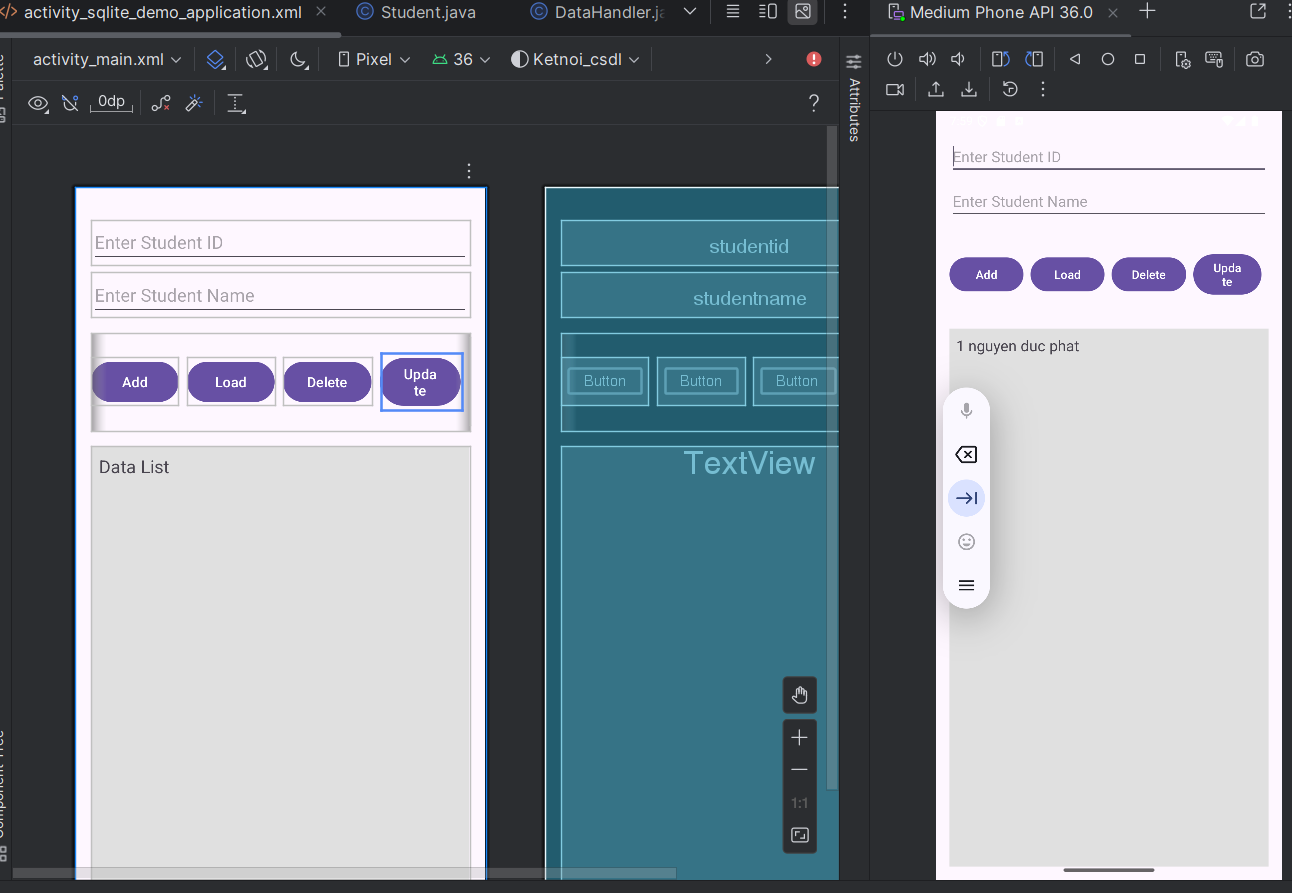
Bài tập tuần 4

Nguyễn Đưc Phát 1150090152

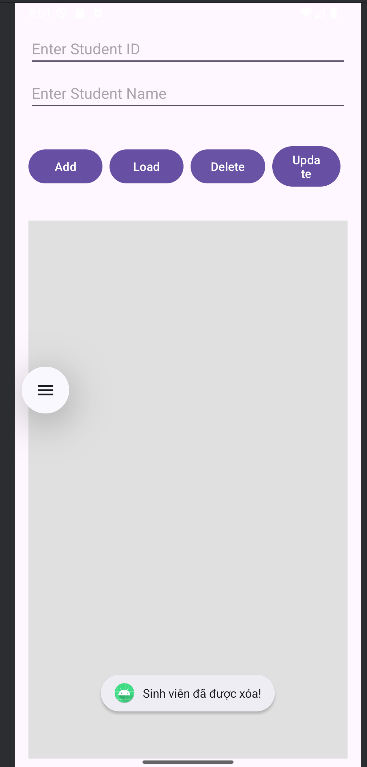
Lớp: THMT

Em xin phép gửi kết quả trước còn cácbước, code em sẽ để ở dưới sau khi trình bày tường phần

