**Ngô Thành Hữu**

**1150070015**

**Câu 2.1:**

A white rectangular box with black text

Description automatically generated

**Câu 2.2**

**CONTACT:**

package com.example.bai3;  
  
public class Contact {  
 private int id;  
 private String name;  
 private String phoneNumber;  
  
 public Contact() {}  
  
 public Contact(String name, String phoneNumber) {  
 this.name = name;  
 this.phoneNumber = phoneNumber;  
 }  
  
 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;  
 }  
  
 public String getPhoneNumber() {  
 return phoneNumber;  
 }  
  
 public void setPhoneNumber(String phoneNumber) {  
 this.phoneNumber = phoneNumber;  
 }  
  
 // ✅ Thêm phương thức toString() để in ra thông tin  
 @Override  
 public String toString() {  
 return "Id: " + id + ", Name: " + name + ", Phone: " + phoneNumber;  
 }  
}

**DatabaseHandler:**

package com.example.bai3;  
  
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 {  
 private static final int *DATABASE\_VERSION* = 1;  
 private static final String *DATABASE\_NAME* = "contactsManager";  
 private static final String *TABLE\_CONTACTS* = "contacts";  
 private static final String *KEY\_ID* = "id";  
 private static final String *KEY\_NAME* = "name";  
 private static final String *KEY\_PHONE* = "phone\_number";  
  
 public DatabaseHandler(Context context) {  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 String CREATE\_CONTACTS\_TABLE = "CREATE TABLE " + *TABLE\_CONTACTS* + "("  
 + *KEY\_ID* + " INTEGER PRIMARY KEY AUTOINCREMENT,"  
 + *KEY\_NAME* + " TEXT,"  
 + *KEY\_PHONE* + " TEXT" + ")";  
 db.execSQL(CREATE\_CONTACTS\_TABLE);  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 db.execSQL("DROP TABLE IF EXISTS " + *TABLE\_CONTACTS*);  
 onCreate(db);  
 }  
  
 // Thêm contact  
 public void addContact(Contact contact) {  
 SQLiteDatabase db = this.getWritableDatabase();  
 ContentValues values = new ContentValues();  
 values.put(*KEY\_NAME*, contact.getName());  
 values.put(*KEY\_PHONE*, contact.getPhoneNumber());  
 db.insert(*TABLE\_CONTACTS*, null, values);  
 db.close();  
 }  
  
 // Lấy tất cả contact  
 public List<Contact> getAllContacts() {  
 List<Contact> contactList = new ArrayList<>();  
 SQLiteDatabase db = this.getReadableDatabase();  
 Cursor cursor = db.rawQuery("SELECT \* FROM " + *TABLE\_CONTACTS*, null);  
 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\_PHONE*)));  
 contactList.add(contact);  
 } while (cursor.moveToNext());  
 }  
 cursor.close();  
 db.close();  
 return contactList;  
 }  
  
 // Xóa contact  
 public void deleteContact(int id) {  
 SQLiteDatabase db = this.getWritableDatabase();  
 db.delete(*TABLE\_CONTACTS*, *KEY\_ID* + "=?", new String[]{String.*valueOf*(id)});  
 db.close();  
 }  
  
}

**MainActivity:**

package com.example.bai3;  
  
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 {  
  
 DatabaseHandler db;  
 ListView lv;  
 ArrayAdapter<String> adapter;  
 List<Contact> contacts;  
 ArrayList<String> contactNames;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 lv = findViewById(R.id.*lv\_contacts*);  
 db = new DatabaseHandler(this);  
  
 // Xóa dữ liệu cũ để tránh nhân đôi  
 db.getWritableDatabase().execSQL("DELETE FROM contacts");  
  
 // Thêm dữ liệu mẫu  
 // Thêm dữ liệu mẫu  
 db.addContact(new Contact("Ngô Thành Hữu", "1150070015"));  
 db.addContact(new Contact("Srinivas", "1150070016"));  
 db.addContact(new Contact("Tommy", "1150070017"));  
 db.addContact(new Contact("Karthik", "1150070018"));  
 db.addContact(new Contact("Lê Văn Minh", "0905123456"));  
 db.addContact(new Contact("Trần Thị Thu Hà", "0912345678"));  
 db.addContact(new Contact("Nguyễn Văn An", "0987654321"));  
 db.addContact(new Contact("Đỗ Hoàng Long", "0977123123"));  
 db.addContact(new Contact("Lý Hải Yến", "0933222111"));  
  
  
  
 // Hiển thị dữ liệu ban đầu  
 loadContacts();  
  
 // Long click để xóa contact  
 lv.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {  
 @Override  
 public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {  
 Contact selectedContact = contacts.get(position);  
  
 // Thông báo trước khi xóa  
 Toast.*makeText*(MainActivity.this,  
 "Xóa: " + selectedContact.getName(),  
 Toast.*LENGTH\_SHORT*).show();  
  
 // Xóa contact trong DB  
 db.deleteContact(selectedContact.getID());  
  
 // Log ra Logcat  
 Log.*e*("Delete", "Đã xóa contact: " + selectedContact);  
  
 // Load lại danh sách để cập nhật giao diện  
 loadContacts();  
  
 return true; // bắt buộc return true để sự kiện long click được xử lý  
 }  
 });  
 }  
  
 private void loadContacts() {  
 // Lấy toàn bộ contact từ DB  
 contacts = db.getAllContacts();  
 contactNames = new ArrayList<>();  
  
 for (Contact c : contacts) {  
 // In ra Logcat với toString()  
 Log.*e*("Contact", c.toString());  
  
 // Hiển thị ID - Name - Phone trong ListView  
 contactNames.add("ID: " + c.getID()  
 + " - " + c.getName()  
 + " - " + c.getPhoneNumber());  
 }  
  
 // Đưa dữ liệu vào ListView  
 adapter = new ArrayAdapter<>(this, android.R.layout.*simple\_list\_item\_1*, contactNames);  
 lv.setAdapter(adapter);  
 }  
}

**KẾT QUẢ:**

**A screenshot of a phone

Description automatically generated**

**Sinh viên tùy chỉnh lại code để hiển thị list Contact ra listview. Khi nhấn giữ lâu  vào 1 contact nào đó (long click) thì ứng dụng xóa contact đó khỏi Database.**

**PHẦN MainActivity:**

// Long click để xóa contact

lv.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {

@Override

public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {

Contact selectedContact = contacts.get(position);

// Thông báo trước khi xóa

Toast.makeText(MainActivity.this,

"Xóa: " + selectedContact.getName(),

Toast.LENGTH\_SHORT).show();

// Xóa contact trong DB

db.deleteContact(selectedContact.getID());

// Log ra Logcat

Log.e("Delete", "Đã xóa contact: " + selectedContact);

// Load lại danh sách để cập nhật giao diện

loadContacts();

return true; // Quan trọng: phải return true

}

});

**Phần DatabaseHandler.java**

// ✅ Hàm xóa contact (cần cho phần long click)

public void deleteContact(int id) {

SQLiteDatabase db = this.getWritableDatabase();

db.delete(TABLE\_CONTACTS, KEY\_ID + "=?", new String[]{String.valueOf(id)});

db.close();

}

**A screenshot of a phone

Description automatically generated**