**Bài tập Lab3**

**Phần I: Tạo 1 CSDL SQLite, thêm vào CSDL danh sách các**

**user, sau đó lấy dữ liệu từ CSDL hiện lên listview**

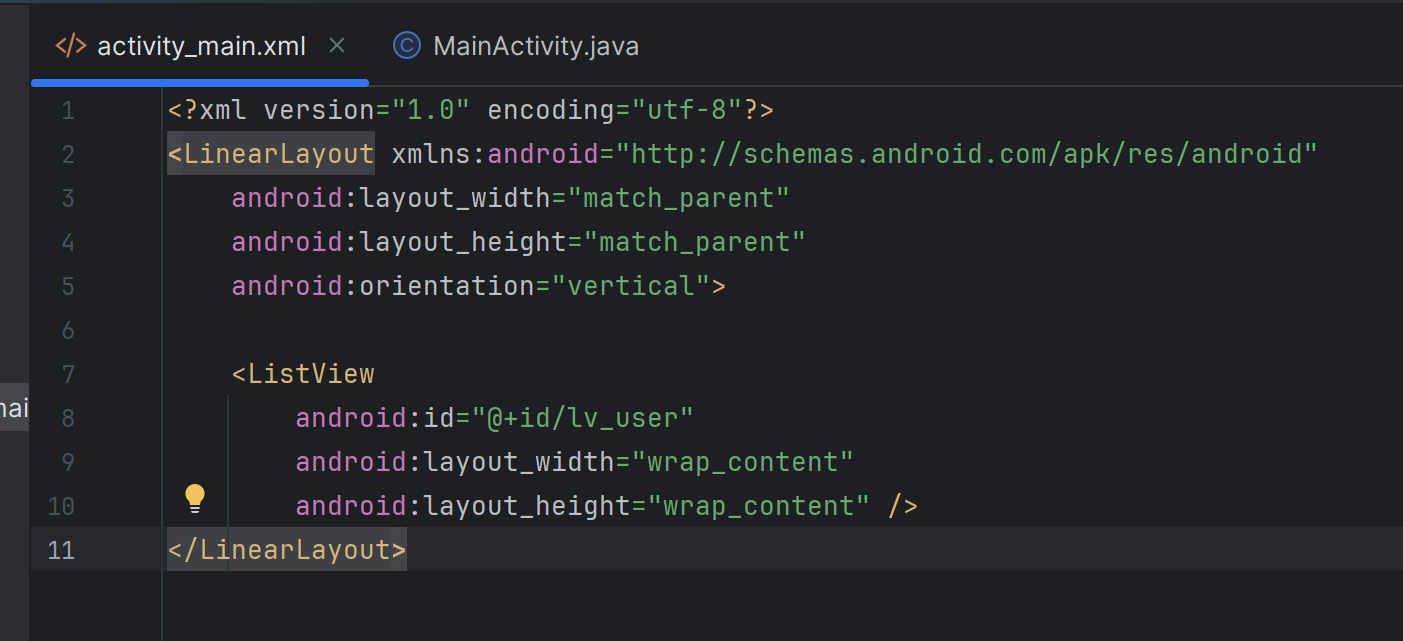
**Bước 1:** Tạo project mới – chọn Empty Views Activity – đặt tên Lab3.

**Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Nội dung do AI tạo ra có thể không chính xác.**

**Bước 2:** Thiết kế giao diện người dùng

**-** Ở **Activity\_main.xml –** viết code sau



**Bước 3**: Ở Layout – chuột phải – New – Layout Resource File – đặt tên **item\_user -** ở **Root element** gõ **Textview.**

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Nội dung do AI tạo ra có thể không chính xác.

