



## **SAMPLE PROJECT**

### **BÀI 5: THỰC HIỆN DỰ ÁN 3**

- ⊙ Kết thúc bài học này bạn có khả năng
  - ⊙ Thực hiện chức năng quản lý thể loại sách
  - ⊙ Hiển thị danh sách các thể loại sách
  - ⊙ Thực hiện chức năng quản lý sách trong cửa hàng
  - ⊙ Hiển thị danh sách các đầu sách
  - ⊙ Thực hiện chức năng quản lý hoá đơn

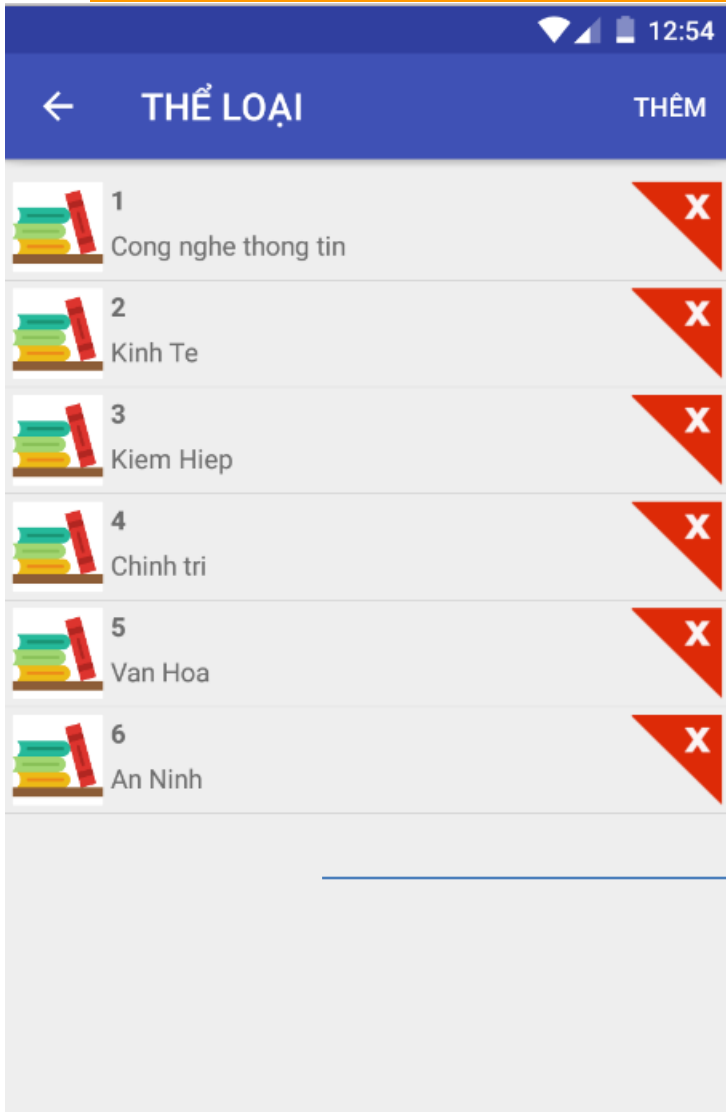


- 📖 Giới thiệu màn hình quản lý thể loại Sách
- 📖 Tổ chức mã cho chức năng quản lý thể loại Sách
- 📖 Giới thiệu màn hình quản lý Sách
- 📖 Tổ chức mã cho chức năng quản lý Sách

