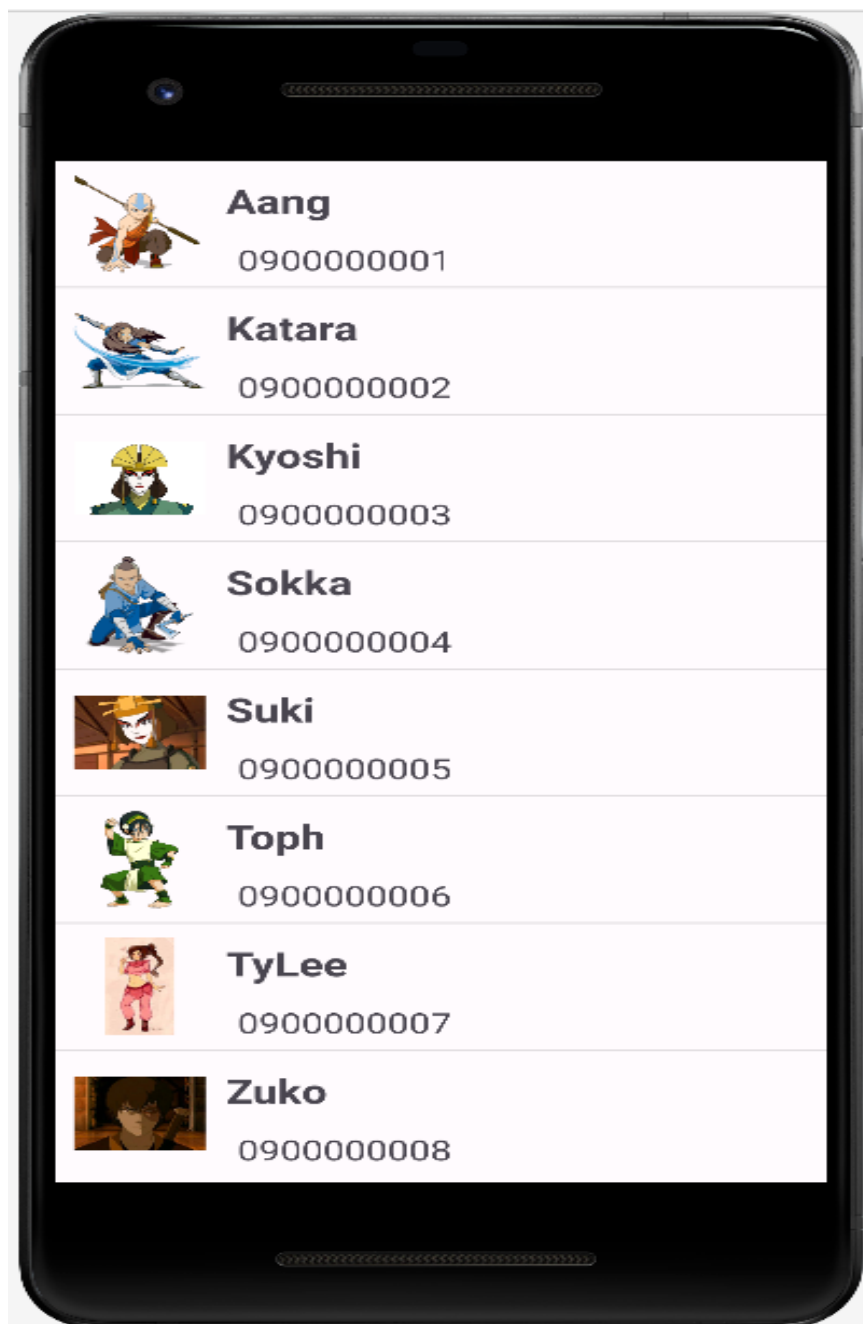
