

BÀI 6. PHIẾU BÀI TẬP

BÀI TẬP TRƯỚC KHI LÊN LỚP: Bài 6.1 và 6.3

BÀI TẬP THỰC HIỆN TRÊN LỚP: Bài 6.2 và 6.4

BÀI TẬP SAU KHI LÊN LỚP: Bài 6.5 và bài 6.6

MỤC LỤC

Bài tập 6.1 . Tạo ứng dụng menu Options.....	1
Bài tập 6.2. Bài tập thực hành Context menu.....	5
Bài tập 6.3. Bài tập thực hành dung Intent mở một Intent khác.....	9
Bài tập 6.4. Bài tập thực hành dung Bundle để truyền dữ liệu giữa 2 Intent	14
Bài tập 6.5. Bài thực hành mở một Activity truyền dữ liệu qua và xử lý kết quả trả về	20
Bài tập 6.6: Bài tập tổng hợp Quản Lý Nhân Viên	25

Bài tập 6.1 . Tạo ứng dụng menu Options

Trong bài này các em sẽ tạo một ứng dụng đơn giản gồm menu Options bao gồm các menu :

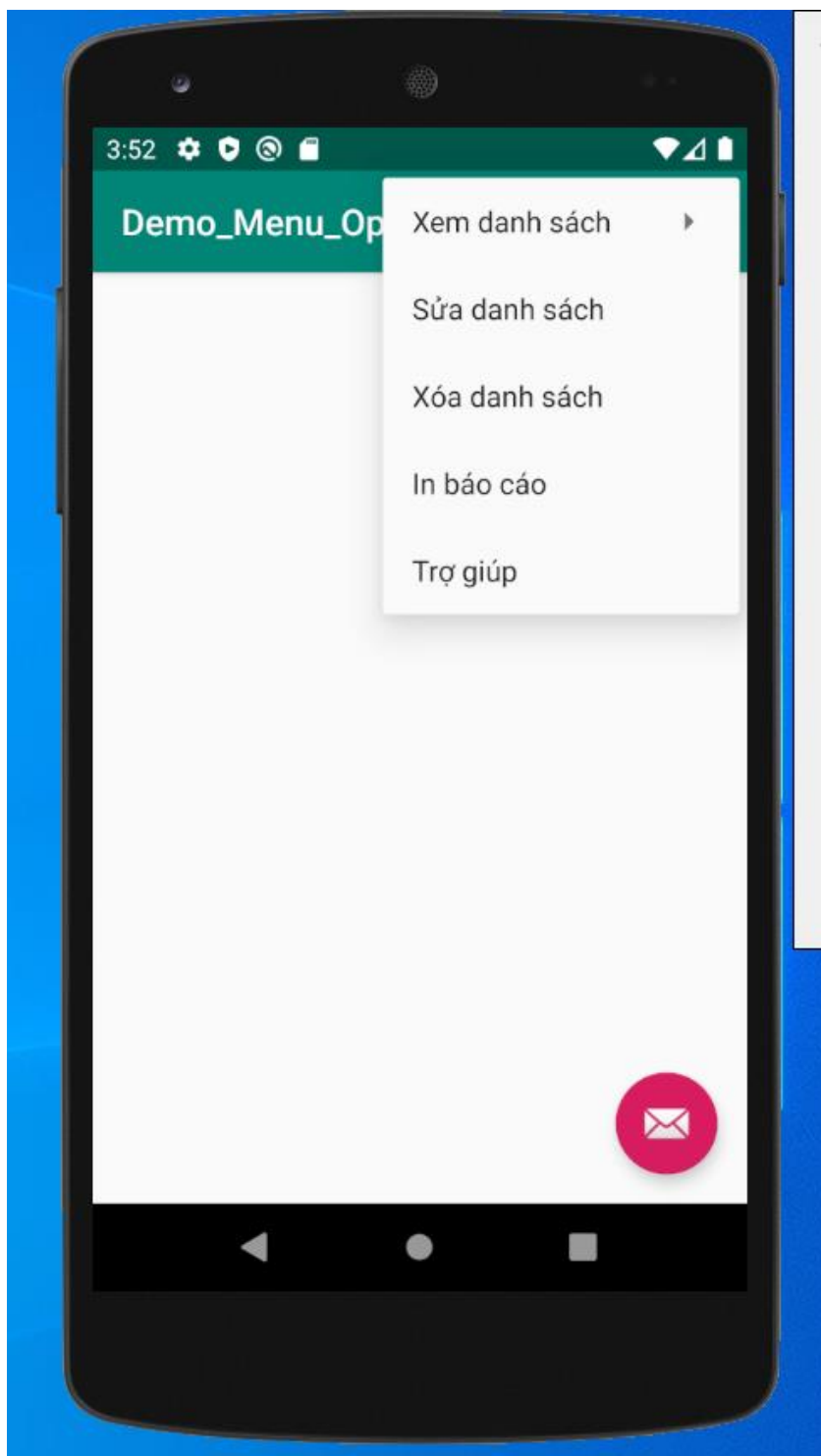
- Xem danh sách
 - Lớp học
 - Sinh viên
- Sửa danh sách
- Xóa danh sách
- In báo cáo
- Trợ giúp

Như vậy chúng ta có một menu Option gồm 5 item đồng cấp là : Xem danh sách, sửa danh sách, xóa danh sách, In báo cáo và trợ giúp. Trong menu xem danh sách có 2 menu item con là : Lớp học và sinh viên.

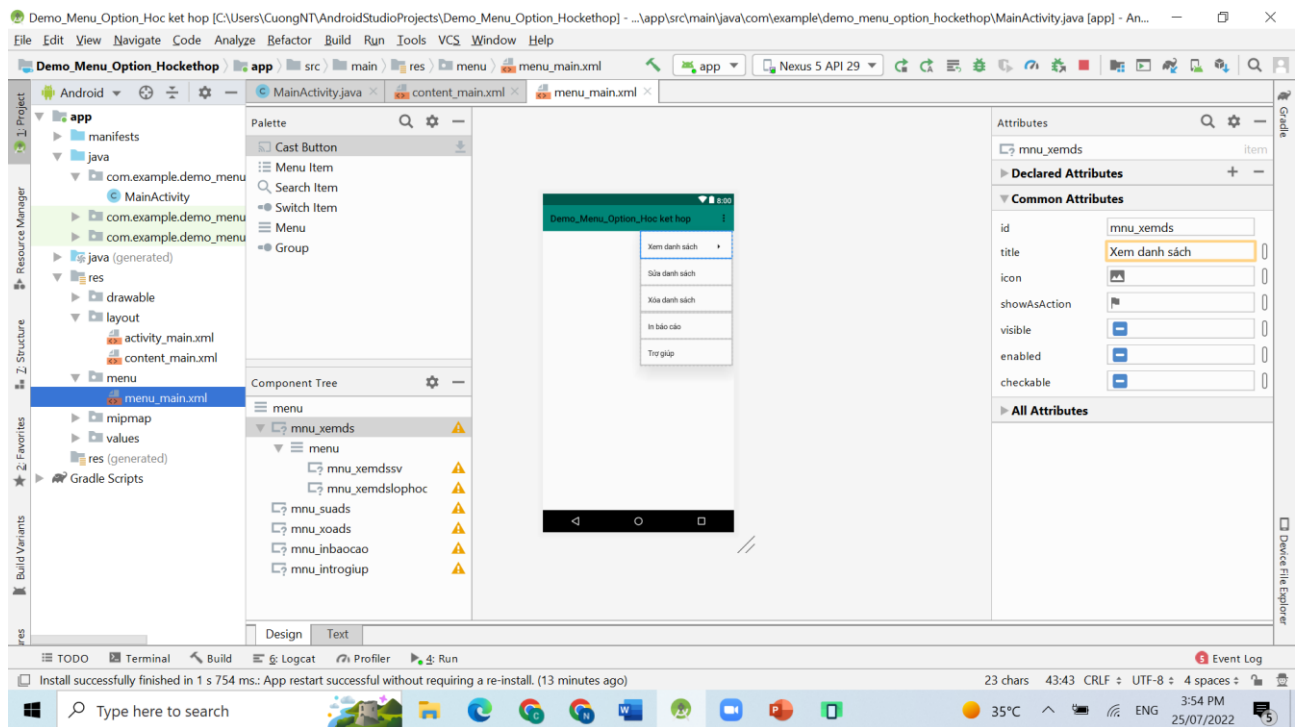
Để hoàn thiện bài thực hành chúng ta cần làm 2 bước:

Bước 1 thiết kế giao diện menu trên file XML

Bước 2: Ánh xạ và điều khiển bằng lập trình java trong file MainActivity.java



Bước 1: Chúng ta vào menu\menu_main.xml



Sau đó chúng ta thêm code thiết kế menu trên XML

```
<menu 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"
```

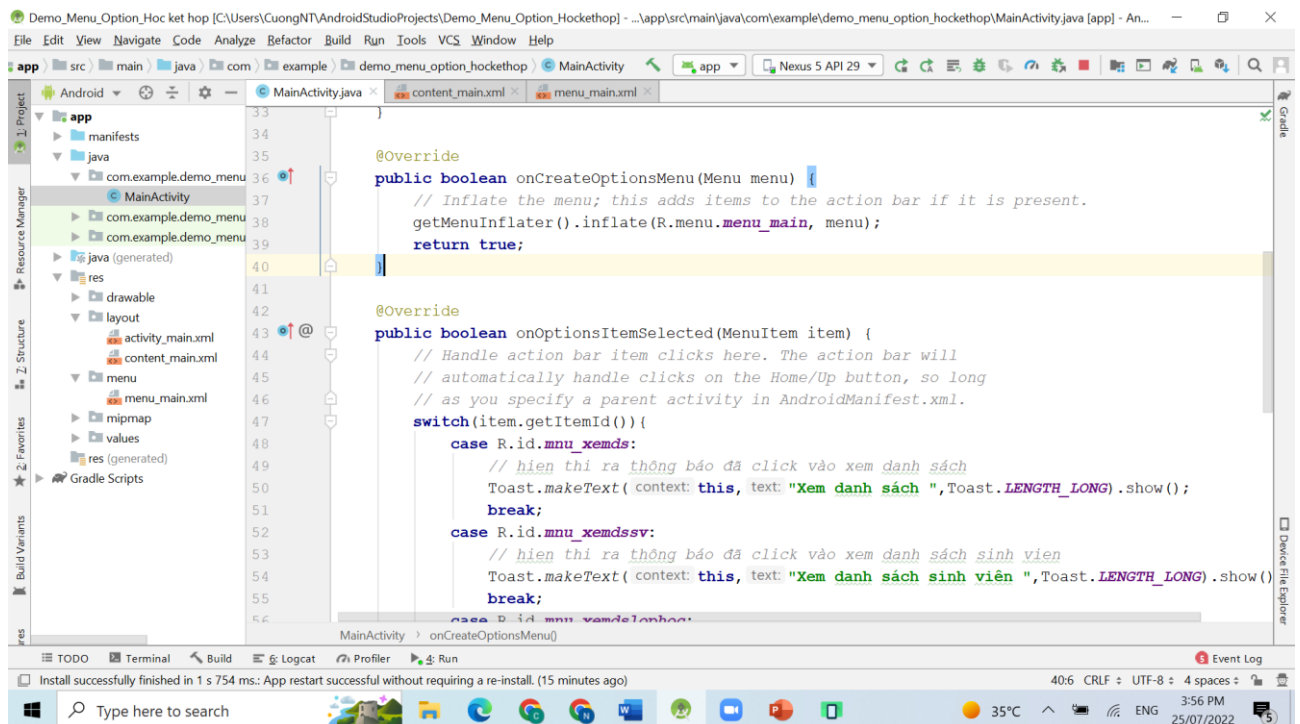
```
    tools:context="com.example.demo_menu_option_hockethop.MainActivity"
>
```

```
<item
    android:id="@+id/mnu_xemds"
    android:title="Xem danh sách">
    <!-- khai báo menu con -->
    <menu>
        <item
            android:id="@+id/mnu_xemdssv"
            android:title="Sinh viên" />
        <item
            android:id="@+id/mnu_xemdslophoc"
            android:title="Lớp học" />
    </menu>
    </item>
    <item
        android:id="@+id/mnu_suads"
        android:title="Sửa danh sách">
    </item>
    <item
        android:id="@+id/mnu_xoads"
        android:title="Xóa danh sách"/>
    <item
        android:id="@+id/mnu_inbaocao"
        android:title="In báo cáo"/>
    <item
        android:id="@+id/mnu_introgiup"
```

```
android:title="Trợ giúp"/>
```

```
</menu>
```

Bước 2 : Tạo menu bên MainActivity. Java và điều khiển menu tạo ra



Trong bài thực hành này các em chú ý đến 2 phương thức là :

`onCreateOptionsMenu(Menu menu)` dùng để tạo menu và phương thức `onOptionsItemSelected(MenuItem item)` dùng để điều khiển menu.

