Môn học: Lập trình Android MOB1032

Gồm: 8 bài Lab + 8 bài quiz

Cần nắm chắc kiến thức cơ bản của môn học này để hoàn thành các môn học tiếp theo!

Sau môn học này, lớp sẽ học tiếp các môn liên quan: Thiết kế giao diện Android, Dự án mẫu, Lập trình android nâng cao, Android networking, Lập trình server cho Android, Dự án 1 : Là những môn liên quan kiến thức android cơ bản.

Cần:

* Kiến thức lập trình Java: Kiến thức cơ bản như sử dụng biến, lệnh điều khiển if..else, lặp for, while, kiến thức về hàm, lớp, đối tượng (thuộc tính, phương thức), kế thừa…
* Kiến thức về CSDL

Công cụ:

* Sử dụng phần mềm Android Studio https://developer.android.com/studio
* Sử dụng Ultraviewer để hỗ trợ remote: <https://ultraviewer.net/en/download.html>
* Hướng dẫn cài đặt phần mềm và máy ảo:  
   <https://zezo.dev/tong-quan-ve-android-24/tao-thiet-bi-ao-chay-ung-dung-android.27.html>   
  Nếu máy tính yếu thì có thể cài đặt https://www.genymotion.com/ hoặc LDPlayer hoặc nox để làm máy ảo thay thế cho máy ảo có sẵn của Android studio.

Trường hợp lỗi máy ảo của android studio không chạy được, bị báo lỗi HAXM thì dùng genymotion thay thế <https://www.genymotion.com/> hoặc nox player

B2. Thực hành

Tạo ứng dụng hiển thị thông tin cá nhân

B3: Làm việc với activity

* Viết ứng dụng hiển thị: Họ tên, điện thoại. Bấm vào họ tên thì thông báo thông tin, bấm vào số điện thoại thì gọi điện

Tìm hiểu các thành phần trong android:

VD ứng dụng bán hàng

- Quản lý sản phẩm: CRUD, thống kê sp...

- Quản lý người dùng: đăng ký, đăng nhập, danh sách, cập nhật ... tách activity

- Quản lý đơn hàng

Activity bao gồm: file code java để điều khiển, file layout xml để thiết kế giao diện

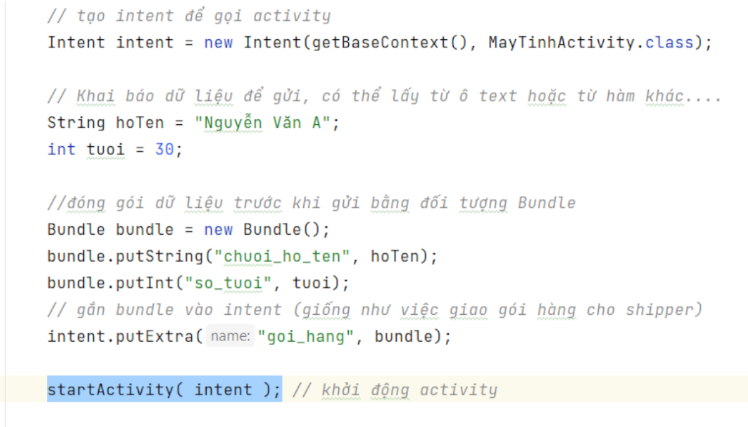
B4 - Vòng đời activity

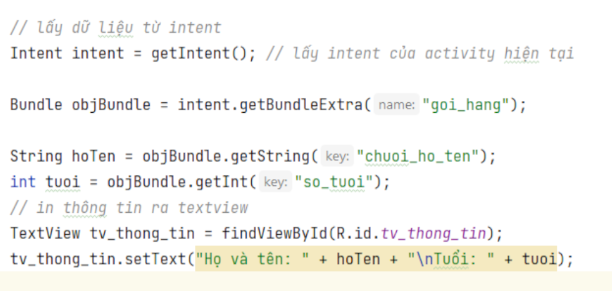
Ví dụ thực hành: Tạo 1 ứng dụng có 2 activity

* Activity main có 1 nút bấm, khi bấm nút sẽ gọi activity máy tính
* Activity máy tính: Hiển thị 2 ô text và nút bấm, khi bấm nút thì tính tổng 2 số nhập ở 2 ô text.

B5 - Thực hành

* Truyền dữ liệu từ activity A -> activity B.



Xử lý ở activity nhận:  


**Bài tự thực hành trực tiếp trên lớp:**

Viết ứng dụng có 2 activity: **Main và Info**

- Khi vào ứng dụng hiển thị activity Main có giao diện gồm 1 ô text để nhập email và 1 ô text nhập password, có 1 nút bấm đăng nhập. Khi người dùng nhập email và password và bấm nút đăng nhập thì kiểm tra nếu đúng là **admin@gmail.com** và pass **123** thì thông báo dạng Toast “Đăng nhập thành công” và tiến hành hiển thị Activity Info. Thực hiện gửi thông tin đăng nhập sang activity Info.

