa nó bằng phương thức **Fragment.getActivity()**.

Sử dụng Fragment

Bước 1: Tạo Fragment: giao diện cho Fragment

Bước 2: Đưa Fragment vào Activity: từ FragmentContainerView

- Đối với static Fragment: chọn Fragment
- Đối với dynamic Fragment: sử dụng **FragmentManager** để quản lý các **Fragment** trong khi đang chạy chương trình, Các bước như sau:

- Dùng FragmentManager.beginTransaction() để lấy 1 instance của Fragment Transaction.
- Dùng FragmentTransaction.**replace**() để thay thế, hoặc ngoài ra có thể dùng 1 số: **add**(), **remove**(), **replace**(), **hide**(), **show**(), **detach**() (api 13), attach() (api 13)..
- Gọi FragmentTransaction.commit() để chấp nhận.

DEMO Static Fragment – Thiết kế fragment

- Bước 1: Tạo mới Fragment

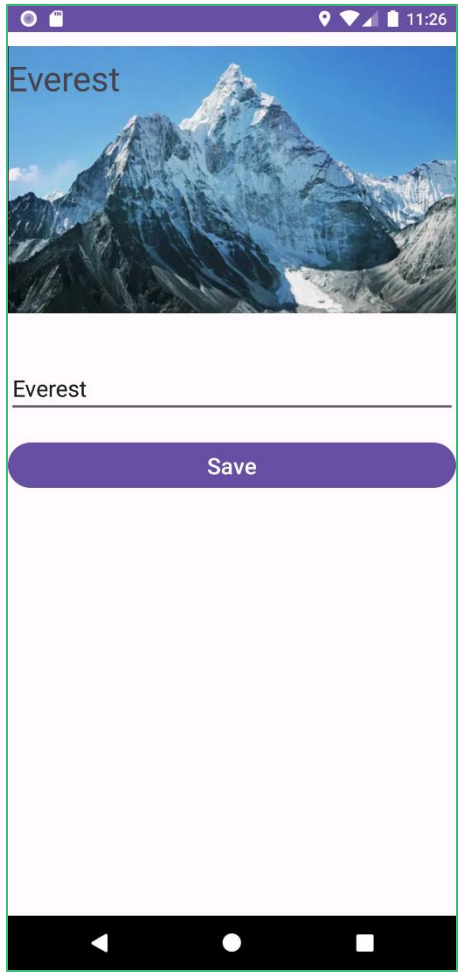
- New/ Fragment/ Frament (blank)

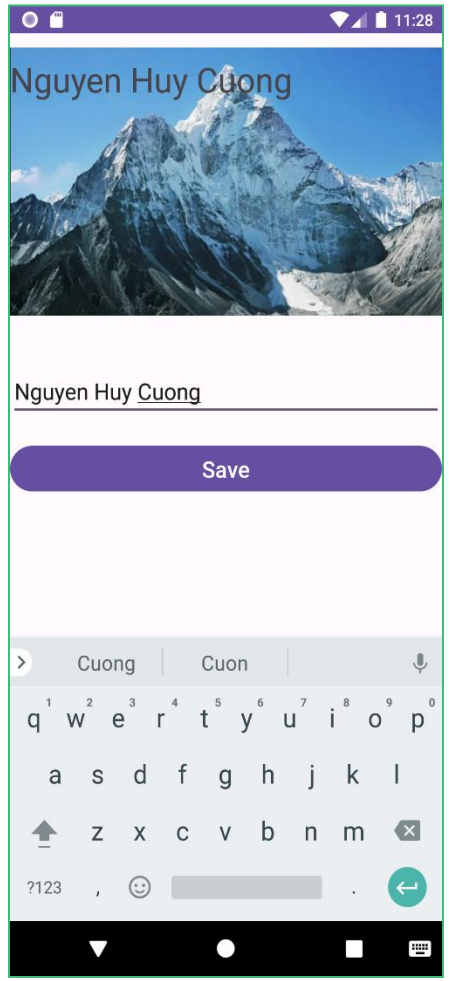
- Đặt tên fragment "fragmentImage"

- Chỉnh sửa layout của fragment

(có Hình từ mipmap/drawable và TextView)

```
<FrameLayout 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=".fragmentImage">
   <ImageView
      android:id="@+id/imageView"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:scaleType="fitCenter"
      android:src="@mipmap/avatar" />
   <TextView
      android:id="@+id/txtImageText"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:paddingTop="20dp"
      android:text="test"
      android:textSize="30sp" />
</FrameLayout>
```



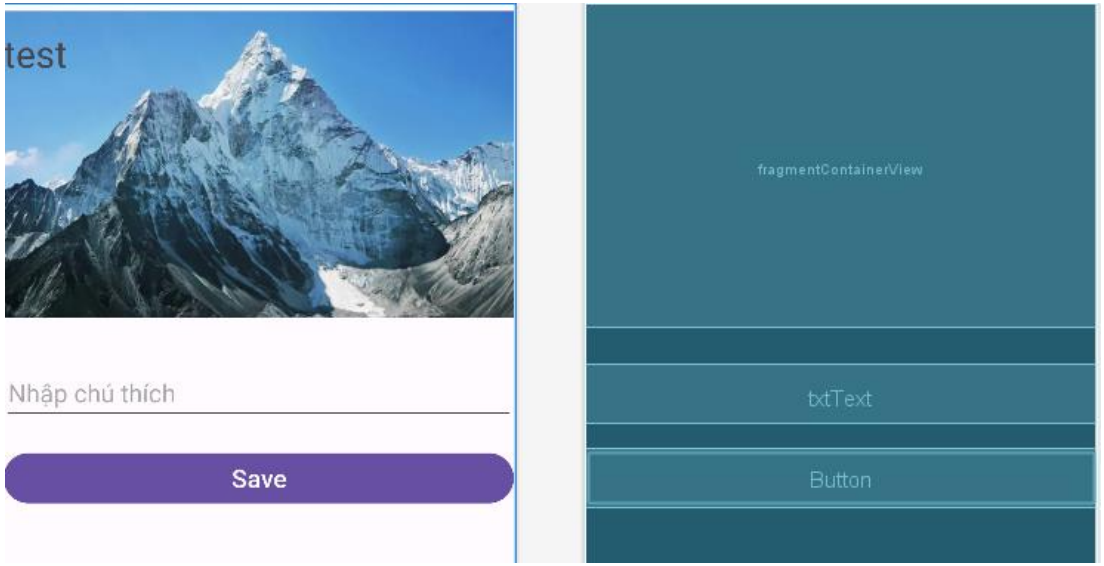


Static Fragment – đưa fragment vào Activity

- Bước 2: Đưa Fragment vào Activity

- Sử dụng fragmentContainerView và chọn framgment tương ứng "fragmentImage"