Trong phương thức điều khiển menu, chúng ta truyền vào tham số `MenuItem` và dùng `item.getItemId` để sử dụng cấu trúc switch điều khiển

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    switch (item.getItemId()) {
        case R.id.mnu_xemds:
            // hiện thi ra thông báo đã click vào xem danh sách
            Toast.makeText(this, "Xem danh sách",
                Toast.LENGTH_LONG).show();
            break;
        case R.id.mnu_xemdssv:
            // hiện thi ra thông báo đã click vào xem danh sách
            sinh vien
            Toast.makeText(this, "Xem danh sách sinh viên",
                Toast.LENGTH_LONG).show();
            break;
        case R.id.mnu_xemdslophoc:
            // hiện thi ra thông báo đã click vào xem danh sách
            lớp học
```

Học kết hợp

```
Toast.makeText(this, "Xem danh sách lớp học", Toast.LENGTH_LONG).show();
break;
case R.id.mnu_suads:
    // hiển thị ra thông báo đã click vào sửa danh sách
    Toast.makeText(this, "Sửa danh sách", Toast.LENGTH_LONG).show();
    break;
case R.id.mnu_xoads:
    // hiển thị ra thông báo đã click vào xóa danh sách
    Toast.makeText(this, "Xóa danh sách", Toast.LENGTH_LONG).show();
    break;
case R.id.mnu_inbaocao:
    // hiển thị ra thông báo đã click vào in báo cáo
    Toast.makeText(this, "In báo cáo", Toast.LENGTH_LONG).show();
    break;
case R.id.mnu_introgiup:
    // hiển thị ra thông báo đã click vào Trợ giúp
    Toast.makeText(this, "Trợ giúp", Toast.LENGTH_LONG).show();
    break;
}
return super.onOptionsItemSelected(item);
}
```

Các em chú ý việc sử dụng break ở mỗi trường hợp case nhằm đảm bảo chương trình chạy đúng logic.

Bài tập 6.2. Bài tập thực hành Context menu

Trong bài này các em sẽ tạo một ứng dụng đơn giản gồm Context menu bao gồm các menu :

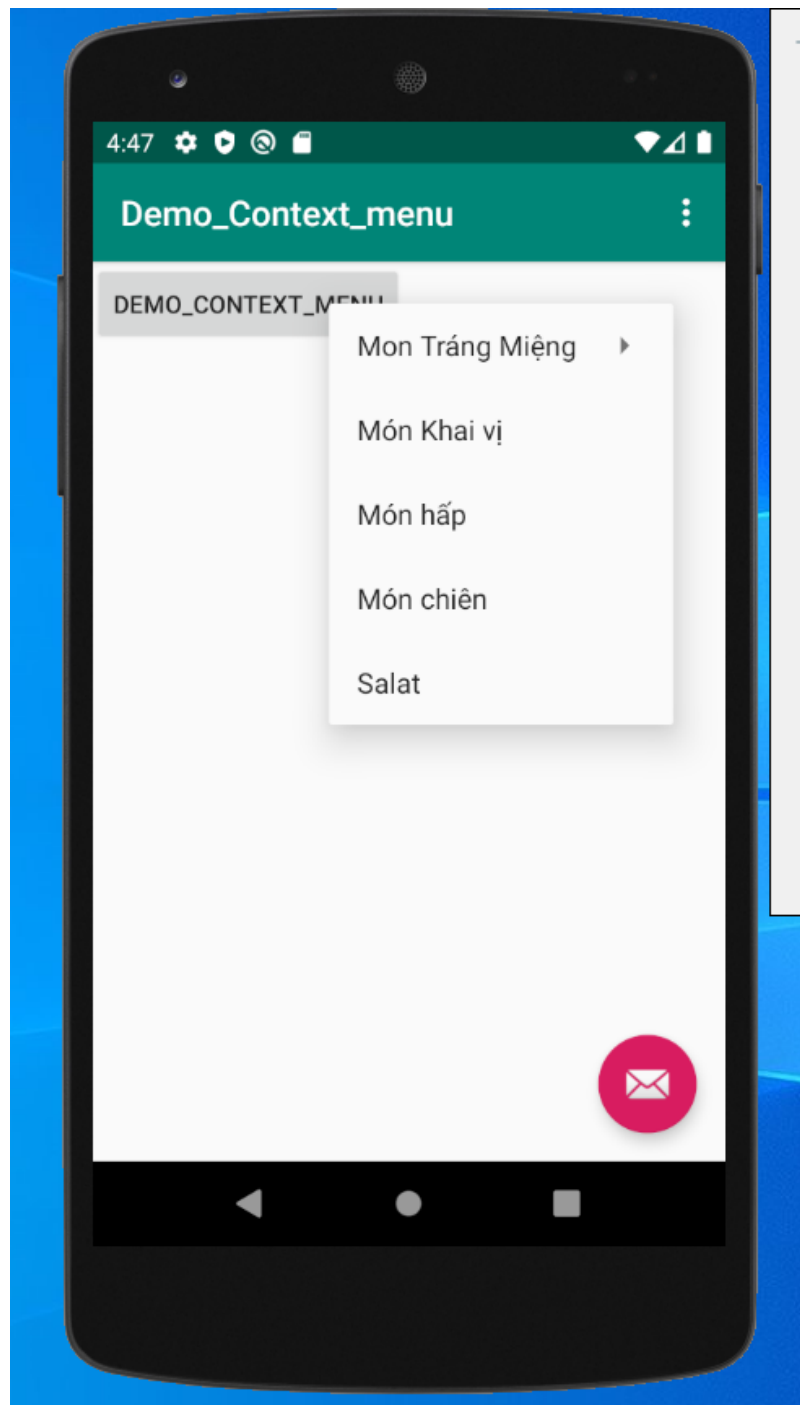
- Món Tráng Miệng
 - Kem Tráng Tiền
 - Nước ép hoa quả theo mùa
- Món Khai vị
- Món Hấp
- Món Chiên
- Salat

Như vậy chúng ta có một menu Option gồm 5 item đồng cấp là : Món Tráng miệng, Món Khai Vị, Món Hấp, Món Chiên và Salat. Trong menu Món Tráng Miệng có 2 menu item con là : Kem Tráng tiền và Nước ép hoa quả theo mùa

Để hoàn thiện bài thực hành chúng ta cần làm 2 bước:

Bước 1 thiết kế giao diện menu trên file XML

Bước 2: Ánh xạ và điều khiển bằng lập trình java trong file MainActivity.java



Bước 1: Chúng ta vào menu\menu_main.xml để thiết kế nội dung menu

```
<menu 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"
      tools:context="com.example.demo_context_menu.MainActivity">
    <item
        android:id="@+id/mnu_xemds"
        android:title="Món Tráng Miệng">
        <!-- khai bao menu con -->
        <menu>
```

```
<item
    android:id="@+id/mnu_xemdsKem"
    android:title="Kem Tràng Tiền" />
<item
    android:id="@+id/mnu_xemdsnuocep"
    android:title="Nước ép hoa quả theo mùa" />
</menu>
</item>
<item
    android:id="@+id/mnu_Khaivi"
    android:title="Món Khai vị">
</item>
<item
    android:id="@+id/mnu_Hấp"
    android:title="Món hấp"/>
<item
    android:id="@+id/mnu_Chien"
    android:title="Món chiên"/>
<item
    android:id="@+id/mnu_Salat"
    android:title="Salat"/>
</menu>
```

Bước 2: Thêm nút ContextMenu vào MainActivity.java và điều khiển

a. Thêm nút lệnh

```
<?xml version="1.0" encoding="utf-8"?>
<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"

    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context=".MainActivity"
    tools:showIn="@layout/activity_main">

    <Button
        android:id="@+id/btnContextMenu"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Demo_Context_Menu"
        tools:layout_editor_absoluteX="115dp"
        tools:layout_editor_absoluteY="82dp" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

b. Gán điều khiển context menu cho nút lệnh

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

```
// khai báo đăng ký btnContextMenu
Button
btn_Context_menu=findViewById(R.id.btnContextMenu) ;
// khai báo menu ngữ cảnh ( context menu)
registerForContextMenu(btn_Context_menu);

FloatingActionButton fab = findViewById(R.id.fab) ;
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Snackbar.make(view, "Replace with your own
action", Snackbar.LENGTH_LONG)
            .setAction("Action", null).show();
    }
});
}
```

c. Xây dựng hàm onCreateContextMenu

```
public void onCreateContextMenu(ContextMenu menu,View
v,ContextMenu.ContextMenuInfo menuInfo){
    super.onCreateContextMenu(menu,v,menuInfo);
    getMenuInflater().inflate(R.menu.menu_main,menu);
}
```

d. Xây dựng hàm onOptionsItemSelected

```
public boolean onOptionsItemSelected(MenuItem item ){
    doSomething(item);
    return super.onOptionsItemSelected(item);
}
```

Trong đó hàm thực thi doSomething được viết riêng ngoài với mục đích tách code, dễ bảo trì

```
// code dùng để điều khiển menu
public void doSomething(MenuItem item){
    switch(item.getItemId()){
        case R.id.mnu_xemds:
            // hiện thị ra thông báo đã click vào xem danh
            sách
            Toast.makeText(this, "Món tráng miệng
",Toast.LENGTH_LONG).show();
            break;
        case R.id.mnu_xemdsKem:
            // hiện thị ra thông báo đã click vào xem danh
            sách sinh viên
            Toast.makeText(this, "Kem Tráng tiên
",Toast.LENGTH_LONG).show();
            break;
        case R.id.mnu_xemdsnuocep:
            // hiện thị ra thông báo đã click vào xem danh
            sách lớp học
            Toast.makeText(this, "Nước ép theo mùa
",Toast.LENGTH_LONG).show();
            break;
        case R.id.mnu_Hấp:
```



```
// hien thi ra thông báo đã click vào sửa danh sách
Toast.makeText(this, "Món hấp", Toast.LENGTH_LONG).show();
break;
case R.id.mnu_Chien:
    // hien thi ra thông báo đã click vào xóa danh sách
    Toast.makeText(this, "Món chiên", Toast.LENGTH_LONG).show();
    break;
case R.id.mnu_KhaiVi:
    // hien thi ra thông báo đã click vào in báo cáo
    Toast.makeText(this, "Món khai vị", Toast.LENGTH_LONG).show();
    break;
case R.id.mnu_Salat:
    // hien thi ra thông báo đã click vào Trợ giúp
    Toast.makeText(this, "Salat", Toast.LENGTH_LONG).show();
    break;
}
}
```

Bài tập 6.3. Bài tập thực hành dung Intent mở một Intent khác

- Intent có rất nhiều tính năng từ việc dùng để mở các Activity và truyền dữ liệu qua lại giữa chúng đến dùng Intent ở những mục đích khác nhau. Về mặt tổng quát Intent được chia ra làm 2 công dụng chính:

- 1) Dùng **Intent** để mở một Activity khác nhưng không kiểm soát kết quả trả về.
- 2) Dùng **Intent** để mở một Activity khác và có kiểm soát kết quả trả về.