- Bên activity Info hiển thị email, password của người vừa đăng nhập trên 1 textview

- Sau khi làm xong 2 chức năng trên, bên activity info: Thiết kế thêm 1 ô text để nhập số điện thoại và nút bấm, nhập vào số điện thoại và bấm nút thì thực hiện gọi điện.

B6 - Làm việc với layout

* Constraint layout
  + Là layout ràng buộc, các view trong layout này phải có ràng buộc mới sắp xếp được vị trí trên giao diện.
  + Các ràng buộc của 1 view trên layout: Tối thiểu phải có 3 ràng buộc: top, left, right. Sẽ có những thuộc tính yêu cầu phải có đủ 4 ràng buộc cho 4 cạnh.
  + Sử dụng GuideLine (Một đường kẻ ảo trong lúc thiết kế) để làm đường lưới giúp ràng buộc các view dễ dàng hơn.   
    <androidx.constraintlayout.widget.Guideline

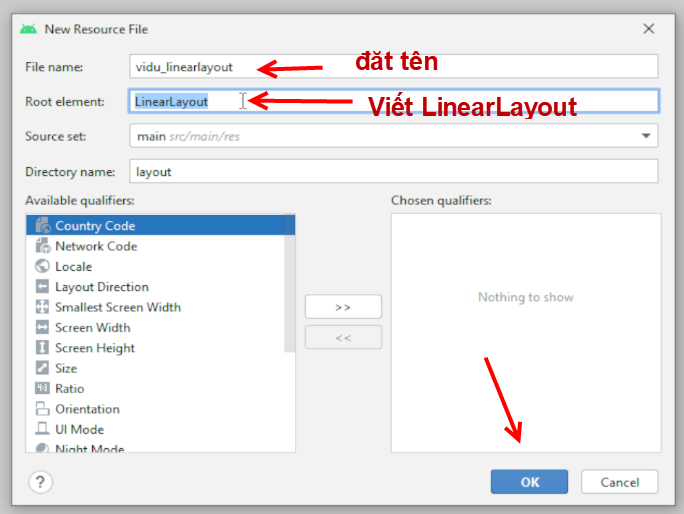
android:id="@+id/guideline3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="horizontal" // xác định hướng dọc hoặc ngang

app:layout\_constraintGuide\_begin="114dp" // khoảng cách từ viền trên hoặc viền trái của màn hình tới đường kẻ  
 />

* + Nếu muốn thiết lập 1 view nào đó chiếm toàn bộ không gian còn lại thì sử dụng thuộc tính layout\_width hoặc layout\_height là 0dp kết hợp phải có đủ 4 ràng buộc cho view đó.
* LinearLayout:
  + Là loại layout cho phép sắp xếp tự động các view con theo thứ tự hàng ngang hoặc hàng dọc.
  + 
  + Thuộc tính quan trọng của layout là: orientation (xác định hướng sắp xếp dọc hoặc ngang)

Nếu muốn thiết lập 1 view nào đó chiếm khoảng trống còn lại thì đặt kích thước là 0dp và kết hợp layout\_weight là 1.   
<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:background="#D3E4FF"

android:layout\_margin="8dp"

>

<TextView

android:layout\_width="100dp"

android:layout\_height="wrap\_content"

android:text="Password"

android:padding="8dp"

android:gravity="right"

/>

<EditText

android:layout\_width="0dp"