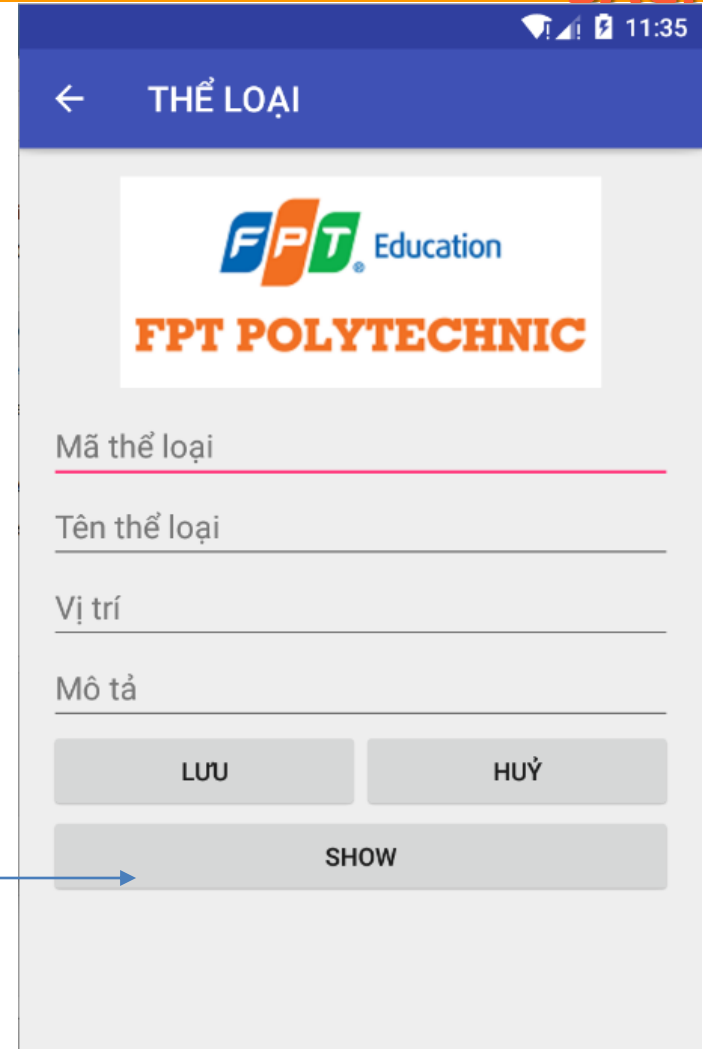


# Hướng dẫn viết mã cho màn hình quản lý thể loại Sách

# MÀN HÌNH CỦA TRANG QUẢN LÝ THỂ LOẠI SÁCH



Khi click thêm màn hình nhập thể loại sẽ xuất hiện



- ❑ Cài đặt mã cho các chức năng trên giao diện là viết mã điều khiển các sự kiện xảy ra trên giao diện nhằm thực hiện đúng mong muốn nghiệp vụ ứng dụng.
- ❑ Trong bài 3, chúng ta đã tạo lớp TheoLoaiDAO, chứa các phương thức:
  - ❖ **int** insertTheLoai(TheLoai theLoai)
  - ❖ **List**<TheLoai> getAllTheLoai()
  - ❖ **int** updateTheLoai(TheLoai theLoai)
  - ❖ **int** deleteTheLoaiByID(String matheloai)
- ❑ Bài này chúng ta viết các các sự kiện trên màn hình gọi các phương thức này

```

public class TheLoaiDAO {
    private SQLiteDatabase db;
    private DatabaseHelper dbHelper;

    public static final String TABLE_NAME = "TheLoai";
    public static final String SQL_THE_LOAI = "CREATE TABLE TheLoai (matheloai text primary key, " +
        "tentheloi text, mota text, vitri int);";
    public static final String TAG = "TheLoaiDAO";
    public TheLoaiDAO(Context context) {
        dbHelper = new DatabaseHelper(context);
        db = dbHelper.getWritableDatabase();
    }
    //insert
    public int insertTheLoai(TheLoai theLoai){
        ContentValues values = new ContentValues();
        values.put("matheloai", theLoai.getMaTheLoai());
        values.put("tentheloi", theLoai.getTenTheLoai());
        values.put("mota", theLoai.getMoTa());
        values.put("vitri", theLoai.getViTri());
        if (checkPrimaryKey(theLoai.getMaTheLoai())){
            int result = db.update(TABLE_NAME, values, "matheloai=?", new String[]{theLoai.getMaTheLoai()});
            if (result == 0){
                return -1;
            }
        }else {
            try {
                if (db.insert(TABLE_NAME, null, values) == -1) {
                    return -1;
                }
            } catch (Exception ex) {
                Log.e(TAG, ex.toString());
            }
        }
        return 1;
    }
}

```

```

//getAllEmployee
public List<TheLoai> getAllTheLoai(){
    List<TheLoai> dsTheLoai = new ArrayList<>();
    Cursor c = db.query(TABLE_NAME,null,null,null,null,null,null);
    c.moveToFirst();
    while (c.isAfterLast()==false){
        TheLoai ee = new TheLoai();
        ee.setMaTheLoai(c.getString(0));
        ee.setTenTheLoai(c.getString(1));
        ee.setMoTa(c.getString(2));
        ee.setViTri(c.getInt(3));
        dsTheLoai.add(ee);
        Log.d("//=====",ee.toString());
        c.moveToNext();
    }
    c.close();
    return dsTheLoai;
}

//update
public int updateTheLoai(TheLoai theLoai){
    ContentValues values = new ContentValues();
    values.put("matheLoai",theLoai.getMaTheLoai());
    values.put("tentheLoai",theLoai.getTenTheLoai());
    values.put("mota",theLoai.getMoTa());
    values.put("vitri",theLoai.getViTri());
    int result = db.update(TABLE_NAME,values,"matheLoai=?", new String[]{theLoai.getMaTheLoai()});
    if (result == 0){
        return -1;
    }
    return 1;
}