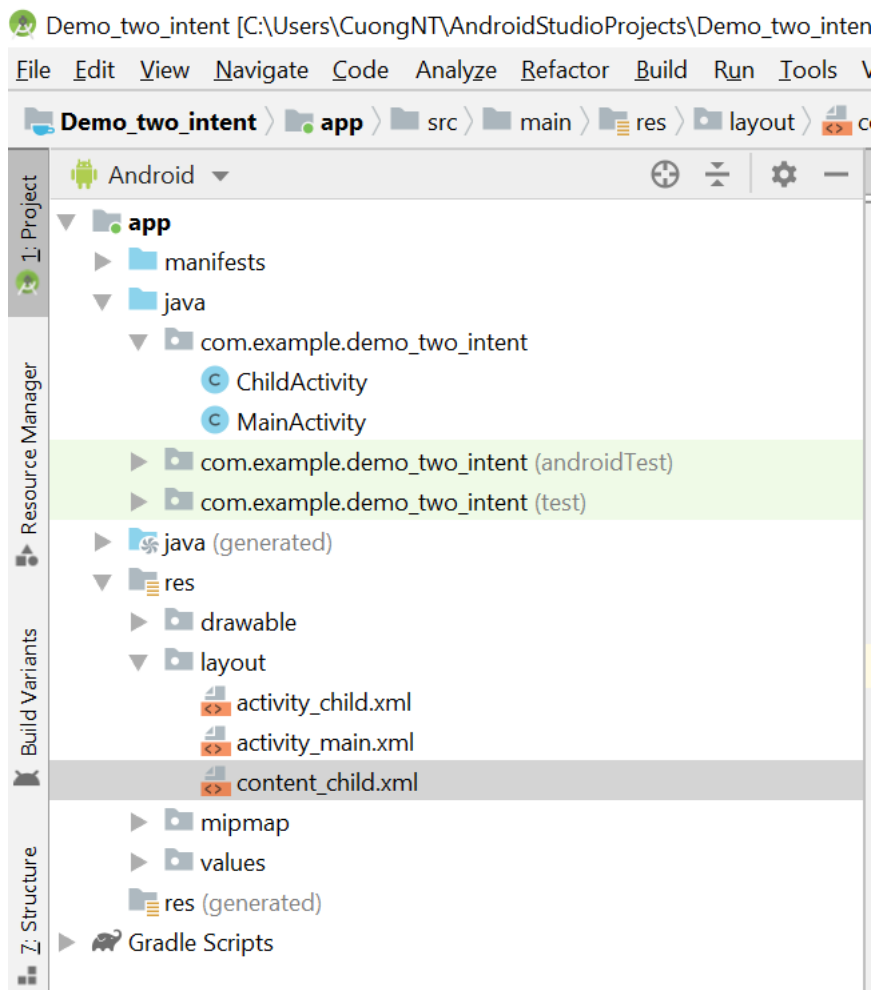
Sau đây là những ví dụ cụ thể: **Đơn thuần là mở một Activity khác.**



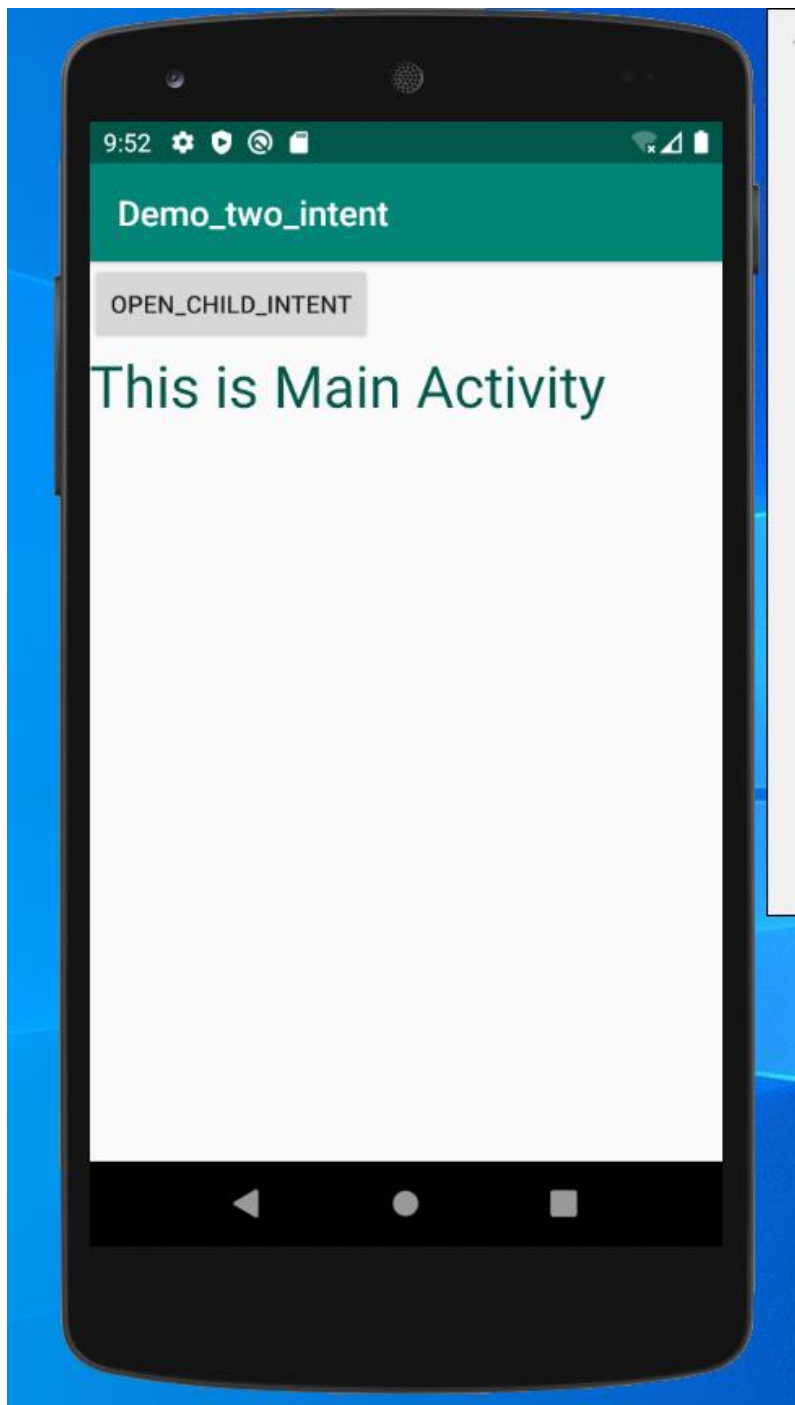
Giải thích hình trên như sau:

- Từ Main Activity có 1 Button, khi nhấn vào Button này thì sẽ mở một Activity khác, tại Activity mới mở này sẽ có 1 Button cho phép quay trở lại MainActivity.

- Tạo một Android Project có cấu trúc như hình bên dưới:



- Xây dựng giao diện của MainActivity



- Code thiết kế trong XML:

```
<?xml version="1.0" encoding="utf-8"?>
<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">

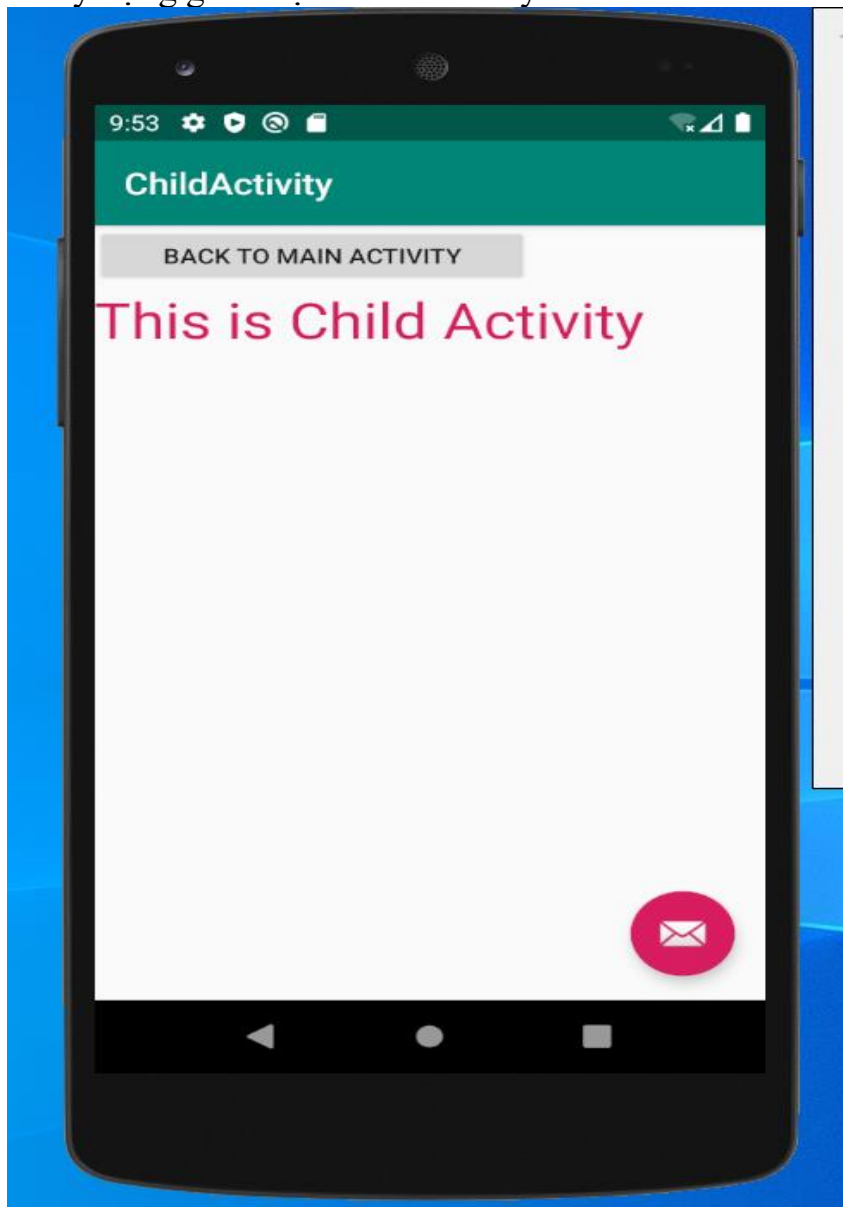
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
```

```
<Button
    android:id="@+id/btnOpenChildIntent"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Open_Child_Intent" />

<TextView
    android:id="@+id/txtVMain"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="This is Main Activity"
    android:textColor="@color/colorPrimaryDark"
    android:textSize="33dp" />
</LinearLayout>
```

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

- Xây dựng giao diện ChildActivity



- Code giao diện ChildActivity trong XML

```
<?xml version="1.0" encoding="utf-8"?>
<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"
app:layout_behavior="@string/appbar_scrolling_view_behavior"
tools:context=".ChildActivity"
tools:showIn="@layout/activity_child">

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

        <Button
            android:id="@+id/btnOpenMainActivity"
            android:layout_width="234dp"
            android:layout_height="40dp"
            android:text="Back to Main Activity"
            />

        <TextView
            android:id="@+id/txtvChild"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="This is Child Activity"
            android:textColor="@color/colorAccent"
            android:textSize="33dp" />
    </LinearLayout>

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

- Để thực thi gọi và chuyển Intent giữa 2 Activity chúng ta phải khai báo 2 button và dùng sự kiện onClick để liên kết
- Thực thi code bên MainActivity.java

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    // khai bao btnOpenChildActivity
    Button
    btnOpenChildActivity=findViewById(R.id.btnOpenChildIntent);
    // gán sự kiện
    btnOpenChildActivity.setOnClickListener(new
    View.OnClickListener() {
        @Override
        // code thực thi sự kiện
        public void onClick(View v) {
            Intent myIntent=new
            Intent(MainActivity.this,ChildActivity.class);
            startActivity(myIntent);
        }
    })
}
```

```
});
```

```
}
```

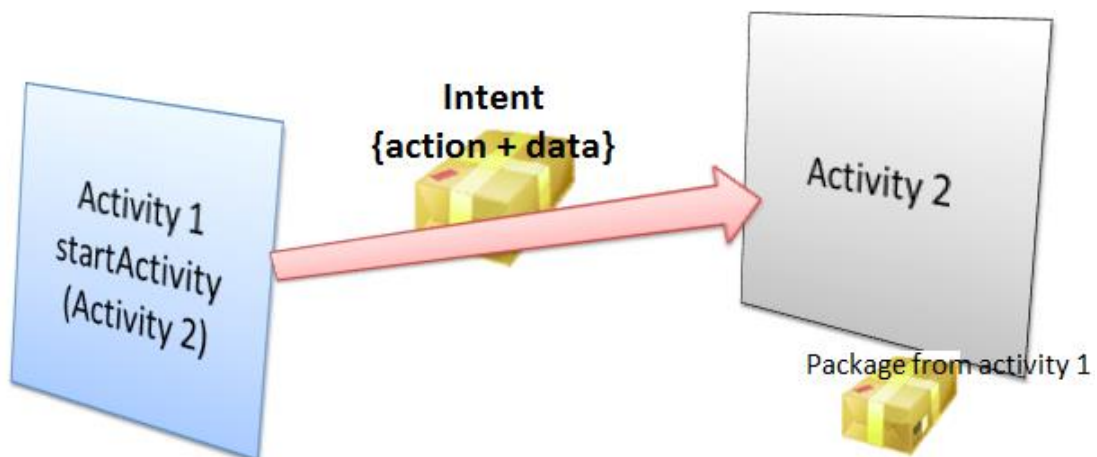
- Thực thi code bên MainActivity.java

```
Button  
btnOpenMainActivity=findViewById(R.id.btnOpenMainActivity);  
btnOpenMainActivity.setOnClickListener(new  
View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent mainIntent=new  
Intent(MainActivity.this, MainActivity.class);  
        startActivity(mainIntent);  
    }  
});
```

Chú ý: Câu lệnh dùng để gọi và chạy Intent thứ 2.