- Thiết kế layout tương tự như dưới




```
<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">

   <androidx.fragment.app.FragmentContainerView
      android:id="@+id/fragmentContainerView"
      android:name="com.example.demostaticfragment.fragmentImage"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      tools:layout="@layout/fragment_image" />

   <EditText
      android:id="@+id/txtText"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginTop="30dp"
      android:hint="Nhập chú thích"
      android:textSize="20sp"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toBottomOf="@+id/fragmentContainerView" />

   <Button
      android:id="@+id/btnSave"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginTop="20dp"
      android:text="Save"
      android:textSize="20sp"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.0"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toBottomOf="@+id/txtText" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Tương tác giữa fragment và Activity

- Bước 3: Để OnClick ở Save, lấy giá trị của "txtText" vào fragment "txtImageText" ?

Ở Fragment:

- Ánh xạ view để lấy txtImageText

(sau **onCreateView** hoặc **onViewCreated**)

- Viết hàm public để set giá trị vào txtImageText

Ở Activity: Viết **onclick** của Save

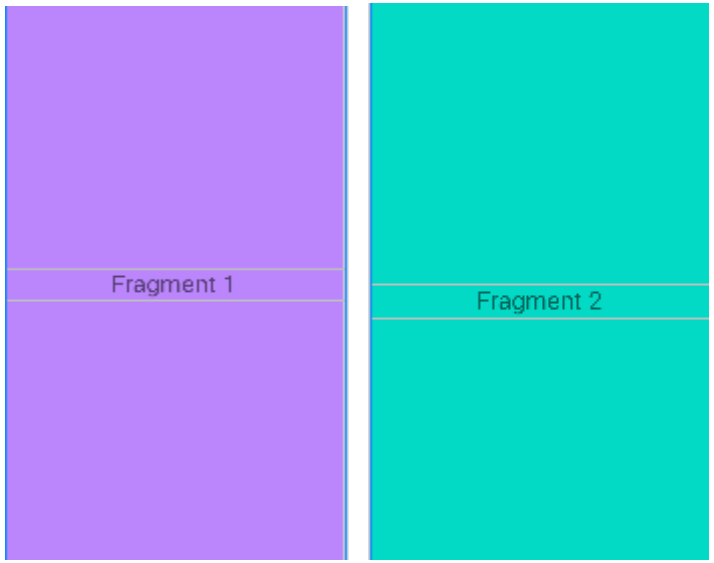
- Tìm fragment tương ứng từ *fragmentContainerView*
- Gọi hàm set giá trị đã viết ở fragment

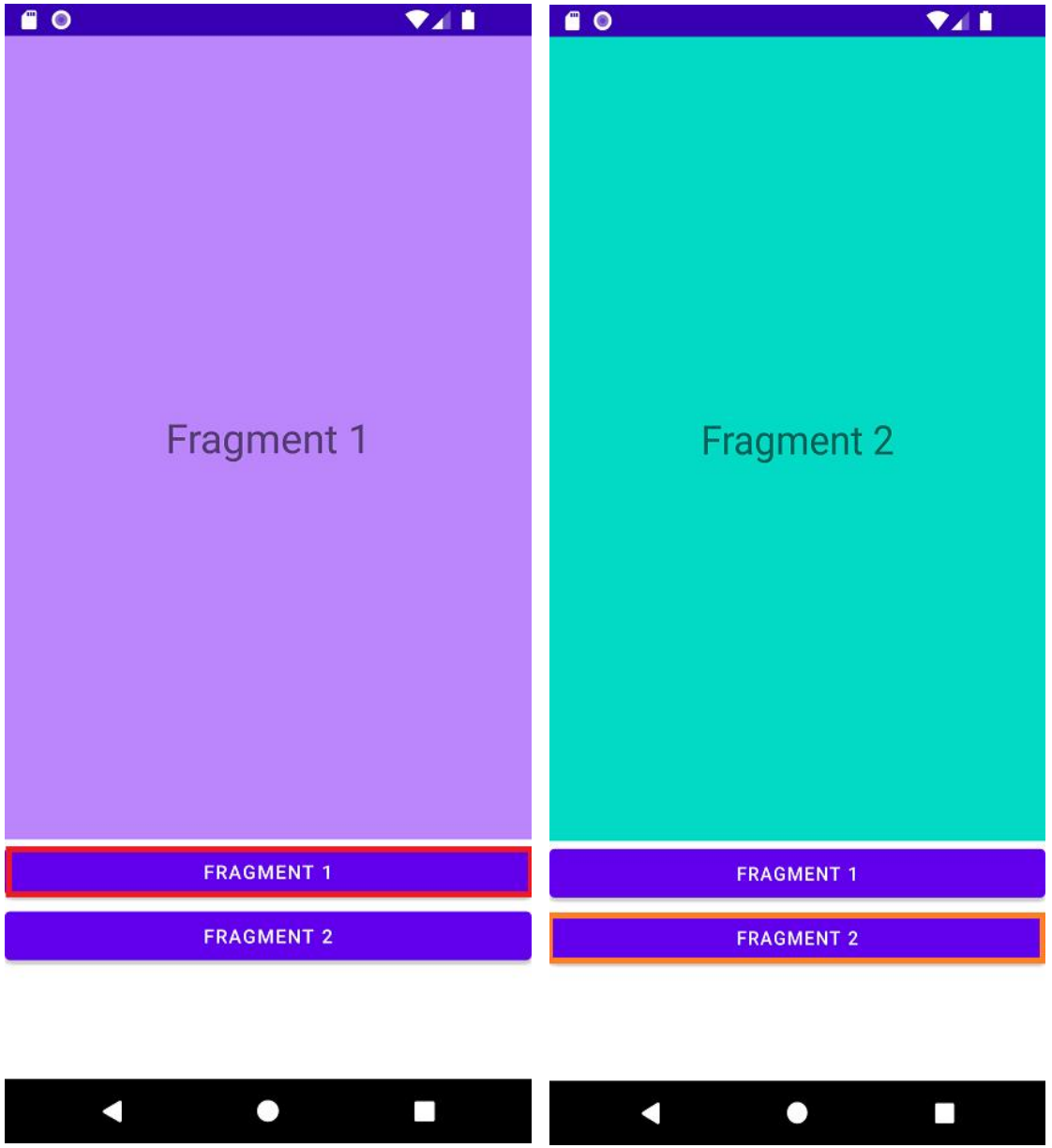
```
4 usages
public class fragmentImage extends Fragment {
    2 usages
    TextView txtImageText;
    public fragmentImage() {
        // Required empty public constructor
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view =  inflater.inflate(R.layout.fragment_image, container,  attachToRoot: false);
        txtImageText = view.findViewById(R.id.txtImageText);
        return view;
    }
    1 usage
    public void SetImageText(String textImage)
    {
        txtImageText.setText(textImage);
    }
}
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    EditText txtText = findViewById(R.id.txtText);
    findViewById(R.id.btnSave).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            fragmentImage myFragment = (fragmentImage)getSupportFragmentManager()
                                        .findFragmentById(R.id.fragmentContainerView);
            if (myFragment != null) {
                myFragment.SetImageText(txtText.getText().toString());
            }
        }
    });
}
```

DEMO dynamic Fragment – Thiết kế fragment

- Bước 1: Tạo mới Fragment1 và Fragment2



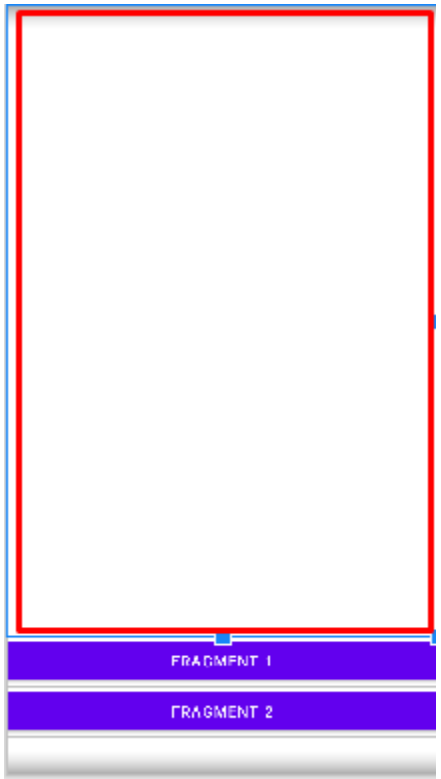


DEMO dynamic Fragment -_Đưa Fragment vào Activity

- Bước 2: Đưa Fragment vào Activity

- Sử dụng fragmentContainerView

- Thiết kế



- Thực hiện code khi click "fragment 1" và "fragment 2"

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        findViewById(R.id.btnFragment1).setOnClickListener(this);
        findViewById(R.id.btnFragment2).setOnClickListener(this);
    }
    @Override
    public void onClick(View view) {
        Fragment fragment = null;
        switch (view.getId()) {
            case R.id.btnFragment1:
                fragment = new fragment1();
                break;
            case R.id.btnFragment2:
                fragment = new fragment2();
                break;
            default:
                fragment = new fragment1();
                break;
        }
        FragmentManager fragmentManager=getSupportFragmentManager();
        FragmentTransaction fragmentTransaction=fragmentManager.beginTransaction();
        fragmentTransaction.replace(R.id.fragmentContainerView,fragment);
        fragmentTransaction.commit();
    } }
```

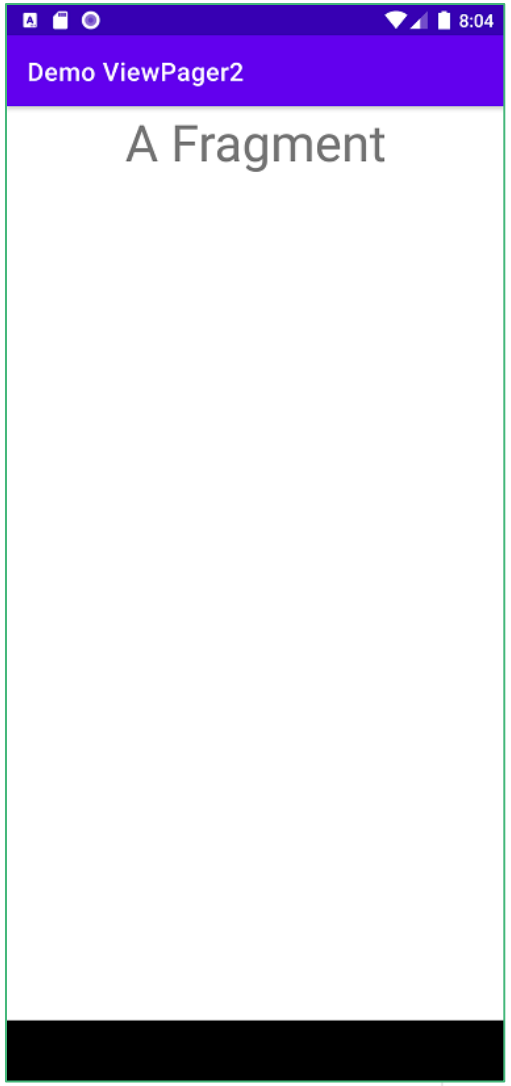
3. ViewPager 2 (ViewPager)

- ViewPager2/ ViewPager là một phần giao diện cho phép người dùng vuốt (**swipe**) để xem một trang (màn hình) hoàn toàn mới.
  - **ViewPager2** được xây dựng trên **RecyclerView**.
  - Khi làm việc với một tập hợp các **Fragment**. Nếu một trong các Fragment thay đổi giao diện của nó, chỉ cần gọi phương thức **notifyDatasetChanged**() để cập nhập giao diện.
- Khai báo trong Activity

```
<androidx.viewpager2.widget.ViewPager2
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
```

- Sử dụng:
  - Xây dựng **Adapter** extends **FragmentStateAdapter**

DEMO 1. ViewPager2 với static Fragment



1. Tạo 3 Fragment A,B,C
2. Thêm Viewpager2 có id = "viewPagerFragment" hiển thị ở MainActivity
3. Tạo Pager2Adapter kế thừa **FragmentStateAdapter**