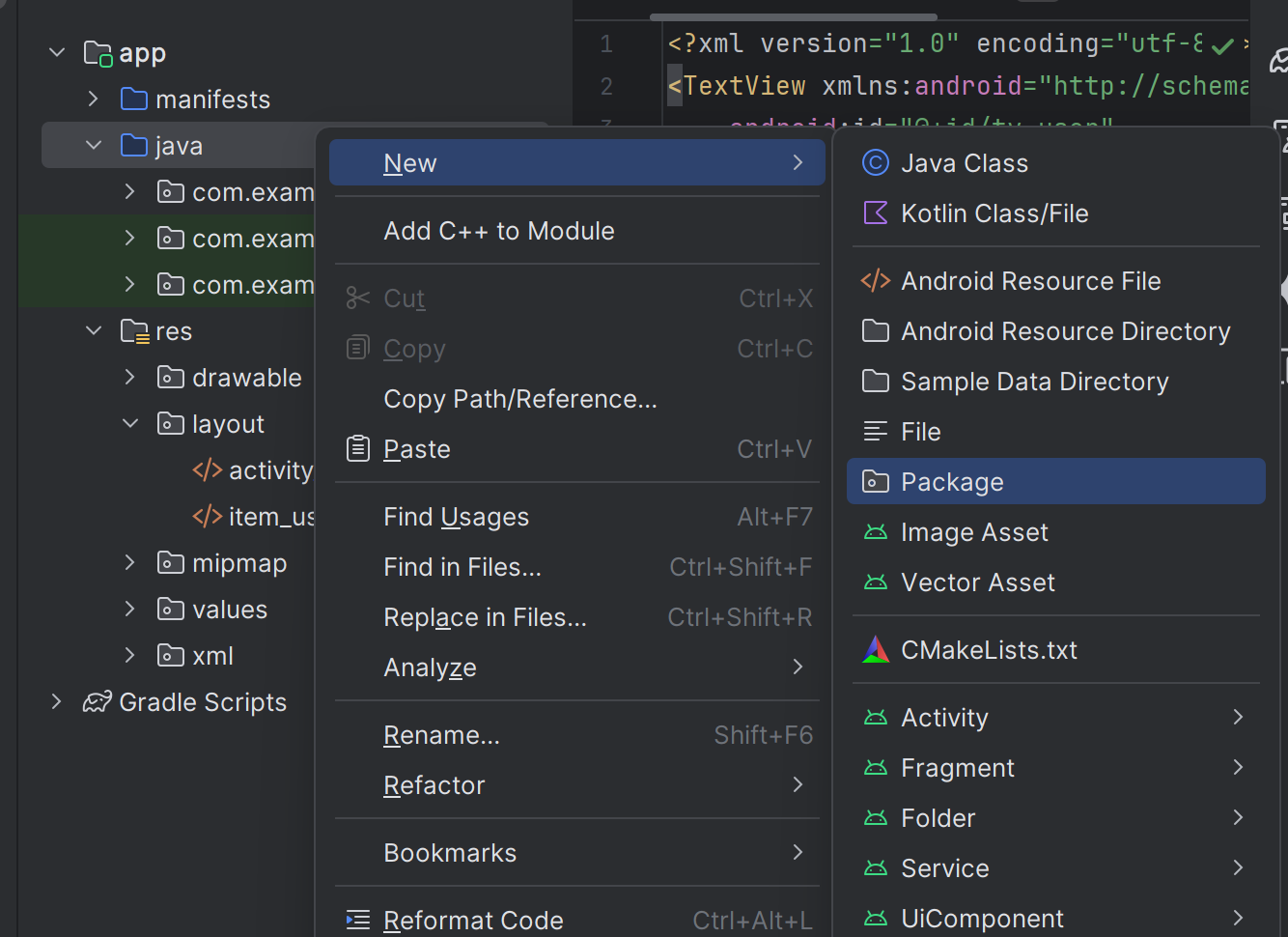
- Ở **item\_user** viết code sau:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Nội dung do AI tạo ra có thể không chính xác.

**Bước 4**: Tạo lớp DatabaseHelper.java

- Ở Java chuột phải – **New** – **Package** – đặt tên là **helper** – **Enter** - nhấp chuột phải vào **package helper** vừa tạo – chọn **New** – **Java Class** – đặt tên là **DatabaseHelper** – **Enter.**

- 

- ở **DatabaseHelper** viết code như sau:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Nội dung do AI tạo ra có thể không chính xác.