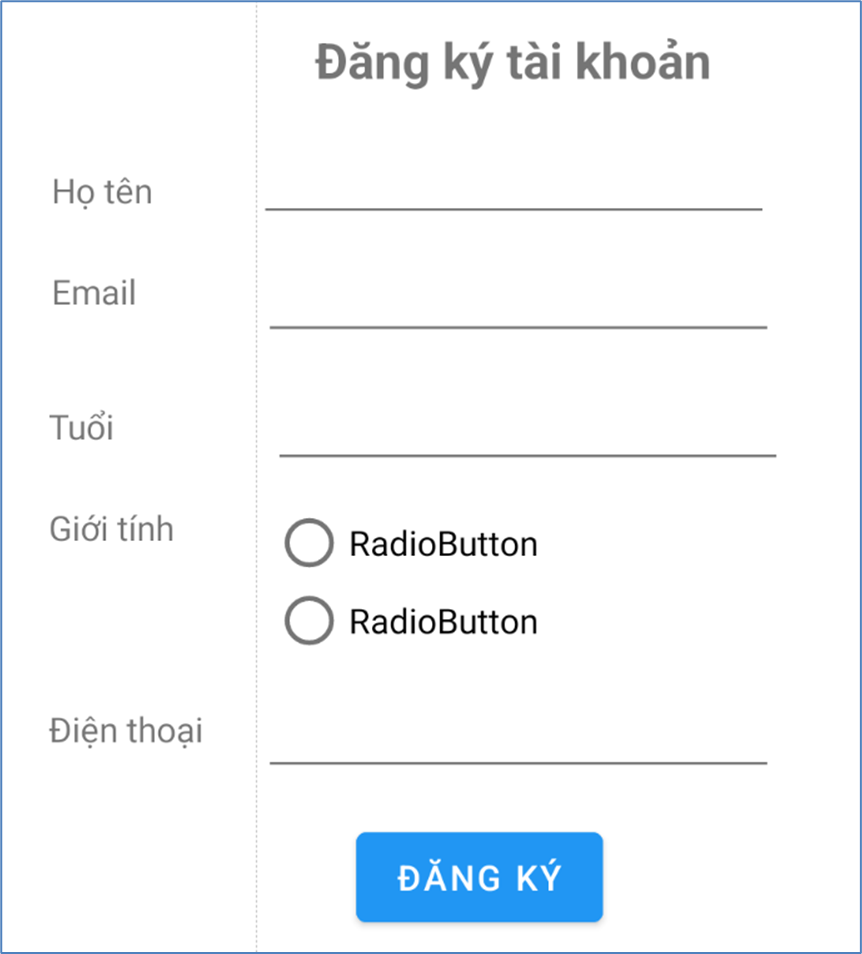
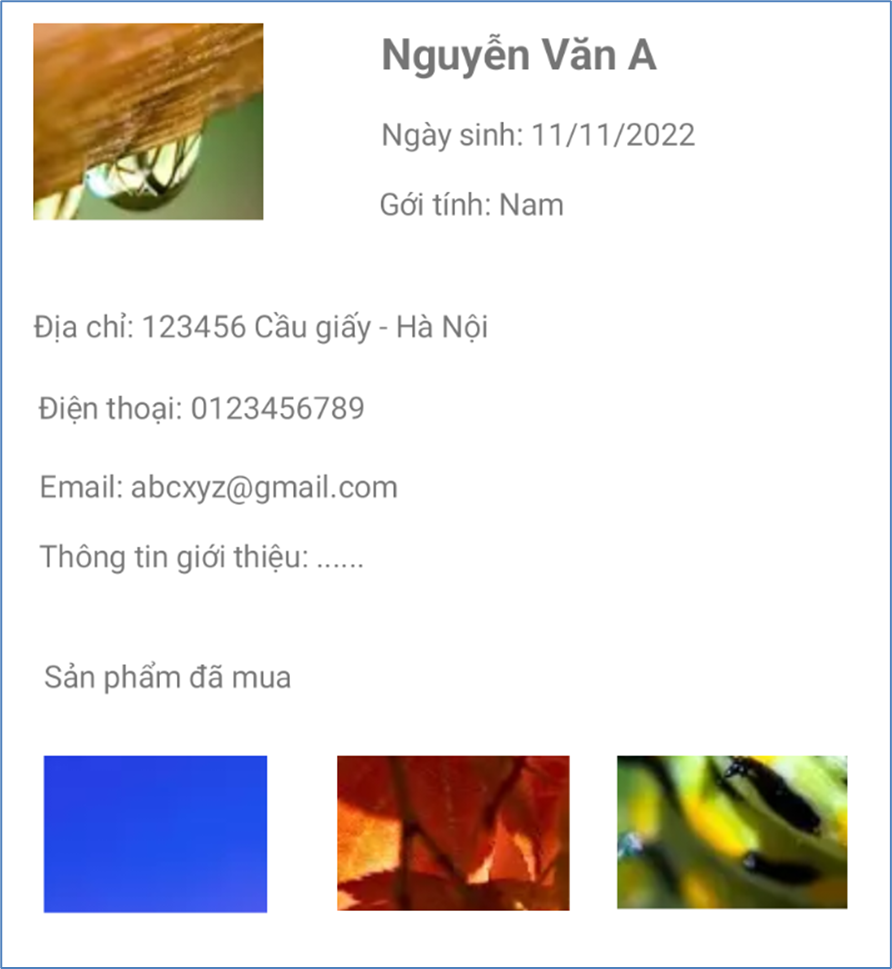
android:layout\_weight="1"