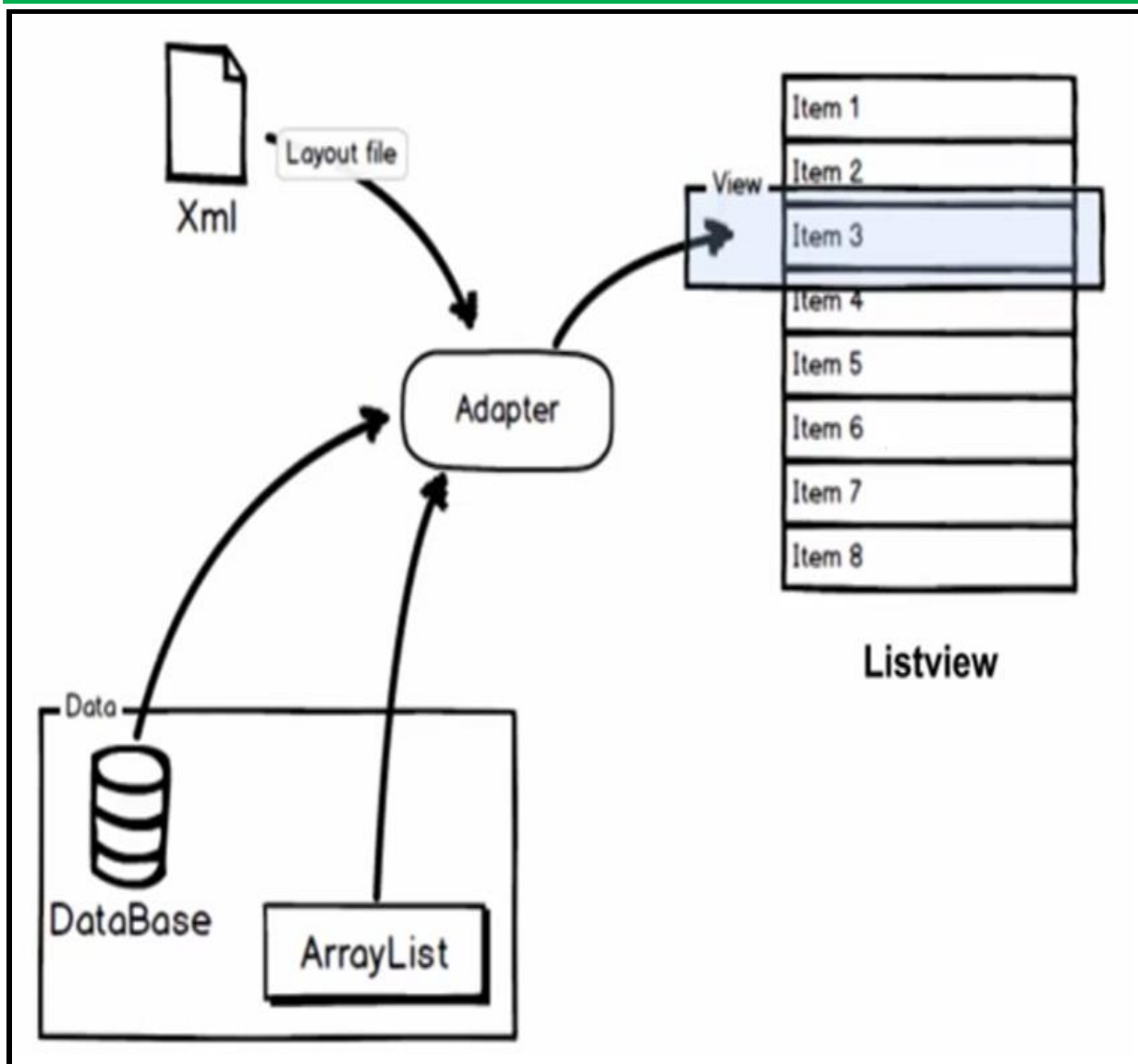