```

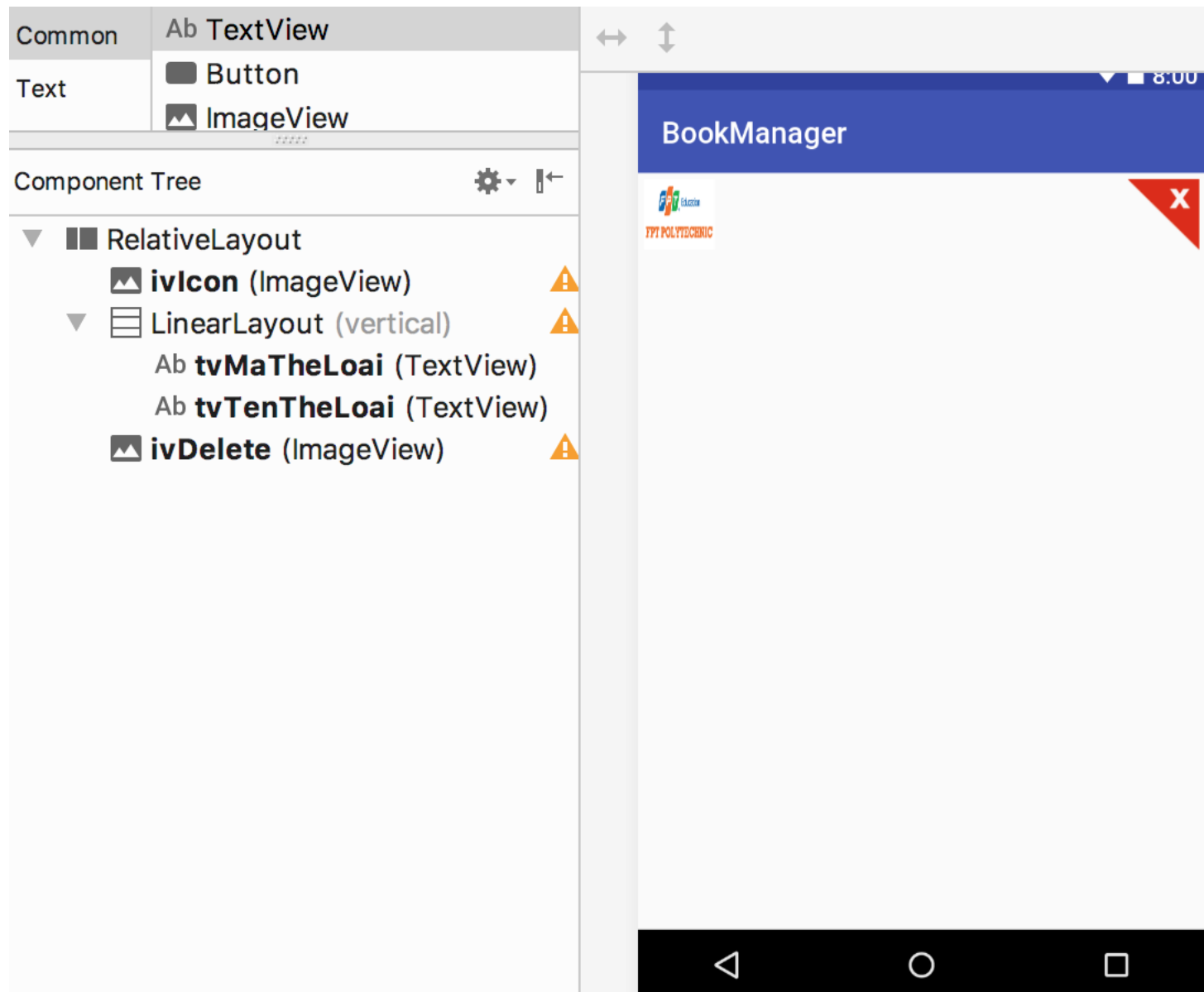


```
//delete
public int deleteTheLoaiByID(String matheloai){
    int result = db.delete(TABLE_NAME,"matheloai=?",new String[]{matheloai});
    if (result == 0)
        return -1;
    return 1;
}

//check
public boolean checkPrimaryKey(String strPrimaryKey){
    //SELECT
    String[] columns = {"matheloai"};
    //WHERE clause
    String selection = "matheloai=?";
    //WHERE clause arguments
    String[] selectionArgs = {strPrimaryKey};
    Cursor c = null;
    try{
        c = db.query(TABLE_NAME, columns, selection, selectionArgs, null, null, null);
        c.moveToFirst();
        int i = c.getCount();
        c.close();
        if(i <= 0){
            return false;
        }
        return true;
    }catch(Exception e){
        e.printStackTrace();
        return false;
    }
}
```