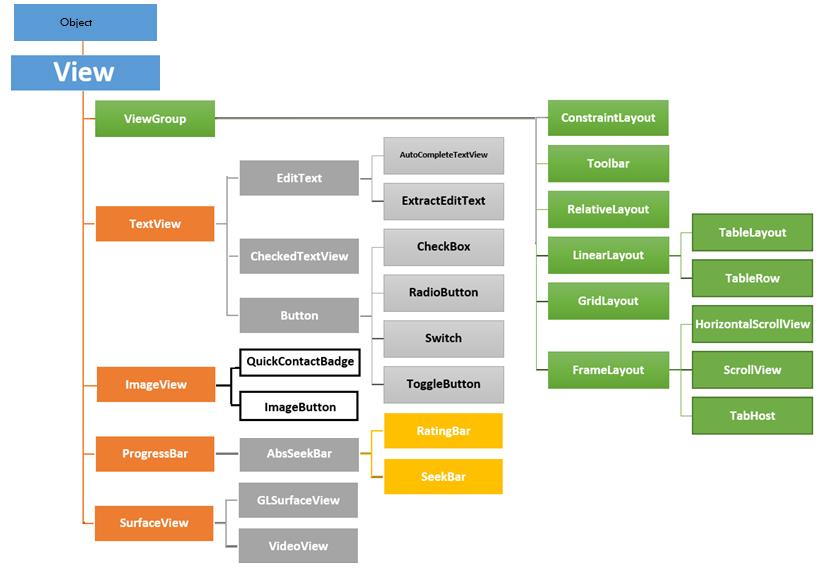
android:layout\_height="wrap\_content"

android:inputType="textPassword"

/>

</LinearLayout>

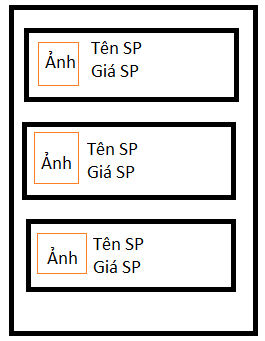
 

* 

====================================

**B7 - Làm việc với ListView**

* Ví dụ 1:   
  Hiển thị danh sách tên màu sắc, sử dụng ArrayAdapter
* Ví dụ 2: Hiển thị danh sách sản phẩm tùy chỉnh



* Bước 1: Tạo layout cho 1 dòng

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

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="horizontal"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_margin="8dp"

android:background="#D3E8F1"

>

<ImageView

android:id="@+id/img\_product"

android:layout\_width="100dp"

android:layout\_height="100dp"

android:src="@drawable/xiao"

android:scaleType="centerCrop"

/>

<LinearLayout

android:layout\_width="0dp"

android:layout\_weight="1"

android:layout\_height="wrap\_content"

android:orientation="vertical"

>

<TextView

android:id="@+id/tv\_name"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="8dp"

android:text="Tên sản phẩm"

/>

<TextView

android:id="@+id/tv\_price"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="8dp"

android:text="Giá tiền"

/>

</LinearLayout>

</LinearLayout>

* Bước 2: Tạo 1 class chứa các thuộc tính của sản phẩm

package vn.edu.spx.vidu6\_custom\_listview;

public class Product {

int id;

String name;*// tên sản phẩm*

double price ; *// giá tiền*

int img\_res; *// ID của ảnh trong thư mục drawable...*

public Product(int id, String name, double price, int img\_res) {

this.id = id;

this.name = name;

this.price = price;

this.img\_res = img\_res;

}

}

* Bước 3: Tạo adapter tùy chỉnh

package vn.edu.spx.vidu6\_custom\_listview;

import android.view.View;

import android.view.ViewGroup;

import android.widget.BaseAdapter;

import android.widget.ImageView;

import android.widget.TextView;

import java.util.ArrayList;

public class ProductAdapter extends BaseAdapter {

ArrayList<Product> listProduct;

*// tạo hàm khởi tạo*

public ProductAdapter(ArrayList<Product> listProduct) {

this.listProduct = listProduct;

}

@Override

public int getCount() {

return listProduct.size(); *// trả về số lượng phần tử trong ds*

}

@Override

public Object getItem(int i) {

*// trả về 1 phần tử tại vị trí thứ i trong ds*

Product objProduct = listProduct.get(i);

return objProduct;

}

@Override

public long getItemId(int i) {

Product objProduct = listProduct.get(i);

return objProduct.id;*// trả về ID của sản phẩm, không phải số thứ tự*

}

@Override

public View getView(int i, View view, ViewGroup viewGroup) {

*// trả về view là 1 dòng*

View item\_view;

if(view == null ){

item\_view = View.*inflate*(viewGroup.getContext(),

R.layout.*item\_row\_listview\_product*, null);

}else{

item\_view = view;

}

*// lấy đối tượng sản phẩm*

Product objProduct = listProduct.get(i);

*// ánh xạ các view*

TextView tv\_name = item\_view.findViewById(R.id.*tv\_name*);

TextView tv\_price = item\_view.findViewById(R.id.*tv\_price*);

ImageView img = item\_view.findViewById(R.id.*img\_product*);

*// gán dữ liệu*

tv\_name.setText( objProduct.name );

tv\_price.setText( objProduct.price + "" );

img.setImageResource( objProduct.img\_res );

*// trả lại view dòng*

return item\_view;