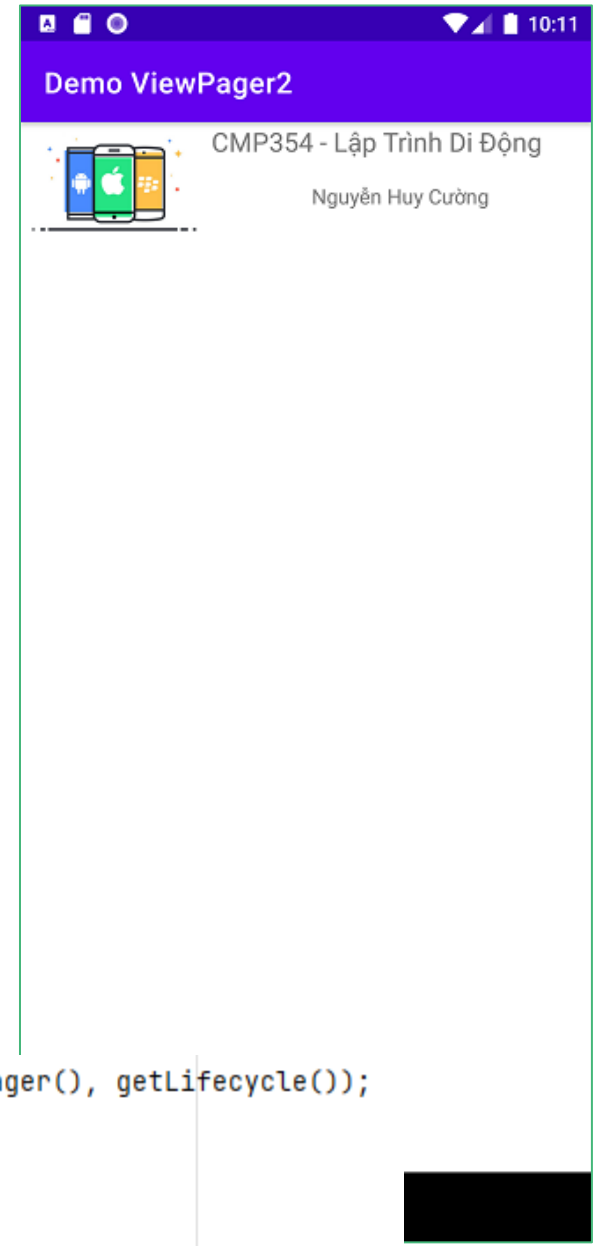
```
public class Pager2Adapter extends FragmentStateAdapter {

    private List<Fragment> fragmentList;
    public Pager2Adapter(@NonNull FragmentManager fragmentManager, @NonNull Lifecycle lifecycle, List<Fragment> _fragmentList) {

        super(fragmentManager, lifecycle);
        fragmentList = fragmentList;
    }

    @NonNull
    @Override
    public Fragment createFragment(int position) {
        return fragmentList.get(position);
    }

    @Override
    public int getItemCount() { return fragmentList.size(); }
}
```

Tạo danh sách Fragment tương ứng

4. Sử dụng ở MainActivity

```
List<Fragment> listFrag = new ArrayList<>();
listFrag.add(new A());
listFrag.add(new B());
listFrag.add(new C());

Pager2Adapter myAdapter = new Pager2Adapter(getSupportFragmentManager(), getLifecycle(), listFrag);
ViewPager2 vpg2  = (ViewPager2) findViewById(R.id.viewPagerFragment);
vpg2.setOrientation(ViewPager2.ORIENTATION_HORIZONTAL);
vpg2.setAdapter(myAdapter);
```

DEMO 2. ViewPager2 với dynamic Fragment



1. Thiết kế 1 Fragment tổng quát để chứa thông tin môn học
2. Tạo Pager2DynamicAdapter kế thừa từ **FragmentStateAdapter**

```
public class Pager2DynamicAdapter extends FragmentStateAdapter {

    private List<MonHoc> lstMonHoc;

    public Pager2DynamicAdapter(@NonNull FragmentManager fragmentManager, @NonNull Lifecycle lifecycle) {
        super(fragmentManager, lifecycle);

        lstMonHoc = MonHoc.LayDSMonHoc();

    }

    @NonNull
    @Override
    public Fragment createFragment(int position) {
        MonHocFragment temp = new MonHocFragment(lstMonHoc.get(position));
        return temp;
    }

    @Override
    public int getItemCount() {
        return lstMonHoc.size();
    }
}
```

3. Gọi ở MainActivity

```
Pager2DynamicAdapter myAdapter = new Pager2DynamicAdapter(getSupportFragmentManager(), getLifecycle());
ViewPager2 vpg2  = (ViewPager2) findViewById(R.id.viewPagerFragment);
vpg2.setOrientation(ViewPager2.ORIENTATION_HORIZONTAL);
vpg2.setAdapter(myAdapter);
```

ASM 4.1

1. Sử dụng **ViewPager2** + **VideoView** để vuốt theo hướng lên-xuống các Video. Mỗi video có (URL , Tittle, Description)

Trong đó: URL có thể là:

(a) Từ local (đặt file mp4 ở folder res/**raw)**

mov_bbb.mp4

ElephantsDream.mp4

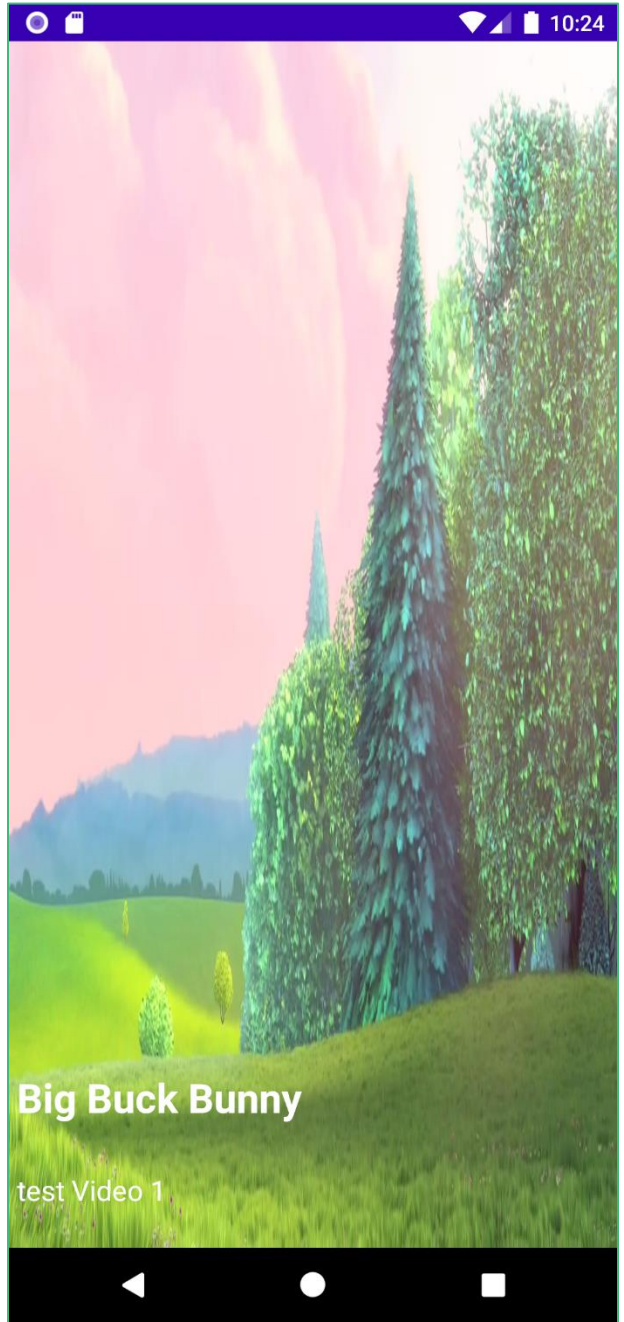
(b) Từ đường dẫn URL:

https://www.w3schools.com/html/mov_bbb.mp4

http://commondatastorage.googleapis.com/gtv-videos-bucket/sample/ElephantsDream.mp4

2. Hiển thị **seeking** của VideoView

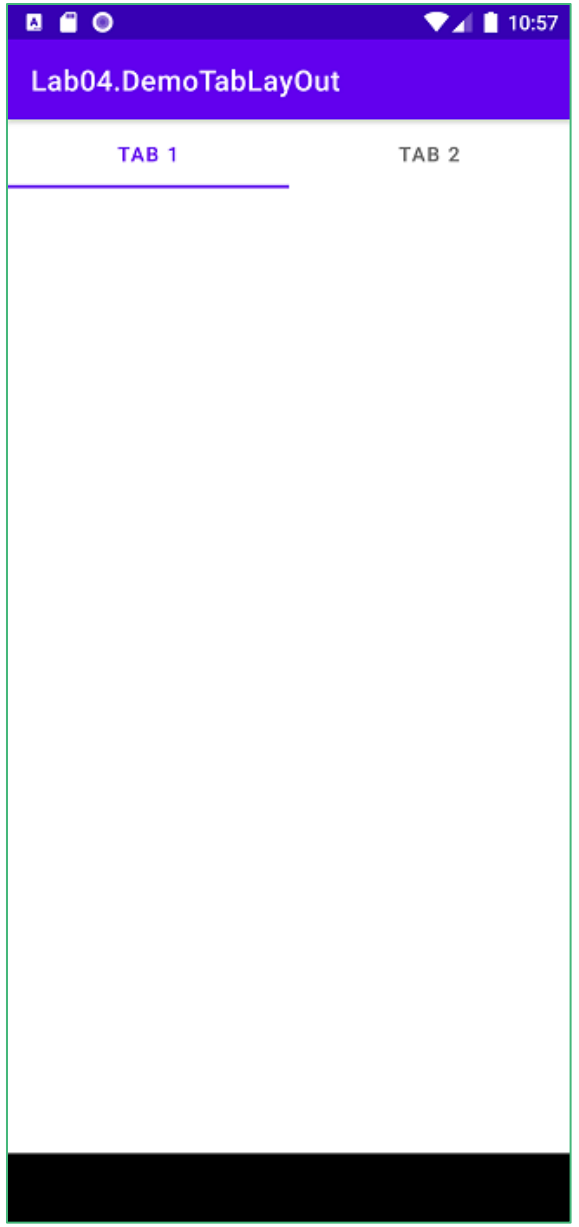
3. Thêm progressBar để loading… khi chưa load được video



4. TabLayout

- TabLayout cung cấp giao diện ngang để hiển thị các tab.
  - ❑ TabLayout thường được kết hợp với fragment và ViewPager.
- Khai báo trong Activity