- ❑ Tạo một file layout cho các dòng (item) trong listView.
- ❑ Tạo class CustomAdapter để custom listview cho layout đã tạo phía trên.
- ❑ Gọi CustomAdapter để lấy dữ liệu

# FILE LAYOUT CHO MỘT ITEM



```

public class TheLoaiAdapter extends BaseAdapter {
    List<TheLoai> arrTheLoai;
    public Activity context;
    public LayoutInflater inflater;
    TheLoaiDAO theLoaiDAO;

    public TheLoaiAdapter(Activity context, List<TheLoai> arrayTheLoai) {
        super();
        this.context = context;
        this.arrTheLoai = arrayTheLoai;
        this.inflater = (LayoutInflater)context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        theLoaiDAO = new TheLoaiDAO(context);
    }

    @Override
    public int getCount() { return arrTheLoai.size(); }

    @Override
    public Object getItem(int position) { return arrTheLoai.get(position); }

    @Override
    public long getItemId(int position) { return 0; }
    public static class ViewHolder {

        ImageView img;
        TextView txtMaTheLoai;
        TextView txtTenTheLoai;
        ImageView imgDelete;
    }
}

```

```

@Override
public View getView(final int position, View convertView, ViewGroup parent) {
    ViewHolder holder;

    if(convertView==null)
    {
        holder = new ViewHolder();
        convertView = inflater.inflate(R.layout.item_theloai, root: null);
        holder.img = (ImageView) convertView.findViewById(R.id.ivIcon);
        holder.txtMaTheLoai = (TextView) convertView.findViewById(R.id.tvMaTheLoai);
        holder.txtTenTheLoai = (TextView) convertView.findViewById(R.id.tvTenTheLoai);
        holder.imgDelete = (ImageView)convertView.findViewById(R.id.ivDelete);
        holder.imgDelete.setOnClickListener((v) -> {
            theLoaiDAO.deleteTheLoaiByID(arrTheLoai.get(position).getMaTheLoai());
            arrTheLoai.remove(position);
            notifyDataSetChanged();
        });
        convertView.setTag(holder);
    }
    else
        holder=(ViewHolder)convertView.getTag();

    TheLoai _entry = (TheLoai) arrTheLoai.get(position);
    holder.img.setImageResource(R.drawable.emone);
    holder.txtMaTheLoai.setText(_entry.getMaTheLoai());
    holder.txtTenTheLoai.setText(_entry.getTenTheLoai());

    return convertView;
}

@Override
public void notifyDataSetChanged() { super.notifyDataSetChanged(); }

public void changeDataset(List<TheLoai> items){
    this.arrTheLoai = items;
    notifyDataSetChanged();
}

```

```
public class ListTheLoaiActivity extends AppCompatActivity {  
    public static List<TheLoai> dsTheLoai = new ArrayList<>();  
    ListView lvTheLoai;  
    TheLoaiAdapter adapter = null;  
    TheLoaiDAO theLoaiDAO;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_list_the_loai);  
        lvTheLoai = (ListView) findViewById(R.id.lvTheLoai);  
  
        theLoaiDAO = new TheLoaiDAO( context: ListTheLoaiActivity.this);  
        dsTheLoai = theLoaiDAO.getAllTheLoai();  
  
        adapter = new TheLoaiAdapter( context: this, dsTheLoai);  
        lvTheLoai.setAdapter(adapter);  
    }  
}
```



