

Bài tập 25: Tiếp tục củng cố kiến thức Intent, ví dụ tổng hợp Quản Lý Nhân Viên

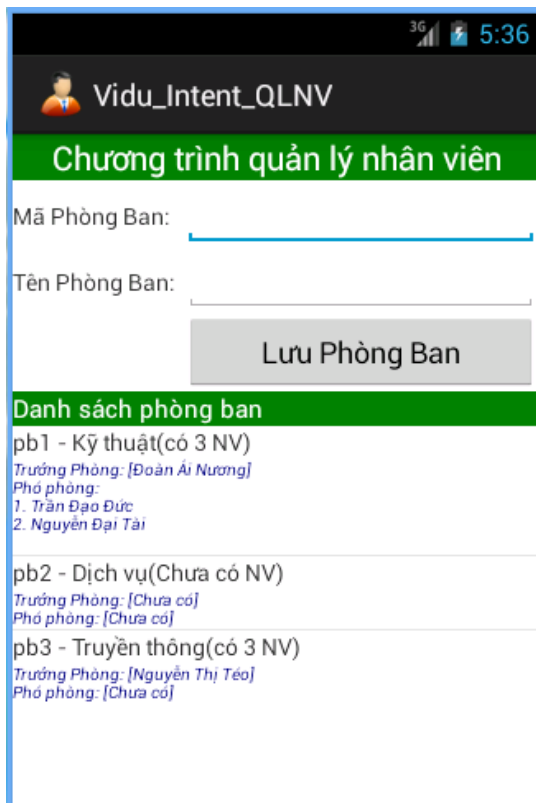
- Mục tiêu của bài tập này:

- 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:

- + Chương trình cho phép quản lý nhân viên + phòng ban.
- + Mỗi phòng ban sẽ có nhiều nhân viên, phòng ban cần lưu trữ (mã phòng ban, tên phòng ban)
- + 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)
- + 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

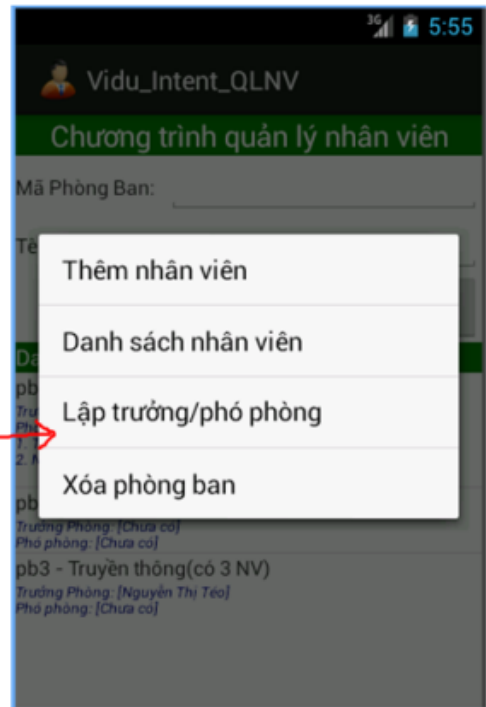
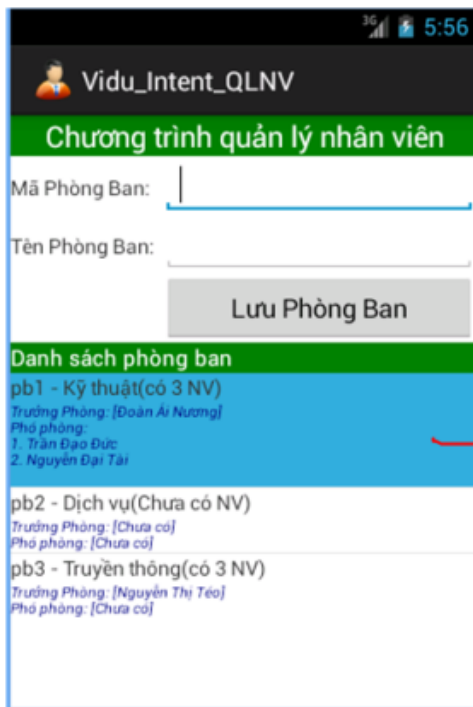
Sau đây là demo giao diện chính của chương trình:



- 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.