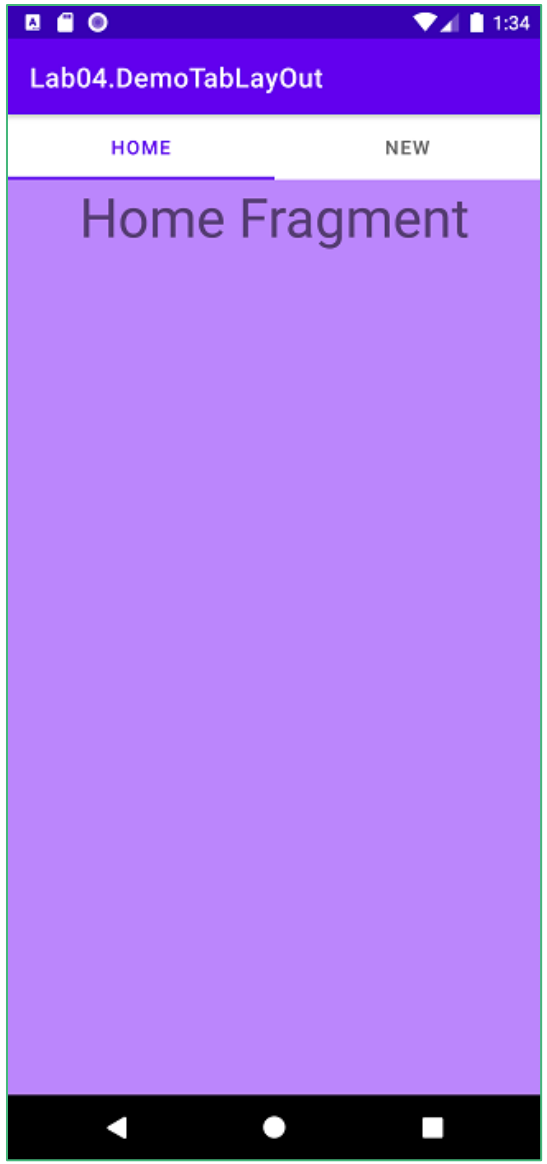
```
<com.google.android.material.tabs.TabLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="paren1
    app:layout_constraintTop_toTopOf="parent">

    <com.google.android.material.tabs.TabItem
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tab 1" />

    <com.google.android.material.tabs.TabItem
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tab 2" />

</com.google.android.material.tabs.TabLayout>
```

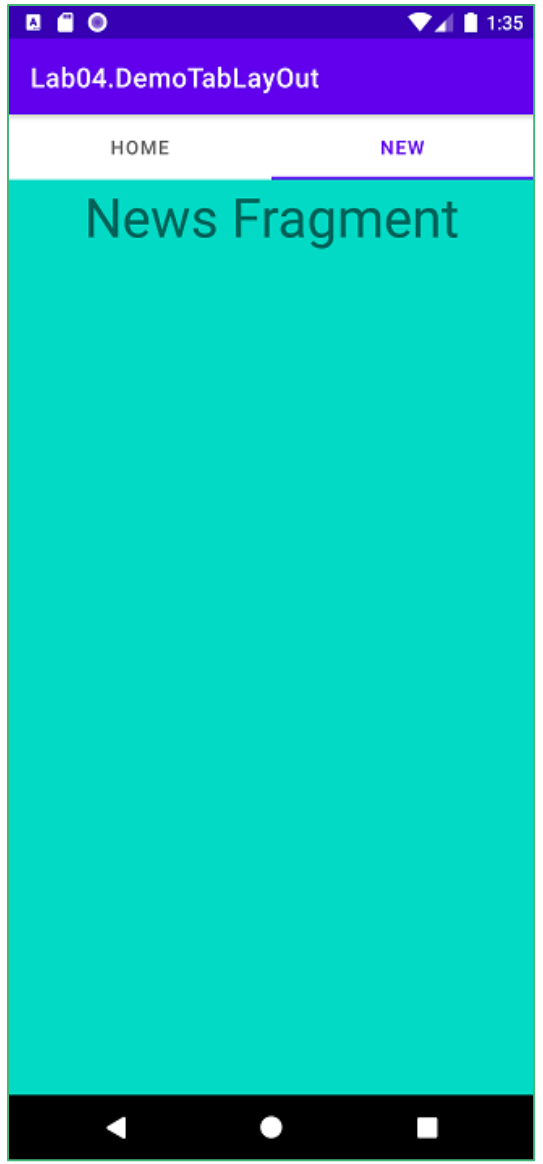


DEMO Tạo TabLayout + Viewpager2 + Fragment

1. Giao diện MainActivity: có Tablayout và viewPager2
2. Tạo 2 Fragement: **HomeFragment** và **NewFragment**
3. Tạo Pager2Adapter extends từ **FragmentStateAdapter**
4. Ở java MainActivity

```
TabLayout mTabLayout = findViewById(R.id.tabLayout);
ViewPager2 mViewPager2 = findViewById(R.id.viewPager2);
Pager2Adapter adapter = new Pager2Adapter(getSupportFragmentManager(), getLifecycle());
mViewPager2.setAdapter(adapter);
new TabLayoutMediator(mTabLayout, mViewPager2, (tab, position) -> {
    if (position == 0)
        tab.setText("Home");
    else
        tab.setText("New");
}).attach();
```





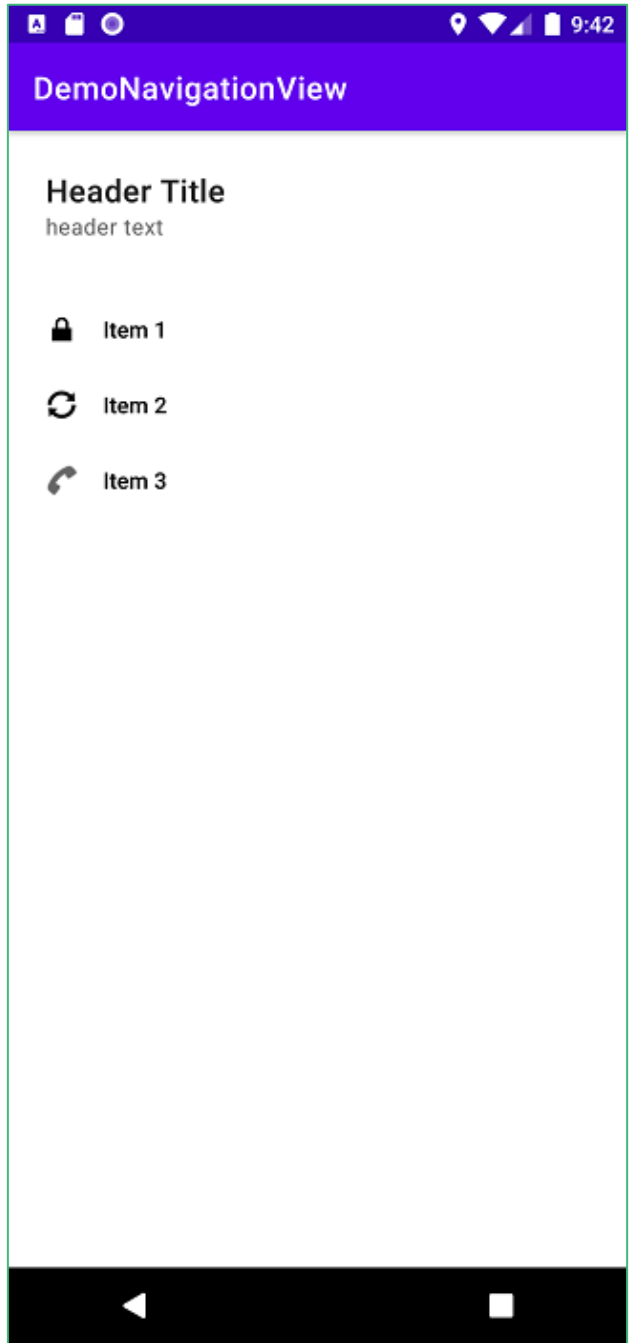
5. Navigation View

- **Navigation View:** dễ để triển khai Navigation, là 1 mở rộng Của **FrameLayout.** Có 2 thành phần chính
  1. Header View [optional]
  2. Menu

Một số thuộc tính

- **Sử dụng trên layout View**

```
<com.google.android.material.navigation.NavigationView
    android:id="@+id/navView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:headerLayout="@layout/header_nav"
    app:menu="@menu/menu_nav" />
```



Navigation View

- **Header layout:** layout/header_nav.xml

header_nav.xml

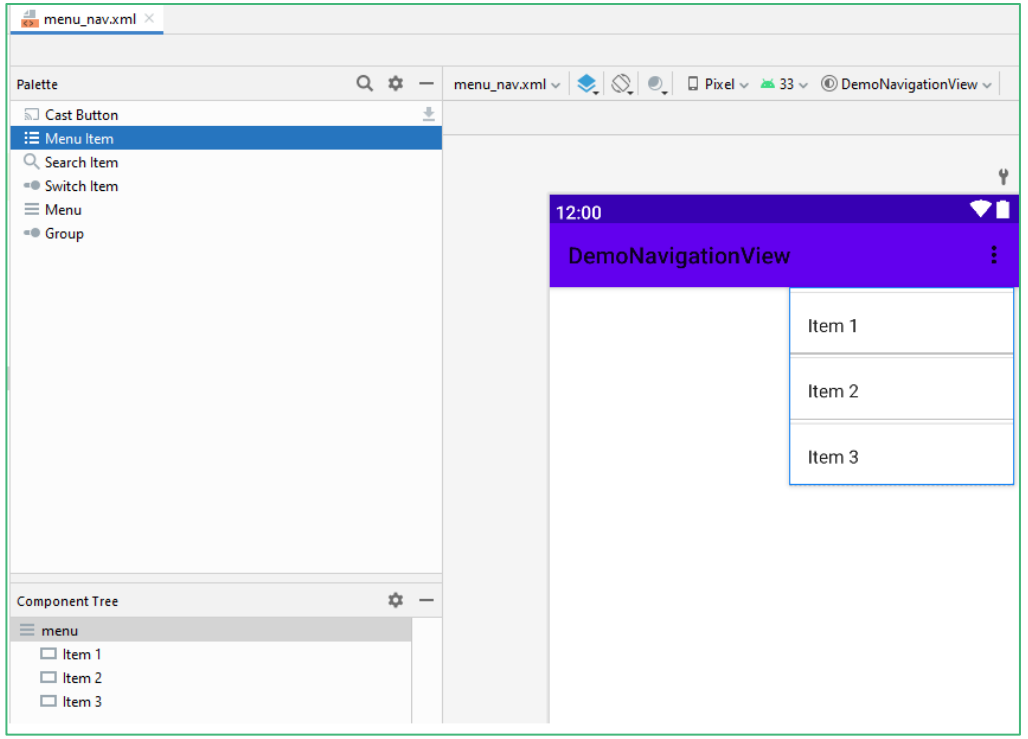
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:layout_marginStart="24dp"
        android:layout_marginEnd="24dp"
        android:textAppearance="?attr/textAppearanceHeadline6"
        android:text="Header Title" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="24dp"
        android:layout_marginStart="24dp"
        android:layout_marginEnd="24dp"
        android:textAppearance="?attr/textAppearanceBody2"
        android:textColor="@color/material_on_surface_emphasis_medium"
        android:text="header text" />
</LinearLayout>
```

- **menu item:** layout/menu/menu_nav.xml

menu_nav.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:icon="@android:drawable/ic_lock_lock"
        android:title="Item 1" />
    <item
        android:icon="@android:drawable/ic_popup_sync"
        android:title="Item 2" />
    <item
        android:icon="@android:drawable/ic_menu_call"
        android:title="Item 3" />
</menu>
```