}

}

* Bước 4: Tạo danh sách dữ liệu để đổ vào Adapter và gắn vào listview (làm trên activity)

package vn.edu.spx.vidu6\_custom\_listview;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.ListView;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

ListView lv\_product;

ProductAdapter adapter;

ArrayList<Product> listProduct;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

*// Ánh xạ view*

lv\_product = findViewById(R.id.*lv\_product*);

*// khởi tạo dữ liệu*

listProduct = new ArrayList<Product>();

listProduct.add( new Product(1,"Điện thoại",200,R.drawable.*xiao*) );

listProduct.add( new Product(2,"Máy tính",200,R.drawable.*xiao*) );

listProduct.add( new Product(3,"Tủ lạnh",200,R.drawable.*xiao*) );

listProduct.add( new Product(4,"Tivi",200,R.drawable.*xiao*) );

listProduct.add( new Product(5,"Xe máy",200,R.drawable.*xiao*) );

*// Khởi tạo Adapter*

adapter = new ProductAdapter( listProduct );

*// gắn adapter vào listview*

lv\_product.setAdapter( adapter );

}

}

=======================================

B8 - Làm việc với CSDL SQLite

* Trang chủ: <https://www.sqlite.org/index.html>
* Tải công cụ trên máy tính: <https://sqlitebrowser.org/dl/>
* CSDL mẫu gồm 2 bảng:   
  **tb\_cat** (**id**, name) // thể loại sản phẩm  
  **tb\_product** (**id**, name, price, img\_res, **id\_cat**) // bảng sản phẩm
* Xây dựng ứng dụng Android làm việc với CSDL
* **B1: Tạo project**, tạo các package cho ứng dụng: DAO, DTO, DbHelper, Adapter
* **B2: Định nghĩa lớp đối tượng dữ liệu DTO**: Cat.java

public class Cat {

int id;

String name;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

* **B3: Tạo lớp MyDbHelper** ở trong thư mục DbHelper để làm việc kết nối csdl, tạo bảng

public class MyDbHelper extends SQLiteOpenHelper {

*// khai báo tên csdl, phiên bản csdl*

static final String *DB\_NAME* = "crud\_product";

static final int *DB\_VERSION* = 1;

*// viết hàm khởi tạo (tự viết, không generate)*

public MyDbHelper(Context context){

super(context, *DB\_NAME*, null, *DB\_VERSION*);

}

@Override

public void onCreate(SQLiteDatabase sqLiteDatabase) {

*// viết lệnh tạo bảng ở trong này*

String sql\_create\_tb\_cat = "CREATE TABLE tb\_cat ( id INTEGER NOT NULL, name TEXT NOT NULL, PRIMARY KEY(id AUTOINCREMENT))";

sqLiteDatabase.execSQL( sql\_create\_tb\_cat );

*// làm tương tư với bảng sản phẩm*

String sql\_create\_tb\_pro ="CREATE TABLE tb\_product ( id INTEGER NOT NULL, name TEXT NOT NULL, price NUMERIC DEFAULT 0, img\_res INTEGER, id\_cat INTEGER, PRIMARY KEY(id AUTOINCREMENT))";

sqLiteDatabase.execSQL( sql\_create\_tb\_pro);

*// Chạy câu lệnh insert tạm một ít dữ liệu vào bảng*

*// Thêm dữ liệu vào bảng thể loại*

String sql\_insert\_cat = "INSERT INTO tb\_cat (name) VALUES('Hàng điện tử'),('Hàng gia dụng')";

sqLiteDatabase.execSQL( sql\_insert\_cat);

}

@Override

public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {

}

}

* B4 - Tạo lớp DAO để thực hiện lấy dữ liệu

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import java.util.ArrayList;

import vn.edu.spx.vd7\_crud\_product.DTO.Cat;

import vn.edu.spx.vd7\_crud\_product.DbHelper.MyDbHelper;

public class CatDAO {

SQLiteDatabase db;

MyDbHelper dbHelper;

*// viết hàm khởi tạo*

public CatDAO(Context context){

dbHelper = new MyDbHelper( context );

db = dbHelper.getWritableDatabase(); *// mở kết nối*

*// nếu chưa có file csdl thì sẽ được tạo mới luôn.*

}

public void close(){

db.close();*// đóng kết nối csdl*

}

*// viết hàm lấy danh sách dữ liệu trong bảng tb\_cat*

public ArrayList<Cat> selectAll(){

*// tạo ra 1 ds rỗng*

ArrayList<Cat> listCat = new ArrayList<Cat>();

*//viết lệnh SQL*

String sql = "SELECT \* FROM tb\_cat ";

*// thực thi cau lệnh*

Cursor c = db.rawQuery( sql, null );