## **SAMPLE PROJECT**

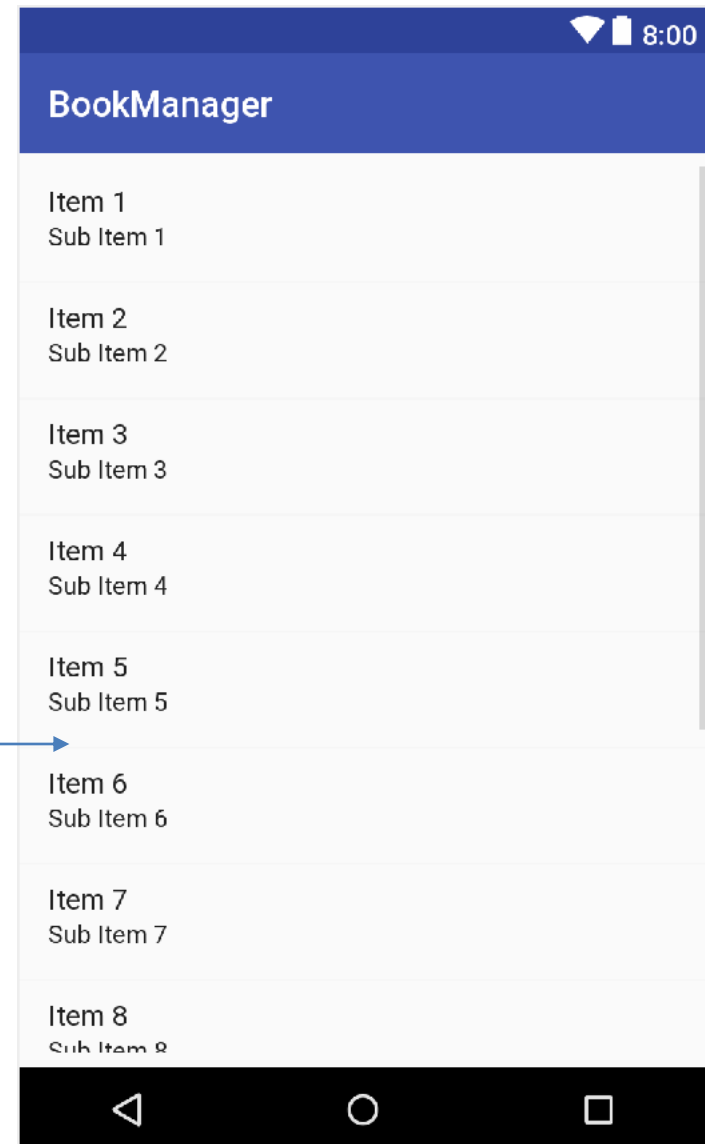
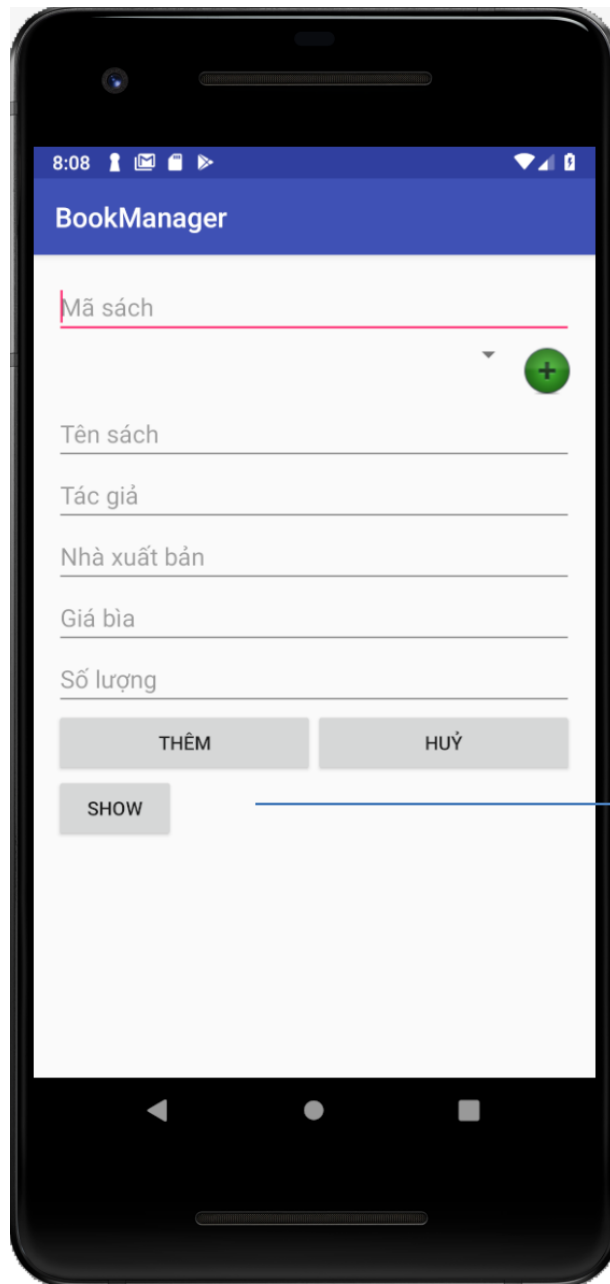
### **BÀI 5: THỰC HIỆN DỰ ÁN**

