Họ và tên : Phạm Ngọc Anh Khôi

MSSV:1150070023

Bước 1 : CSDL dữ liệu có cấu trúc như sau:

A close-up of a data type

AI-generated content may be incorrect.

Bước 2 : Tạo một lớp Contact:

file Contact.java là một mô hình dữ liệu để biểu diễn một liên hệ. Lớp này chứa các thuộc tính (id, name, phoneNumber) và các phương thức cần thiết để làm việc với dữ liệu

package com.example.baitaplythuyet3;  
  
public class Contact {  
 private int id;  
 private String name;  
 private String phoneNumber;  
  
 // Constructor rỗng (bắt buộc cho SQLite)  
 public Contact() {  
 }  
  
 // Constructor khi thêm contact mới (không cần id vì SQLite tự sinh)  
 public Contact(String name, String phoneNumber) {  
 this.name = name;  
 this.phoneNumber = phoneNumber;  
 }  
  
 // Constructor đầy đủ (dùng khi lấy dữ liệu từ DB)  
 public Contact(int id, String name, String phoneNumber) {  
 this.id = id;  
 this.name = name;  
 this.phoneNumber = phoneNumber;  
 }  
  
 // Getter và Setter  
 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;  
 }  
}

A screenshot of a computer program

AI-generated content may be incorrect.

Bước 3 : Tạo 1 lớp DatabaseHandler kế thừa từ SQLiteOpenHelper

package com.example.baitaplythuyet3;  
  
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* = 2;  
 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 (chỉ thêm nếu chưa có số điện thoại)  
 public void addContact(Contact contact) {  
 if (getContactByPhone(contact.getPhoneNumber()) == null) {  
 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;  
 }  
  
 // Lấy contact theo phone number  
 public Contact getContactByPhone(String phone) {  
 SQLiteDatabase db = this.getReadableDatabase();  
 Cursor cursor = db.query(*TABLE\_CONTACTS*,  
 new String[]{*KEY\_ID*, *KEY\_NAME*, *KEY\_PHONE*},  
 *KEY\_PHONE* + "=?",  
 new String[]{phone}, null, null, null, null);  
  
 if (cursor != null && cursor.moveToFirst()) {  
 Contact contact = new Contact(  
 cursor.getInt(cursor.getColumnIndexOrThrow(*KEY\_ID*)),  
 cursor.getString(cursor.getColumnIndexOrThrow(*KEY\_NAME*)),  
 cursor.getString(cursor.getColumnIndexOrThrow(*KEY\_PHONE*))  
 );  
 cursor.close();  
 db.close();  
 return contact;  
 }  
  
 if (cursor != null) cursor.close();  
 db.close();  
 return null;  
 }  
  
 // Xóa contact  
 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 computer program

AI-generated content may be incorrect.

Bước 4 : Code MainActivity.java

MainActivity.java là màn hình chính của ứng dụng, chịu trách nhiệm xử lý logic và hiển thị dữ liệu.

package com.example.baitaplythuyet3;  
  
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 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);  
  
 // 🔹 Chỉ thêm dữ liệu mẫu khi bảng rỗng  
 if (db.getAllContacts().isEmpty()) {  
 addSampleContact("Pham Ngoc Anh Khôi", "1150070023");  
 addSampleContact("Jimmy", "1150070024");  
 addSampleContact("Tomm", "1150070025");  
 addSampleContact("Chris", "1150070026");  
 }  
  
 // Hiển thị dữ liệu lần đầu  
 loadContacts();  
  
 // Xóa contact khi nhấn giữ (long click)  
 lv.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {  
 @Override  
 public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {  
 Contact contact = contacts.get(position);  
 db.deleteContact(contact.getID());  
 loadContacts();  
 return true;  
 }  
 });  
 }  
  
 // Hàm chỉ thêm contact nếu chưa có số điện thoại đó  
 private void addSampleContact(String name, String phone) {  
 Contact c = db.getContactByPhone(phone);  
 if (c == null) {  
 db.addContact(new Contact(name, phone));  
 }  
 }  
  
 private void loadContacts() {  
 contacts = db.getAllContacts();  
 contactNames = new ArrayList<>();  
  
 for (Contact c : db.getAllContacts()) {  
 contactNames.add(c.getID() + " - " + c.getName() + " - " + c.getPhoneNumber());  
 }  
  
  
 adapter = new ArrayAdapter<>(this, android.R.layout.*simple\_list\_item\_1*, contactNames);  
 lv.setAdapter(adapter);  
 }  
}

A screenshot of a computer program

AI-generated content may be incorrect.