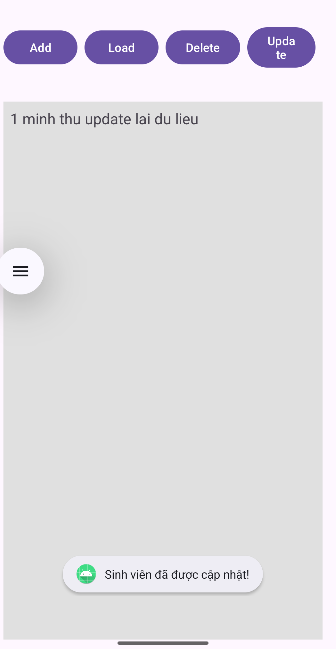
1. Ảnh kết quả khi thêm dữ liệu



1. Kết quả khi xóa sinh viên



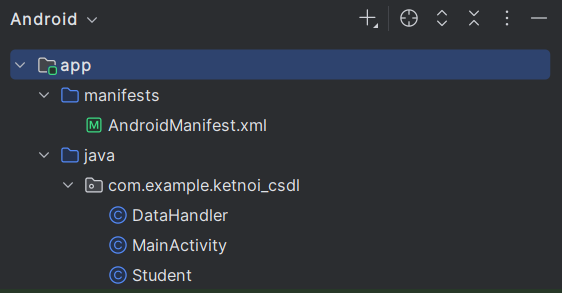
1. Kết quả khi update lại



**I/ Tạo lớp mô hình dữ liệu và lớp xử lý dữ liệu**

**(Student.java và DataHandler.java)**

Tại dự án em tạo studen và datahandler trong thu com.example.ketnoi\_csdl (gói dữ liệu đầu tiên trong thư mục JAVA)



Nhấp chuột phải vào gói đó, chọn **New** > **Java Class**. Đặt tên lớp là Student và DataHandler

**Mã code của Student**

package com.ngocminhtran.sqlitedemoapplication;

public class Student {

// Biến tương ứng với các cột trong bảng Students

private int \_studentid;

private String \_studentname;

// Constructor mặc định

public Student() {

}

// Constructor có tham số

public Student(int id, String name) {

this.\_studentid = id;

this.\_studentname = name;

}

// Các phương thức getter và setter

public int getStudentID() {

return this.\_studentid;

}

public void setStudentID(int id) {

this.\_studentid = id;

}

public String getStudentName() {

return this.\_studentname;

}

public void setStudentName(String name) {

this.\_studentname = name;

}

}

**Mã code của DataHandler**

package com.ngocminhtran.sqlitedemoapplication;

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 DataHandler extends SQLiteOpenHelper {

// Các biến mô tả CSDL

private static final int DATABASE\_VERSION = 1;

private static final String DATABASE\_NAME = "StudentsDB.db";

public static final String TABLE\_NAME = "Students";

public static final String COLUMN\_ID = "StudentID";

public static final String COLUMN\_NAME = "StudentName";

// Constructor

public DataHandler(Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {

super(context, DATABASE\_NAME, factory, DATABASE\_VERSION);

}

@Override

public void onCreate(SQLiteDatabase db) {

// Câu lệnh SQL để tạo bảng Students

String CREATE\_STUDENTS\_TABLE = "CREATE TABLE " +

TABLE\_NAME + "(" +

COLUMN\_ID + " INTEGER PRIMARY KEY," +

COLUMN\_NAME + " TEXT )";

db.execSQL(CREATE\_STUDENTS\_TABLE);

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

// Xóa bảng nếu đã tồn tại

db.execSQL("DROP TABLE IF EXISTS " + TABLE\_NAME);

// Tạo lại bảng mới

onCreate(db);

}

// Các phương thức CRUD (Create, Read, Update, Delete)

public void addDataHandler(Student student) {

ContentValues values = new ContentValues();

values.put(COLUMN\_ID, student.getStudentID());

values.put(COLUMN\_NAME, student.getStudentName());