- Tiếp tục xử lý Context Menu cho ListView Phòng ban:



nhấn
lâu...

- Khi

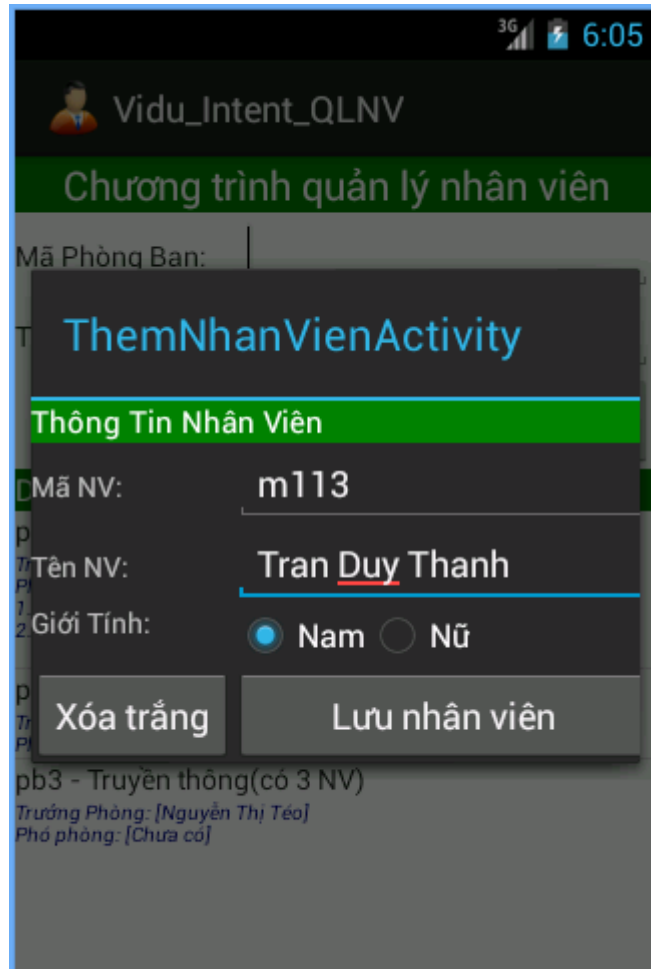
nhấn thật lâu vào một phòng ban nào đó, ví dụ trường hợp trên là phòng Kỹ Thuật, chương trình sẽ hiển thị context Menu có 3 chức năng:

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

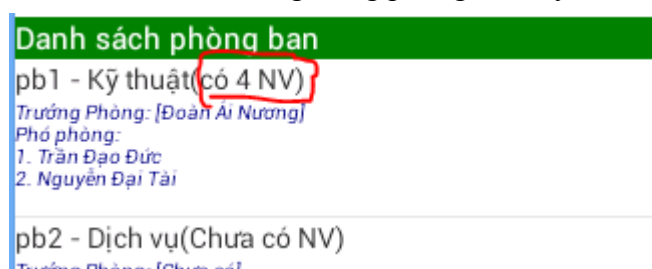
- Ứng với mỗi chức năng ta có:

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):



- 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). - Ví dụ ở hình trên nếu chọn “Lưu nhân viên”, màn hình này sẽ tắt đi để hiển thị lại màn hình chính, nhưng trong phòng ban Kỹ Thuật số lượng nhân viên sẽ tăng lên là 4, xem hình:



2. Xem danh sách nhân viên:

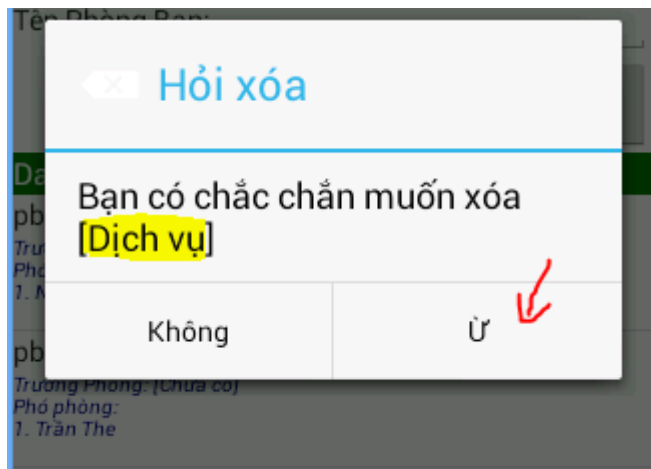
- Chương trình sẽ hiển thị danh sách nhân viên của phòng ban đang chọn, ví dụ trong trường hợp này là xem danh sách nhân viên của phòng ban Kỹ thuật:



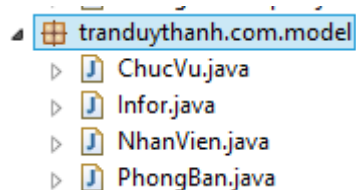
- Như bạn thấy đó. Màn hình xem danh sách nhân viên có các chức năng sau:
- + Nút mũi tên “Back” cho phép trở về màn hình chính.
- + Nhân viên có hình theo giới tính (nam, nữ)
- + Nhân viên có mô tả: Chức vụ, giới tính
- + Vậy ta cũng phải dùng Custom layout. (nhưng thời gian k cho phép nên ta bỏ qua phần custom này)

4. và xóa phòng ban.

- Ta qua chức năng cuối cùng: Xóa phòng ban, tương tự như xóa nhân viên. Chương trình cũng hỏi ta có chắc chắn muốn xóa hay không?



- Như vậy Tôi đã mô tả xong yêu cầu của bài toán:
- Bạn quan sát trong phần Source Code chia ra làm 3 Package:
- Bây giờ ta vào package : **model** :



- Package này có 3 class và 1 enum chức vụ (thực ra enum cũng là một class đặc biệt).
- Coding enum ChucVu.java:

```
Package tranduythanh.com.model;
```

```

/**
 * Enum này để thiết lập chức vụ cho nhân viên
 * @author drthanh
 *
 */
//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");
    PrivateString cv;
    ChucVu(String cv)
    {
        this.cv=cv;
    }
    Public String getChucVu()
    {
        returnthis.cv;
    }
}

```

- Coding 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 thôn tin /*
chung là mã, tên *
 */
Public class Infor implements Serializable{
    Private String ma;
    Private String ten;
    //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 ChucVu chucvu;
    // xây dựng hàm tạo
    Public boolean isGioitinh() {
        Return gioitinh;
    }
}

```