LAB DEMO LISTVIEW CONTACT

MÔ HÌNH LISTVIEW

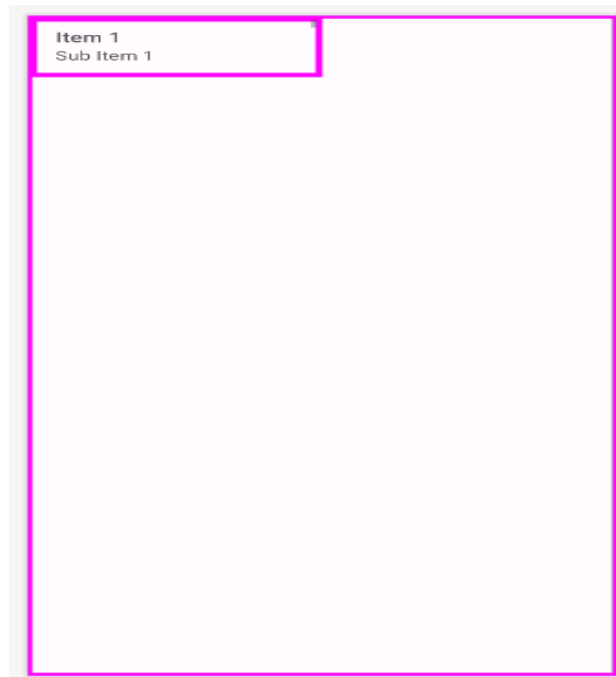




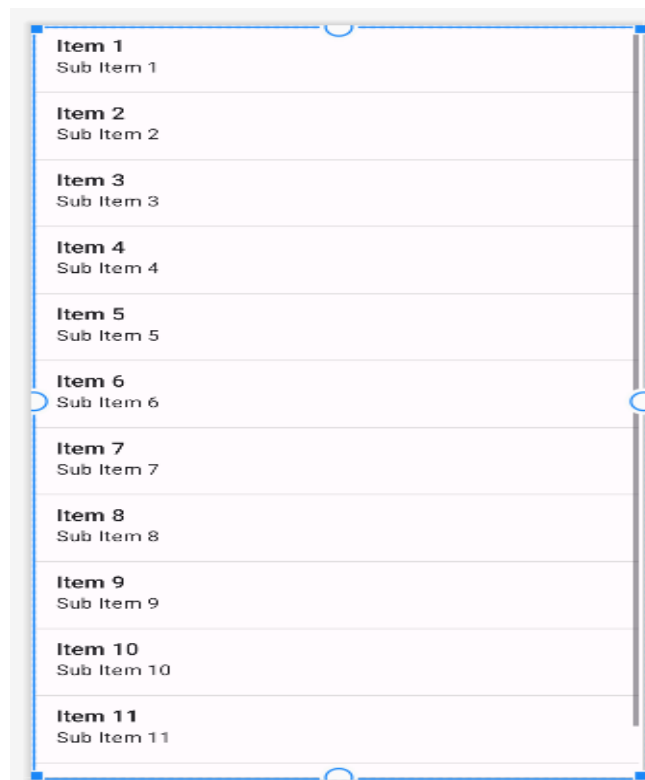
TẠO GIAO DIỆN

BƯỚC 1: Tạo 1 project mới: demoListViewContact

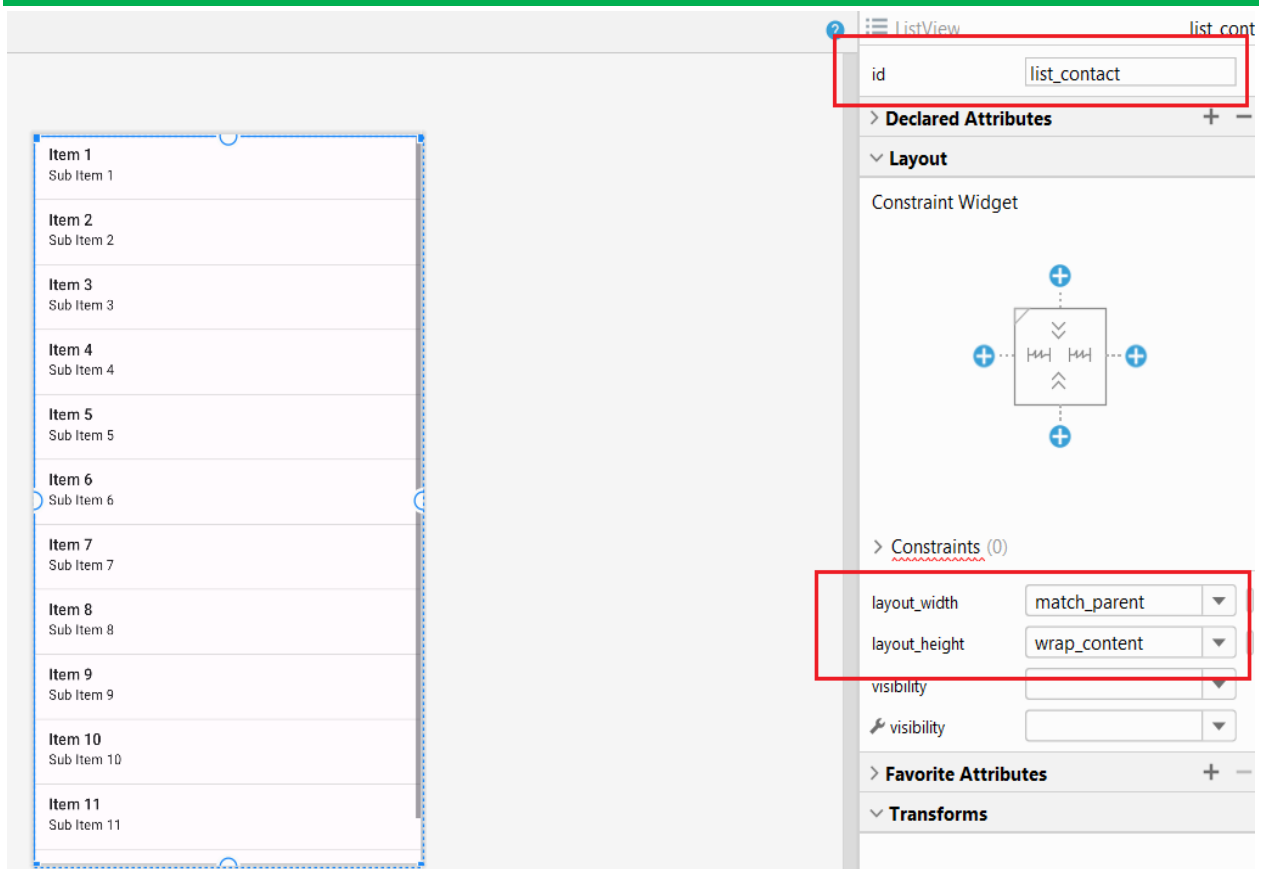
BƯỚC 2: Kéo ListView vào file layout XML như hình