*// kiểm tra có dữ liệu thì lấy ra*

if(c.moveToFirst()){

*// có dữ liêu*

while ( !c.isAfterLast() ){

*// tạo DTO để làm việc*

Cat objCat = new Cat();

objCat.setId( c.getInt( 0 ) );

objCat.setName( c.getString( 1 ) );

*// cho đối tượng vào trong danh sách*

listCat.add( objCat );

*// chuyển con trỏ đọc sang dòng tiếp theo*

c.moveToNext();

}

}

return listCat;

}

}

* B5 - Tạo đối tượng DAO để thực thi ở trong MainActivity

*// Tạo đối tượng DAO*

CatDAO catDAO = new CatDAO(this);

ArrayList<Cat> listCat = catDAO.selectAll();

*// ghi log để xem kết quả*

Log.*d*(TAG, "onCreate: số lượng bản ghi: " + listCat.size() );

*// dùng vòng lặp in tất cả bản ghi*

for(int i = 0; i< listCat.size(); i++){

Cat objCat = listCat.get(i);

Log.*d*(TAG, "onCreate: Tên cat = " + objCat.getName() );

}

* B6: Chạy thử ở chế độ debug, trên android studio, bấm vào Log Cat để xem log.

========================================

**B9 - Hiển thị danh sách Category lên listview, viết các hàm thêm, sửa, xóa dữ liệu trong 1 bảng**

**\*\* Hiển thị ds category lên listview**

* Tạo activity riêng cho Category
* Tạo cái listview trong layout của activity mới, tạo layout cho 1 dòng trên listview

*<?*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=".CategoryActivity">

<TextView

android:id="@+id/tv\_title"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Danh sách thể loại"

android:textSize="30dp"

android:gravity="center"

android:layout\_margin="20dp"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

/>

<ListView

android:id="@+id/**lv\_cat**"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_margin="20dp"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/tv\_title"

app:layout\_constraintBottom\_toBottomOf="parent"/>

</androidx.constraintlayout.widget.ConstraintLayout>

Layout của 1 dòng

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

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="horizontal"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="8dp"

>

<TextView

android:id="@+id/tv\_id"

android:layout\_width="40dp"

android:layout\_height="wrap\_content"

android:text="ID"

android:padding="8dp"

/>

<TextView

android:id="@+id/tv\_name"

android:layout\_width="0dp"

android:layout\_weight="1"

android:layout\_height="wrap\_content"

android:padding="8dp"

/>

<TextView

android:id="@+id/tv\_edit"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Sửa"

android:padding="8dp"

/>

<TextView

android:id="@+id/tv\_delete"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Xóa"

android:padding="8dp"

/>

</LinearLayout>

* Tạo adapter cho listview

package vn.edu.spx.vd7\_crud\_product.Adapter;

import android.view.View;

import android.view.ViewGroup;

import android.widget.BaseAdapter;

import android.widget.TextView;

import java.util.ArrayList;

import vn.edu.spx.vd7\_crud\_product.DTO.Cat;

import vn.edu.spx.vd7\_crud\_product.R;

public class CategoryAdapter extends BaseAdapter {

ArrayList<Cat> listCat ;

public CategoryAdapter(ArrayList<Cat> listCat) {

this.listCat = listCat;

}

@Override

public int getCount() {

return listCat.size();

}

@Override

public Object getItem(int i) {

Cat objCat = listCat.get(i);

return objCat;

}

@Override

public long getItemId(int i) {

Cat objCat = listCat.get(i);

return objCat.getId() ;

}

@Override

public View getView(int i, View view, ViewGroup viewGroup) {

*//1. tạo 1 dòng view*

View row;

if(view == null)

row = View.*inflate*(viewGroup.getContext(), R.layout.*row\_list\_view*, null);

else

row = view;

*//2. Ánh xạ các view trong dòng vào biến*

TextView tv\_id = row.findViewById(R.id.*tv\_id*);

TextView tv\_name = row.findViewById(R.id.*tv\_name*);

TextView tv\_edit = row.findViewById(R.id.*tv\_edit*);

TextView tv\_delete = row.findViewById(R.id.*tv\_delete*);

*// 3. gán dữ liệu cho textview*

Cat objCat = listCat.get(i);

tv\_id.setText( objCat.getId() + "");

tv\_name.setText( objCat.getName() );

*/// trả về dòng*

return row;

}

}

* Vào activity tạo đối tượng DAO, adapter để gắn dữ liệu vào listview

package vn.edu.spx.vd7\_crud\_product;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.util.Log;

import android.widget.ListView;

import vn.edu.spx.vd7\_crud\_product.Adapter.CategoryAdapter;

import vn.edu.spx.vd7\_crud\_product.DAO.CatDAO;

public class CategoryActivity extends AppCompatActivity {

ListView lv\_cat;

CatDAO dao;

CategoryAdapter adapter;

String TAG = "zzzzzzzzzzzzzzz";