```

Public void setGioitinh(boolean gioitinh) {
    this.gioitinh = gioitinh;
}
Public ChucVu getChucvu() {
    Return chucvu;
}
//xây dựng các hàm get/set khác nếu cần
//ghi đè toString để quy định hiển thị dữ liệu
}
}

```

- Coding 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
     * @param nv
     */
    Public void themNv(NhanVien nv){
        //sv tự triển khai
    }
    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+="(Chua có NV) ";
        else str+="(có "+dsnv.size()+" NV) ";
        return str;
    }
}

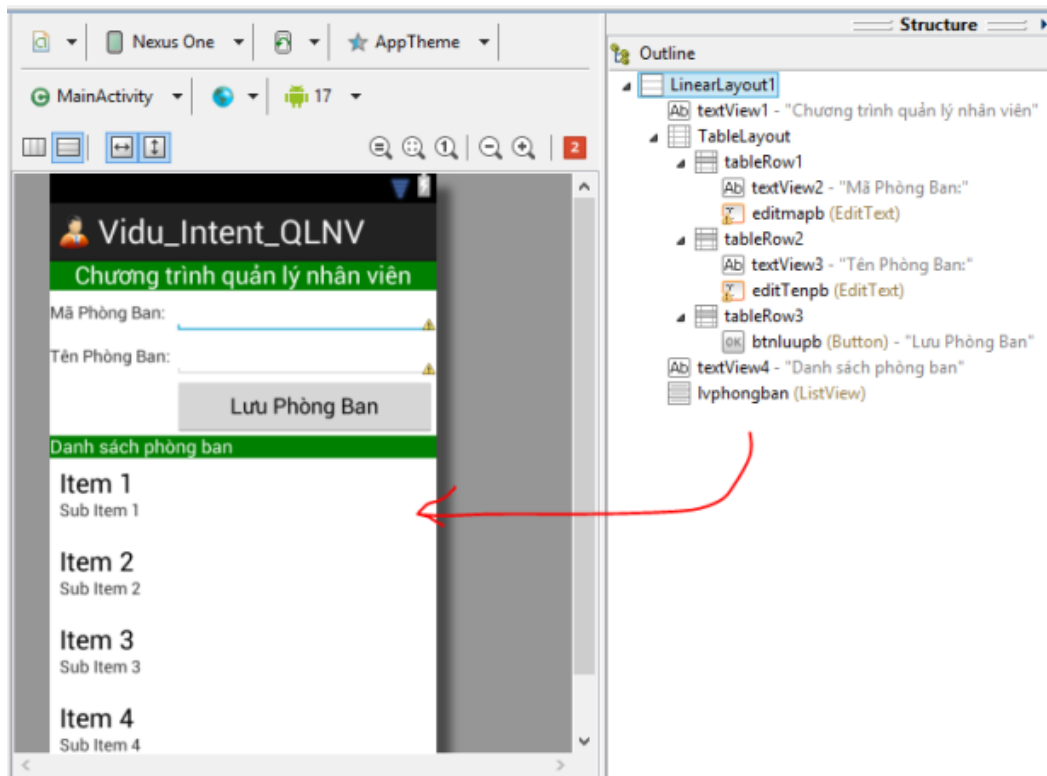
```

- Bây giờ ta qua package lưu các **activity**:

- Ta lần lượt vào từng màn hình theo thứ tự.

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

- Bạn xem Outline của màn hình chính chính:



- Ta xem xử lý coding của **MainActivity**:

```
Public class MainActivity extends Activity {
    //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();
    }
}
```

```

}
/**
 * Hàm này dùng để tải dữ liệu tạm (đỡ phải nhập mỗi tay)
 * Tôi dùng hàm này để ví dụ cho nhanh
 */
Public void fakeData()
{
    NhanVien nv=null;
    PhongBan pb=new PhongBan("pb1", "Kỹ thuật");
    nv=new NhanVien("m4", "Đoàn Ái Nương", true);
    pb.themNv(nv);
    nv=new NhanVien("m5", "Trần Đạo Đức", false);
    pb.themNv(nv);
    nv=newNhanVien("m6", "Nguyễn Đại Tài", false);
    pb.themNv(nv);
    arrPhongBan.add(pb);
    pb=newPhongBan("pb2", "Dịch vụ");
    arrPhongBan.add(pb);
    pb=newPhongBan("pb3", "Truyền thông");
    arrPhongBan.add(pb);
    nv=newNhanVien("m1", "Nguyễn Văn Téo", false);
    pb.themNv(nv);
    nv=newNhanVien("m2", "Nguyễn Thị Téo", true);
    pb.themNv(nv);
    nv=newNhanVien("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=??;
    editMapb??;
    editTenpb=??;
    lvpb=??;
    //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
 */
@Override
Public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenuInfo 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
 * ta dựa vào Id để xử lý
 */
@Override
Public boolean onContextItemSelected(MenuItem item) {
    switch(item.getItemId())
    {
        Case R.id.mnuthemnv:
            doThemNhanVien();
            break;
        case R.id.mnudanhsachnv:
            doDanhSachNhanVien();
            break;
        case R.id.mnuxoapb:
            doXoaPhongBan();
            break;
    }
    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
 * do đó ta 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
*/
@Override
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 học rồi, ko nói lại
        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);
}

