



Lập trình Android

Bài 10: *Các đối tượng giao diện nâng cao*

Phòng LT & Mạng

<http://csc.edu.vn/lap-trinh-va-csdl>





Nội dung

1. Toolbar
2. Menu
3. ViewPager
4. TabLayout
5. Navigation Drawer
6. SwipeRefreshLayout



Toolbar

- Là một trong các thiết kế quan trọng, cung cấp hình dung về cấu trúc ứng dụng, các thành phần tương tác thân thiện với người dùng.
- Thay thế cho ActionBar
- Thuộc gói thư viện **appcompat support v7**
- Ý nghĩa áp dụng của **Toolbar**:
 - Cho biết ứng dụng đang ở màn hình nào
 - Truy cập nhanh các tính năng thường sử dụng, như Tìm kiếm
 - Hỗ trợ điều hướng





Các bước thêm Toolbar và activity

- **Bước 1:** Thêm v7 appcompat support library vào project (build.gradle)

```
implementation 'com.android.support:appcompat-v7:28.0.0'
```

- **Bước 2:** Activity phải extends từ AppCompatActivity:

```
public class MyActivity extends AppCompatActivity {  
    // ...  
}
```

MyActivity.java

- **Bước 3:** Sử dụng theme **NoActionBar** tương ứng với appcompat

```
<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">  
    <item name="colorPrimary">@color/colorPrimary</item>  
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
    <item name="colorAccent">@color/colorAccent</item>  
</style>
```

res/values/styles.xml

- Có thể điều chỉnh màu sắc của Toolbar thông qua cấu hình **colorPrimary** ở style



Các bước thêm Toolbar và activity (tiếp)

○ Bước 4: Thêm Toolbar vào layout của activity

```
<?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"
    tools:context=".MainActivity">

    <android.support.v7.widget.Toolbar
        android:id="@+id/my_toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        android:elevation="4dp"
        android:theme="@style/ThemeOverlay.AppCompat.ActionBar"
        app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />

</LinearLayout>
```



activity_main.xml

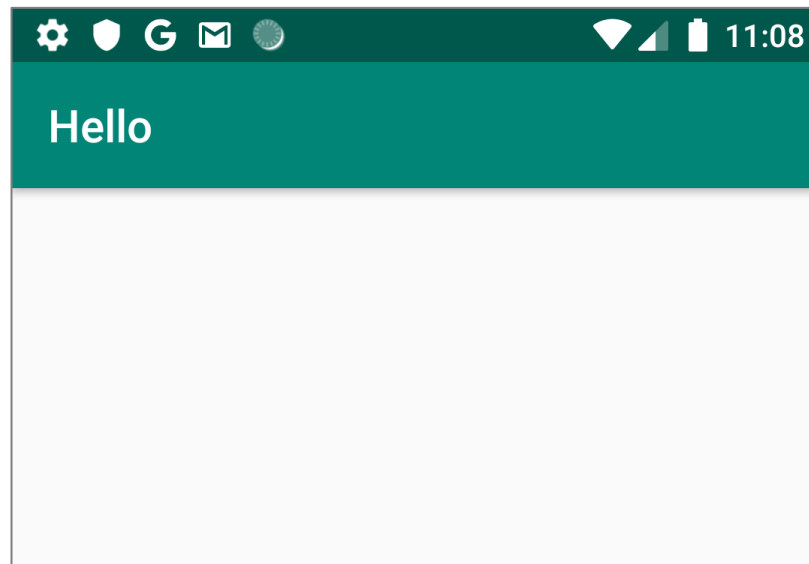


Các bước thêm Toolbar và activity (tiếp)

- **Bước 5:** Trong hàm **onCreate()** của activity, gọi **setSupportActionBar()** để toolbar trở thành app bar của activity