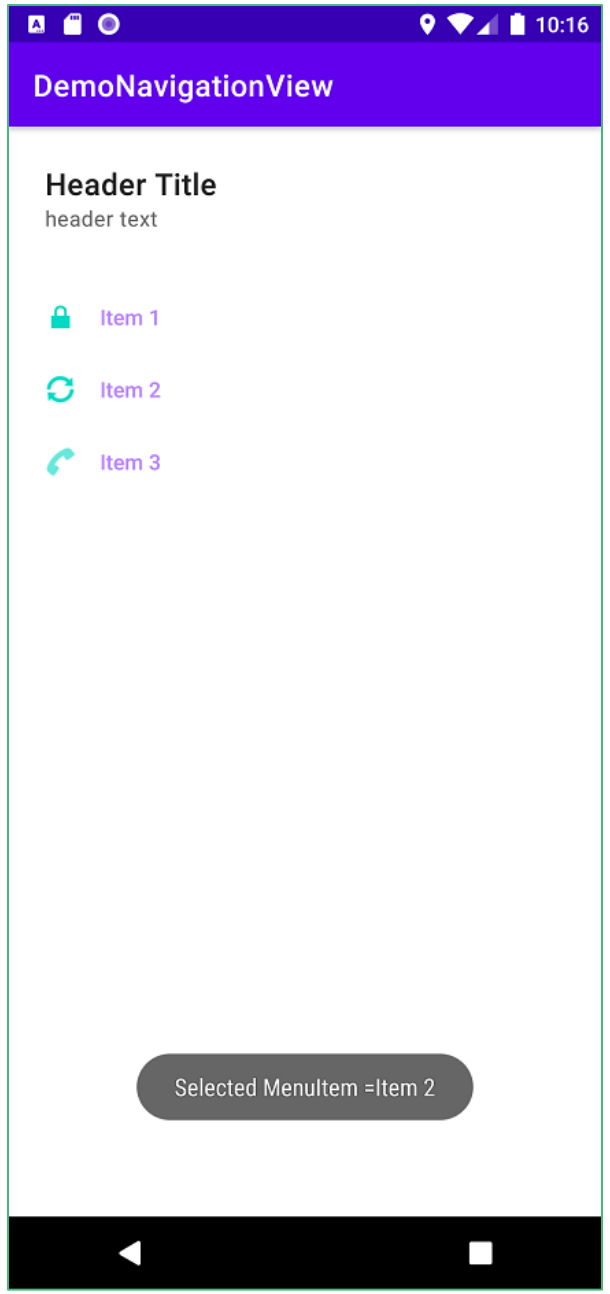
- **Một số thuộc tính của navigationView**
  - app:itemTextColor
  - app:itemIconTint
  - app:itemBackground



Navigation View

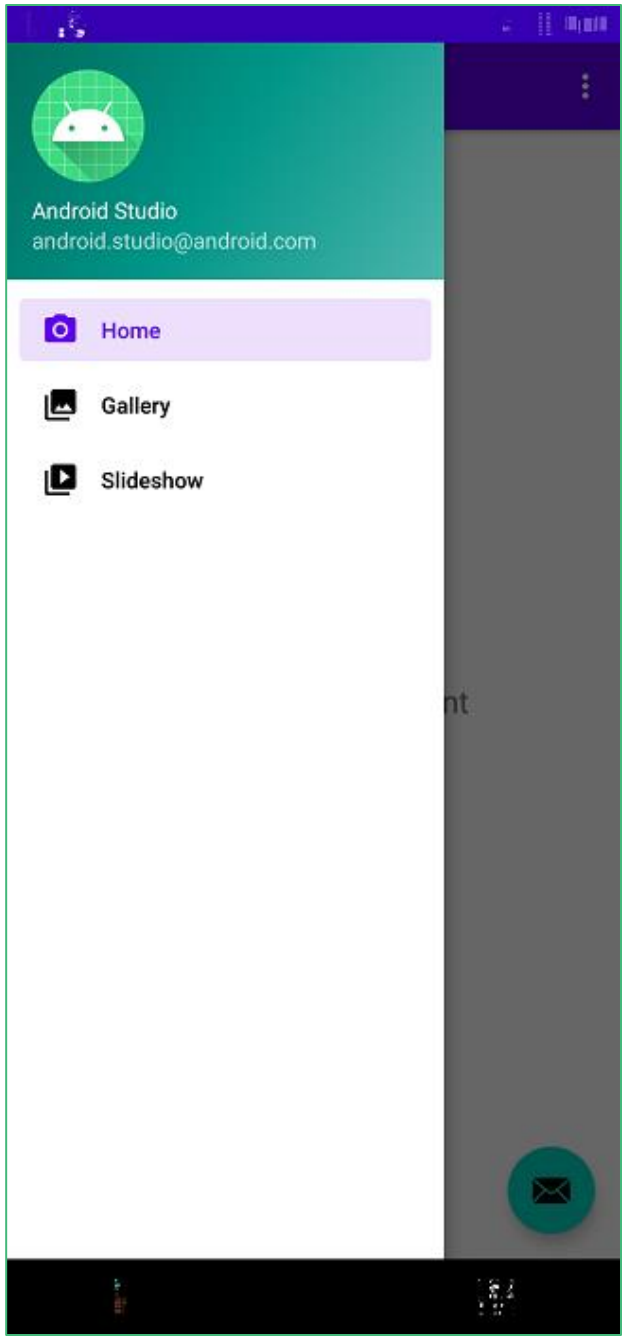
- **Item Selected Listener:** setNavigationItemSelectedListener