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_category*);

lv\_cat = findViewById(R.id.*lv\_cat*);

*// tạo DAO*

dao = new CatDAO(this);

*//--- test thử kiểm tra dữ liệu*

Log.*d*(TAG, "onCreate: Số lượng dòng = " + dao.selectAll().size() );

*// Tạo adapter*

adapter = new CategoryAdapter( dao.selectAll() );

*// gắn vào listview*

lv\_cat.setAdapter( adapter );

}

}

* Vào activity main viết lệnh gọi category activity để chạy thử nghiệm

**\*\* Viết các hàm xử lý thêm, sửa, xóa, lấy 1 dòng trong DAO**

**===================================**

**B12: Tổng hợp các vấn đề liên quan tới tương tác cơ sở dữ liệu**

1. **Vấn đề tạo project:**

Đặt tên theo quy tắc: Tên lớp, tên gói viết chữ hoa tất cả các chữ cái đầu các từ.  
Tên biến: Viết chữ thường ký tự đầu tiên, các ký tự đầu các từ tiếp theo thì viết hoa.  
VD: Tên class: DanhBa, DanhBaDAO, DanhBaAdapter…, tên biến: danhBa, danhBaAdapter…

1. **Các thành phần cần có khi làm việc với CSDL**
   1. **DbHelper**: Tạo csdl, tạo bảng SQLiteOpenHelper
      1. Hàm onCreate: viết lệnh tạo bảng ( hàm này 1 lần duy nhất khi chưa có CSDL, muốn chạy lại hàm này thì đổi tên CSDL hoặc xóa file trong thiết bị)
      2. Hàm onUpgrade: Chạy khi thay đổi version của CSDL.
   2. **DTO**: Tạo đối tượng dữ liệu (các thuộc tính thường tương ứng với các cột trong bảng dữ liệu)
   3. **DAO**: Tạo các hàm thực hiện câu lệnh SQL thêm, sửa, xóa, đọc dữ liệu (CRUD)
      1. Cần có đối tượng SqliteDatabase
      2. Đối tượng DbHelper
      3. Hàm dbHelper.**getWritableDatabase**(); dùng để mở kết nối CSDL, tạo mới file CSDL nếu chưa có file.
      4. Chú ý: Tên bảng trong lệnh select hoặc các lệnh xóa sửa phải trùng khớp với tên bảng ở trong câu lệnh tạo bảng (create table ….) no such **table** name… abc
      5. Hàm lấy danh sách:
         1. Có 2 cách để lấy dữ liệu

*// Cách 1: Dùng hàm query để lấy dữ liệu*

String[] ds\_cot = new String[] { "\*" };

*//// // tạo đối tượng con trỏ đọc dữ liệu*

Cursor cursor = db.query(GroupUser.*TB\_NAME*, ds\_cot,null, null,null,null, null);

*// cách 2: Dùng hàm rawQuery để chạy câu lệnh SQL tự viết*

Cursor cursor = db.rawQuery("SELECT \* FROM " + GroupUser.*TB\_NAME*, null);

* + - 1. Sau câu lệnh gọi query thì luôn có lệnh kiểm tra if(c.**moveToFirst**()){...}
      2. Trong lệnh lặp while đọc dữ liệu luôn phải nhớ có câu lệnh **cursor.moveToNext()**; ở cuối vòng lặp.
  1. **Adapter**: Là lớp điều khiển hiển thị dữ liệu
     1. Cần danh sách dữ liệu dạng List hoặc ArrayList , có hàm khởi tạo để gán dữ liệu từ DAO
     2. Hàm getCount() phải return về size của danh sách, getItem() trả về đối tượng trong ds
     3. Hàm **getView**():
        1. Cần chọn đúng layout của dòng (tạo file layout riêng)
        2. Các xử lý sửa, xóa, hiển thị chi tiết 1 dòng thì viết trong hàm này
        3. Cần nghiên cứu kỹ thuật dùng Dialog: Có 2 loại: Dialog đơn giản và Dialog tùy chỉnh.

Tạo dialog đơn gản:

AlertDialog.Builder builder = new AlertDialog.Builder( Truyền\_vào\_dây\_là\_context );

builder.setTitle("Tiêu đề dialog");

builder.setMessage("Nội dung câu thông báo");

AlertDialog dialog = builder.create();

dialog.show();

======================================

**B13: Intent - options menu - Nhận dữ liệu từ activity con**

* Làm việc với intent
  + Tham khảo bảng các action: <https://zezo.dev/note/bang-cac-action-intent>
  + Sử dụng intent-filter: <https://zezo.dev/note/huong-dan-su-dung-intent-filter-de-nhan-du-lieu-vao-ung-dung-android>
* Menu options:

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

<menu xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto">

<item android:id="@+id/m\_add"

android:title="Thêm mới"

android:icon="@android:drawable/ic\_menu\_add"

app:showAsAction="always"/>

<item android:id="@+id/m\_2"

android:title="Item 2"

android:icon="@android:drawable/ic\_menu\_add"

app:showAsAction="never"/>

<item android:id="@+id/m\_3"

android:title="Item 3"

android:icon="@android:drawable/ic\_menu\_add"

app:showAsAction="never"/>

</menu>

Code java viết trong activity:

@Override

public boolean onCreateOptionsMenu(Menu menu) {

MenuInflater inflater = getMenuInflater();

inflater.inflate(R.menu.*opt\_menu01*, menu );

return super.onCreateOptionsMenu(menu);

}

@Override

public boolean onOptionsItemSelected(@NonNull MenuItem item) {

switch ( item.getItemId() ){

case R.id.*m\_add*:

*// code xử lý làm gì đó thì viết ở đây*

Toast.*makeText*(this, "Bạn chọn thêm", Toast.*LENGTH\_SHORT*).show();

break;

*// làm tương tự với các menu item khác*

}

return super.onOptionsItemSelected(item);

}

* Truyền ngược dữ liệu từ activity con sang activity cha.
  + Tham khảo <https://zezo.dev/note/su-dung-intent-truyen-nguoc-du-lieu-tu-activity-con-cho-activity-cha>
  + Nhiệm vụ 1: Thực hành triển khai code trong link trên để chạy được ứng dụng
  + Nhiệm vụ 2:
  + Yêu cầu LAB:
  + Viết 1 ứng dụng có main activity là chạy chính, khi activity main chạy sẽ tự động gọi activity Login hiển thị lên, người dùng nhập username và password. Sau khi bấm nút login ở màn hình đăng nhập thì gửi dữ liệu về activity chính và trên màn hình chính hiển thị họ tên người đăng nhập. (Để quay trở lại thì gọi hàm onBackPressed() ở trong sự kiện bấm nút login)

B14 - Thao tác lưu trữ dữ liệu

* Lưu trữ File
  + Đọc ghi ở bộ nhớ trong: <https://zezo.dev/note/doc-ghi-file-o-bo-nho-trong>
  + Đọc ghi ở bộ nhớ ngoài: <https://zezo.dev/note/doc-ghi-file-o-bo-nho-ngoai>
  + Đọc ghi cache: <https://zezo.dev/note/doc-ghi-file-cache>
* Đọc ghi sharedPreferences
  + Tham khảo <https://zezo.dev/note/doc-ghi-sharedpreferences>

Yêu cầu Lab:

1. Viết chương trình đăng nhập đơn giản gồm username, password và một nút checkbox là “Ghi nhớ đăng nhập”. Khi đăng nhập kiểm tra username và password (admin/123) nếu không đúng thì yêu cầu nhập lại, nếu đúng thì kiểm tra tiếp nếu có check Ghi nhớ thì lưu username và cả password vào sharedPref, nếu không check thì chỉ lưu username vào mà không lưu password.
2. Sau khi chạy thành công chương trình, cải tiến ứng dụng: Lúc vào màn hình đăng nhập thì kiểm tra sharedPref nếu có lưu username thì hiển thị username lên ô text, nếu có password thì điền password vào ô text. Trường hợp có password thì sẽ tự check vào ô checkbox.
3. Cải tiến bài Assignment thêm 1 màn hình chào full màn hình trước khi vào ứng dụng. (tham khảo <https://zezo.dev/note/an-thanh-tieu-de-lam-ung-dung-tran-man-hinh> )

**B15 - Làm việc với Webview - Luyện tập**

Cách sử dụng webview: <https://zezo.dev/note/su-dung-webview-don-gian>

Tham khảo code demo <https://zezo.dev/note/code-demo-dung-webview>

**Bài luyện tập:**

Viết ứng dụng gồm các tính năng như sau:

1. Khi vào ứng dụng sẽ hiển thị màn hình đăng ký gồm: Họ tên, Email, Điện thoại, Giới thiệu và nút bấm Đăng ký. Bấm nút Đăng ký thì thực hiện lưu thông tin vào sharedPref, sau đó đóng màn activity đăng ký lại (dùng hàm finish()).

2. Trên activity\_main có 1 webview, thực hiện đọc nội dung trong sharePref ra và trình bày lên webview. Yêu cầu: Dùng kỹ thuật ghép chuỗi trong 1 biến kiểu chuỗi để tạo chuỗi html thông tin định dạng như sau:



Gợi ý: Viết code trong hàm onResume()

3. Áp dụng vào bài assignment, tạo 1 trang html và gắn vào ứng dụng, tạo menu option là: Giới thiệu, khi bấm vào sẽ hiển thị 1 activity giới thiệu, trong activity này sẽ dùng webview hiển thị trang html giới thiệu vừa tạo.