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_my);  
    Toolbar myToolbar = (Toolbar) findViewById(R.id.my_toolbar);  
    setSupportActionBar(myToolbar);  
}
```

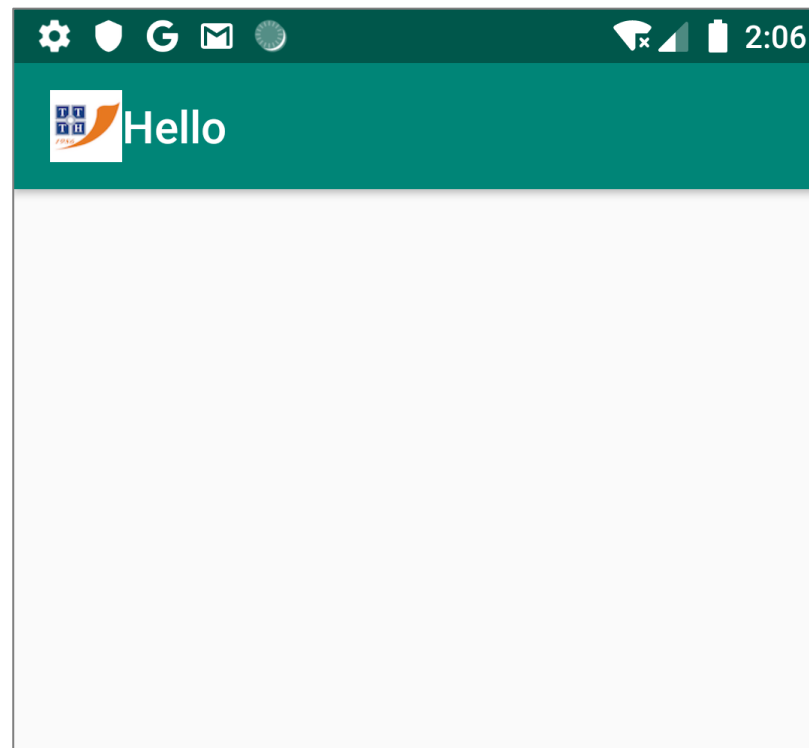




Một số tùy chỉnh cho Toolbar

- Thiết lập logo, title, title text color:

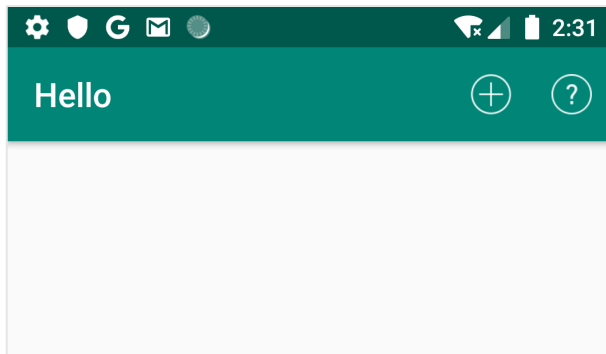
```
Toolbar myToolbar = findViewById(R.id.my_toolbar);  
myToolbar.setTitle("Hello");  
myToolbar.setTitleTextColor(Color.WHITE);  
myToolbar.setLogo(getDrawable(R.drawable.t3h));  
setSupportActionBar(myToolbar);
```



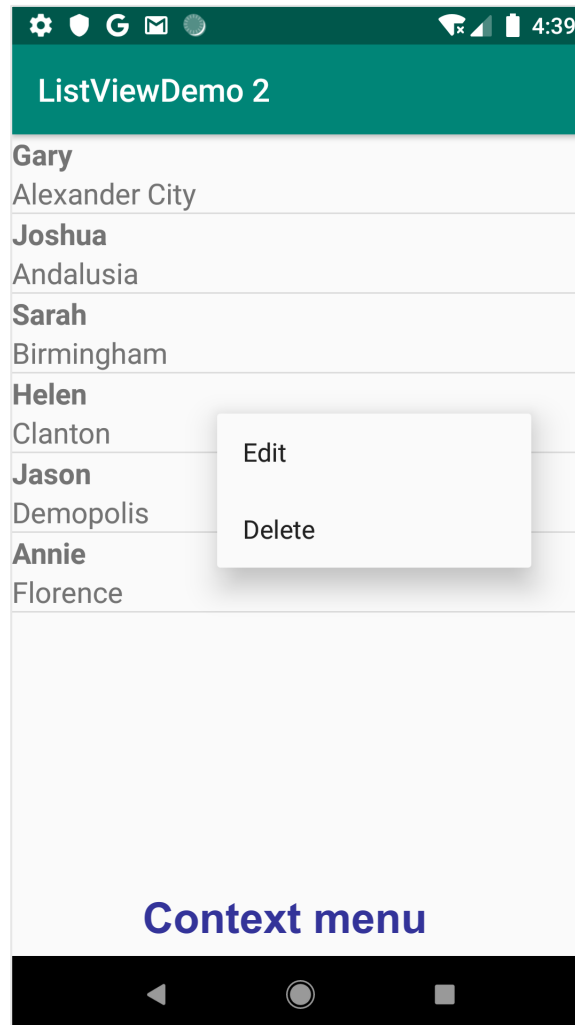


Menu

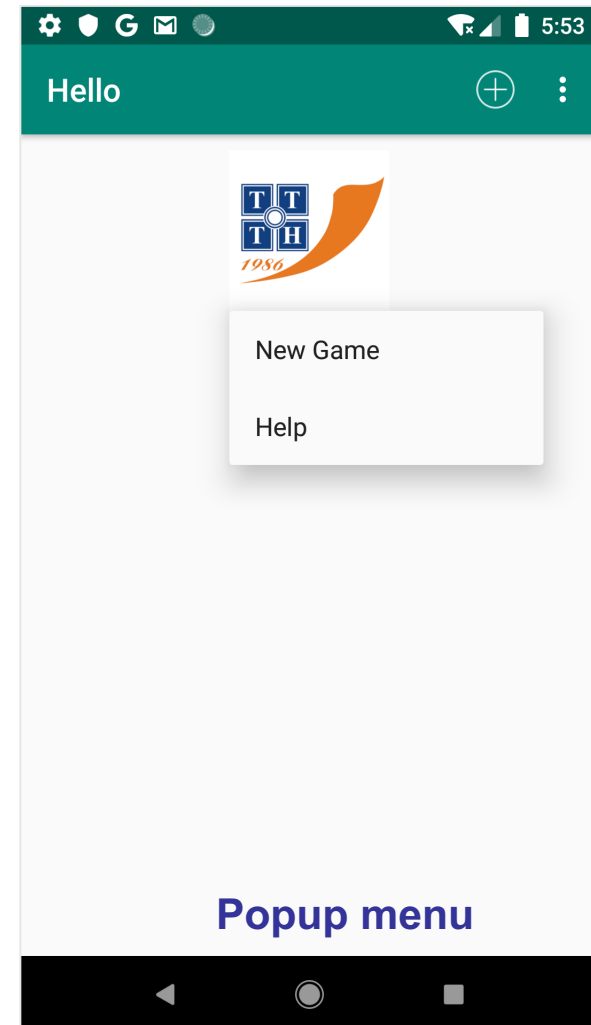
- Có 3 loại menu:
 - Option menu
 - Context menu
 - Popup menu



Option menu



Context menu



Popup menu



Option menu

- Thêm option menu vào Toolbar:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+id/new_game"
        android:icon="@drawable/ic_new" android:title="@string/new_game"
        android:showAsAction="ifRoom"/>
  <item android:id="@+id/help"
        android:icon="@drawable/ic_help" android:title="@string/help"
        android:showAsAction="ifRoom" />
</menu>
```

res/menu/game_menu.xml

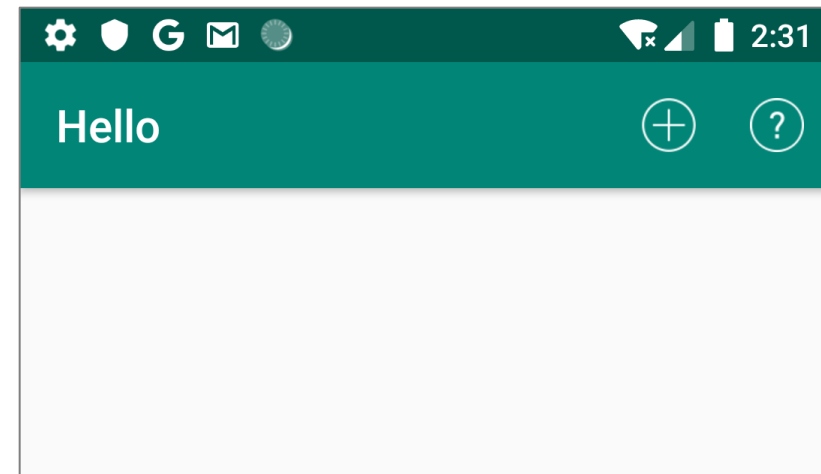
(1) Định nghĩa menu, menu items

```
public class MainActivity extends AppCompatActivity {
  // ...
  @Override
  public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.game_menu, menu);
    return true;
  }
  // ...
}
```

MainActivity.java

(2) Gắn menu vào activity

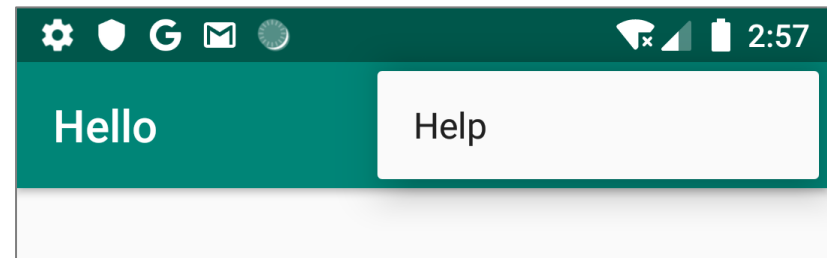
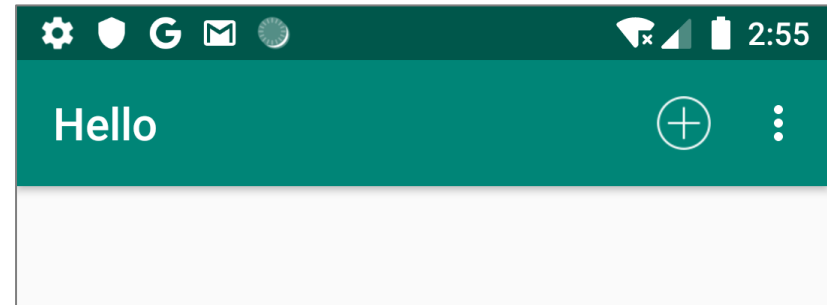
(3) Kết quả chạy thử