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    NavigationView mNavigationView = (NavigationView) findViewById(R.id.navView);
    mNavigationView.setNavigationItemSelectedListener(new NavigationView.OnNavigationItemSelectedListener() {
        @Override
        public boolean onNavigationItemSelected(@NonNull MenuItem item) {
            Toast.makeText( context: MainActivity.this,  text: "Selected MenuItem =" + item.getTitle(), Toast.LENGTH_SHORT).show();
            return false;
        }
    });
}
```



6. NavigationDrawer

- Navigation Drawer: Navigation Drawer chỉ đơn giản là một thanh menu được ẩn đi về phía bên trái màn hình

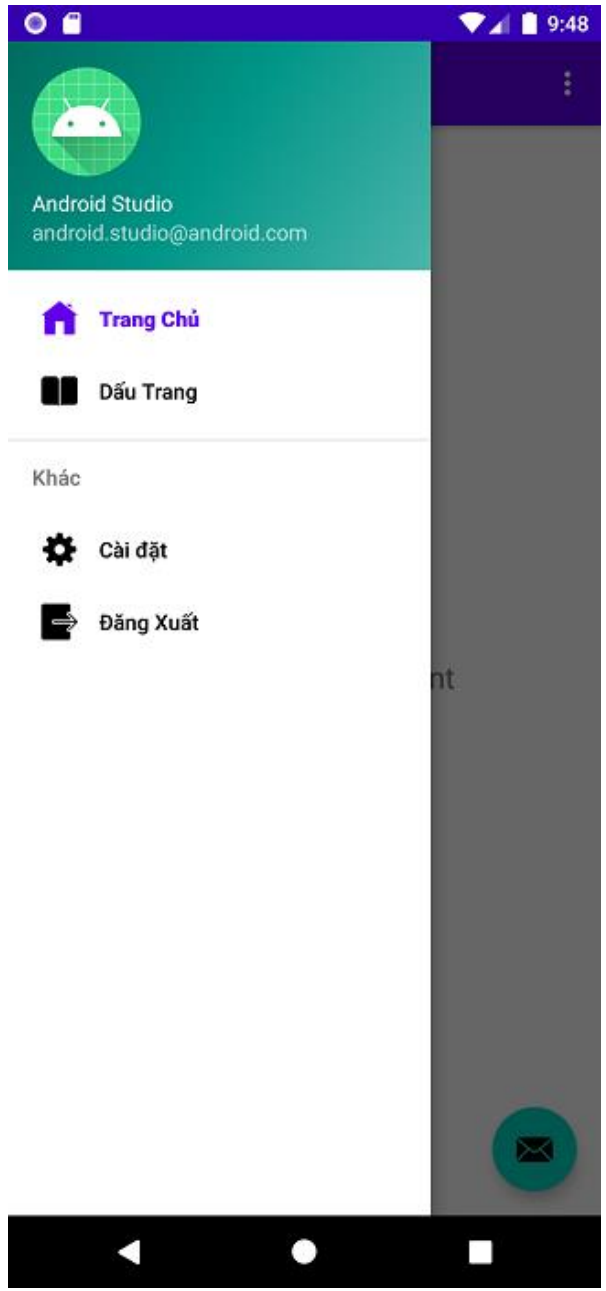


DEMO Xây dựng NavigationDrawer

Ví dụ: Xây dựng giao diện sử dụng **DrawerNavigation**

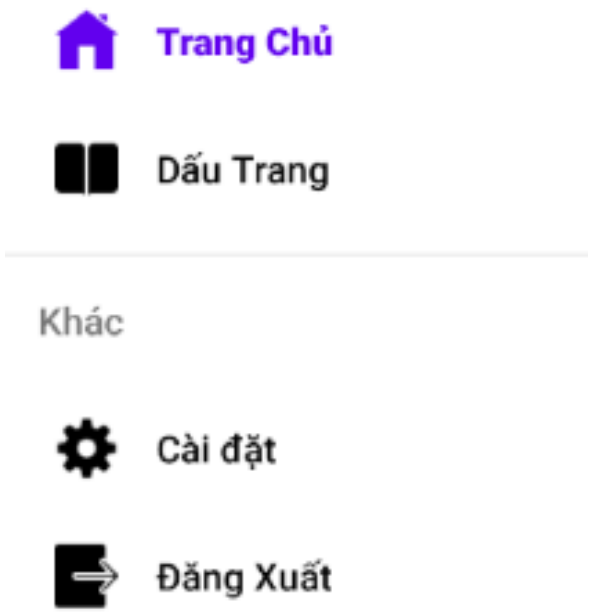
Yêu cầu:

- Mỗi item phải có icon
- Chia item trong DrawerNavigation thành 2 phần
- Trên DrawerNavigation có headerLayout



Xây dựng NavigationDrawer

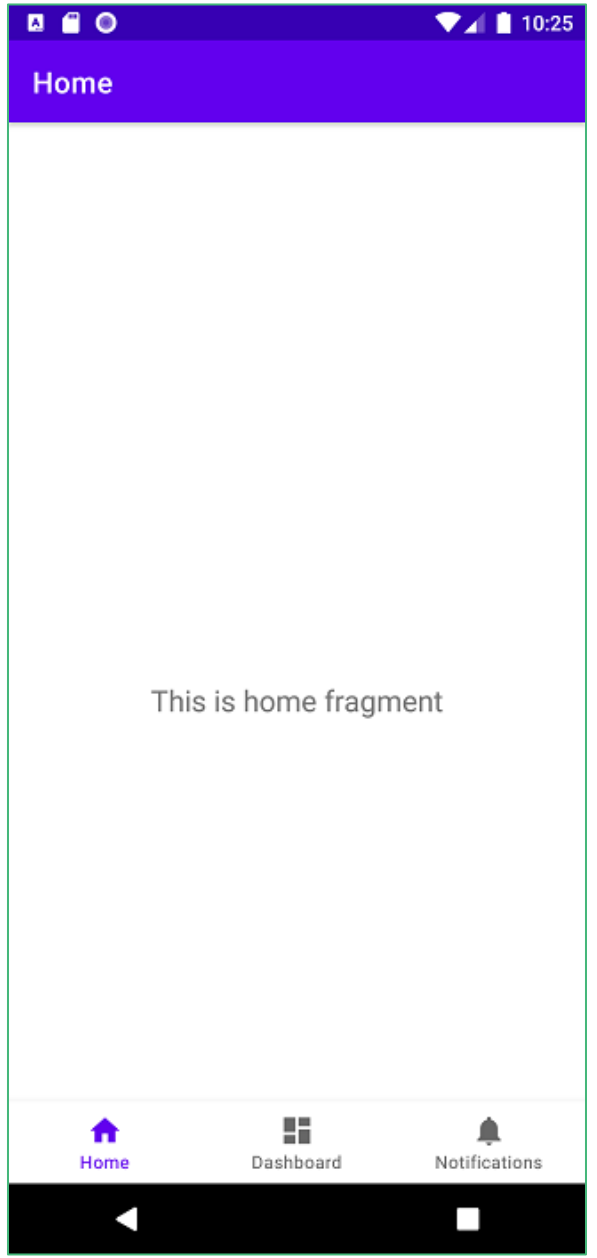
Tạo menu và icon cho menu
trong DrawerNavigation




```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:showIn="navigation_view">

    <item
        android:id="@+id/nav_home"
        android:icon="@drawable/ic_home"
        android:title="Trang Chủ" />
    <item
        android:id="@+id/nav_bookmark"
        android:icon="@drawable/ic_bookmark"
        android:title="Dấu Trang" />
    <group android:checkableBehavior="single">
        <item android:title="Khác">
            <menu>
                <item
                    android:id="@+id/nav_setting"
                    android:icon="@drawable/ic_setting"
                    android:title="Cài đặt" />
                <item
                    android:id="@+id/nav_logout"
                    android:icon="@drawable/ic_logout"
                    android:title="Đăng Xuất" />
            </menu>
        </item>
    </group>
</menu>
```

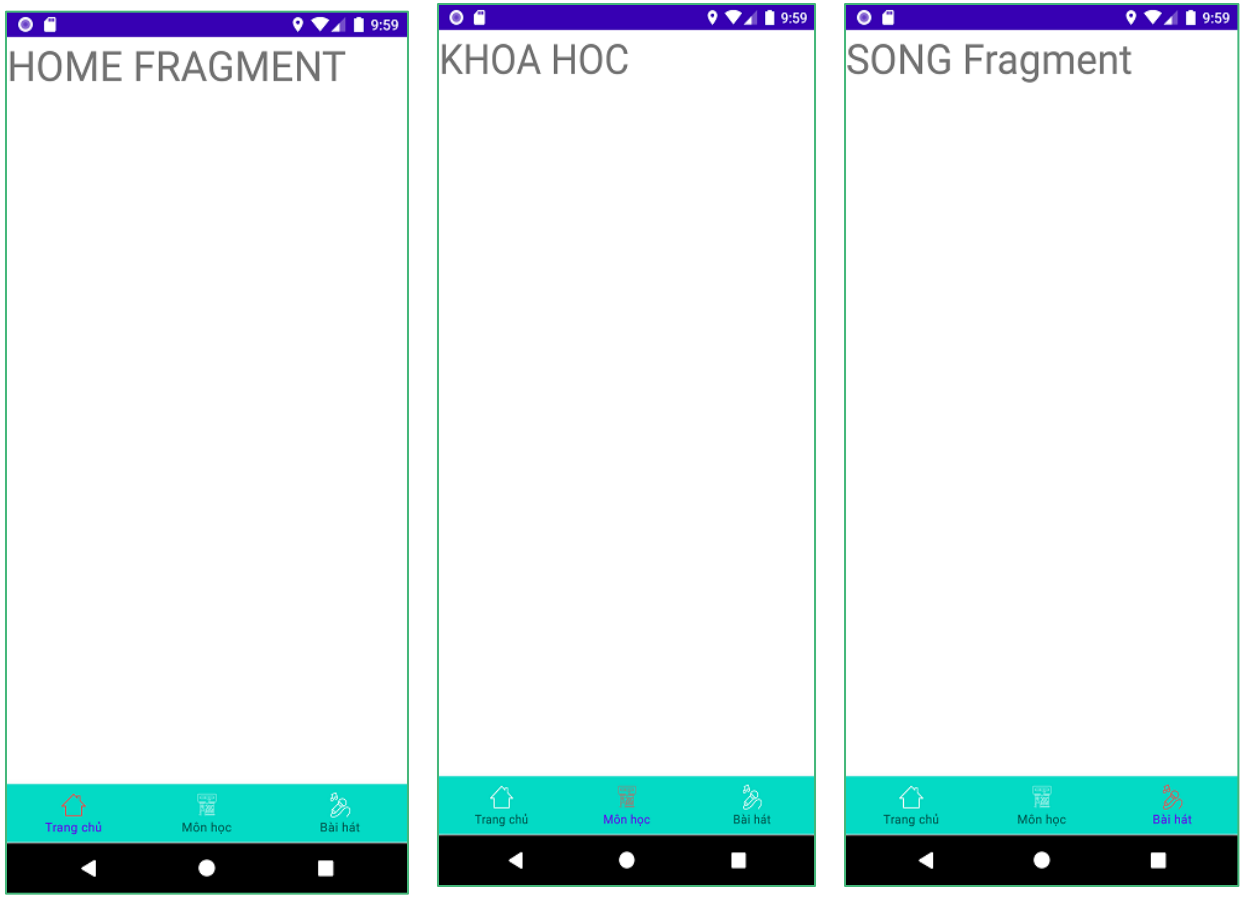
7. BottomNavigationView

- Bottom NavigationView:là một thanh menu ở cuối màn hình cung cấp điều hướng giữa các chế độ views ở top-level trong ứng dụng



ASM 4.2

1. Thiết kế giao diện **BottomNavigation**
Gồm 3 fragment tương ứng
2. Hãy thêm ViewPager2 để sử dụng khi người dùng vuốt (swipe)
3. Giữ trạng thái checked khi vuốt như BottomNavigation được chọn
4. Custom lại màu icon khi checked
5. Tìm hiểu để thay đổi hiệu ứng vuốt (**Zoom-out page transformer**)
https://developer.android.com/develop/ui/views/animations/screen-slide-2
6. Thực hiện fragment Khóa học hoặc Song đã làm (đưa danh sách trên recyclerView.)

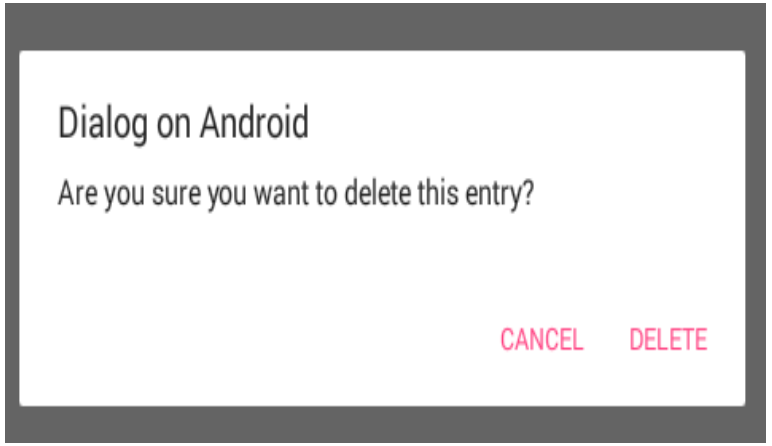


8. AlertDialog

- AlertDialog là lớp con của Dialog. Dialog có thể coi là một hộp thoại mà người dùng có thể tương tác trực tiếp được.

Ví dụ: khi xóa một tập tin quan trọng hay muốn thoát một chương trình nào đó thì cần một thông báo để confirm ?

- AlertDialog gồm có 3 vùng:
  - Tiêu đề (title)
  - Nội dung (content)
  - Các button hành động.
- Để tạo Alert Dialog thường sử dụng thành phần **AlertDialog.Builder**