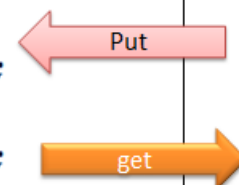
```
Intent myIntent=new  
Intent(MainActivity.this, MainActivity.class);  
startActivity(myIntent)
```

Bài tập 6.4. Bài tập thực hành dung Bundle để truyền dữ liệu giữa 2 Intent



- Ở đây Ta thường dùng đối tượng **Bundle** để đóng gói dữ liệu để truyền tải qua các Activity khác.
- Trong Android khi truyền dữ liệu người dùng cũng có thể truyền từng thông số riêng lẻ nhưng nên thực hiện đóng gói dữ liệu vào 1 **Bundle** rồi gửi **Bundle** này qua Activity nào đó. Bên kia chỉ cần lấy Bundle này và mọi dữ liệu đều nằm trong Bundle nên dễ dàng xử lý. (như túi đựng dữ liệu)

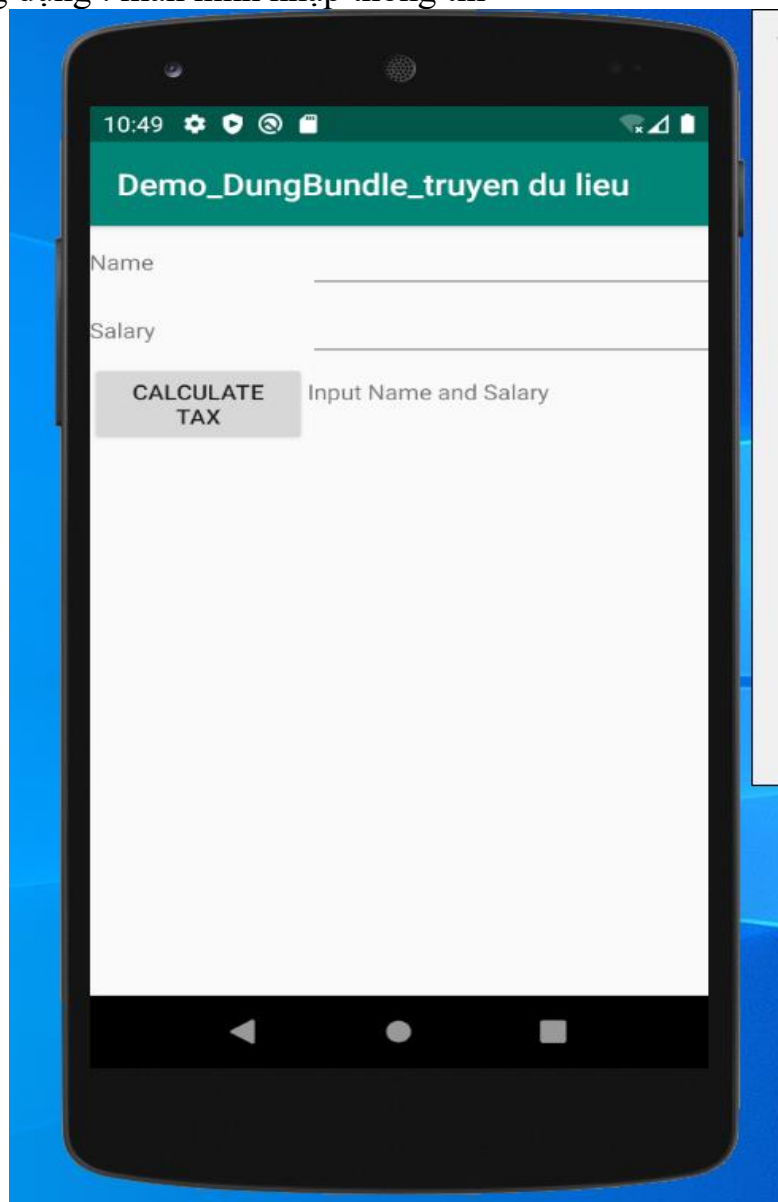
```
Bundle bundle=new Bundle();  
bundle.putInt("soa", 5);  
bundle.putDouble("sob", 5.7);  
//...  
int soa=bundle.getInt("soa");  
double sob=bundle.getDouble("sob");
```



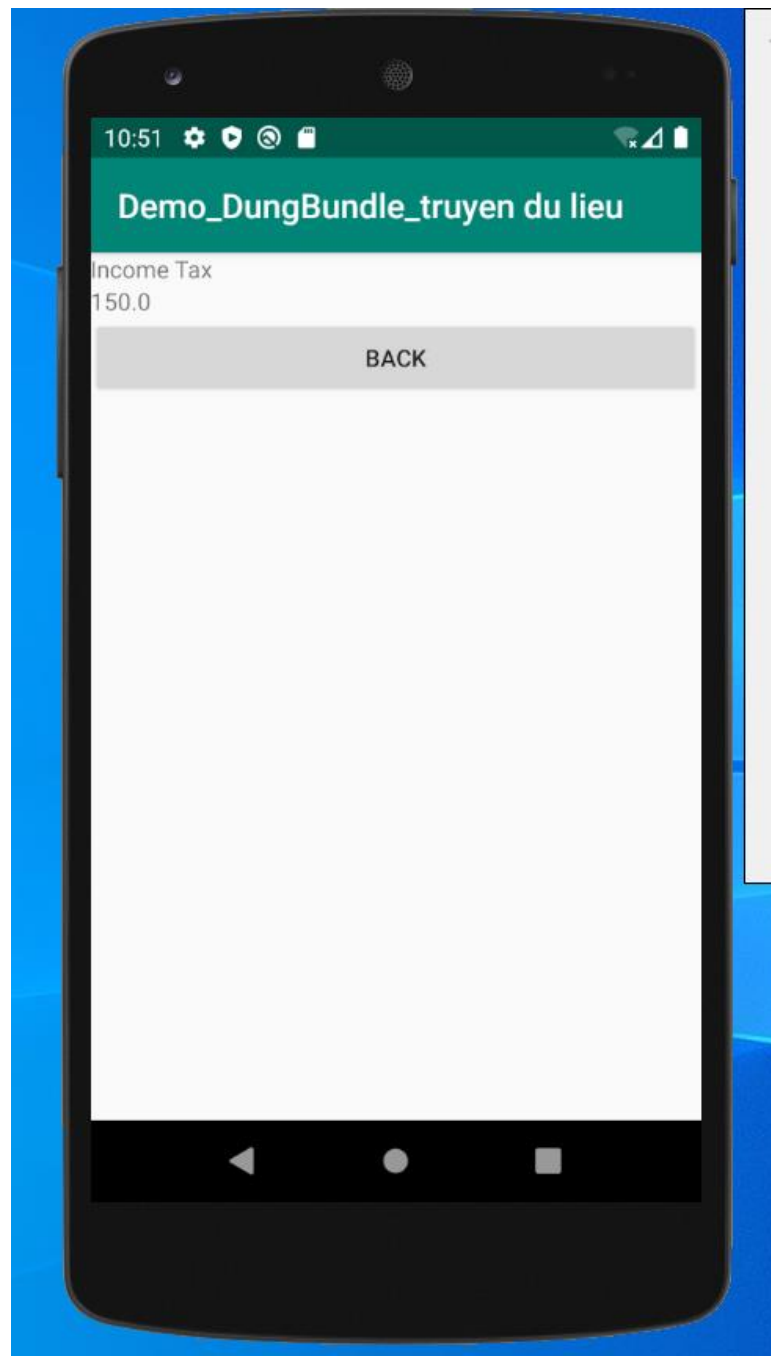
Dưới đây minh họa một ứng dụng tính thuế đơn giản, người dùng phải nhập vào tên và lương, thông báo nếu người dùng không nhập đầy đủ thông tin.

Khi người dùng nhập đủ thông tin, thuế sẽ được tính bằng 10% nếu lương >1000 trái lại bằng 0. Kết quả tính thuế sẽ được hiển thị ở màn hình thứ hai.

Giao diện ứng dụng : màn hình nhập thông tin



Màn hình tính toán và hiển thị kết quả:



- Thiết kế giao diện XML bên MainActivity:

```
<?xml version="1.0" encoding="utf-8"?>
<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">

    <TableLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TableRow
```



```
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <TextView
            android:id="@+id/txtName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Name" />

        <EditText
            android:id="@+id/edtName"
            android:layout_width="330dp"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="textPersonName"
            android:text="" />
    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/txtSalary"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Salary" />

        <EditText
            android:id="@+id/edtSalary"
            android:layout_width="330dp"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="numberDecimal"
            android:text="" />
    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <Button
            android:id="@+id/btnCal"
            android:layout_width="127dp"
            android:layout_height="wrap_content"
            android:text="Calculate Tax" />

        <TextView
            android:id="@+id/txtNotify"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="" />
    </TableRow>
```

```
</TableLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
- Thiết kế giao diện XML bên ResultActivity:
<?xml version="1.0" encoding="utf-8" ?>
<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=".ResultActivity">

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

        <TextView
            android:id="@+id/txtTax"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Income Tax" />

        <TextView
            android:id="@+id/txtResult"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="" />

        <Button
            android:id="@+id/btnBack"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Back" />

    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
- Mã nguồn khai báo và truyền dữ liệu qua bundle bên MainActivity.java
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    // khai báo các biến
    final EditText edtName=findViewById(R.id.edtName);
    final EditText edtSalary=findViewById(R.id.edtSalary);
    final TextView txtNotify=findViewById(R.id.txtNotify);
    Button btnCheck=findViewById(R.id.btnCal);
    // cài đặt tính toán logic btnCheck
    btnCheck.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // kiểm tra người dùng nhập dữ liệu chưa

            if (edtName.getText().length()==0||edtSalary.getText().length(
            )==0) {
```

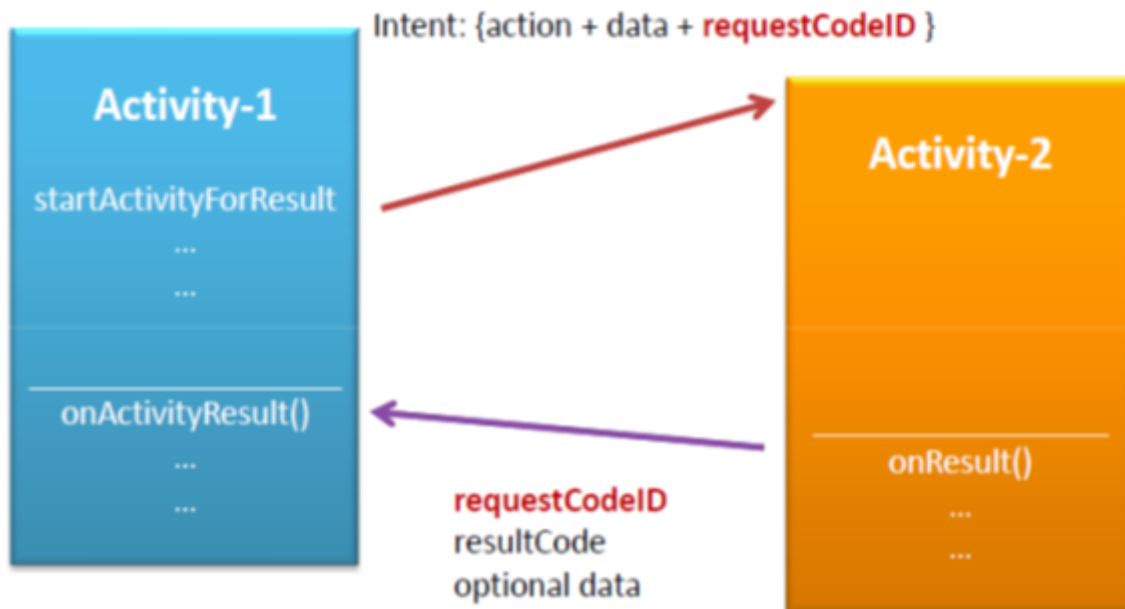
```
txtNotify.setText("Input Name and Salary");
}
else{
    // gọi sang Intent ResultActivity
    Intent myIntent=new
Intent(MainActivity.this,ResultActivity.class);
    // khai báo Bundle để truyền dữ liệu
    Bundle bundle=new Bundle();
    // lấy dữ liệu lương, ép kiểu từ string sang
float
float
salary=Float.parseFloat(edtSalary.getText().toString());
    // truyền dữ liệu qua bundle
    bundle.putFloat("Salary_send",salary);
    myIntent.putExtra("Main_send",bundle);
    startActivity(myIntent);
}
}
});
```