Option menu (tiếp)

- Các thuộc tính của menu item:
 - **id**: id của menu item
 - **icon**: icon của menu item (drawable)
 - **showAsAction**: cho biết có hiển thị menu item thành 1 nút nhấn trên toolbar hay không. Có thể nhận các giá trị như ifRoom, always...



```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+id/new_game"
        android:icon="@drawable/ic_new"
        android:title="@string/new_game"
        android:showAsAction="ifRoom"/>
  <item android:id="@+id/help"
        android:icon="@drawable/ic_help"
        android:title="@string/help" />
</menu>
```

res/menu/game_menu.xml

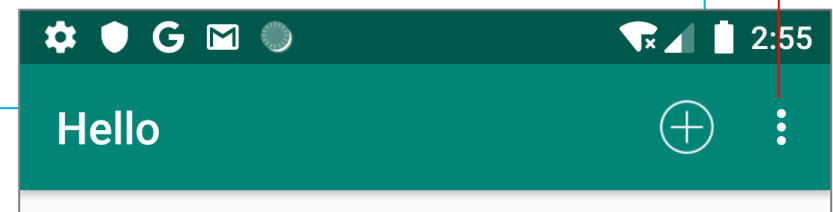


Option menu (tiếp)

- Xử lý sự kiện click lên menu item → override phương thức `onOptionsItemSelected`

```
public class MainActivity extends AppCompatActivity {  
    // ...  
    @Override  
    public boolean onOptionsItemSelected(MenuItem item) {  
        // Handle item selection  
        switch (item.getItemId()) {  
            case R.id.new_game: ← Cần thiết lập id cho các menu item ở XML menu  
                newGame();  
                return true;  
            case R.id.help:  
                showHelp();  
                return true;  
            default:  
                return super.onOptionsItemSelected(item);  
        }  
    }  
}
```

MainActivity.java





Context menu

- Minh họa sau đây trình bày cách thêm một context menu vào ListView
- **Bước 1:** Định nghĩa XML menu

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/edit"
          android:title="@string/edit" />
    <item android:id="@+id/delete"
          android:title="@string/delete" />
</menu>
```

res/menu/context_menu.xml

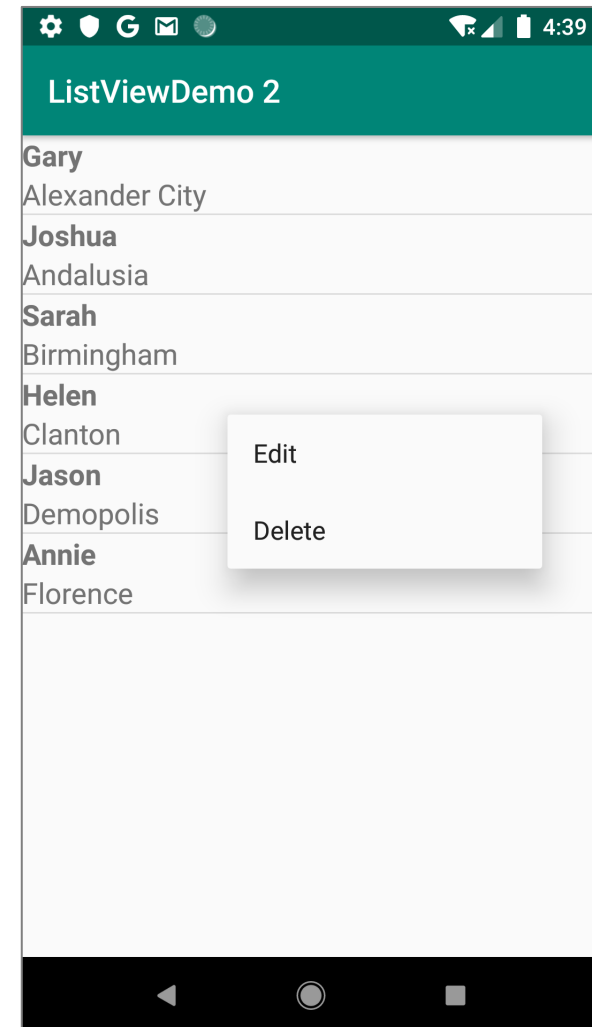
- **Bước 2:** Đăng ký listview với context menu

```
public class MainActivity extends AppCompatActivity {
    ListView lvItems;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        lvItems = this.findViewById(R.id.lvItems);
        // ...
        registerForContextMenu(lvItems);
    }
}
```

MainActivity.java

Lập trình Android





Context menu (tiếp)

- **Bước 3:** Tạo menu và gắn vào activity

```
public class MainActivity extends AppCompatActivity {  
    ListView lvItems;  
    ArrayList<User> data;  
  
    // ...  
  
    @Override  
    public void onCreateContextMenu(ContextMenu menu, View v,  
                                   ContextMenu.ContextMenuInfo menuInfo) {  
        super.onCreateContextMenu(menu, v, menuInfo);  
        MenuInflater inflater = getMenuInflater();  
        inflater.inflate(R.menu.context_menu, menu);  
    }  
}
```

MainActivity.java

Context menu (tiếp)



- **Bước 4:** Xử lý sự kiện click lên menu item

```
public class MainActivity extends AppCompatActivity {
    ListView lvItems;
    ArrayList<User> data;

    // ...