```
AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(this);
```

AlertDialog

Sử dụng AlertDialog

- **setTitle(CharSequence title)** – Phương thức ***tùy chọn*** dùng để thiết lập title.
- **setIcon(Drawable icon)** – Phương thức dùng để thêm icon trước tiêu đề. Icon được lưu trong drawable.
- **setMessage(CharSequence message)** – Phương thức **bắt buộc** này hiển thị nôi dung trong Alert Dialog.
- **setCancelable(boolean cancelable)** –có 2 giá trị **true/false**.
  - Nếu **false** thì khi show dialog lên người dùng click ra bên ngoài dialog thì vẫn không bị mất
  - Nếu **true** thì sẽ mất khi click vào bất kì đâu ngoài dialog.

AlertDialog

Sử dụng AlertDialog

❑ setPositiveButton(CharSequence text, DialogInterface.OnClickListener listener)

Phương thức thiết lập một Button Positive (Yes/Agree/…)

```
builder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int id) {
        Toast.makeText(MainActivity.this,"Đã chọn YES là THÍCH", Toast.LENGTH_SHORT).show();
    }
});
```

❑ setNegativeButton(CharSequence text, DialogInterface.OnClickListener listener)

Phương thức này thiết lập một Button Negative (No/Không/…)

```
builder.setNegativeButton("No", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int id) {
        Toast.makeText(MainActivity.this,"Đã chọn No tức là THÍCH NHIỀU", Toast.LENGTH_SHORT).show();
        // Cancel
        dialog.cancel();
    }
});
```

AlertDialog

Sử dụng AlertDialog

❑ setNeutralButton(CharSequence text, DialogInterface.OnClickListener listener)

Phương thức này đơn giản là thêm một Button mới, với button này chúng ta có thể xử lý sự kiện cho nó.

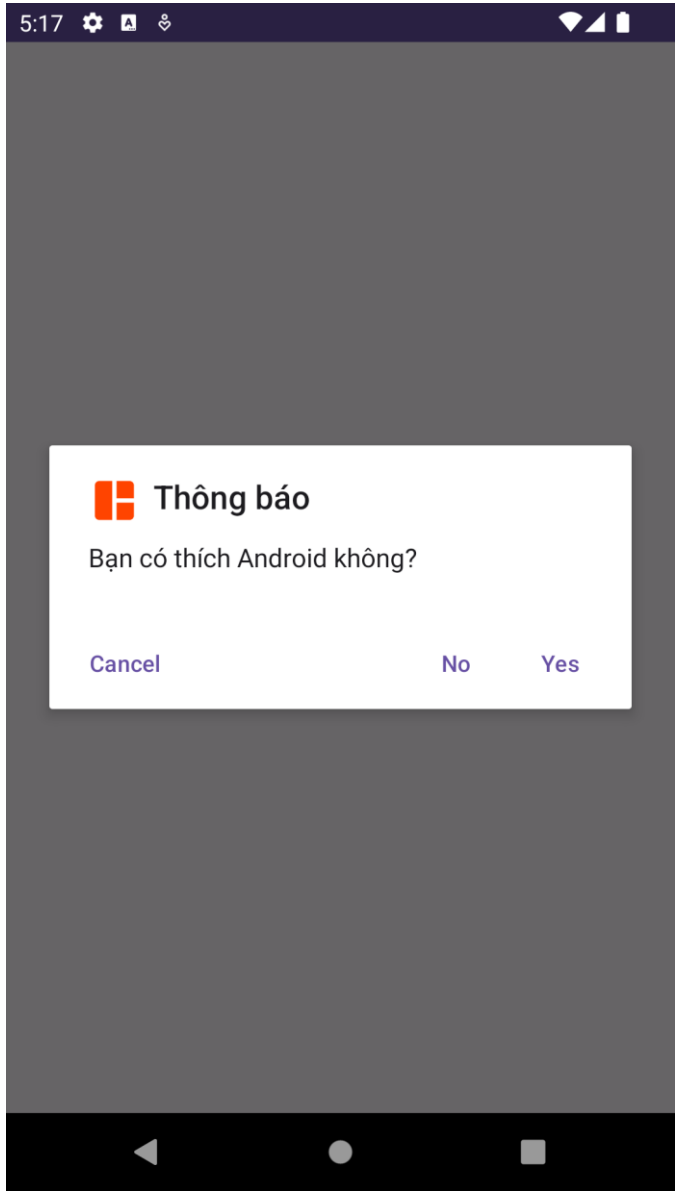
*Ví dụ 2 phương thức trên bạn đã thiết lập 2 Button "**Yes**" và "**No**", trong phương thức này chúng ta thiết lập Button "**Cancel**".*

```
alertDialogBuilder.setNeutralButton( text: "Cancel", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        //...
    }
});
```

AlertDialog

Ví dụ-1

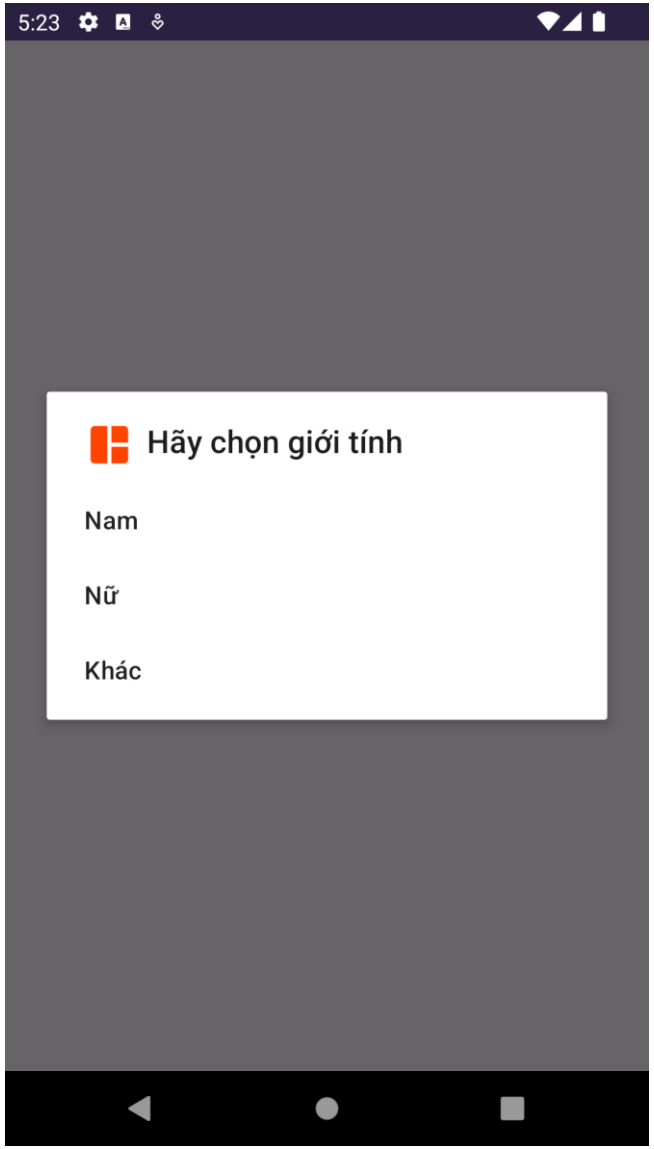
```
AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setTitle("Thông Báo");
builder.setIcon(R.drawable.ic_question);
builder.setMessage("Bạn đã thích Android chưa ?");
builder.setCancelable(false);

// Create "Positive" button with OnClickListener.
builder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
   public void onClick(DialogInterface dialog, int id) {
      Toast.makeText(MainActivity.this,"Đã chọn YES là THÍCH", Toast.LENGTH_SHORT).show();
   }
});
// Create "No" button with OnClickListener.
builder.setNegativeButton("No", new DialogInterface.OnClickListener() {
   public void onClick(DialogInterface dialog, int id) {
      Toast.makeText(MainActivity.this,"Đã chọn No tức là THÍCH NHIỀU", Toast.LENGTH_SHORT).show();
      //  Cancel
      dialog.cancel();
   }
});