KẾT QUẢ NHƯ SAU:

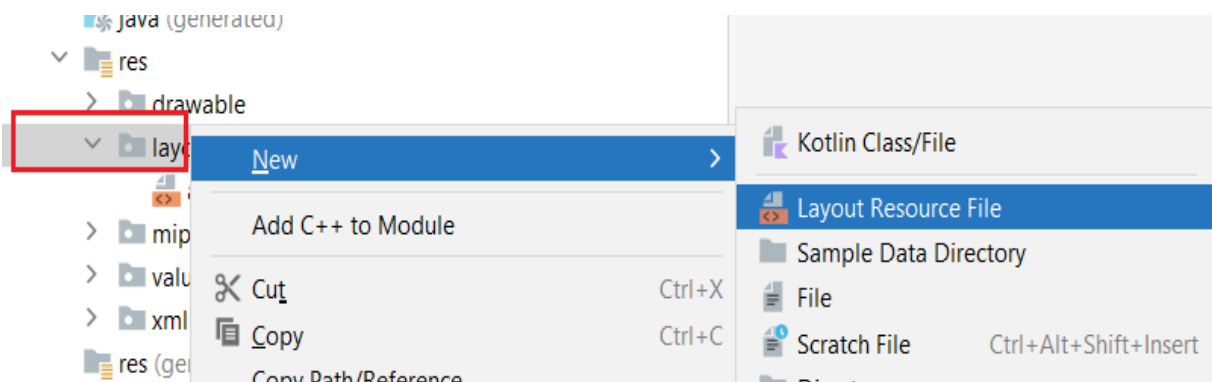


BƯỚC 3: Chỉnh layout_width, layout_height, đặt id

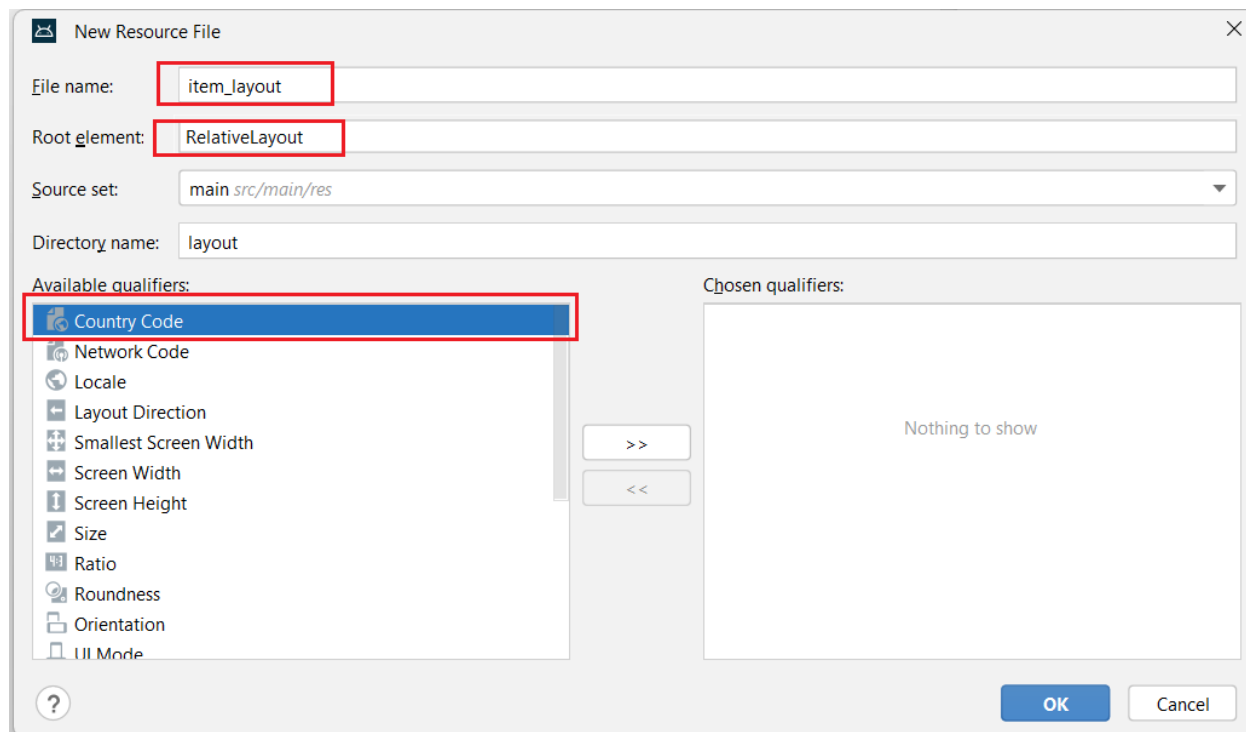


BƯỚC 4: TẠO LAYOUT CHO 1 ITEM

Vào layout -> New -> Layout Resource File



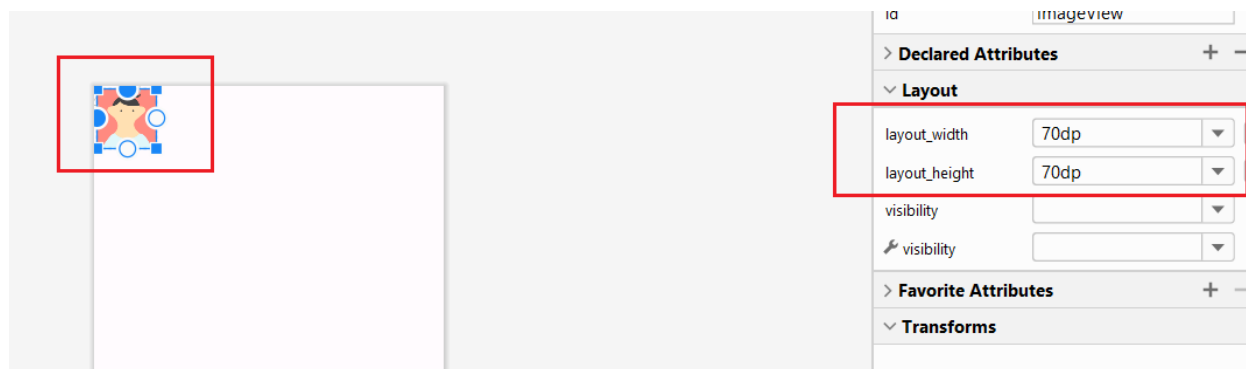
Đặt tên: item_layout; chọn RelativeLayout