    @Override
    public boolean onContextItemSelected(MenuItem item) {
        AdapterView.AdapterContextMenuInfo info =
            (AdapterView.AdapterContextMenuInfo) item.getMenuInfo();
        switch (item.getItemId()) {
            case R.id.edit:
                editUser(data.get(info.position));
                return true;
            case R.id.delete:
                deleteUser(data.get(info.position));
                return true;
            default:
                return super.onContextItemSelected(item);
        }
    }
}
```

MainActivity.java



Popup menu

- Popup menu là một dạng context menu, neo vào một View nào đó

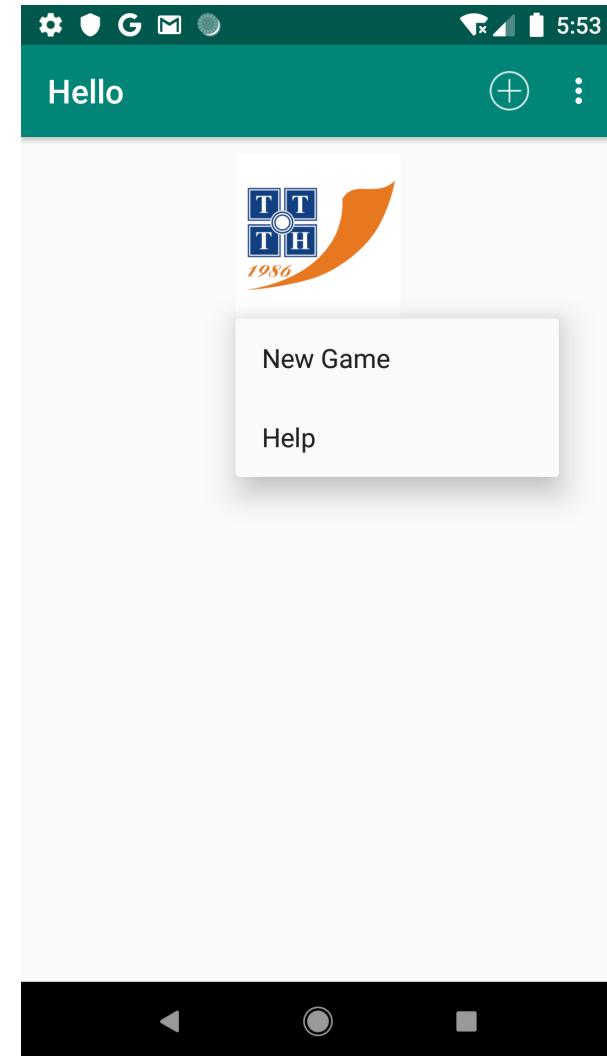
```
public class MainActivity extends AppCompatActivity
    implements View.OnClickListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        findViewById(R.id.ivImage).setOnClickListener(this);
    }

    public void showMenu(View v) {
        PopupMenu popup = new PopupMenu(this, v);
        // popup.setOnMenuItemClickListener(this);
        popup.inflate(R.menu.game_menu);
        popup.show();
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.ivImage)
            showMenu(v);
    }
}
```

MainActivity.java





Popup menu (tiếp)

- Xử lý sự kiện click lên menu item

```
public class MainActivity extends AppCompatActivity
    implements PopupMenu.OnMenuItemClickListener, View.OnClickListener {

    // ...

    public void showMenu(View v) {
        PopupMenu popup = new PopupMenu(this, v);
        popup.setOnMenuItemClickListener(this);
        popup.inflate(R.menu.game_menu);
        popup.show();
    }

    @Override
    public boolean onMenuItemClick(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.new_game:
                newGame();
                return true;
            case R.id.help:
                showHelp();
                return true;
            default:
                return false;
        }
    }
}
```

MainActivity.java





ViewPager

- **ViewPager** là một layout manager cho phép người dùng duyệt qua từng trang nội dung (page), bằng thao tác vuốt trái/phải (swipe)
- Mỗi trang nội dung khác nhau thường được quản lý bởi 1 fragment
- Minh họa sau trình bày một



ViewPager gồm 5 trang, mỗi trang trình bày một nội dung dự báo thời tiết



Rainy
27 °C



Sunny
29 °C





Các bước tích hợp ViewPager

- **Bước 1:** Thêm cấu hình thư viện support design vào build.grade của module app

```
implementation 'com.android.support:design:28.0.0'
```

- **Bước 2:** Thêm ViewPager vào XML layout

```
<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=".MainActivity">

    <android.support.v4.view.ViewPager
        android:id="@+id/pagerWeather"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

activity_main.xml



Các bước tích hợp ViewPager (tiếp)

- **Bước 3:** Định nghĩa 1 page là 1 Fragment - (3a) Tạo XML layout file cho fragment

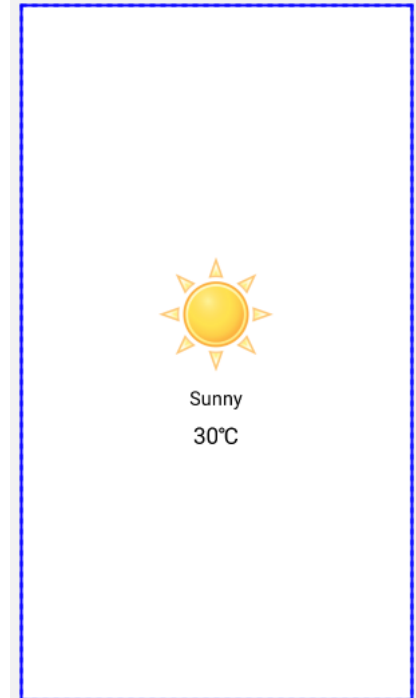
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/white"
    android:gravity="center"
    android:orientation="vertical">
```

```
<ImageView
    android:id="@+id/ivWeather"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_sunny" />
```

```
<TextView
    android:id="@+id/tvStatus"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Sunny"
    android:layout_marginTop="10dp"
    android:textColor="@android:color/black"
    android:textSize="20sp" />
```

```
<TextView
    android:id="@+id/tvTemperature"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="30\u2103"
    android:layout_marginTop="10dp"
    android:textColor="@android:color/black"
    android:textSize="24sp" />
```

```
</LinearLayout>
```





Các bước tích hợp ViewPager (tiếp)

- **Bước 3:** Định nghĩa 1 page là 1 Fragment - (3b) Tạo Java class cho fragment

```
public class WeatherFragment extends Fragment {  
    int[] images = {R.drawable.ic_cloudy, R.drawable.ic_few_clouds, R.drawable.ic_rainy,  
R.drawable.ic_storm, R.drawable.ic_sunny};  
    String[] status = {"Cloudy", "Few Clouds", "Rainy", "Storm", "Sunny"};  
  
    @Nullable  
    @Override  
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup  
container, @Nullable Bundle savedInstanceState) {  
        View view = inflater.inflate(R.layout.fragment_weather, container, false);  
        ImageView ivWeather = view.findViewById(R.id.ivWeather);  
        TextView tvStatus = view.findViewById(R.id.tvStatus);  
        TextView tvTemperature = view.findViewById(R.id.tvTemperature);  
        int randNum = (new Random()).nextInt(5);  
        ivWeather.setImageResource(images[randNum]);  
        tvStatus.setText(status[randNum]);  
        tvTemperature.setText(String.format("%d \u2103", 25 + randNum));  
        return view;  
    }  
}
```

WeatherFragment.java



Các bước tích hợp ViewPager (tiếp)

- **Bước 4:** Tạo class adapter cho ViewPager kế thừa từ `FragmentStatePagerAdapter`
 - Tương tự `ListView`, thì `ViewPager` cũng cần một adapter để cung cấp nội dung cho từng trang

```
public class WeatherPagerAdapter extends FragmentStatePagerAdapter {  
    private static final int NUM_PAGES = 5;  
  
    public WeatherPagerAdapter(FragmentManager fm) {  
        super(fm);  
    }  
  
    @Override  
    public Fragment getItem(int position) { // trả về 1 fragment cho 1 trang  
        return new WeatherFragment();  
    }  
  