// Create AlertDialog
AlertDialog alert = builder.create();
alert.show();
```



AlertDialog

Ví dụ-2

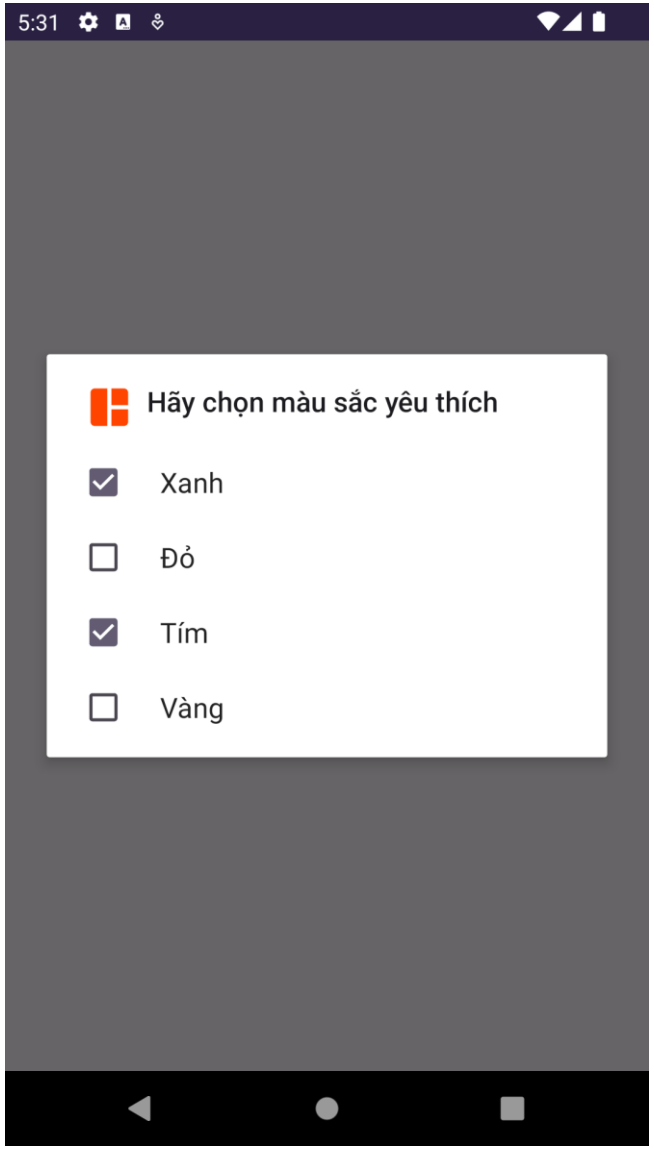
```
String[] gioiTinh = new String[]{"Nam", "Nữ", "Khác"};

AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder( context: this);
alertDialogBuilder.setTitle("Hãy chọn giới tính");
alertDialogBuilder.setIcon(R.drawable.ic_question);

alertDialogBuilder.setItems(gioiTinh, new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        Toast.makeText( context: MainActivity.this,  text: "Giới tính: " + gioiTinh[which], Toast.LENGTH_SHORT).show();
    }
});

AlertDialog alertDialog = alertDialogBuilder.create();
alertDialog.show();
```



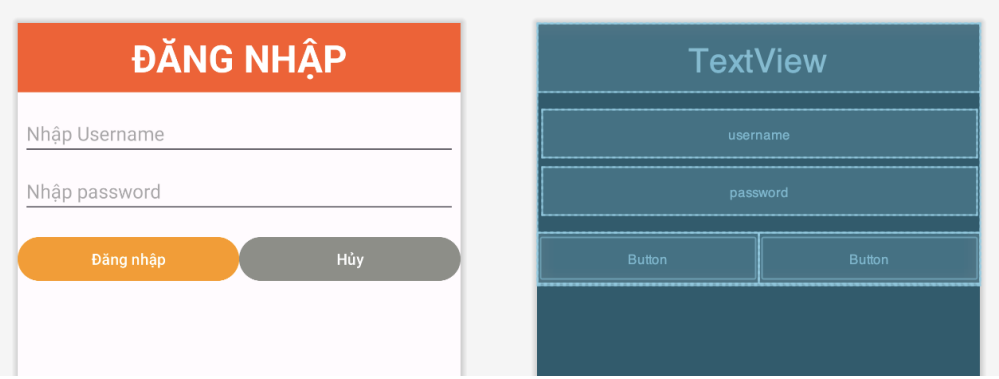
AlertDialog

Ví dụ-3

```
String[] mauSac = new String[]{"Xanh", "Đỏ", "Tím", "Vàng"};

AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder( context: this);
alertDialogBuilder.setTitle("Hãy chọn màu sắc yêu thích");
alertDialogBuilder.setIcon(R.drawable.ic_question);

alertDialogBuilder.setMultiChoiceItems(mauSac,  checkedItems: null, new DialogInterface.OnMultiChoiceClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which, boolean isChecked) {
        if (isChecked)
            Toast.makeText( context: MainActivity.this,  text: "Bạn chọn màu " + mauSac[which], Toast.LENGTH_SHORT).show();
        else
            Toast.makeText( context: MainActivity.this,  text: "Bạn bỏ chọn màu " + mauSac[which], Toast.LENGTH_SHORT).show();
    }
});

AlertDialog alertDialog = alertDialogBuilder.create();
alertDialog.show();
```

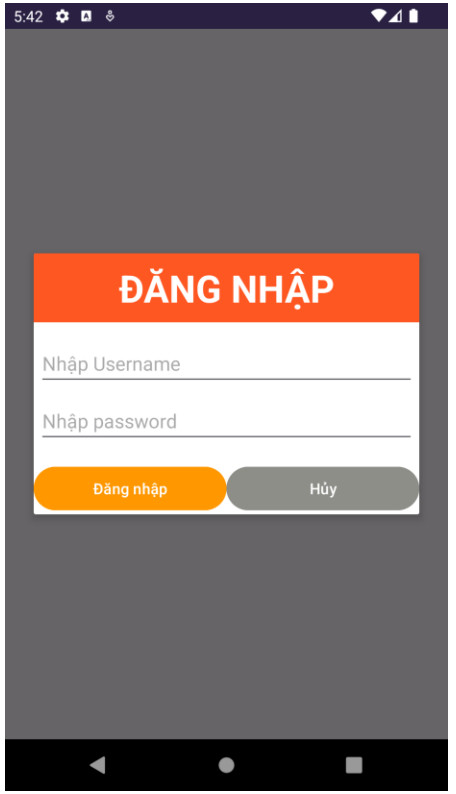


9. Custom AlertDialog



- Tạo một layout cho dialog
- Tạo dialog, gắn layout và gán view

```
AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder( context: this);
LayoutInflater layoutInflater = getLayoutInflater();
View view = layoutInflater.inflate(R.layout.dialog_login,  root: null);
alertDialogBuilder.setView(view);

AlertDialog alertDialog = alertDialogBuilder.create();
alertDialog.show();
```



- Ánh xạ và xử lý chức năng trong layout

9. Custom AlertDialog

- Ánh xạ và xử lý chức năng trong layout

```
AlertDialog alertDialog = alertDialogBuilder.create();

EditText edtUser = view.findViewById(R.id.edtUser);
EditText edtPass = view.findViewById(R.id.edtPass);
Button btnLogin = view.findViewById(R.id.btnLogin);
Button btnCancel = view.findViewById(R.id.btnCancel);

btnLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String user = edtUser.getText().toString();
        String pass = edtPass.getText().toString();

        if (user.equals("poly") && pass.equals("android@2")) {
            Toast.makeText( context: MainActivity.this,  text: "Đăng nhập thành công", Toast.LENGTH_SHORT).show();
            alertDialog.dismiss();
        } else {
            Toast.makeText( context: MainActivity.this,  text: "Đăng nhập thất bại", Toast.LENGTH_SHORT).show();
        }
    }
});

btnCancel.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        alertDialog.dismiss();
    }
});

alertDialog.show();
```

ASM 4.3

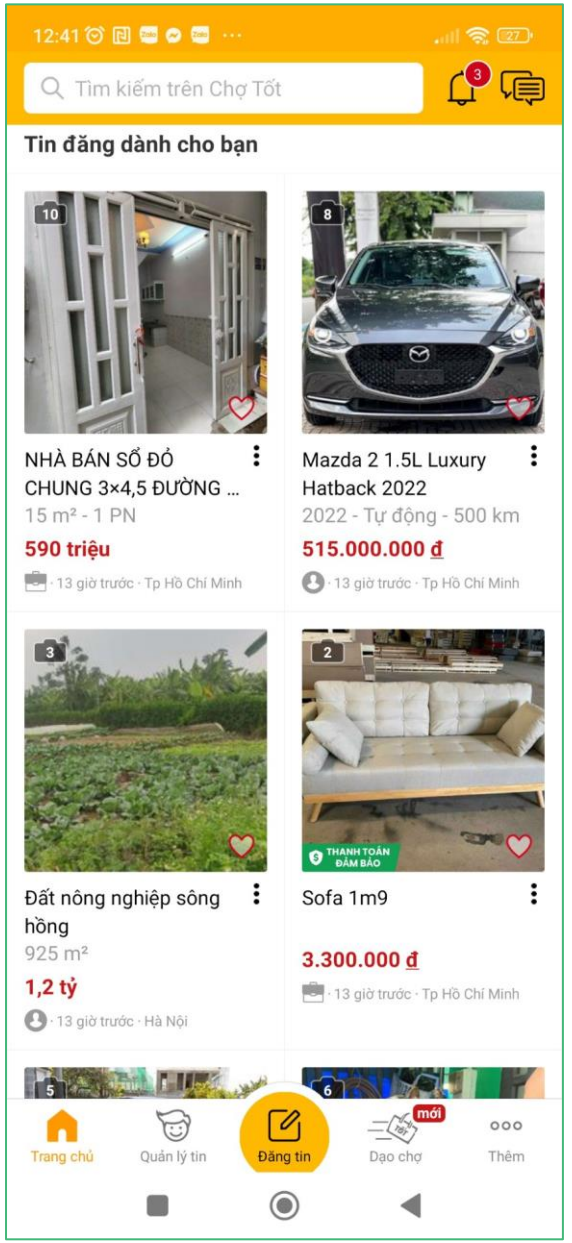
1. Thiết kế giao diện có **BottomNavigation**

Ở Trang Chủ

- SearchView
- Slider Image

https://www.youtube.com/watch?v=HkwMzElr6j4

- Truy cập nhanh
- Khám phá danh mục
- Tin dành cho bạn



ASM 4.4

1. Thiết kế giao diện ứng dụng như **e-HUTECH**

- Trang Chủ
- Thời khóa biểu (ngày)
- Cài đặt