# Hướng dẫn viết mã cho màn hình quản lý Sách



# CHỨC NĂNG THÊM MỚI SÁCH

Khi người  
dùng  
kích vào  
nút  
"Show"  
thì màn  
hình  
danh  
sách các  
quyển  
sách hiển  
thị



- ❑ Cài đặt mã cho các chức năng trên giao diện là viết mã điều khiển các sự kiện xảy ra trên giao diện nhằm thực hiện đúng mong muốn nghiệp vụ ứng dụng.
- ❑ Trong bài 3, chúng ta đã tạo lớp SachDAO, chứa các phương thức:
  - ❖ **int** insertSach(Sach s)
  - ❖ **List**<Sach> getAllSach()
  - ❖ **int** updateSach(Sach s)
  - ❖ **int** deleteSachByID(String maSach)
- ❑ Bài này chúng ta viết các các sự kiện trên màn hình gọi các phương thức này

# CÀI ĐẶT MÃ CHO CHỨC NĂNG THÊM SÁCH

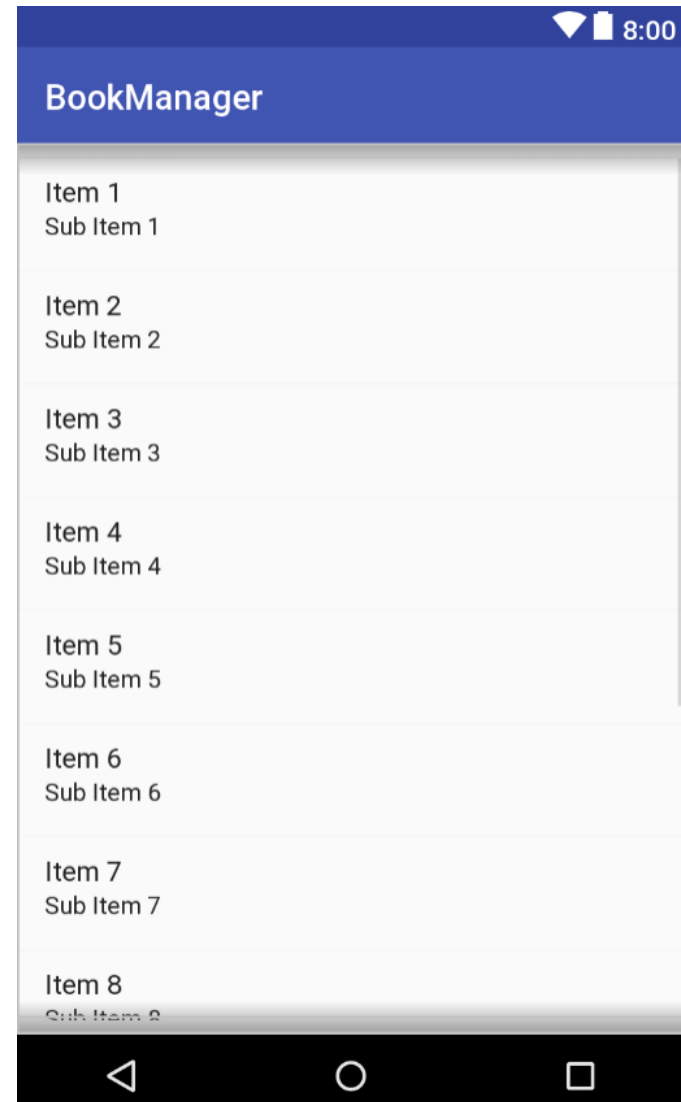
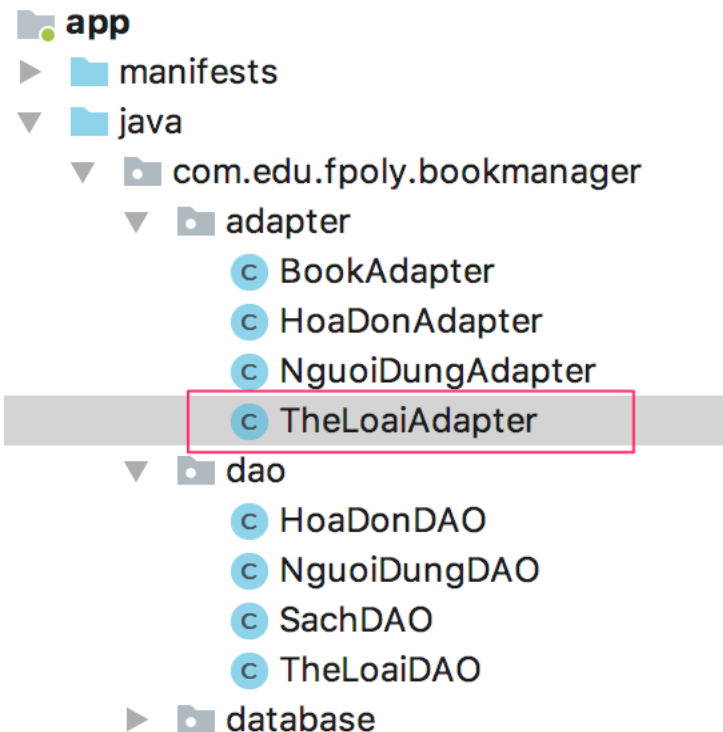
```
public class ThemSachActivity extends AppCompatActivity {
    SachDAO sachDAO;
    TheLoaiDAO theLoaiDAO;
    Spinner spnTheLoai;
    EditText edMaSach, edTenSach, edNXB, edTacGia, edGiaBia, edSoLuong;
    String maTheLoai = "";
    List<TheLoai> listTheLoai = new ArrayList<>();
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sach);
        spnTheLoai = (Spinner) findViewById(R.id.spnTheLoai);
        getTheLoai();
        edMaSach = (EditText) findViewById(R.id.edMaSach);
        edTenSach = (EditText) findViewById(R.id.edTenSach);
        edNXB = (EditText) findViewById(R.id.edNXB);
        edTacGia = (EditText) findViewById(R.id.edTacGia);
        edGiaBia = (EditText) findViewById(R.id.edGiaBia);
        edSoLuong = (EditText) findViewById(R.id.edSoLuong);
        //