- Mã nguồn nhận dữ liệu qua bundle và xử lý hiển thị bên ResultActivity.java

```
- protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_result);
    TextView txtResult=findViewById(R.id.txtResult);
    Button btnBack=findViewById(R.id.btnBack);
    // lấy kết quả từ Intent
    Intent callerIntent=getIntent();
    // nhận kết quả từ Activity_send
    Bundle
    getResultfromCaller=callerIntent.getBundleExtra("Main_send");
    // lấy salary_send từ bên MainActivity truyền qua bundle
    và xử lý
    float
    result_salary=getResultfromCaller.getFloat("Salary_send");
    float tax=0;
    // tính thuế
    if(result_salary>1000){
        tax=0.1f*result_salary;
    }
    // hiển thị kết quả
    txtResult.setText(tax+"");
    // cài đặt sự kiện nút Back
    btnBack.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent backIntent=new
Intent(ResultActivity.this,MainActivity.class);
            startActivity(backIntent);
        }
    });
}
```

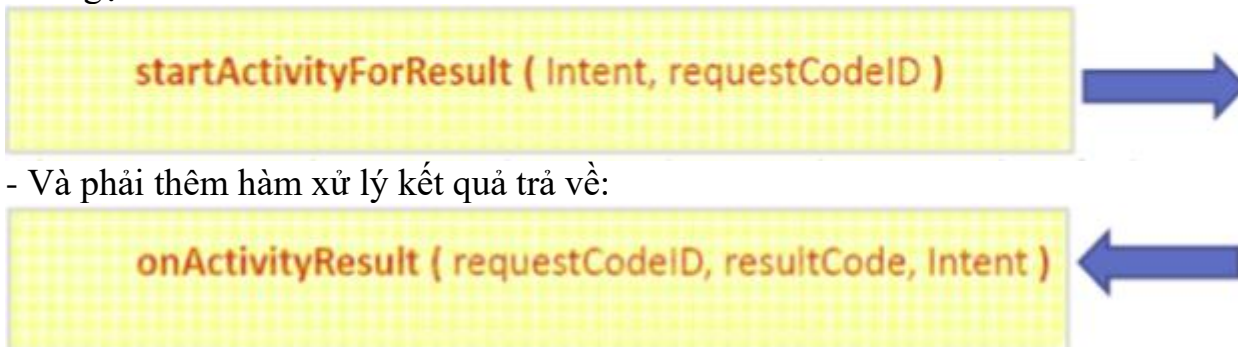
Bài tập 6.5. Bài thực hành mở một Activity truyền dữ liệu qua và xử lý kết quả trả về .

Để dễ hiểu sinh viên có thể xem hình minh họa sau :



Chúng ta sẽ dựa vào requestCodeID và resultCode để xử lý.

- Việc tạo Intent trong trường hợp này cũng y xì như trường hợp trước. Nó chỉ khác hàm gọi :

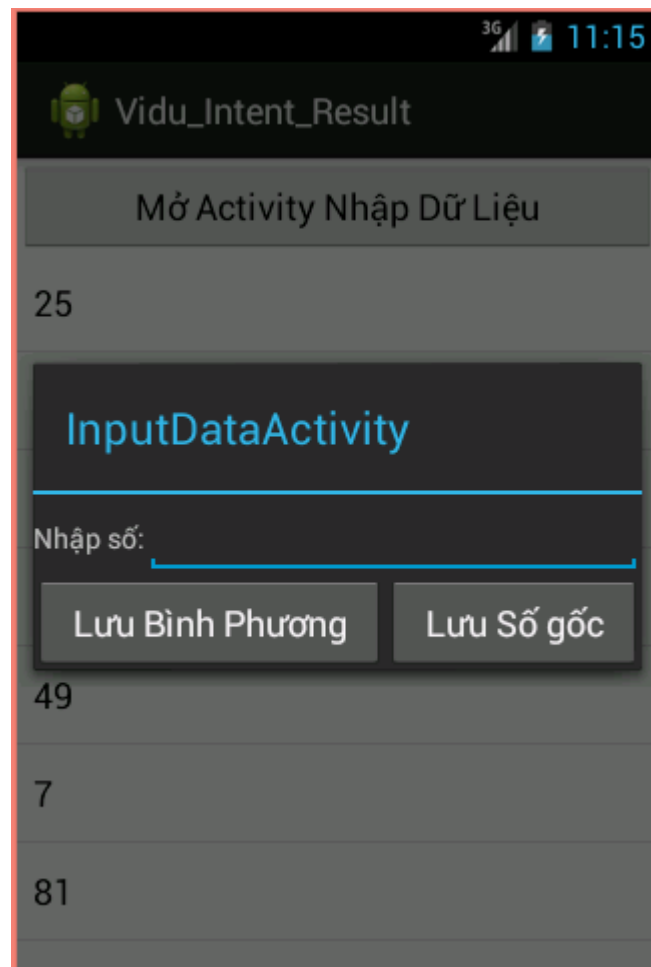


- Và phải thêm hàm xử lý kết quả trả về:

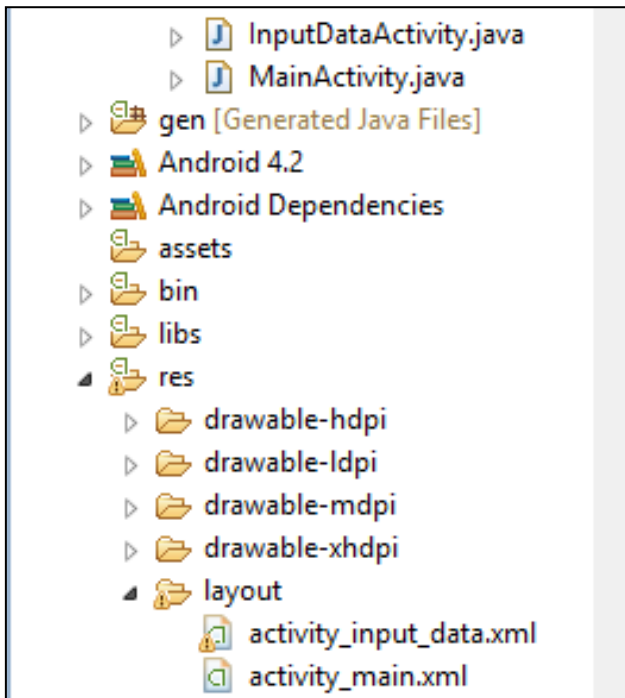
Chương trình minh họa đơn giản là MainActivity sẽ có 1 Button và ListView như hình bên dưới:



- Khi nhấn “Mở Activity nhập dữ liệu”, chương trình sẽ mở Activity mới dưới dạng Dialog:



- Khi nhập số và nhấn “Lưu bình phương” thì nó sẽ truyền số này qua MainActivity và cập nhật ListView với số này là nhân bình phương.
- Khi nhập số và nhấn “lưu số gốc” thì nó sẽ truyền số này qua MainActivity và cập nhật ListView với đúng số gốc này.
- Chú ý là phải đóng **Dialog** ngay, vì nếu không đóng thì **onActivityResult** sẽ không xảy ra. **onActivityResult** chỉ xảy ra trong **foreground life time**.
- Xem cấu trúc chương trình:



-Xử lý code của **MainActivity.java**:

```
1      public class MainActivity extends Activity {
2
3          public static final intREQUEST_CODE_INPUT=113;
4          public static final intRESULT_CODE_SAVE1=115;
5          public static final intRESULT_CODE_SAVE2=116;
6
7          Button btnInputData;
8          ListView lvData;
9          ArrayList<Integer>arrData=newArrayList<Integer>();
10         ArrayAdapter<Integer>adapter=null;
11         @Override
12         Protected void onCreate(Bundle savedInstanceState)
13     {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16
17         btnInputData =(Button)
18         findViewById(R.id.btnopenactivity);
19         btnInputData.setOnClickListener(newOnClickListen
20         er() {
21
22             @Override
23             public void onClick(View arg0) {
24                 //Mở Activity với REQUEST_CODE_INPUT
25                 Intent intent=new Intent(MainActivity.this,
26                     InputDataActivity.class );
27                 //gọi startActivityForResultsẽ yêu cầu ghi đè
28                 startActivityForResult
29                 startActivityForResult(intent,
30                 REQUEST_CODE_INPUT);
```

```
31         }
32     });
33     lvData=(ListView) findViewById(R.id.lvdata);
34     adapter=new ArrayAdapter<Integer>(
35         this,
36         android.R.layout.simple_list_item_1,
37         arrData );
38     lvData.setAdapter(adapter);
39 }
40 /**
41  * Xử lý kết quả trả về ở đây
42  */
43 @Override
44     Protected void onActivityResult(int requestCode, int
45 resultCode, Intent data) {
46
47     super.onActivityResult(requestCode, resultCode,
48 data);
49     //Kiểm tra có đúng requestCode =REQUEST_CODE_INPUT
50 hay không
51     //Vì ta có thể mở Activity với những RequestCode
52 khác nhau
53     if(requestCode==REQUEST_CODE_INPUT)
54     {
55         //Kiểm tra ResultCode trả về, cái này ở bên
56 InputDataActivity truyền về. Có nó rồi thì xử lý trở
57 lên thông thường
58         switch(resultCode)
59         {
60             Case RESULT_CODE_SAVE1:
61                 //giá trị từ InputDataActivity
62                 intv1= data.getIntExtra("data", 0);
63                 arrData.add(v1*v1);
64                 adapter.notifyDataSetChanged();
65                 break;
66             case RESULT_CODE_SAVE2:
67                 //giá trị từ InputDataActivity
68                 intv2= data.getIntExtra("data", 0);
69                 arrData.add(v2);
70                 adapter.notifyDataSetChanged();
71                 break;
72         }
73     }
74 }
```

- Chúng ta nên khai báo hằng số cho **requestCode** và **resultCode** như ở trên, dựa vào nó để xử lý kết quả trả về. Mã nguồn của **InputDataActivity.java**:

```
public class InputDataActivity extends Activity {

    Button btnSave1, btnSave2;
    EditText editNumber;
    @Override
    Protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_input_data);
        btnSave1=(Button) findViewById(R.id.btnSave1);
        btnSave2=(Button) findViewById(R.id.btnSave2);

        btnSave1.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View arg0) {
                //Gửi thông điệp là lưu bình phương
                sendToMain(MainActivity.RESULT_CODE_SAVE1);
            }
        });
        editNumber=(EditText) findViewById(R.id.editNumber);

        btnSave2.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                //Gửi thông điệp là lưu số gốc
                sendToMain(MainActivity.RESULT_CODE_SAVE2);
            }
        });
    }
    /**
     * hàm xử lý gửi kết quả về mainactivity
     * khi hàm này được gọi thì lập tức onActivityResult
     * ở MainActivity sẽ xảy ra đem theo ResultCode và Intent
     * @param resultcode
     */
    Public void sendToMain(int resultcode)
    {
        Intent intent=getIntent();
        Intvalue= Integer.parseInt(editNumber.getText()+"");
        intent.putExtra("data", value);
        setResult(resultcode, intent);
        finish();
    }
}
```

Chú ý: hàm **sendToMain**, khi hàm này vừa kết thúc thì sự kiện **onActivityResult** của **MainActivity** sẽ được gọi **onActivityResult(int requestCode, int resultCode, Intent data)**

Ta kiểm tra requestCode và resultCode cho đúng

- **requestCode** là bên MainActivity dùng để triệu gọi một Activity bất kỳ nào đó
- **resultCode** là kết quả trả về trong hàm **setResult(resultcode, intent)**; của sub Activity nào đó. Hàm này cho chúng ta biết kết quả trả về là **code** nào và đồng

thời cho ta biết luôn **Intent** của nó. Dựa vào **Intent** này mà trong **onActivityResult** ta có thể dễ dàng lấy thông số ra (đối số thứ 3).