Bước 5: Code Activity\_main.xml để thiết kế layout

<?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:id="@+id/main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
  
 <ListView  
 android:id="@+id/lv\_contacts"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Bước 6: Kết Quả:

A screenshot of a cell phone

AI-generated content may be incorrect.

Bài thực hành nâng cao:

Quản lý danh sách sinh viên, thêm, xem, xóa, sửa, nhấn vào mỗi 1 sinh viên sẽ

rat rang chi tiết 1 sinh viên có sử dung Recyclerview và SQLite. Thiết kế giao diện

theo cảm nhận của mỗi bạn.

Bước 1 Tạo Activity2\_main.xml

activity2\_main.xml định nghĩa giao diện người dùng cho ứng dụng quản lý sinh viên. Giao diện bao gồm các thành phần:

* Ba trường

EditText để nhập mã số sinh viên (MSSV), tên và lớp.

* Bốn nút

Button cho các chức năng: INSERT (thêm), UPDATE (cập nhật), DELETE (xóa), và QUERY (truy vấn).

* Một

RecyclerView để hiển thị danh sách sinh viên một cách hiệu quả

A screen shot of a computer

AI-generated content may be incorrect.

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <EditText  
 android:id="@+id/edtMssv"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:minHeight="48dp"  
 android:hint="MSSV" />  
  
 <EditText  
 android:id="@+id/edtName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:minHeight="48dp"  
 android:hint="Tên sinh viên" />  
  
 <EditText  
 android:id="@+id/edtLop"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:minHeight="48dp"  
 android:hint="Lớp" />  
  
 <LinearLayout  
 android:orientation="horizontal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center">  
  
 <Button android:id="@+id/btnInsert" android:text="INSERT"  
 android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"/>  
 <Button android:id="@+id/btnUpdate" android:text="UPDATE"  
 android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"/>  
 <Button android:id="@+id/btnDelete" android:text="DELETE"  
 android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"/>  
 <Button android:id="@+id/btnQuery" android:text="QUERY"  
 android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"/>  
 </LinearLayout>  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recyclerView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"/>  
 </LinearLayout>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Bước 2 : Tạo DatabaseHelper.java

DatabaseHelper.java được tạo để quản lý CSDL cho danh sách sinh viên

A screenshot of a computer program

AI-generated content may be incorrect.

package com.example.baitaplythuyet3;  
  
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;  
  
public class DatabaseHelper extends SQLiteOpenHelper {  
  
 private static final String *DATABASE\_NAME* = "StudentDB";  
 private static final int *DATABASE\_VERSION* = 1;  
  
 private static final String *TABLE\_STUDENT* = "students";  
 private static final String *COL\_ID* = "mssv";  
 private static final String *COL\_NAME* = "name";  
 private static final String *COL\_CLASS* = "lop";  
  
 public DatabaseHelper(Context context) {  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 String CREATE\_TABLE = "CREATE TABLE " + *TABLE\_STUDENT* + "("  
 + *COL\_ID* + " TEXT PRIMARY KEY,"  
 + *COL\_NAME* + " TEXT,"  
 + *COL\_CLASS* + " TEXT)";  
 db.execSQL(CREATE\_TABLE);  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 db.execSQL("DROP TABLE IF EXISTS " + *TABLE\_STUDENT*);  
 onCreate(db);  
 }  
  
 // Insert  
 public void insertStudent(Student student) {  
 SQLiteDatabase db = this.getWritableDatabase();  
 ContentValues values = new ContentValues();  
 values.put(*COL\_ID*, student.getMssv());  
 values.put(*COL\_NAME*, student.getName());  
 values.put(*COL\_CLASS*, student.getLop());  
 db.insert(*TABLE\_STUDENT*, null, values);  
 db.close();  
 }  
  
 // Get all  
 public ArrayList<Student> getAllStudents() {  
 ArrayList<Student> list = new ArrayList<>();  
 SQLiteDatabase db = this.getReadableDatabase();  
 Cursor cursor = db.rawQuery("SELECT \* FROM " + *TABLE\_STUDENT*, null);  
 if (cursor.moveToFirst()) {  
 do {  
 list.add(new Student(  
 cursor.getString(0),  
 cursor.getString(1),  
 cursor.getString(2)  
 ));  
 } while (cursor.moveToNext());  
 }  
 cursor.close();  
 db.close();  
 return list;  
 }  
  