**Bước 5**: Tạo lớp **DbAdapter.java**

- Nhấp chuột phải vào package **Helper – New – Package – Đặt tên adapter – Enter**

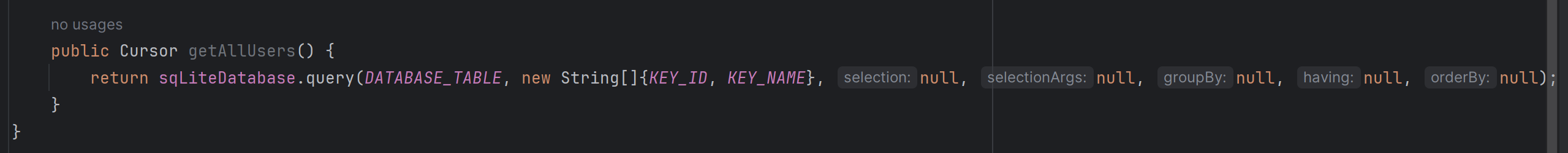
**-** nhấp chuột phải vào package **adapter** vừa tạo – **New** – **Java Class** – Viết code.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.



**Bước 6:** Cập nhật **MainActivity,java**

**-** Trong **Project panel,** mở file **MainActivity.java** trong package **Helper –** viết mã như hình

package com.example.lab3;  
  
import android.database.Cursor;  
import android.os.Bundle;  
import androidx.appcompat.app.AppCompatActivity; // Sửa lại import  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
import java.util.ArrayList;  
import java.util.List;  
import adapter.DbAdapter;  
  
public class MainActivity extends AppCompatActivity { // Sửa lại kế thừa  
 private DbAdapter dbAdapter;  
 private Cursor cursor;  
 private List<String> users;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 dbAdapter = new DbAdapter(this);  
 dbAdapter.open();  
 dbAdapter.deleteAllUsers();  
  
 // Thêm 10 người dùng mẫu  
 for (int i = 0; i < 10; i++) {  
 dbAdapter.createUser("Nguyễn Văn An " + i);  
 }  
  
 users = getData();  
 showData();  
 }  
  
 private List<String> getData() {  
 List<String> users = new ArrayList<>();  
 cursor = dbAdapter.getAllUsers();  
 while (cursor.moveToNext()) {  
 // Lấy chuỗi từ cột có tên là "name"  
 users.add(cursor.getString(cursor.getColumnIndexOrThrow(DbAdapter.*KEY\_NAME*)));  
 }  
 cursor.close(); // Đóng cursor sau khi dùng xong  
 return users;  
 }  
  
 private void showData() {  
 ListView lvUser = (ListView) findViewById(R.id.*lv\_user*);  
 // Sử dụng item\_user.xml đã tạo và android.R.id.text1 vì nó là một TextView đơn giản  
 ArrayAdapter<String> userAdapter = new ArrayAdapter<String>(MainActivity.this, R.layout.*item\_user*, R.id.*tv\_user*, users);  
 lvUser.setAdapter(userAdapter);  
 }  
}