        spnTheLoai.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
                maTheLoai = listTheLoai.get(spnTheLoai.getSelectedItemPosition()).getMaTheLoai();
            }

            @Override
            public void onNothingSelected(AdapterView<?> parent) {
            }
        });
    }
}
```

```
public void showSpinner(View view){
    sachDAO = new SachDAO( context: ThemSachActivity.this);
    sachDAO.getAllSach();
}
public void getTheLoai(){
    theLoaiDAO = new TheLoaiDAO( context: ThemSachActivity.this);

    listTheLoai = theLoaiDAO.getAllTheLoai();
    ArrayAdapter<TheLoai> dataAdapter = new ArrayAdapter<~>( context: this,
        android.R.layout.simple_spinner_item, listTheLoai);
    dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spnTheLoai.setAdapter(dataAdapter);
}
public void addBook(View view){
    sachDAO = new SachDAO( context: ThemSachActivity.this);
    Sach sach = new Sach(edMaSach.getText().toString(), maTheLoai, edTenSach.getText().toString(),
        edTacGia.getText().toString(), edNXB.getText().toString(),
        Double.parseDouble(edSoLuong.getText().toString()), Integer.parseInt(edSoLuong.getText().toString()));
    try {
        if (sachDAO.inserSach(sach) > 0) {
            Toast.makeText(getApplicationContext(), text: "Thêm thành công", Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(getApplicationContext(), text: "Thêm thất bại", Toast.LENGTH_SHORT).show();
        }
    } catch (Exception ex) {
        Log.e( tag: "//=====", ex.toString());
    }
}
```