```

```

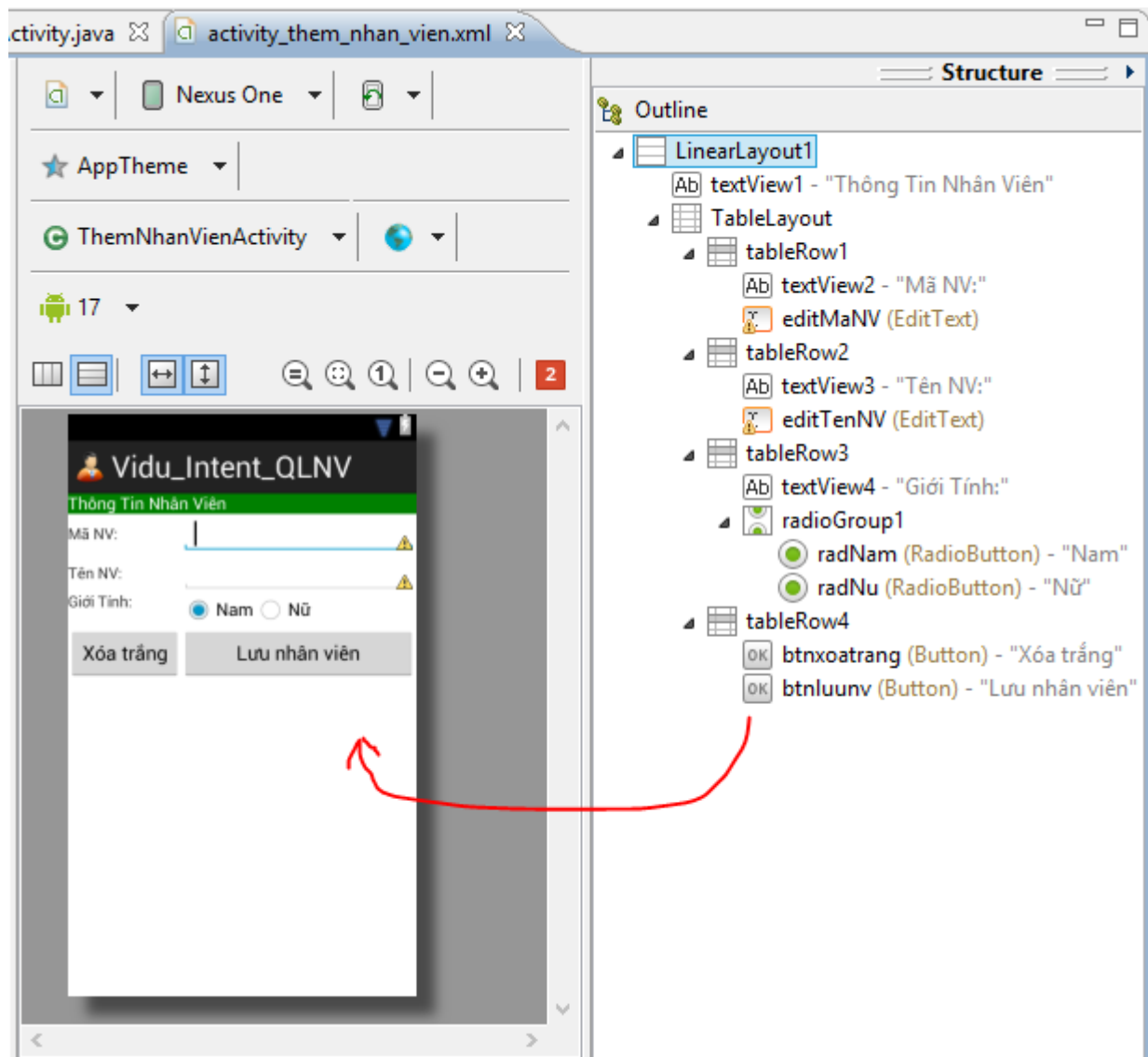
        startActivityForResult(i, XEM_DS_NHAN_VIEN);
    }

/**
 * hàm cho phép xóa phòng ban đang chọn
 */
Public void doXoaPhongBan()
{
    AlertDialog.Builder builder=newAlertDialog.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, int arg1) {
                arg0.cancel();
            }
        });
    builder.setPositiveButton("có",
        new DialogInterface.OnClickListener() {
            @Override
            Public void onClick(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;
}
}