**Bước 7:** Chạy ứng dụng

**-** Mở máy ảo **– Run App**

**Ảnh có chứa văn bản, ảnh chụp màn hình, đa phương tiện, đồ phụ tùng

Nội dung do AI tạo ra có thể không chính xác.**

**Phần II: Thực hành quản lý Contact**

**Bước 1: Tạo Project mới**

**-** Chọn **Empty Views Activity** – đặt tên **Lab3\_ContactManager – Finish.**

**Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

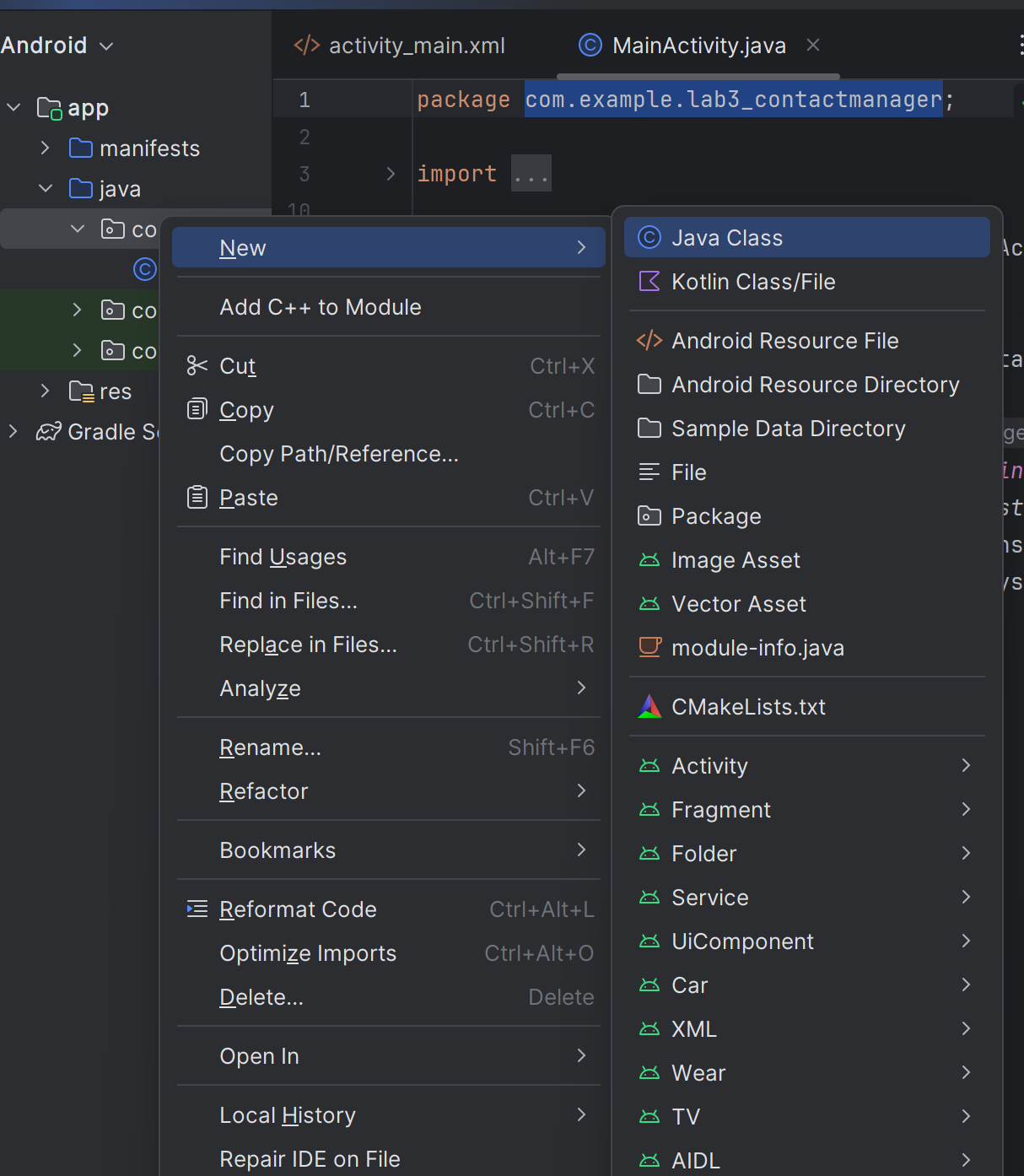
Nội dung do AI tạo ra có thể không chính xác.**

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Nội dung do AI tạo ra có thể không chính xác.

**Bước 2:** Tạo lớp Contact.java ( lớp mô hình dữ liệu )

- Chuột phải vào package **com.example.lab3\_contactmanager** – **New** – **Java Class** – đặt tên **Contact** – **Enter**



- Ở **Contact.java** – viết code như hình

package com.example.lab3\_contactmanager;  
  
public class Contact {  
 private int id;  
 private String name;  
 private String phoneNumber;  
  
 // Constructor (phương thức khởi tạo) rỗng  
 public Contact() {  
 }  
  
 // Constructor để tạo đối tượng Contact mới (chưa có ID)  
 public Contact(String name, String phoneNumber) {  
 this.name = name;  
 this.phoneNumber = phoneNumber;  
 }  
  
 // Constructor đầy đủ các thuộc tính  
 public Contact(int id, String name, String phoneNumber) {  
 this.id = id;  
 this.name = name;  
 this.phoneNumber = phoneNumber;  
 }  
  
 // Các phương thức để lấy và gán giá trị (Getters and Setters)  
 public int getID() {  
 return this.id;  
 }  
  
 public void setID(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return this.name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getPhoneNumber() {  
 return this.phoneNumber;  
 }  
  
 public void setPhoneNumber(String phoneNumber) {  
 this.phoneNumber = phoneNumber;  
 }  
}