BẮT ĐẦU THIẾT KẾ CÁC VIEW CHO item_layout, bao gồm 1 ảnh lưu hình, 1 textview lưu tên và 1 textview lưu số điện thoại. Lưu ý bạn cần chuẩn bị bộ ảnh có sẵn dạng file .png

KÉO THẢ ImageView VÀO item_layout

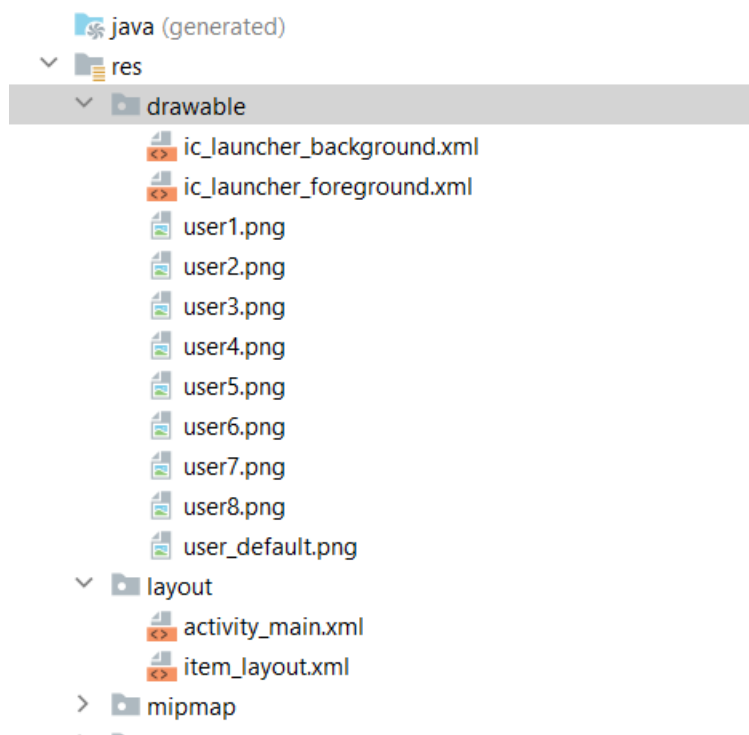
Set layout_width: 70dp và layout_height: 70dp



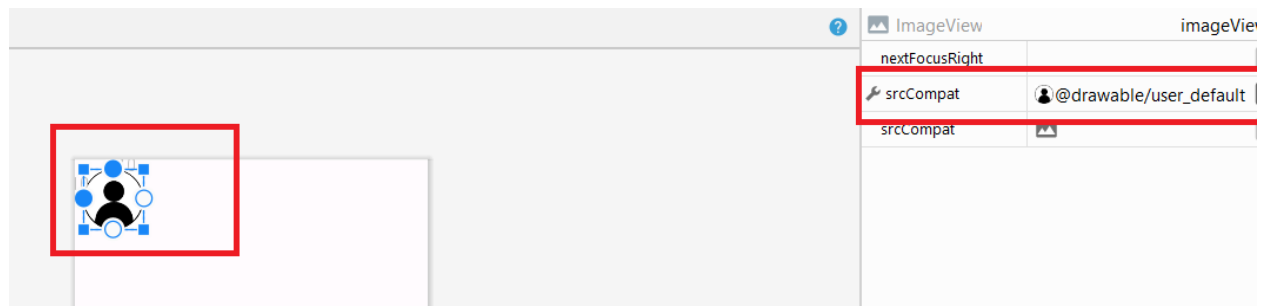
Chỉn layout_margin: 10dp để tạo khoảng cách với các view khác