    @Override  
    public int getCount() { // cho biết số lượng trang  
        return NUM_PAGES;  
    }  
}
```

WeatherPagerAdapter.java



Các bước tích hợp ViewPager (tiếp)

- **Bước 5:** Gán adapter vào ViewPager

```
public class MainActivity extends AppCompatActivity {  
    private ViewPager mPager;  
    private PagerAdapter pagerAdapter;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        mPager = findViewById(R.id.pagerWeather);  
        pagerAdapter = new WeatherPagerAdapter(getSupportFragmentManager());  
        mPager.setAdapter(pagerAdapter);  
    }  
}
```

MainActivity.java

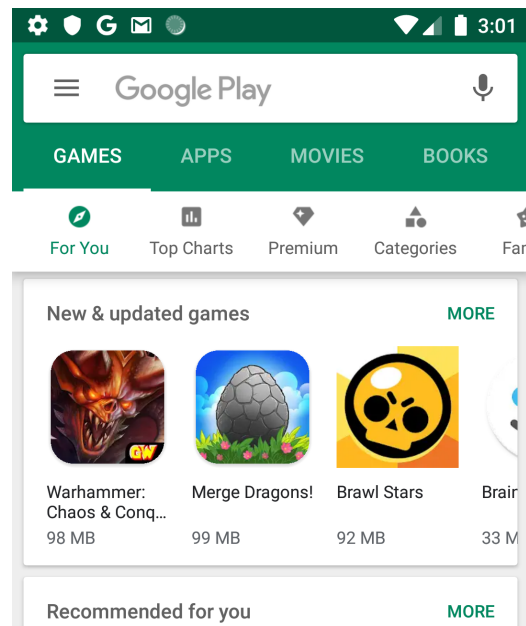


TabLayout

- **TabLayout** là một design được sử dụng khá phổ biến để xây dựng các ứng dụng Android
- Thường được kết hợp với **ViewPager**, theo đó mỗi trang của **ViewPager** tương ứng với nội dung 1 tab của **TabLayout**
- Minh họa sau kết hợp **ViewPager** và **TabLayout** với mỗi tab là một thành phố, mỗi page là thông tin thời tiết của thành phố tương ứng



Sunny
29 °C



Google Play





Minh hoạ TabLayout

- Tiếp tục với minh hoạ ViewPager, các bước tích hợp **TabLayout** cũng cần đảm bảo **bước 1** (cấu hình thư viện support design). **Bước 2**: Thêm TabLayout vào XML layout

```
<?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=".MainActivity">

    <android.support.design.widget.TabLayout
        android:id="@+id/tabWeather"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:tabIndicatorColor="@color/colorPrimary"
        app:tabMode="scrollable" />

    <android.support.v4.view.ViewPager
        android:id="@+id/pagerWeather"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

activity_main.xml



Minh hoạ TabLayout (tiếp)

- **Bước 3:** Tiêu đề cho các tab – override hàm **getPageTitle** cho pager adapter

```
public class WeatherPagerAdapter extends FragmentStatePagerAdapter {
    private static final int NUM_PAGES = 5;
    private String[] cities = {"Hanoi", "Hue", "Danang", "Saigon", "Cantho"};

    public WeatherPagerAdapter(FragmentManager fm) {
        super(fm);
    }

    @Override
    public Fragment getItem(int position) {
        return new WeatherFragment();
    }

    @Override
    public int getCount() {
        return NUM_PAGES;
    }

    @Nullable
    @Override
    public CharSequence getPageTitle(int position) { // → hàm này trả về tiêu đề cho mỗi tab
        return cities[position];
    }
}
```





Minh hoạ TabLayout (tiếp)

- **Bước 4:** Kết nối TabLayout và ViewPager – sử dụng hàm **setupWithViewPager**

```
public class MainActivity extends AppCompatActivity {  
    private ViewPager mPager;  
    private PagerAdapter pagerAdapter;  
    private TabLayout tabLayout;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        mPager = findViewById(R.id.pagerWeather);  
        pagerAdapter = new WeatherPagerAdapter(getSupportFragmentManager());  
        mPager.setAdapter(pagerAdapter);  
        tabLayout = findViewById(R.id.tabWeather);  
        tabLayout.setupWithViewPager(mPager);  
    }  
}
```

MainActivity.java



TabLayout – một số thuộc tính quan trọng

Tên	Giá trị	Mô tả
tabBackground	@drawable/image	Background của tab
tabGravity	center, fill	Gravity của các tab
tabIndicatorColor	@color/blue	Màu của indicator (cho biết tab nào đang được chọn)
tabIndicatorHeight	@dimen/tabh	Chiều cao của indicator
tabMaxWidth	@dimen/tabmaxw	Chiều dài tối đa của 1 tab
tabMode	fixed, scrollable	Các tab fix trong chiều dài của tablayout hay có thể cuộn được
tabTextColor	@color/blue	Màu chữ của tiêu đề tab



Tuỳ chỉnh TabLayout - Thêm icon vào các tab

- **Không** override hàm **getPageTitle** từ **FragmentManager** cho class pager adapter; hoặc override nhưng trả về **null** (return null)
- Sử dụng đối tượng **tabLayout**, duyệt qua từng tab và gán icon tương ứng:

```
public class MainActivity extends AppCompatActivity {  
    // ...
```

```
    private int[] imageResId = {R.drawable.ic_one, R.drawable.ic_two, R.drawable.ic_three,  
    R.drawable.ic_four, R.drawable.ic_five};
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        // ...
```

```
        tabLayout = findViewById(R.id.tabWeather);
```

```
        tabLayout.setupWithViewPager(mPager);
```

```
        for (int i = 0; i < imageResId.length; i++) {
```

```
            tabLayout.getTabAt(i).setIcon(imageResId[i]);
```

```
        }
```

```
    }
```

```
}
```

MainActivity.java



1 2 3 4 5



Sunny

29 °C



TabLayout – Thêm icon + text vào các tab

- Sử dụng **SpannableString**

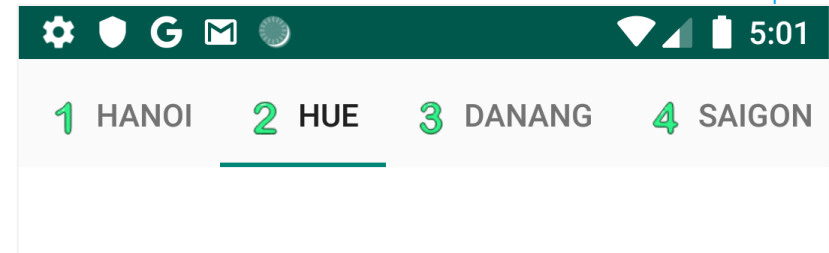
```
public class WeatherPagerAdapter extends FragmentStatePagerAdapter {  
    private Context context;  
    private String[] cities = {"Hanoi", "Hue", "Danang", "Saigon", "Cantho"};  
    private int[] imageResId = {R.drawable.ic_one, R.drawable.ic_two, R.drawable.ic_three,  
R.drawable.ic_four, R.drawable.ic_five};
```

```
    public WeatherPagerAdapter(Context context, FragmentManager fm) {  
        super(fm);  
        this.context = context;  
    }  
}
```