**Bước 3**: Tạo lớp **DatabaseHandler.java** (Lớp Xử lý Cơ sở dữ liệu)

- Thực hiện giống bước 2 – đặt tên **DatabaseHandler – Enter**

- Ở **DatabaseHandler.java** viết code như hình sau:

package com.example.lab3\_contactmanager;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class DatabaseHandler extends SQLiteOpenHelper {  
 // Phiên bản của CSDL  
 private static final int *DATABASE\_VERSION* = 1;  
 // Tên của CSDL  
 private static final String *DATABASE\_NAME* = "contactsManager";  
 // Tên bảng  
 private static final String *TABLE\_CONTACTS* = "contacts";  
 // Tên các cột trong bảng  
 private static final String *KEY\_ID* = "id";  
 private static final String *KEY\_NAME* = "name";  
 private static final String *KEY\_PH\_NO* = "phone\_number";  
  
 public DatabaseHandler(Context context) {  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }  
  
 // Phương thức này được gọi khi CSDL được tạo lần đầu tiên  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 // Câu lệnh SQL để tạo bảng contacts  
 String CREATE\_CONTACTS\_TABLE = "CREATE TABLE " + *TABLE\_CONTACTS* + "("  
 + *KEY\_ID* + " INTEGER PRIMARY KEY," + *KEY\_NAME* + " TEXT,"  
 + *KEY\_PH\_NO* + " TEXT" + ")";  
 db.execSQL(CREATE\_CONTACTS\_TABLE);  
 }  
  
 // Phương thức này được gọi khi nâng cấp phiên bản CSDL  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 // Xóa bảng cũ nếu nó đã tồn tại  
 db.execSQL("DROP TABLE IF EXISTS " + *TABLE\_CONTACTS*);  
 // Tạo lại bảng  
 onCreate(db);  
 }  
  
 */\*\*  
 \* Các hàm CRUD (Create, Read, Update, Delete)  
 \*/* // Thêm một contact mới  
 void addContact(Contact contact) {  
 SQLiteDatabase db = this.getWritableDatabase();  
 ContentValues values = new ContentValues();  
 values.put(*KEY\_NAME*, contact.getName()); // Tên contact  
 values.put(*KEY\_PH\_NO*, contact.getPhoneNumber()); // Số điện thoại  
  
 // Chèn một dòng mới vào bảng  
 db.insert(*TABLE\_CONTACTS*, null, values);  
 db.close(); // Đóng kết nối CSDL  
 }  
  
 // Lấy tất cả các contact  
 public List<Contact> getAllContacts() {  
 List<Contact> contactList = new ArrayList<>();  
 // Câu lệnh truy vấn  
 String selectQuery = "SELECT \* FROM " + *TABLE\_CONTACTS*;  
  
 SQLiteDatabase db = this.getWritableDatabase();  
 Cursor cursor = db.rawQuery(selectQuery, null);  
  
 // Lặp qua tất cả các dòng và thêm vào danh sách  
 if (cursor.moveToFirst()) {  
 do {  
 Contact contact = new Contact();  
 contact.setID(cursor.getInt(cursor.getColumnIndexOrThrow(*KEY\_ID*)));  
 contact.setName(cursor.getString(cursor.getColumnIndexOrThrow(*KEY\_NAME*)));  
 contact.setPhoneNumber(cursor.getString(cursor.getColumnIndexOrThrow(*KEY\_PH\_NO*)));  
 // Thêm contact vào danh sách  
 contactList.add(contact);  
 } while (cursor.moveToNext());  
 }  
 cursor.close();  
 return contactList;  
 }  
  
 // Xóa một contact  
 public void deleteContact(Contact contact) {  
 SQLiteDatabase db = this.getWritableDatabase();  
 db.delete(*TABLE\_CONTACTS*, *KEY\_ID* + " = ?",  
 new String[]{String.*valueOf*(contact.getID())});  
 db.close();  
 }  
  
 // Xóa tất cả contact (hàm này tự thêm để tiện cho việc test)  
 public void deleteAllContacts() {  
 SQLiteDatabase db = this.getWritableDatabase();  
 db.delete(*TABLE\_CONTACTS*, null, null);  
 db.close();  
 }  
}

Bước 4: Thiết kế Giao diện **activity\_main.xml**

- Ở **activity\_main.xml** viết code như hình sau:

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:fitsSystemWindows="true">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:background="#F0F0F0">  
  