 // Update  
 public void updateStudent(Student student) {  
 SQLiteDatabase db = this.getWritableDatabase();  
 ContentValues values = new ContentValues();  
 values.put(*COL\_NAME*, student.getName());  
 values.put(*COL\_CLASS*, student.getLop());  
 db.update(*TABLE\_STUDENT*, values, *COL\_ID* + "=?", new String[]{student.getMssv()});  
 db.close();  
 }  
  
 // Delete  
 public void deleteStudent(String mssv) {  
 SQLiteDatabase db = this.getWritableDatabase();  
 db.delete(*TABLE\_STUDENT*, *COL\_ID* + "=?", new String[]{mssv});  
 db.close();  
 }  
}

Bước 3:Tạo MainActivity2.java

MainActivity2.java chứa logic chính để tương tác với người dùng và CSDL. Các chức năng nổi bật gồm:

* Ánh xạ các thành phần giao diện (EditText, Button, RecyclerView).
* Khởi tạo

RecyclerView và gắn StudentAdapter để hiển thị danh sách sinh viên.

* Xử lý sự kiện

onClick cho từng nút INSERT, UPDATE, DELETE, QUERY.

A screenshot of a computer program

AI-generated content may be incorrect.

package com.example.baitaplythuyet3;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
import java.util.ArrayList;  
  
public class MainActivity2 extends AppCompatActivity {  
  
 private EditText edtMssv, edtName, edtLop;  
 private Button btnInsert, btnUpdate, btnDelete, btnQuery;  
 private RecyclerView recyclerView;  
 private StudentAdapter adapter;  
 private ArrayList<Student> students;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity2\_main*); // layout bạn vừa gửi  
  
 // Ánh xạ view  
 edtMssv = findViewById(R.id.*edtMssv*);  
 edtName = findViewById(R.id.*edtName*);  
 edtLop = findViewById(R.id.*edtLop*);  
 btnInsert = findViewById(R.id.*btnInsert*);  
 btnUpdate = findViewById(R.id.*btnUpdate*);  
 btnDelete = findViewById(R.id.*btnDelete*);  
 btnQuery = findViewById(R.id.*btnQuery*);  
 recyclerView = findViewById(R.id.*recyclerView*);  
  
 // Khởi tạo danh sách sinh viên  
 students = new ArrayList<>();  
  
 // Gắn adapter cho RecyclerView  
 adapter = new StudentAdapter(this, students);  
  
 recyclerView.setLayoutManager(new LinearLayoutManager(this));  
 recyclerView.setAdapter(adapter);  
  
 // Xử lý nút Insert  
 btnInsert.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String mssv = edtMssv.getText().toString();  
 String name = edtName.getText().toString();  
 String lop = edtLop.getText().toString();  
  
 students.add(new Student(mssv, name, lop));  
 adapter.notifyDataSetChanged();  
 }  
 });  
  
 // Xử lý nút Update  
 btnUpdate.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String mssv = edtMssv.getText().toString();  
 for (Student s : students) {  
 if (s.getMssv().equals(mssv)) {  
 s.setName(edtName.getText().toString());  
 s.setLop(edtLop.getText().toString());  
 break;  
 }  
 }  
 adapter.notifyDataSetChanged();  
 }  
 });  
  
 // Xử lý nút Delete  
 btnDelete.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String mssv = edtMssv.getText().toString();  
 for (int i = 0; i < students.size(); i++) {  
 if (students.get(i).getMssv().equals(mssv)) {  
 students.remove(i);  
 break;  
 }  
 }  
 adapter.notifyDataSetChanged();  
 }  
 });  
  
 // Xử lý nút Query (ở đây mình chỉ hiển thị lại danh sách)  
 btnQuery.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 adapter.notifyDataSetChanged();  
 }  
 });  
 }  
}

Bước 4 :Tạo Student.java

Student (tên file Student.java) là mô hình dữ liệu (model) để biểu diễn một sinh viên. Nó chứa các thuộc tính như

mssv, name, và lop cùng với các phương thức getter/setter tương ứng.

package com.example.baitaplythuyet3;  
  
public class Student {  
 private String mssv;  
 private String name;  
 private String lop;  
  
 public Student(String mssv, String name, String lop) {  
 this.mssv = mssv;  
 this.name = name;  
 this.lop = lop;  
 }  
  
 public String getMssv() { return mssv; }  
 public String getName() { return name; }  
 public String getLop() { return lop; }  
  
 public void setMssv(String mssv) { this.mssv = mssv; }  
 public void setName(String name) { this.name = name; }  
 public void setLop(String lop) { this.lop = lop; }  
}

A screenshot of a computer program

AI-generated content may be incorrect.