@Nullable

@Override

```
    public CharSequence getPageTitle(int position) {  
        Drawable image = context.getResources().getDrawable(imageResId[position]);  
        image.setBounds(0, 0, image.getIntrinsicWidth(), image.getIntrinsicHeight());  
        SpannableString sb = new SpannableString(" " + cities[position]);  
        ImageSpan imageSpan = new ImageSpan(image, ImageSpan.ALIGN_BOTTOM);  
        sb.setSpan(imageSpan, 0, 1, Spannable.SPAN_EXCLUSIVE_EXCLUSIVE);  
        return sb;  
    }  
}
```



WeatherPagerAdapter.java





TabLayout – Thêm icon + text vào các tab

- Để điều chỉnh thuộc tính **textAllCaps** của các tab, có thể dùng style để thay đổi như sau:

```
<resources>

<style name="MyCustomTabLayout" parent="Widget.Design.TabLayout">
    <item name="tabTextAppearance">@style/MyCustomTextAppearance</item>
</style>

<style name="MyCustomTextAppearance" parent="TextAppearance.Design.Tab">
    <item name="textAllCaps">false</item>
</style>

</resources>
```

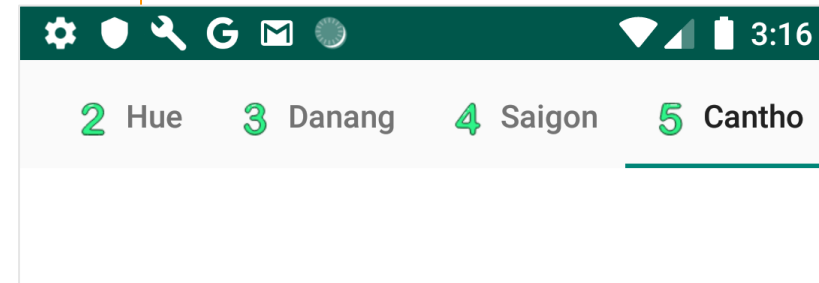
res/values/styles.xml

```
<LinearLayout ...>

<android.support.design.widget.TabLayout
    android:id="@+id/tabWeather"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:tabIndicatorColor="@color/colorPrimary"
    style="@style/MyCustomTabLayout"
    app:tabMode="scrollable" />

</LinearLayout>
```

activity_main.xml





TabLayout – Tạo custom view cho các tab

- Có thể áp dụng 1 custom XML layout cho các tab, bằng cách sử dụng hàm **setCustomView(...)** tương ứng với mỗi tab của TabLayout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="vertical">

    <ImageView
        android:id="@+id/ivTabIcon"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/tvTabTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

layout_tab_item.xml





TabLayout – Tạo custom view cho các tab (tiếp)

- Viết phương thức tạo custom view cho từng tab (getTabView) ở pager adapter

```
public class WeatherPagerAdapter extends FragmentStatePagerAdapter {  
    private Context context;  
    private String[] cities = {"Hanoi", "Hue", "Danang", "Saigon", "Cantho"};  
    private int[] imageResId = {R.drawable.ic_one, R.drawable.ic_two, R.drawable.ic_three,  
    R.drawable.ic_four, R.drawable.ic_five};  
  
    public View getTabView(int position) {  
        View v = LayoutInflater.from(context).inflate(R.layout.layout_tab_item, null);  
        TextView tvTitle = v.findViewById(R.id.tvTabTitle);  
        tvTitle.setText(cities[position]);  
        ImageView ivIcon = v.findViewById(R.id.ivTabIcon);  
        ivIcon.setImageResource(imageResId[position]);  
        return v;  
    }  
}
```

WeatherPagerAdapter

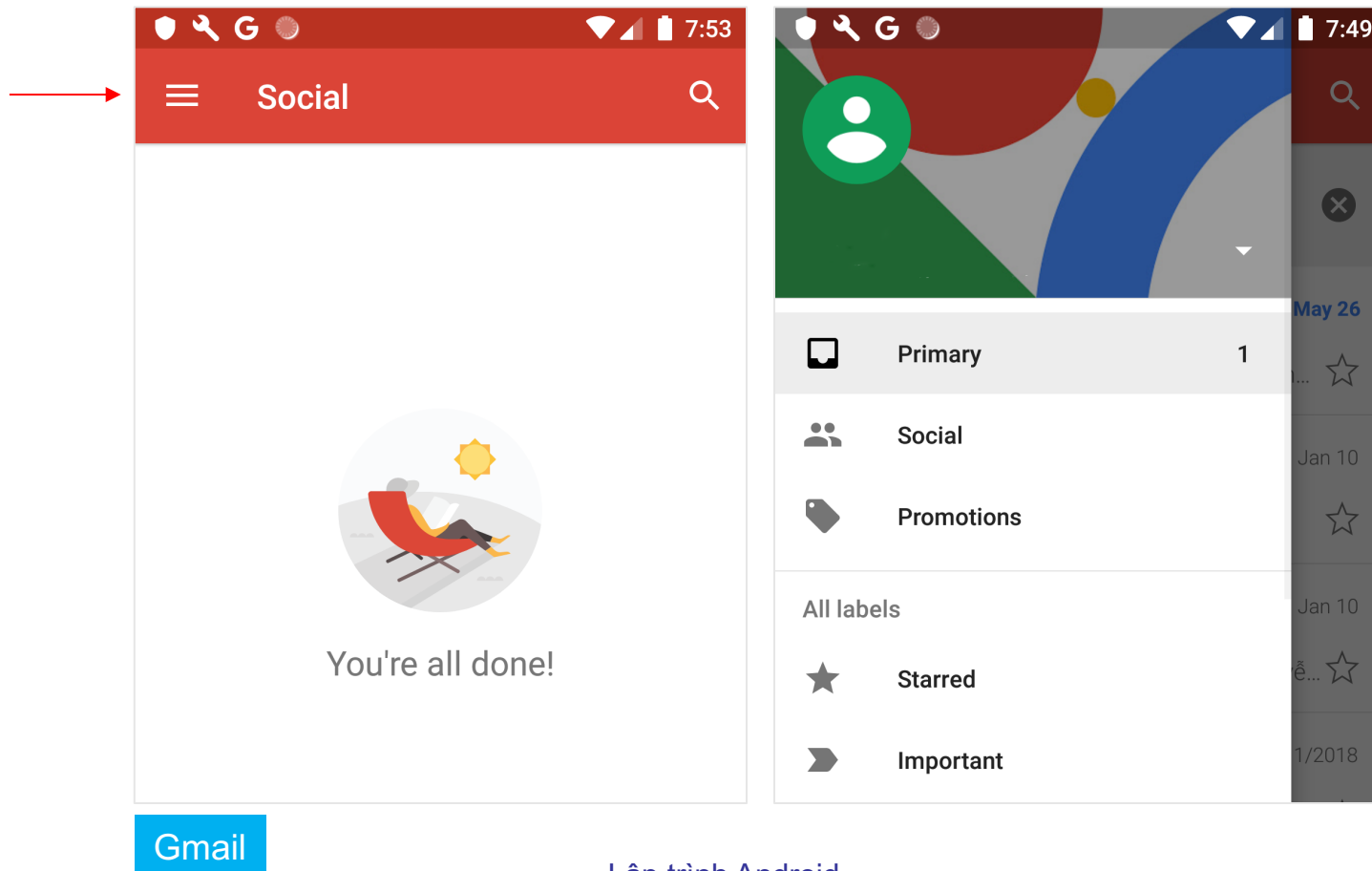
```
public class MainActivity extends AppCompatActivity {  
    private WeatherPagerAdapter pagerAdapter;  
    private TabLayout tabLayout;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        // ...  
        for (int i = 0; i < tabLayout.getTabCount(); i++) {  
            tabLayout.getTabAt(i).setCustomView(pagerAdapter.getTabView(i));  
        }  
    }  
}
```