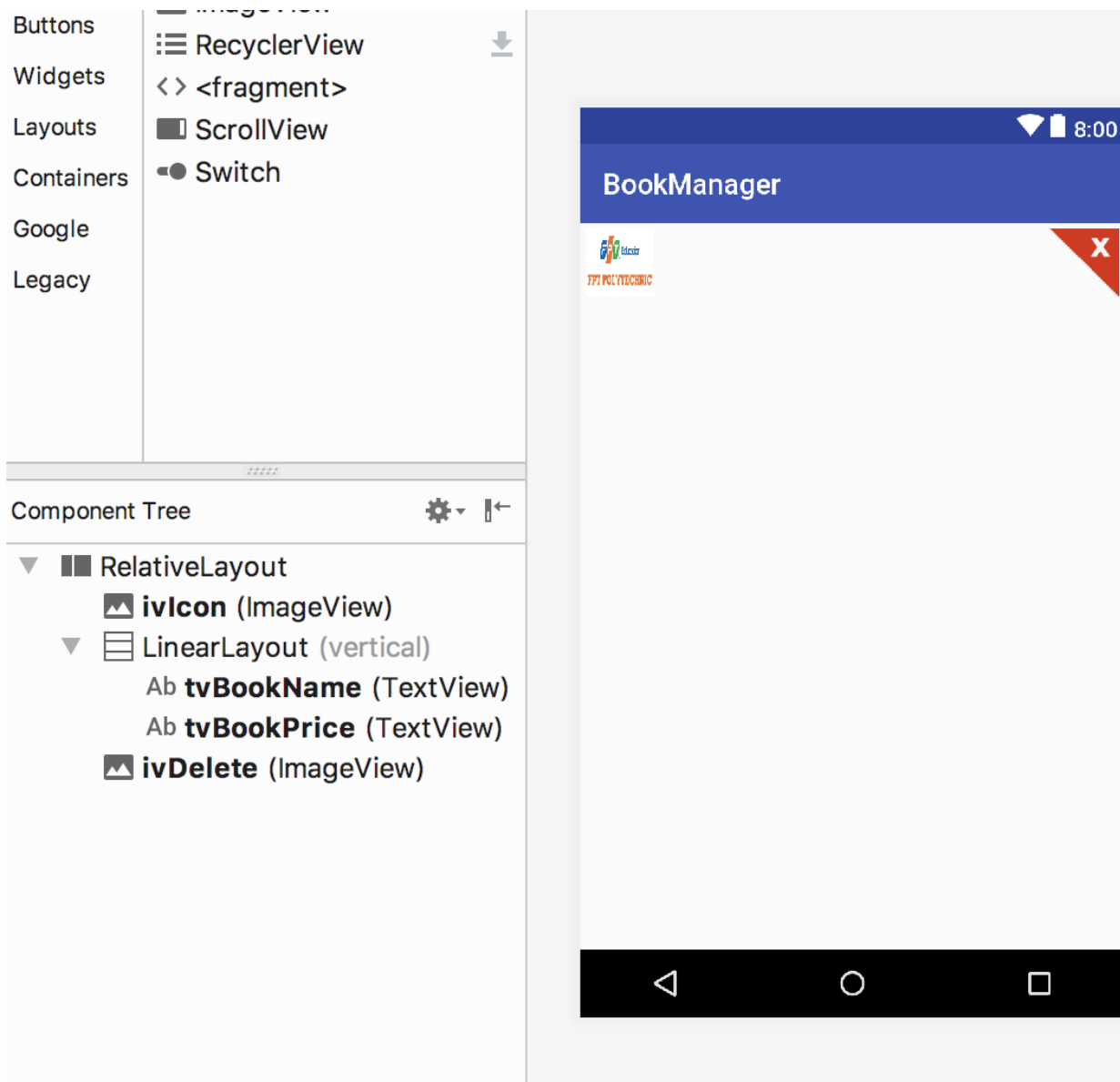
Danh sách các sách được hiển thị dưới dạng các ListView.  
Cần phải tạo lớp BookAdapter để custom



- ❑ Tạo một file layout cho các dòng (item) trong listView.
- ❑ Tạo class CustomAdapter để custom listview cho layout đã tạo phía trên.
- ❑ Gọi CustomAdapter để lấy dữ liệu

Tương tự như cách làm TheLoaiAdapter

# FILE LAYOUT CHO MỘT ITEM



- ❑ Kết nối giữa 3 tầng của ứng dụng
  - ❖ Giao diện của ứng dụng
  - ❖ Các lớp model
  - ❖ Các lớp DAO
- ❑ Tổ chức và điều khiển sự kiện
- ❑ Custom giao diện ListView







**Cảm ơn**