TẠO CÁC AVARTA MẶC ĐỊNH

- Copy các ảnh có sẵn vào thư mục drawable -> được như hình sau

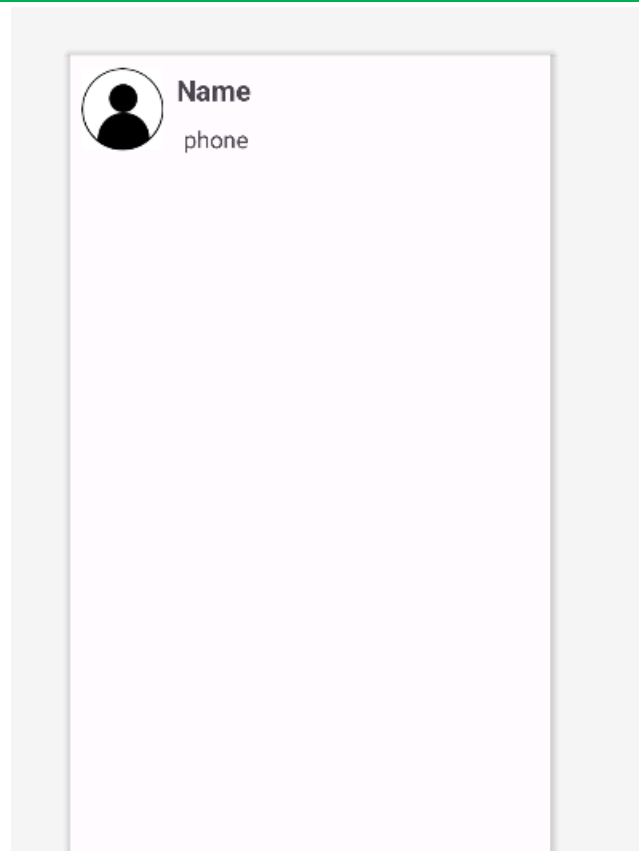


Set user default như sau:



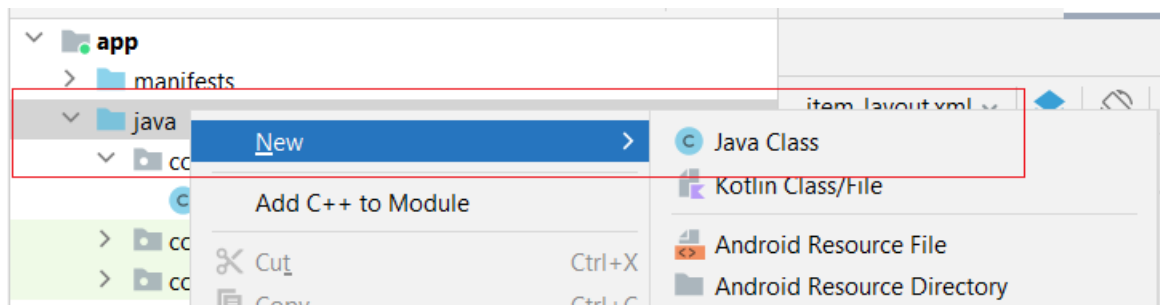
THÊM 2 TEXTVIEW ĐỂ HIỆN TÊN VÀ SỐ ĐIỆN THOẠI

- Đặt các id: txt_name, txt_phone
- Textsize: 24, text-bold



BUƯỚC 5: TẠO ContactModule

- Tại thư mục Java tạo mới 1 class java, đặt tên: contactModel.java



- Tạo các biến lưu các giá trị: name, phone, image

```
no usages
public class contactModel {
    no usages
    private String name;
    no usages
    private String phone;
    no usages
    private int image;
}
```

- Tạo các phương thức Getter và Setter

```
package com.example.demolistviewcontacts;
```

no usages

```
public class contactModel {
```

no usages

```
private String name;
```

no usages

```
private String phone;
```

no usages

```
private int imageId;
```

```
}
```

Show Context Actions Alt+Enter

Cut Ctrl+X

Copy Ctrl+C

Paste Ctrl+V

Copy / Paste Special >

Column Selection Mode Alt+Shift+Insert

Find in Files

Find Usages Alt+F7

Go To >

Folding >

Search with Google

Refactor >

Generate... Alt+Insert

Toggle Field Watchpoint

Open In >

Local History >

Compare with Clipboard

Create Gist...


```
package com.example.demolistviewcontacts;
```

no usages

```
public class contactModel {
```

no usages

```
private String name;
```

no usages

```
private String phone;
```

no usages

```
private int image;
```

```
}
```

Generate

Constructor

Getter

Setter

Getter and Setter

equals() and hashCode()