MainActivity



Navigation Drawer

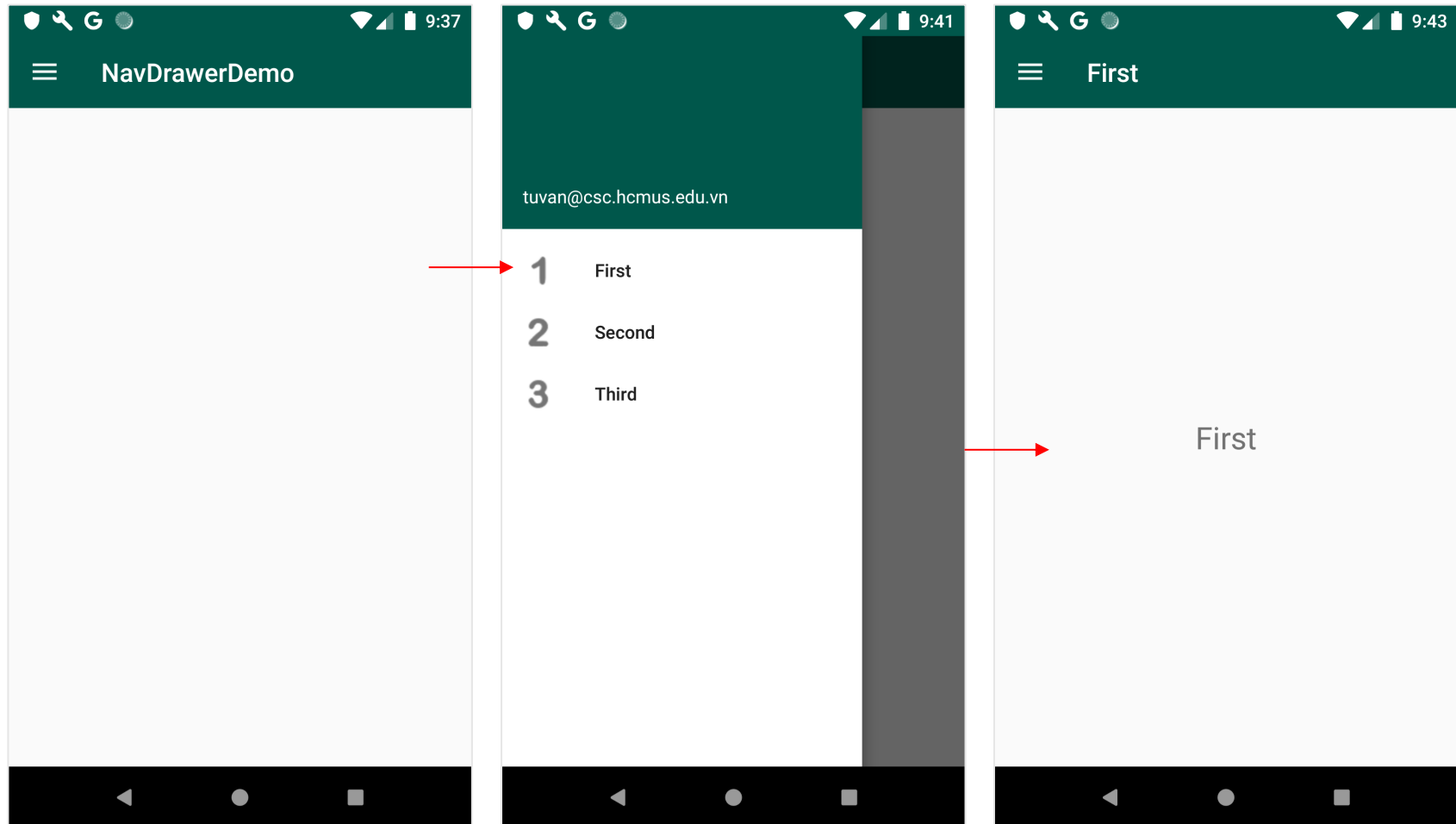
- Navigation Drawer là một kiểu giao diện cho phép trình bày một menu và giúp điều hướng trong ứng dụng
- Người dùng nhấn vào drawer icon hoặc dùng thao tác vuốt (swipe) để mở menu này





Minh hoạ Navigation Drawer

- Minh hoạ sau xây dựng một ứng dụng có sử dụng navigation drawer. Với mỗi menu item, khi được chọn, màn hình chính sẽ thay đổi nội dung tương ứng. Mỗi nội dung được quản lý bởi một fragment.





Minh hoạ Navigation Drawer (tiếp)

- **Bước 1:** Cấu hình **build.gradle** sử dụng thư viện **support design**
implementation 'com.android.support:design:28.0.0'
- **Bước 2:** Tạo **Toolbar** layout

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.Toolbar xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?attr/colorPrimaryDark"
    android:fitsSystemWindows="true"
    android:minHeight="?attr/actionBarSize"
    app:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar" />
```

res/layout/toolbar.xml



Minh hoạ Navigation Drawer (tiếp)

○ Bước 3: Tạo XML layout cho màn hình ứng dụng

activity_main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
  <android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/layoutDrawer"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
2 <LinearLayout
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical">
```

```
a <include
  layout="@layout/toolbar"
  android:layout_width="match_parent"
  android:layout_height="wrap_content" />
```

```
b <FrameLayout
  android:id="@+id/container"
  android:layout_width="match_parent"
  android:layout_height="match_parent" />
</LinearLayout>
```