Bước 5 :Tạo Student Adapter.java

StudentAdapter (tên file StudentAdapter.java) là thành phần cốt lõi để hiển thị dữ liệu trong RecyclerView. Adapter này có nhiệm vụ:

* Tạo ra các

view holder (khung chứa các view con) cho mỗi mục trong danh sách.

* Gán dữ liệu của từng sinh viên vào các view tương ứng.
* Xử lý sự kiện khi người dùng nhấn vào một sinh viên, chuyển sang màn hình chi tiết (

StudentDetailActivity) và truyền dữ liệu đi.

package com.example.baitaplythuyet3;  
  
import android.content.Context;  
import android.content.Intent;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
import java.util.ArrayList;  
  
public class StudentAdapter extends RecyclerView.Adapter<StudentAdapter.ViewHolder> {  
  
 private ArrayList<Student> students;  
 private Context context;  
  
 public StudentAdapter(Context context, ArrayList<Student> students) {  
 this.context = context;  
 this.students = students;  
 }  
  
 @NonNull  
 @Override  
 public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View v = LayoutInflater.*from*(context).inflate(R.layout.*activity\_student*, parent, false);  
 return new ViewHolder(v);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull ViewHolder holder, int position) {  
 Student s = students.get(position);  
 holder.txtName.setText(s.getMssv() + " - " + s.getName());  
  
 holder.itemView.setOnClickListener(v -> {  
 Intent intent = new Intent(context, StudentDetailActivity.class);  
 intent.putExtra("mssv", s.getMssv());  
 intent.putExtra("name", s.getName());  
 intent.putExtra("lop", s.getLop());  
 context.startActivity(intent);  
 });  
 }  
  
 @Override  
 public int getItemCount() {  
 return students.size();  
 }  
  
 public static class ViewHolder extends RecyclerView.ViewHolder {  
 TextView txtName;  
 public ViewHolder(@NonNull View itemView) {  
 super(itemView);  
 txtName = itemView.findViewById(R.id.*txt\_student*);  
 }  
 }  
}

A screenshot of a computer screen

AI-generated content may be incorrect.

Bước 6 : Tạo item Student.xml

item\_student.xml định nghĩa layout cho mỗi mục (item) trong RecyclerView. Layout này đơn giản chỉ chứa một

TextView để hiển thị thông tin tóm tắt của sinh viên.

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="8dp">  
  
 <TextView  
 android:id="@+id/txt\_student"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Sinh viên"  
 android:textSize="18sp"  
 android:textColor="#000"/>  
</LinearLayout>

A screen shot of a computer program

AI-generated content may be incorrect.

Bước 7 : Tạo StudentDetailActivy.java

StudentDetailActivity.java là màn hình dùng để hiển thị thông tin chi tiết của một sinh viên khi người dùng nhấn vào. Nó nhận dữ liệu được gửi từ

StudentAdapter và hiển thị lên một TextView.

package com.example.baitaplythuyet3;  
  
import android.os.Bundle;  
import android.util.Log;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class StudentDetailActivity extends AppCompatActivity {  
 private static final String *TAG* = "StudentDetailAct";  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
 // dùng layout detail chính xác  
 setContentView(R.layout.*activity\_student*);  
  
 TextView txtDetail = findViewById(R.id.*txtDetail*);  
 if (txtDetail == null) {  
 Log.*e*(*TAG*, "txtDetail == null -> kiểm tra activity\_student\_detail.xml");  
 return;  
 }  
  
 String mssv = getIntent().getStringExtra("mssv");  
 String name = getIntent().getStringExtra("name");  
 String lop = getIntent().getStringExtra("lop");  
  
 if (mssv == null) mssv = "";  
 if (name == null) name = "";  
 if (lop == null) lop = "";  
  
 txtDetail.setText("MSSV: " + mssv + "\nHọ tên: " + name + "\nLớp: " + lop);  
 }  
}

A screenshot of a computer program

AI-generated content may be incorrect.

Bước 8 : Vào Manifests và chọn AndroidManifests chỉnh lại thành .MainActivity2 để chạy

A screenshot of a computer program

AI-generated content may be incorrect.

Kết quả :