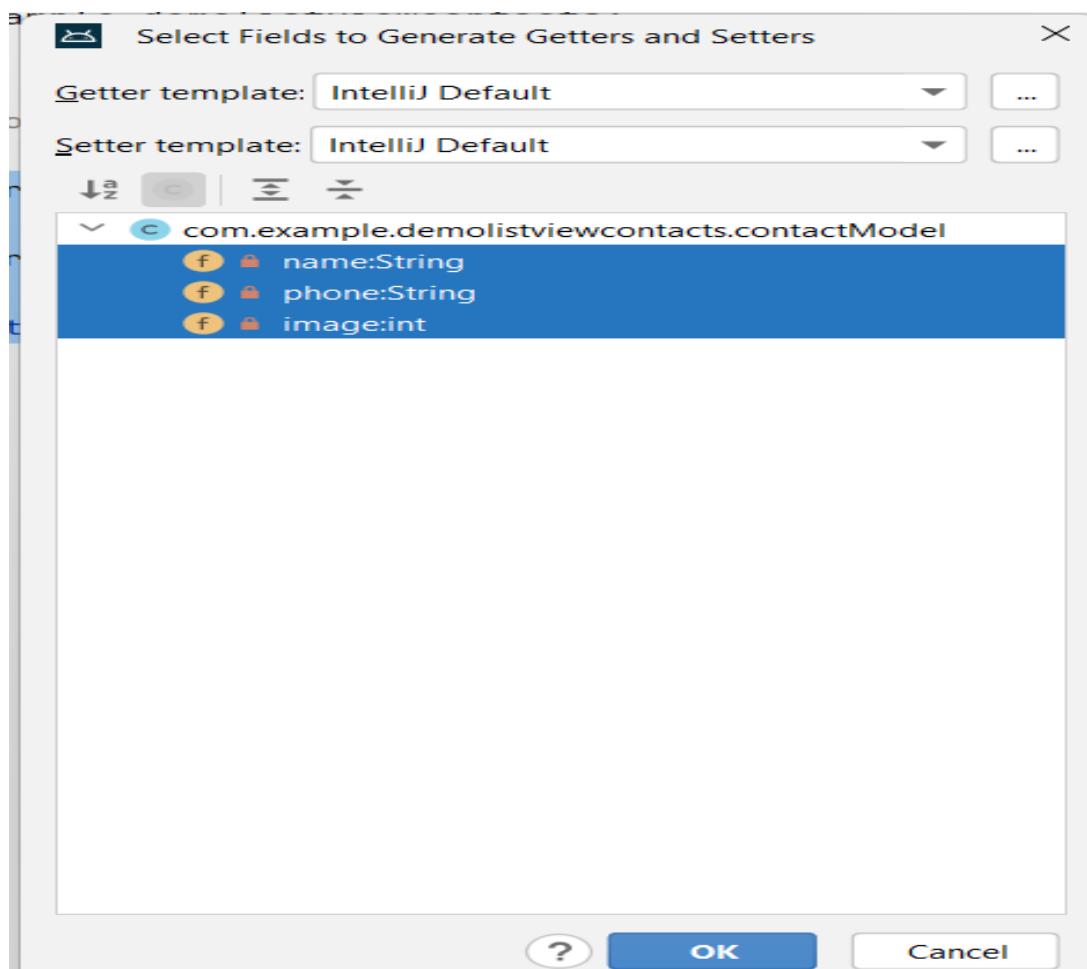
toString()

Override Methods... Ctrl+O

Delegate Methods...

Test...

Copyright



- Tạo Hàm khởi tạo các giá trị

no usages

```
public class contactModel {
```

3 usages

```
    private String name;
```

3 usages

```
    private String phone;
```

3 usages

```
    private int image;
```

```
    // Hàm khởi tạo
```

no usages

```
    public contactModel(String name, String phone, int image){
```

```
        this.name=name;
```

```
        this.phone=phone;
```

```
        this.image=image;
```

```
    }
```

no usages

```
    public String getName() {
```

```
        return name;
```

```
    }
```

no usages

```
    public void setName(String name) {
```

```
        this.name = name;
```

```
    }
```