Bài tập 6.6: Bài tập tổng hợp Quản Lý Nhân Viên

- Mục tiêu của bài tập này:

- Thao tác nhuần nhuyễn với Intent
- Ôn tập lại các control thường hay sử dụng
- Nhuần nhuyễn với ListView
- Xử lý Checked (radio, Checkbox) trong ListView

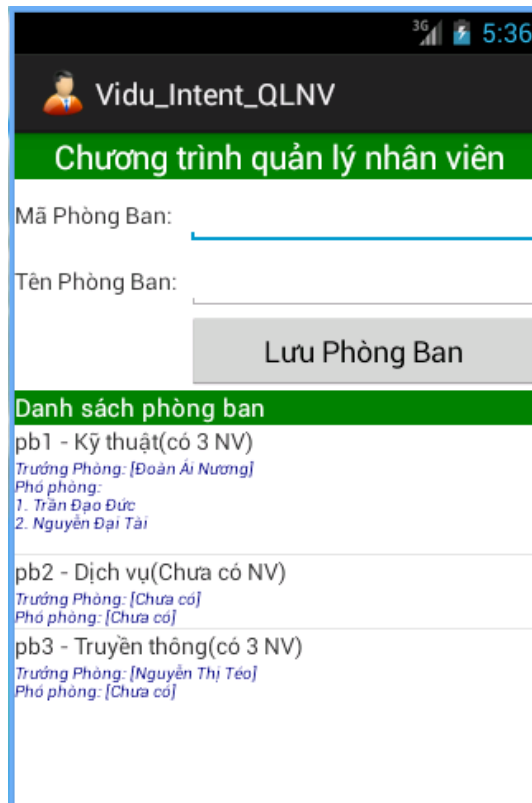
Mô tả chương trình:

- o Chương trình cho phép quản lý nhân viên + phòng ban.
- o Mỗi phòng ban sẽ có nhiều nhân viên, phòng ban cần lưu trữ (mã phòng ban, tên phòng ban)
- o Một phòng ban sẽ có Trưởng phòng và Phó phòng. Trưởng phòng thì chỉ có 1, phó phòng thì số lượng không giới hạn (hoặc không có)
- o Thông tin nhân viên : Mã nhân viên, tên nhân viên, giới tính (nam, nữ), chức vụ (trưởng phòng, phó phòng, nhân viên)

Chương trình cho phép:

- Thêm/sửa/ xóa phòng ban
- Thêm / sửa / xóa nhân viên
- và một số chức năng khác

Demo giao diện chính của chương trình:



Vidu_Intent_QLNV

Chương trình quản lý nhân viên

Mã Phòng Ban:

Tên Phòng Ban:

Lưu Phòng Ban

Danh sách phòng ban

pb1 - Kỹ thuật(có 3 NV)
Trưởng Phòng: [Đoàn Ái Nương]
Phó phòng:
1. Trần Đạo Đức
2. Nguyễn Đại Tài

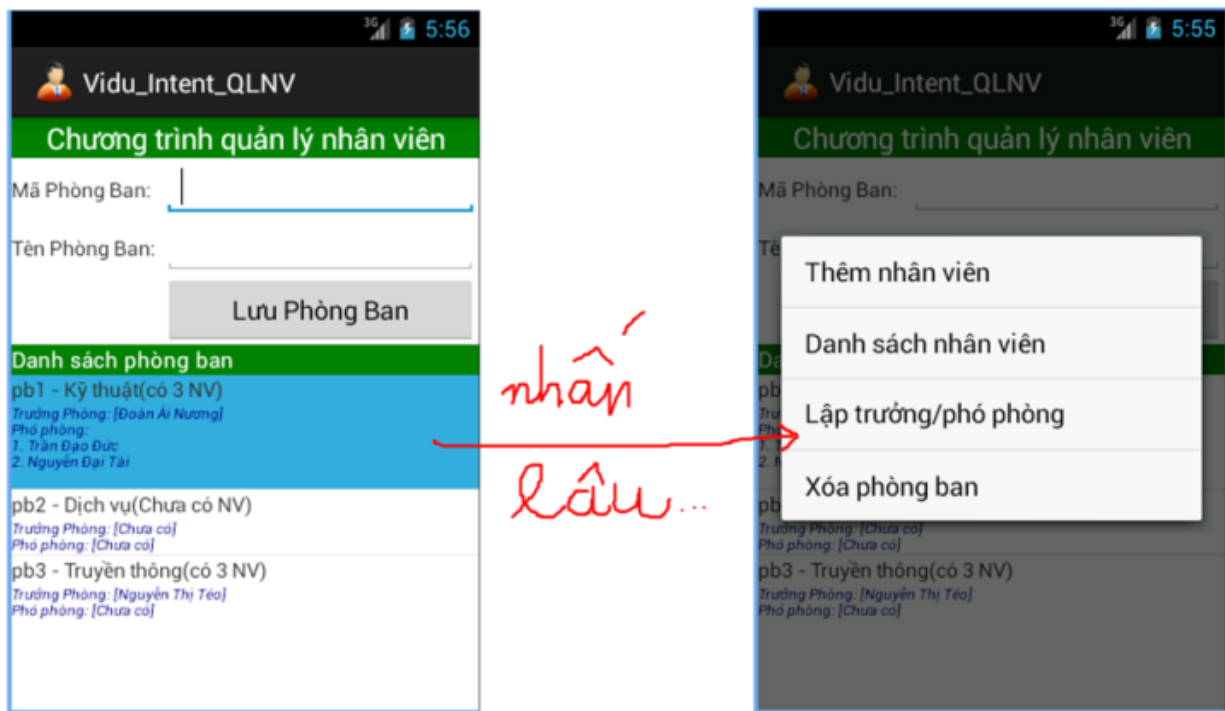
pb2 - Dịch vụ(Chưa có NV)
Trưởng Phòng: [Chưa có]
Phó phòng: [Chưa có]

pb3 - Truyền thông(có 3 NV)
Trưởng Phòng: [Nguyễn Thị Téo]
Phó phòng: [Chưa có]

- Với giao diện trên ta có:

- Nút “**Lưu Phòng Ban**“: Cho phép lưu phòng ban và cập nhật vào ListView. Nếu mã phòng ban đã tồn tại thì tự động cập nhật, còn mã phòng ban chưa tồn tại thì phải thêm mới.
- Sự kiện nhấn trên **Listview**: nhấn vào phòng ban nào thì hiển thị lại thông tin của phòng ban đó vào phần **edittext** bên trên.

Xử lý Context Menu cho ListView Phòng ban:



- Nhấn thật lâu vào một phòng ban nào đó, ví dụ trường hợp trên là phòng Kỹ Thuật, chương trình sẽ hiển thị context Menu có 3 chức năng:

- o 1. Thêm nhân viên
- o 2. Xem danh sách nhân viên
- o 3. và xóa phòng ban.

- Ứng với mỗi chức năng ta có:

1. Thêm nhân viên:

- Mặc định mỗi một nhân viên mới được tạo ra thì chỉ có chức vụ là Nhân Viên (không có chức quản lý):

3G 6:05

Vidu_Intent_QLNV

Chương trình quản lý nhân viên

Mã Phòng Ban:

ThêmNhânVienActivity

Thông Tin Nhân Viên

Mã NV: m113

Tên NV: Tran Duy Thanh

Giới Tính: ☒ Nam ☐ Nữ

Xóa trắng Lưu nhân viên

pb3 - Truyền thông(có 3 NV)

Trưởng Phòng: [Nguyễn Thị Téo]

Phó phòng: [Chưa có]

- Nút “xóa trắng “: cho phép xóa dữ liệu trên EditText và focus tới mã nhân viên
- Nút “Lưu nhân viên”, cho phép thêm mới nhân viên vào phòng ban, Giới Tính Nam, Nữ để hiển thị hình ảnh cho đúng.
- Sau khi bấm nút “Lưu nhân viên”, thì cập nhật lại danh sách phòng ban (cập nhập số lượng nhân viên cho phòng ban).
- Minh họa hình trên nếu chọn “Lưu nhân viên”, màn hình này sẽ đóng lại và trở về màn hình chính, số lượng nhân viên phòng ban Kỹ Thuật số lượng nhân viên sẽ tăng lên là 4, xem hình:

Danh sách phòng ban

pb1 - Kỹ thuật(có 4 NV)

Trưởng Phòng: [Đoàn Ai Nương]

Phó phòng:

1. Trần Đạo Đức

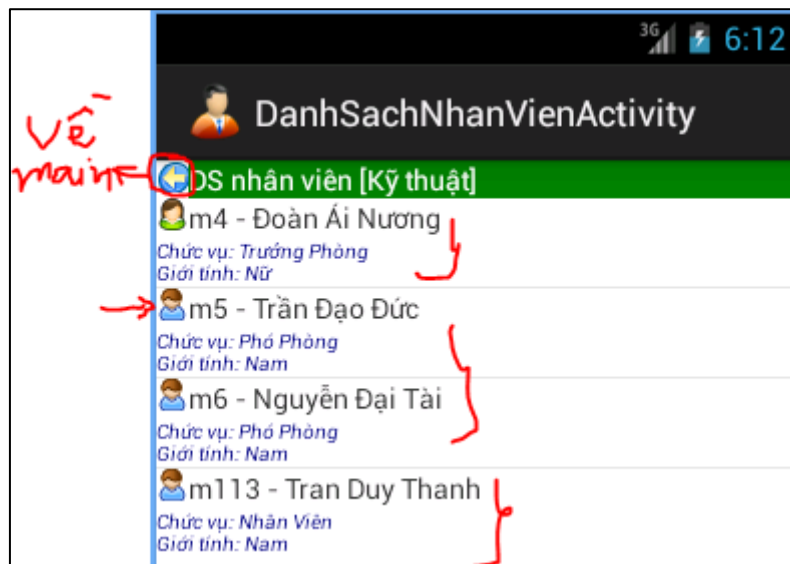
2. Nguyễn Đại Tài

pb2 - Dịch vụ(Chưa có NV)

Trưởng Phòng: [Chưa có]

2. Xem danh sách nhân viên:

- Chương trình sẽ hiển thị danh sách nhân viên của phòng ban đang chọn (minh họa danh sách nhân viên của phòng ban Kỹ thuật):



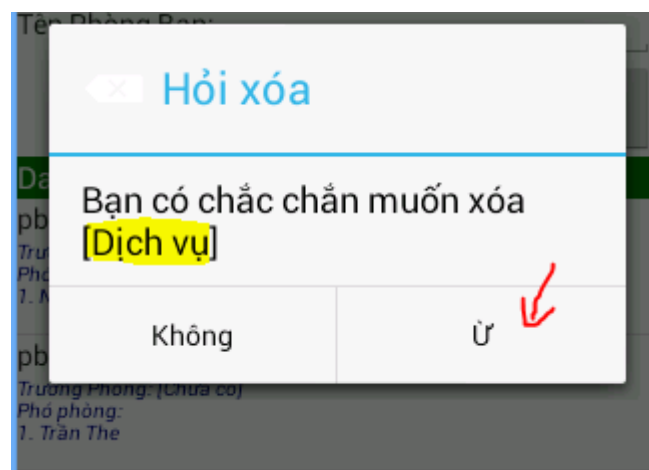
Màn hình xem danh sách nhân viên có các chức năng sau:

- Nút mũi tên “Back” cho phép trở về màn hình chính.
- Nhân viên có hình theo giới tính (nam, nữ)
- Nhân viên có mô tả: Chức vụ, giới tính

Vậy ta cũng phải dùng Custom layout để biểu diễn thông tin hiển thị như minh họa.

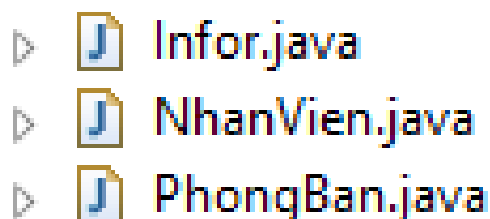
4. Xóa phòng ban.

Xóa phòng ban, tương tự như xóa nhân viên. Chương trình cũng hỏi ta có chắc chắn muốn xóa hay không?



Hướng dẫn thực hiện:

- Source Code chia ra làm 3 Package:
 - o Package **model** :



- Package này có 3 class.