- (1) DrawerLayout là node gốc, gồm 2 node con
- (2) Node con 1: Layout chứa nội dung chính của màn hình
 - a) Toolbar
 - b) Layout chứa các fragment
- (3) Node con 2: Layout chứa menu điều hướng xuất hiện từ bên trái màn hình

```
3 <android.support.design.widget.NavigationView
  android:id="@+id/leftView"
  android:layout_width="wrap_content"
  android:layout_height="match_parent"
  android:layout_gravity="start"
  android:background="@android:color/white"
  app:headerLayout="@layout/nav_header"
  app:menu="@menu/main_menu" />
</android.support.v4.widget.DrawerLayout>
```





Minh hoạ Navigation Drawer (tiếp)

- **Bước 4:** Tạo XML layout cho menu

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <group android:checkableBehavior="single">
    <item
      android:id="@+id/nav_first"
      android:icon="@drawable/ic_one"
      android:title="@string/first" />
    <item
      android:id="@+id/nav_second"
      android:icon="@drawable/ic_two"
      android:title="@string/second" />
    <item
      android:id="@+id/nav_third"
      android:icon="@drawable/ic_three"
      android:title="@string/third" />
  </group>
</menu>
```

res/menu/main_menu.xml

○ Bước 5: Thiết lập toolbar, hamburger icon



1 import android.support.v7.app.ActionBarDrawerToggle;

```
public class MainActivity extends AppCompatActivity {  
    private DrawerLayout drawer;  
    private Toolbar toolbar;  
    private NavigationView leftView;  
    private ActionBarDrawerToggle hamburger;
```

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    toolbar = findViewById(R.id.toolbar);  
    setSupportActionBar(toolbar);  
    drawer = findViewById(R.id.layoutDrawer);  
    leftView = findViewById(R.id.leftView);  
    hamburger = new ActionBarDrawerToggle(this, drawer, toolbar, R.string.drawer_open,  
R.string.drawer_close);  
    drawer.addDrawerListener(hamburger);  
}
```

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    hamburger.syncState();  
}
```

- (1) Lưu ý sử dụng thư viện support v7 thay vì v4
- (2) Thiết lập hamburger icon
- (3) Cần override phương thức này để xuất hiện hamburger icon

MainActivity





Minh hoạ Navigation Drawer (tiếp)

- **Bước 6:** Thiết lập left view (menu view), xử lý sự kiện chọn menu item:

```
public class MainActivity extends AppCompatActivity {  
    // ...  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        //...  
        setupDrawerContent(leftView);  
    }  
  
    private void setupDrawerContent(NavigationView navigationView) {  
        navigationView.setNavigationItemSelectedListener(  
            new NavigationView.OnNavigationItemSelectedListener() {  
                @Override  
                public boolean onNavigationItemSelected(MenuItem menuItem) {  
                    selectDrawerItem(menuItem);  
                    return true;  
                }  
            });  
    }  
}
```

MainActivity



- **Bước 6:** (tiếp) Tùy theo menu item được chọn, **thay thế fragment** tương ứng vào layout nội dung chính (R.id.container)

```
public class MainActivity extends AppCompatActivity {  
    // ...  
    public void selectDrawerItem(MenuItem menuItem) {  
        Fragment fragment = null;  
        Class fragmentClass;  
        switch(menuItem.getItemId()) {  
            case R.id.nav_first:  
                fragmentClass = FirstFragment.class;  
                break;  
            // ... thêm xử lý cho các menu item khác  
        }  
  
        try {  
            fragment = (Fragment) fragmentClass.newInstance();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
  
        FragmentManager fragmentManager = getSupportFragmentManager();  
        fragmentManager.beginTransaction().replace(R.id.container, fragment).commit();  
        menuItem.setChecked(true); // đánh dấu, highlight menu item được chọn  
        setTitle(menuItem.getTitle()); // đổi tiêu đề màn hình trên toolbar  
        drawer.closeDrawers(); // đóng left view  
    }  
}
```

MainActivity



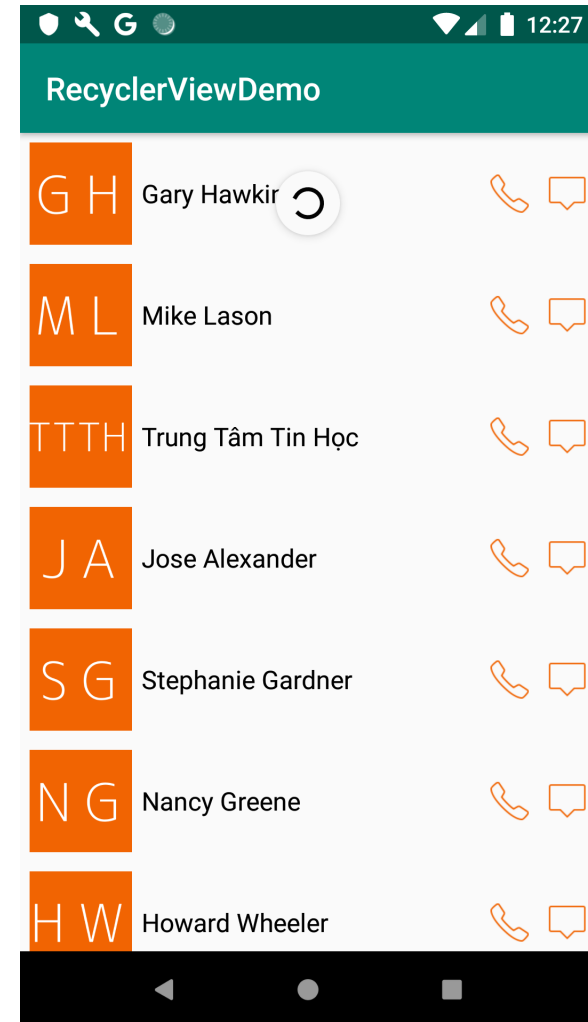
SwipeRefreshLayout

- **SwipeRefreshLayout** cung cấp trải nghiệm “pull to refresh” trên ứng dụng Android
- Được cung cấp theo gói thư viện design support v4
implementation 'com.android.support:support-v4:28.0.0'
- Thường được sử dụng kết hợp với ListView hoặc RecyclerView

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.SwipeRefreshLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/swipeView"
    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">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/rvContacts"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</android.support.v4.widget.SwipeRefreshLayout>
```

activity_main.xml





SwipeRefreshLayout (tiếp)

- Xử lý sự kiện “pull to refresh”:

```
public class MainActivity extends AppCompatActivity implements
    SwipeRefreshLayout.OnRefreshListener { // MainActivity đóng vai trò OnRefreshListener
    SwipeRefreshLayout swipeView;
    // ...
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // ...
        swipeView = findViewById(R.id.swipeView);
        swipeView.setOnRefreshListener(this); // gán đối tượng lắng nghe sự kiện pull to refresh
    }

    @Override
    public void onRefresh() { // override phương thức onRefresh()
        // do a slow task
        swipeView.setRefreshing(false); // ẩn SwipeRefreshLayout sau khi hoàn tất cập nhật dữ liệu
    }
}
```

MainActivity