BUƯỚC 6: TẠO ARRAYLIST

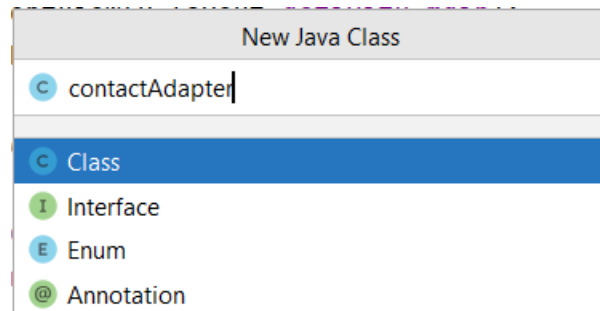
```
public class MainActivity extends AppCompatActivity {
    //KHỎI TẠO 1 ARRAYLIST ĐỂ CHỨA CONTACT
    no usages
    private ArrayList<contactModel> arraycontact = new ArrayList<>();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

- Tạo hàm khởi tạo data cho arraycontact

```
public class MainActivity extends AppCompatActivity {
    //KHỎI TẠO 1 ARRAYLIST ĐỂ CHỨA CONTACT
    8 usages
    private ArrayList<contactModel> arraycontact = new ArrayList<>();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        khoiTaoData();
    }
    //Khởi tạo data cho arraycontact
    1 usage
    private void khoiTaoData(){
        arraycontact.add(new contactModel( name: "Aang", phone: "0900000001",R.drawable.user1));
        arraycontact.add(new contactModel( name: "Katara", phone: "0900000002",R.drawable.user2));
        arraycontact.add(new contactModel( name: "Kyoshi", phone: "0900000003",R.drawable.user3));
        arraycontact.add(new contactModel( name: "Sokka", phone: "0900000004",R.drawable.user4));
        arraycontact.add(new contactModel( name: "Suki", phone: "0900000005",R.drawable.user5));
        arraycontact.add(new contactModel( name: "Toph", phone: "0900000006",R.drawable.user6));
        arraycontact.add(new contactModel( name: "TyLee", phone: "0900000007",R.drawable.user7));
        arraycontact.add(new contactModel( name: "Zuko", phone: "0900000008",R.drawable.user8));
    }
}
```

BƯỚC 7: TẠO CONTACT ADEPTER

- Tại thư mục Java tạo mới 1 class java, đặt tên: contactAdapter.java



- Extends class BaseAdapter và Implement methods như hình dưới

no usages

```
public class contactAdapter extends BaseAdapter {  
  
    @Override  
    public int getCount() {  
        return 0;  
    }  
  
    @Override  
    public Object getItem(int position) {  
        return null;  
    }  
  
    @Override  
    public long getItemId(int position) {  
        return 0;  
    }  
  
    @Override  
    public View getView(int position, View convertView, ViewGroup parent) {  
        return null;  
    }  
}
```

- Khởi tạo hàm contactAdapter

```
no usages
public class contactAdapter extends BaseAdapter {
    1 usage
    private Context context;
    1 usage
    private ArrayList<contactModel> arrayContact;
    //HÀM KHỞI TẠO CONTACTADAPTER
    no usages
    public contactAdapter(Context context, ArrayList<contactModel> array){
        this.context=context;
        this.arrayContact=array;
    }
}
```

Xây dựng các hàm Implement

- Hàm getCount()

```
// Hàm getCount đếm số item có trong listView do lấy dữ liệu từ arraylist
@Override
public int getCount() {
    if(arrayContact != null) {
        return arrayContact.size();
    }else
        return 0;
}
```

- Hàm getItem

```
//Trả về ModelData
@Override
public Object getItem(int position) {
    return arrayContact.get(position);
}
```

- Hàm getView

```
//TRẢ VỀ VIEW GIAO DIỆN CỦA 1 ITEM TRÊN LISTVIEW
```

```
@Override
```

```
public View getView(int position, View convertView, ViewGroup parent) {  
    LayoutInflater inflater = ((Activity)context).getLayoutInflater();  
    View rowView = inflater.inflate(R.layout.item_layout, parent, attachToRoot: false);  
  
    ImageView img = (ImageView) rowView.findViewById(R.id.imageView);  
    TextView lblName = (TextView) rowView.findViewById(R.id.txt_name);  
    TextView lblPhone = (TextView) rowView.findViewById(R.id.txt_phone);  
  
    contactModel model = arrayContact.get(position);  
    img.setImageResource(model.getImage());  
    lblName.setText(model.getName());  
    lblPhone.setText(model.getPhone());  
  
    return rowView;  
}
```

BƯỚC 8: TÌM LISTVIEW TRÊN GIAO DIỆN

```
private ArrayList<contactModel> arraycontact = new ArrayList<>();  
// KHAI BAO BIẾT LIST CONTACT  
2 usages  
private ListView listContact;  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    khoiTaoData();  
  
    listContact = (ListView) findViewById(R.id.list_contact);  
    contactAdapter adapter = new contactAdapter( context: this, arraycontact);  
  
    listContact.setAdapter(adapter);  
}
```

RUN BUILD