- enum ChucVu.java:

```
//Muốn gán được như thế này: TruongPhong("Trưởng Phòng")
//thì phải có constructor ChucVu(String cv)

public enum ChucVu {
    TruongPhong("Trưởng Phòng"), PhoPhong("Phó
    Phòng"), NhanVien("Nhân Viên");
    private String cv;
    ChucVu(String cv){
        this.cv=cv;
    }
    public String getChucVu(){
        return this.cv;
    }
}
```

Infor.java:

```
/**
 * Class này để lưu mã và tên là class cha của NhanVien
 * và PhongBan. Vì cả nhân viên hay phòng ban đều có thông tin
:mã, tên *
 */
public class Infor implements Serializable{
    private String mã;
    private String tên;
    //xây dựng hàm tạo
    //xây dựng hàm get/set
    //ghi đè toString
}
```

- Coding NhanVien.java:

```
public class NhanVien extends infor implements Serializable{
    private boolean gioitinh;
    private String chucvu;
    // xây dựng hàm tạo
    //xây dựng các hàm get/set khác nếu cần
    //ghi đè toString để quy định hiển thị dữ liệu
}
```

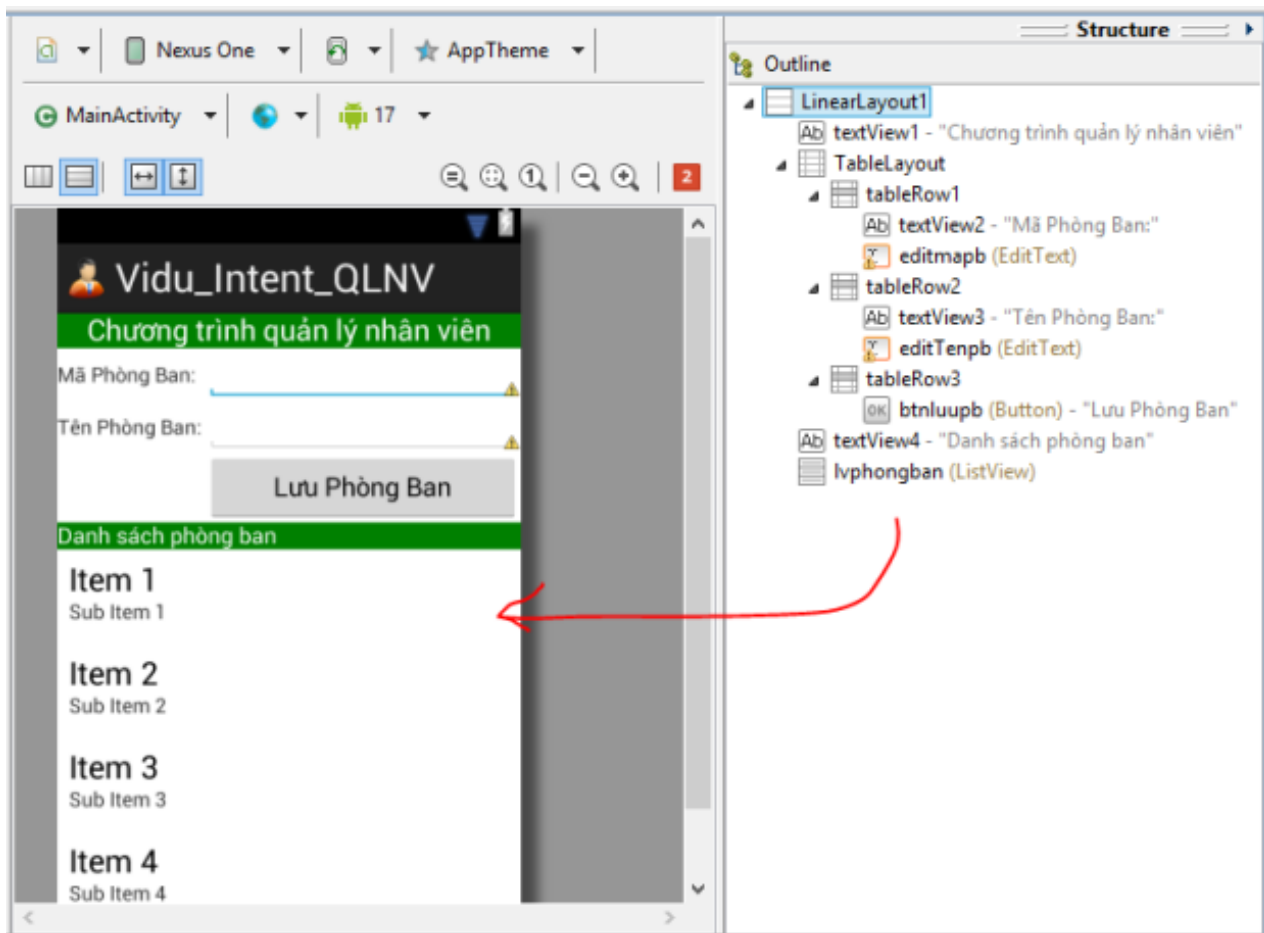
```
}  
  
}
```

- PhongBan.java:

```
public class PhongBan extends Infor implements Serializable{  
    private ArrayList<NhanVien>dsnv = new ArrayList<NhanVien>();  
    //xây dựng hàm tạo. sv tự xd 2 hàm tạo. 1 hàm k đối số, 1 hàm có  
    các đối số truyền vào  
  
    /** Hàm cho phép thêm một nhân viên mới  
     * nếu nhân viên có mã tồn tại rồi thì tự động trở thành cập nhật  
     */  
    public void themNv(NhanVien nv){  
        //sv tự triển khai theem nv vào dsnv  
    }  
    public NhanVien get(intindex) {  
        return dsnv.get(index);  
    }  
    public int size() {  
        return dsnv.size();  
    }  
    public ArrayList<NhanVien> getListNhanVien() {  
        return this.dsnv;  
    }  
    public String toString() {  
        String str=super.toString();  
        if(dsnv.size()==0) str+="(Chưa có NV)";  
        else str+="(có "+dsnv.size()+" NV)";  
        return str;  
    }  
}
```

Package lưu các activity:

- Màn hình chính (**MainActivity.java**):



- **MainActivity**:

```
public class MainActivity extends AppCompatActivity {  
    //Khai báo các Request + Result code để xử lý Intent for  
    result  
  
    public static final int MO_ACTIVITY_THEM_NHAN_VIEN=1;  
    public static final int MO_ACTIVITY_SUA_NHAN_VIEN=2;  
    public static final int THEM_NHAN_VIEN_THANHCONG=3;  
    public static final int SUA_NHAN_VIEN_THANHCONG=4;  
    public static final int XEM_DS_NHAN_VIEN=5;  
    public static final int CAPNHAT_DS_NHAN_VIEN_THANHCONG=6;  
    private Button btnLuuPb;  
    private EditText editMapb, editTenpb;  
    private ListView lvpb;
```



```
private static ArrayList<PhongBan> arrPhongBan = new  
ArrayList<PhongBan>();  
//Khai báo adapter cho phòng ban  
private ArrayAdapter<PhongBan> adapterPb=null;  
private PhongBan pbSelected=null;
```

```
@Override  
Protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    getFormWidgets();  
    addEvents();  
    fakeData();  
}
```

```
Private void fakeData() {  
    pb= new PhongBan("pb2", "Dịch vụ");  
    arrPhongBan.add(pb);  
    pb= new PhongBan("pb3", "Truyền thông");  
    arrPhongBan.add(pb);  
    nv= new NhanVien("m1", "Nguyễn Văn Tèo", false);  
    pb.themNv(nv);  
    nv= new NhanVien("m2", "Nguyễn Thị Téo", true);  
    pb.themNv(nv);  
    nv= new NhanVien("m3", "Nguyễn Văn Teo", false);  
    pb.themNv(nv);  
    adapterPb.notifyDataSetChanged();  
}
```

```
/**  
 * - Hàm này để load các control theo Id  
 * - Thiết lập Adapter cho ListView phòng ban  
 * - thiết lập context Menu cho ListView  
 */  
public void getFormWidgets() {  
    btnLuuPb=findViewById(R.id.??);
```

```
editMapb=findViewById(R.id.??);
editTenpb= findViewById(R.id.??);
lvpb= findViewById(R.id.??);
//khởi tạo đối tượng phòng ban adapter
adapterPb=new ??(?, ?, arrPhongBan);
lvpb.setAdapter(adapterPb);
//Đăng ký contextmenu cho ListView
registerForContextMenu(lvpb);
}
```

```
/**
 * hàm gán sự kiện cho các control: button, ListView
 */
public void addEvents() {
    //Bấm lưu để lưu phòng ban
    btnLuuPb.setOnClickListener(new OnClickListener() {
        public void onClick(View arg0) {
            doLuuPhongBan();
        }
    });
    //xử lý lưu biến tạm khi nhấn long - click
    //để biết được trước đó đã chọn item nào
    lvpb.setOnItemLongClickListener(new OnItemLongClickListener() {
        @Override
        public boolean onItemLongClick(AdapterView<?> arg0, View
arg1,
        Int arg2, long arg3) {
            //lưu vết lại đối tượng thứ arg2 trong danh sách
            pbSelected=arrPhongBan.get(arg2);
            return false;
        }
    });
}
```

```
/**
 * hàm xử lý lưu phòng ban
 * đơn thuần là đưa phòng ban mới vào ArrayList
 * chú ý bạn phải kiểm tra trùng lặp Id, hay các
 * thông tin đã đầy đủ chưa ở đây
 * gọi adapterPb.notifyDataSetChanged(); để cập nhật ListView
 */
public void doLuuPhongBan() {
    String mapb=editMapb.getText()+" ";
    String tenpb=editTenpb.getText()+" ";
    PhongBan pb=new PhongBan(mapb, tenpb);
    arrPhongBan.add(pb);
    adapterPb.notifyDataSetChanged();
}
```

```
/**
 * hàm này để gán ContextMenu vào ứng dụng
Sv tự tạo nội dung menu như ví dụ trước
Thực hiện:
- Tạo 1 file contextmenu_phongban.xml trong package menu
- Nội dung có 3 item: mnuthemnv; mnudanhhsachnv và mnuxoaphong ứng
với 3 hiển thị.
```

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenuItem menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    getMenuInflater().inflate(R.menu.contextmenu_phongban, menu);
}
```

```
/**
 * hàm này để xử lý người sử dụng vừa chọn menuitem nào theo Id
 */
@Override
```

```
public boolean onContextItemSelected(MenuItem item) {  
    switch(item.getItemId()) {  
    Case R.id.mnuthemnv:  
        doThemNhanVien();  
        break;  
    case R.id.mnudanhsachnv:  
        doDanhSachNhanVien();  
        break;  
    case R.id.mnuxoaphong:  
        //sv xử lý tiếp nút xóa  
    }  
    return super.onContextItemSelected(item);  
}
```

```
/**  
 * khi chọn menu Thêm nhân viên, đơn thuần chỉ là hiển thị  
 * màn hình thêm nhân viên (với dạng Dialog)  
 * nhớ là dùng startActivityForResult  
 * và lắng nghe kết quả tại hàm onActivityResult  
 */  
public void doThemNhanVien(){  
    Intent i=new Intent(this, ThemNhanVienActivity.class);  
    startActivityForResult(i, MO_ACTIVITY_THEM_NHAN_VIEN);  
}
```