```

- Ta qua màn hình thêm nhân viên. Chú ý là màn hình này bạn phải cấu hình mở dạng Dialog.
- xem layout:



- Giờ ta xem xử lý coding cho phần thêm nhân viên (**ThemNhanVienActivity**):

```
/**
 * class này cho phép thêm mới một nhân viên vào
 * Phòng ban đang chọn
 *
 */
Public class ThemNhanVienActivity extends Activity {

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();
}
```

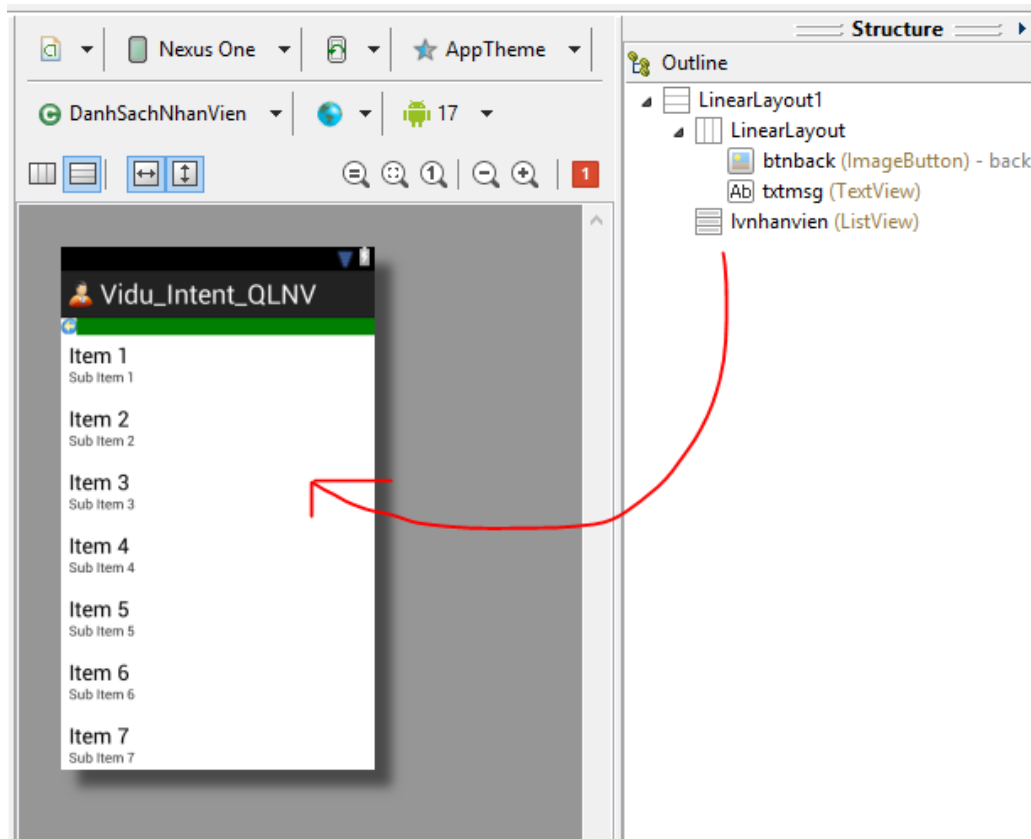
```

}
/**
 * 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();
}
}

```

- Tiếp tục ta xem màn hình Xem danh sách nhân viên:

- bạn xem outline:



- Xem coding xử lý phần xem danh sách nhân viên (**DanhSachNhanVienActivity**) :

```
/**
 * class này dùng để hiển thị danh sách nhân viên
 */
Public class DanhSachNhanVienActivity extends Activity {

    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 (đã quá quen thuộc rồi)
    * có lưu vết lại nhân viên vừa chọn để xử lý
    * cho contextmenu
    */
    Public void addEvents()
    {
        btnback.setOnClickListener(new OnClickListener() {
            @Override
            Public void onClick(View arg0) {
                finish() ;
            }
        });
    }
}

```