 <TextView  
 android:id="@+id/tv\_student\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Họ và Tên: Võ Quỳnh Chi"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 android:gravity="center"  
 tools:ignore="HardcodedText" />  
  
 <TextView  
 android:id="@+id/tv\_student\_id"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="MSSV: 1150070004"  
 android:textSize="16sp"  
 android:gravity="center"  
 android:layout\_marginTop="4dp"  
 tools:ignore="HardcodedText" />  
 </LinearLayout>  
  
 <View  
 android:layout\_width="match\_parent"  
 android:layout\_height="1dp"  
 android:background="@android:color/darker\_gray" />  
  
 <ListView  
 android:id="@+id/lv\_contacts"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1" />  
  
</LinearLayout>

**Bước 5:** Mở **MainActivity.java**

**-** Viết code như hình sau

package com.example.lab3\_contactmanager;  
  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import java.util.ArrayList;  
import java.util.List;  
  
public class MainActivity extends AppCompatActivity {  
  
 private DatabaseHandler db;  
 private List<Contact> contacts; // Danh sách các đối tượng Contact  
 private ArrayAdapter<String> adapter;  
 private ArrayList<String> contactDisplayList; // Danh sách chuỗi để hiển thị  
 private ListView lvContacts;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 // Ánh xạ ListView từ file layout  
 lvContacts = findViewById(R.id.*lv\_contacts*);  
 db = new DatabaseHandler(this);  
  
 // Xóa dữ liệu cũ để mỗi lần chạy lại ứng dụng sẽ không bị lặp lại dữ liệu  
 db.deleteAllContacts();  
  
 // Thêm dữ liệu mẫu vào CSDL như trong tài liệu  
 Log.*d*("Insert: ", "Đang thêm dữ liệu mẫu...");  
 db.addContact(new Contact("Ravi", "9100000000"));  
 db.addContact(new Contact("Srinivas", "9199999999"));  
 db.addContact(new Contact("Tommy", "9522222222"));  
 db.addContact(new Contact("Karthik", "9533333333"));  
  
 // Gọi hàm để tải và hiển thị dữ liệu  
 loadDataToListView();  
  
 // Cài đặt sự kiện nhấn giữ (Long Click) cho ListView  
 lvContacts.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {  
 @Override  
 public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {  
 // Lấy đối tượng Contact tại vị trí được nhấn giữ  
 Contact contactToDelete = contacts.get(position);  
  
 // Gọi hàm xóa contact khỏi CSDL  
 db.deleteContact(contactToDelete);  
  
 // Hiển thị một thông báo ngắn để người dùng biết  
 Toast.*makeText*(MainActivity.this, "Đã xóa " + contactToDelete.getName(), Toast.*LENGTH\_SHORT*).show();  
  
 // Tải lại danh sách để cập nhật giao diện  
 loadDataToListView();  
  
 return true; // Trả về true để báo hiệu rằng sự kiện đã được xử lý  
 }  
 });  
 }  
  
 // Hàm để tải dữ liệu từ CSDL và hiển thị lên ListView  
 private void loadDataToListView() {  
 Log.*d*("Reading: ", "Đang đọc tất cả contact...");  
 // Lấy danh sách contact mới nhất từ CSDL  
 contacts = db.getAllContacts();  
 contactDisplayList = new ArrayList<>();  
  
 // Chuyển đổi danh sách đối tượng Contact thành danh sách chuỗi để hiển thị  
 for (Contact cn : contacts) {  
 String log = "Id: " + cn.getID() + ", Name: " + cn.getName() + ", Phone: " + cn.getPhoneNumber();  
 Log.*d*("Contact Info: ", log); // Ghi log để kiểm tra (có thể xem ở tab Logcat)  
 contactDisplayList.add(cn.getName() + "\n" + cn.getPhoneNumber());  
 }  
  
 // Tạo Adapter để kết nối dữ liệu với ListView  
 adapter = new ArrayAdapter<>(this, android.R.layout.*simple\_list\_item\_1*, contactDisplayList);  
 lvContacts.setAdapter(adapter);  
 }  
}

**Bước 6: Chạy ứng dụng**

**Ảnh có chứa đồ điện tử, văn bản, ảnh chụp màn hình, Thiết bị liên lạc

Nội dung do AI tạo ra có thể không chính xác.**