```
/**  
 * hàm onActivityResult để xử lý kết quả trả về  
 * sau khi gọi startActivityForResult kết thúc  
 */  
protected void onActivityResult(int requestCode, int resultCode,  
Intent data) {  
    super.onActivityResult(requestCode, resultCode, data);  
    //màn hình thêm mới nhân viên trả kết quả về  
    if(resultCode==THEM_NHAN_VIEN_THANHCONG){  
        //Cách lấy dữ liệu đã được đề cập trong bài thực hành intent  
        Bundle bundle= data.getBundleExtra("DATA");  
        NhanVien nv= (NhanVien) ;
```

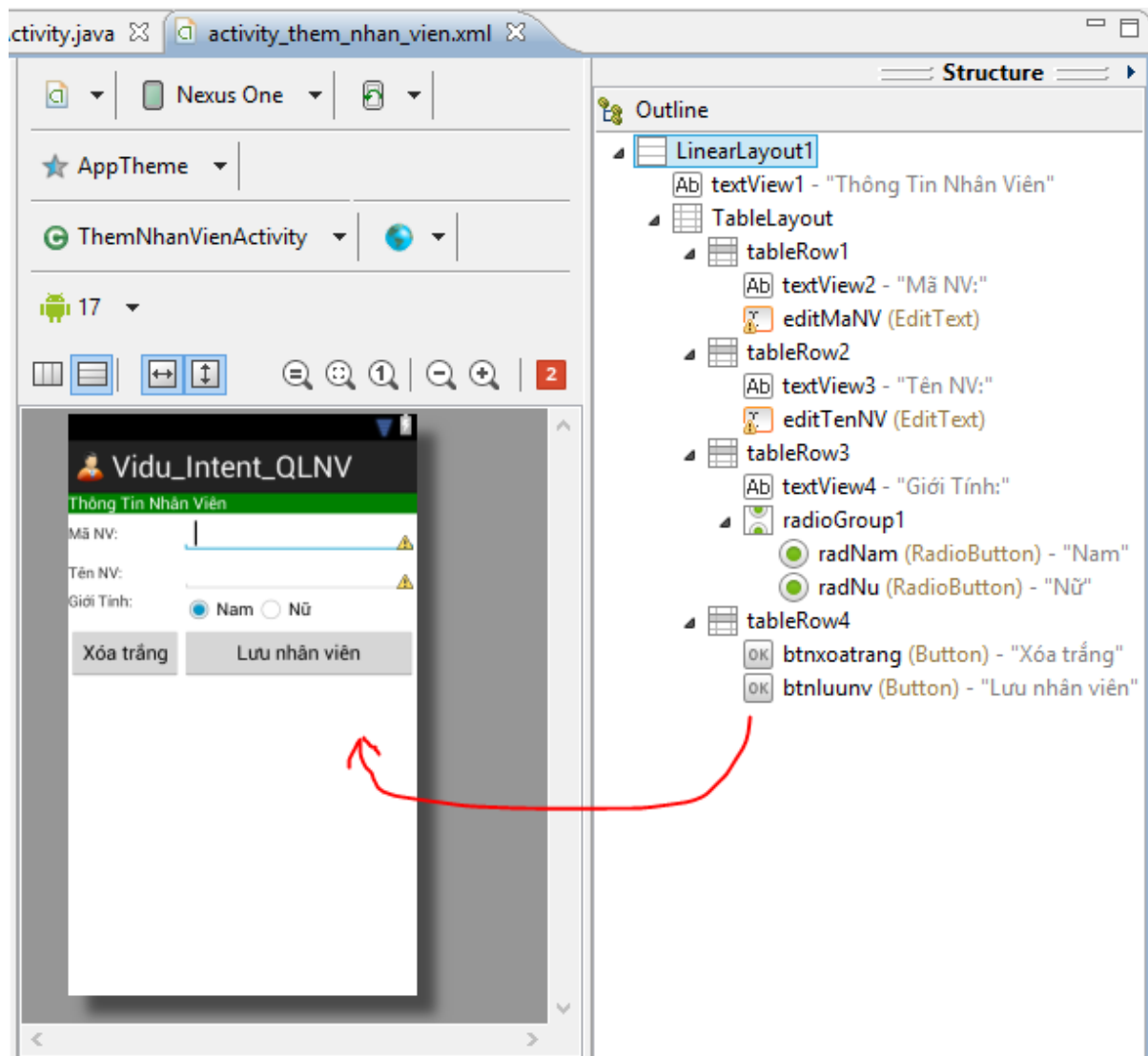
```
bundle.getSerializable("NHANVIEN");
pbSelected.themNv(nv);
adapterPb.notifyDataSetChanged();
}
else if(resultCode == CAPNHAT_DS_NHAN_VIEN_THANHCONG) {
//Cách lấy dữ liệu đã học rồi, ko nói lại
Bundle bundle= data.getBUNDLEExtra("DATA");
PhongBan pb= (PhongBan) bundle.getSerializable("PHONGBAN");
//đơn thuần là xóa danh sách cũ
pbSelected.getListNhanVien().clear();
//rồi cập nhật lại toàn bộ danh sách mới
//sinh viên có thể chọn giải pháp khác thay vì xóa đi thêm lại.
pbSelected.getListNhanVien().addAll(pb.getListNhanVien());
adapterPb.notifyDataSetChanged();
}
}

/**
 * hàm cho phép danh xem danh sách nhân viên của phòng
 * ban đang chọn, đơn thuần là mở DanhSachNhanVienActivity
 * và truyền phòng ban qua -->toàn bộ danh sách nhân viên
 * sẽ được hiển thị trong DanhSachNhanVienActivity
 */
public void doDanhSachNhanVien(){
Intent i=new Intent(this, DanhSachNhanVienActivity.class);
Bundle bundle=new Bundle();
bundle.putSerializable("PHONGBAN", pbSelected);
i.putExtra("DATA", bundle);
startActivityForResult(i, XEM_DS_NHAN_VIEN);
}
```

```
/**
 * hàm cho phép xóa phòng ban đang chọn
 */
public void doXoaPhongBan() {
    AlertDialog.Builder builder=new AlertDialog.Builder (this);
    builder.setTitle("Hỏi xóa");
    builder.setMessage("Bạn có chắc chắn muốn xóa"
    +pbSelected.getTen());
    builder.setIcon(android.R.drawable.ic_input_delete);
    builder.setNegativeButton("Không", new
    DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface arg0, intarg1) {
            arg0.cancel();
        }
    });
    builder.setPositiveButton("có", new
    DialogInterface.OnClickListener() {
        @Override
        public voidonClick(DialogInterface dialog, int which) {
            arrPhongBan.remove(pbSelected);
            adapterPb.notifyDataSetChanged();
        }
    });
    builder.show();
}

/**
 * hàm này cho phép truy suất được danh sách phòng ban tổng thể
 */
public static ArrayList<PhongBan> getListPhongBan() {
    return arrPhongBan;
}
}
```

- Màn hình thêm nhân viên cấu hình mở dạng Dialog, layout như sau:

**ThemNhanVienActivity.java:**

```
/**
 * class này cho phép thêm mới một nhân viên vào
 * Phòng ban đang chọn
 */

public class ThemNhanVienActivity extends AppCompatActivity{
    private Button btnXoaTrang,btnLuuNhanVien;
    private EditText editManv,editTenNv;
    private RadioButton radNam;

    @Override
    Protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_them_nhan_vien);
        getFormWidgets();
        addEvents();
    }
}
```

```
}
```

```
/**
 * hàm lấy các control theo id
 */
public void getFormWidgets() { ??? Sv tự làm }
```

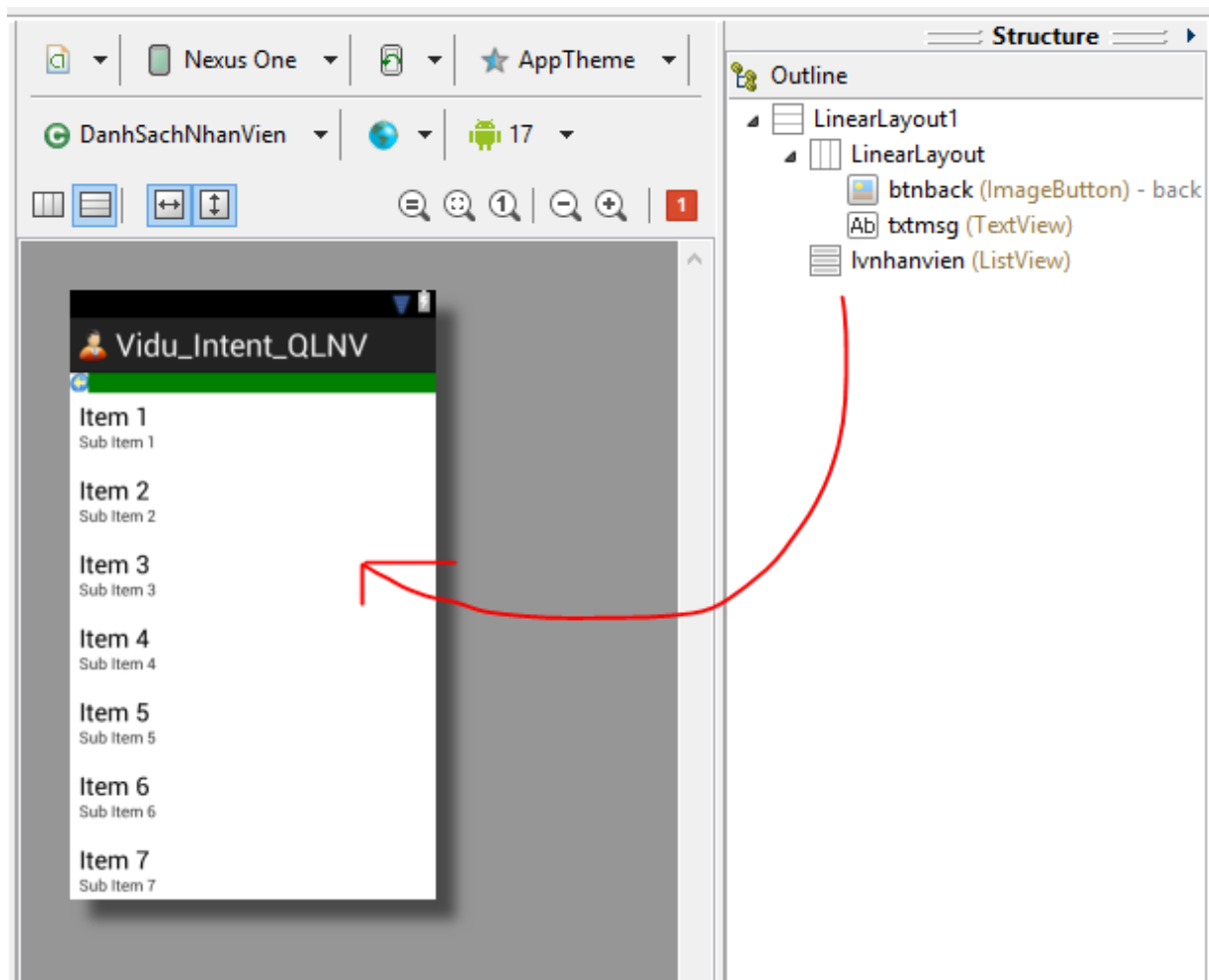
```
/**
 * hàm gán sự kiện
 */
public void addEvents() {
    btnXoaTrang.setOnClickListener(??);
    btnLuuNhanVien.setOnClickListener(??) và gọi hàm
    doLuuNhanVien;
}
```

```
/**
 * hàm lưu nhân viên theo phòng ban
 * truyền nhân viên qua MainActivity
 * MainActivity có nhiệm vụ lấy được nhân viên này
 * và đưa họ vào phòng ban chọn lúc này
 */
public void doLuuNhanVien() {
    NhanVien nv=new NhanVien();
    nv.setMa(editManv.getText()+"");
    nv.setTen(editTenNv.getText()+"");
    nv.setChucvu(ChucVu.NhanVien);
    nv.setGioitinh(!radNam.isChecked());
    Intent i=getIntent();
    Bundle bundle=new Bundle();
    bundle.putSerializable("NHANVIEN", nv);
```



```
i.putExtra("DATA", bundle);  
setResult(MainActivity.THEM_NHAN_VIEN_THANHCONG, i);  
finish();  
}  
}
```

Xem danh sách nhân viên. outline:



DanhSachNhanVienActivity.java :

```
/**  
 * class này dùng để hiển thị danh sách nhân viên  
 */  
  
public class DanhSachNhanVienActivity extends  
AppCompatActivity {  
  
    TextView txtmsg;  
    ImageButton btnback;  
    ListView lvNhanvien;  
    ArrayList<NhanVien> arrNhanvien=null;
```

```
//Nhân viên Adapter để hiển thị thông tin
//và chi tiết : chức vụ, giới tính
ArrayAdapter<NhanVien> adapter=null;
PhongBan pb=null;
private NhanVien nvSelected=null;
private int position=-1;
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_danh_sach_nhan_vien);
    txtmsg=(TextView) findViewById(R.id.txtmsg);
    btnback=(ImageButton) findViewById(R.id.btnback);
    lvNhanvien=(ListView) findViewById(R.id.lvnhanvien);
    getDataFromMain();
    addEvents();
    registerForContextMenu(lvNhanvien);
}
```

```
/**
 * hàm lấy giá trị từ MainActivity
 * sẽ truyền phòng ban qua
 * bên này lấy phòng ban và dựa vào phòng ban này
 * lấy ra danh sách nhân viên
 */
public void getDataFromMain(){
    Intent i=getIntent();
    Bundle b=i.getBundleExtra("DATA");
    pb= (PhongBan) b.getSerializable("PHONGBAN");
    arrNhanvien=pb.getListNhanVien();
    adapter=new ArrayAdapter<NhanVien>(this,
        android.R.layout.simple_list_item_1,
        arrNhanvien);
    lvNhanvien.setAdapter(adapter);
    txtmsg.setText("DS nhân viên ["+pb.getTen()+"]");
}
```

```
/**
 * hàm gán sự kiện cho contextmenu
 */
public void addEvents() {
    btnback.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View arg0) {
            finish() ;
        }
    });
}
```