SQLiteDatabase db = this.getWritableDatabase();

db.insert(TABLE\_NAME, null, values);

db.close();

}

public String loadDataHandler() {

String result = "";

String query = "SELECT \* FROM " + TABLE\_NAME;

SQLiteDatabase db = this.getWritableDatabase();

Cursor cursor = db.rawQuery(query, null);

while (cursor.moveToNext()) {

int result\_0 = cursor.getInt(0);

String result\_1 = cursor.getString(1);

result += String.valueOf(result\_0) + " " + result\_1 + System.getProperty("line.separator");

}

cursor.close();

db.close();

return result;

}

public void deleteDataHandler(int id) {

SQLiteDatabase db = this.getWritableDatabase();

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

db.close();

}

public void updateDataHandler(int id, String name) {

ContentValues values = new ContentValues();

values.put(COLUMN\_NAME, name);

SQLiteDatabase db = this.getWritableDatabase();

db.update(TABLE\_NAME, values, COLUMN\_ID + "=?", new String[]{String.valueOf(id)});

db.close();

}

public Student findFirstDataHandler(String studentname) {

String query = "SELECT \* FROM " + TABLE\_NAME + " WHERE " + COLUMN\_NAME + " = \"" + studentname + "\"";

SQLiteDatabase db = this.getWritableDatabase();

Cursor cursor = db.rawQuery(query, null);

Student student = null;

if (cursor.moveToFirst()) {

int result\_0 = cursor.getInt(0);

String result\_1 = cursor.getString(1);

student = new Student(result\_0, result\_1);

}

cursor.close();

db.close();

return student;

}

public List<Student> findAllDataHandler(String studentname) {

String query = "SELECT \* FROM " + TABLE\_NAME + " WHERE " + COLUMN\_NAME + " = \"" + studentname + "\"";

SQLiteDatabase db = this.getWritableDatabase();

Cursor cursor = db.rawQuery(query, null);

List<Student> students = new ArrayList<>();

if (cursor.moveToFirst()) {

do {

int id = cursor.getInt(0);

String name = cursor.getString(1);

students.add(new Student(id, name));

} while (cursor.moveToNext());

}

cursor.close();

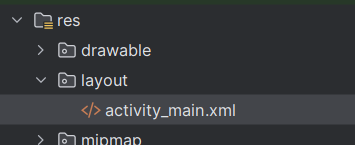
db.close();

return students;

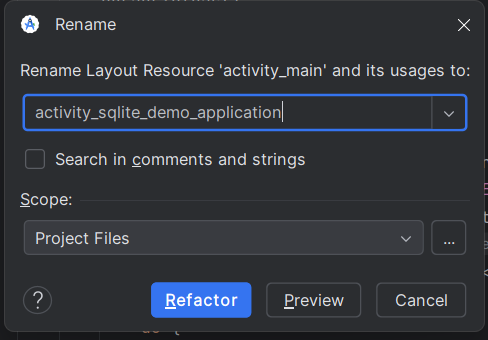
}

}

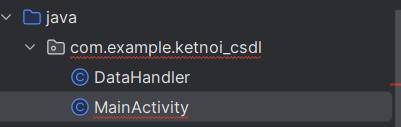
II/ Thiết kế giao diện người dùng (activity\_sqlite\_demo\_application.xml)



Tại trong lớp layout khi tạo dự án mới mặc dịnh có tên là activity\_main.xml để thuận tiện với tài liệu của thầy em xẽ đổi tên giống thầy ghi trong tài liệu



Khi đổi tên thì nên tự viết tay không nên copy-paste để tránh gặp lỗi



-sẵn tiện đổi tên file MainActivity.java và chọn **Refactor** > **Rename** nhập tên mới là **SQLiteDemoApplicationActivity**

**Mã Code activity\_sqlite\_demo\_application.xml**

<?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"  
 android:padding="16dp"  
 tools:context=".SQLiteDemoApplicationActivity">  
  
 <EditText  
 android:id="@+id/studentid"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:hint="Enter Student ID"  
 android:inputType="number"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <EditText  
 android:id="@+id/studentname"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:hint="Enter Student Name"  
 android:inputType="textPersonName"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/studentid" />  
  
 <LinearLayout  
 android:id="@+id/button\_layout"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:orientation="horizontal"  
 android:gravity="center"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/studentname">  
  
 <Button  
 android:id="@+id/btnAdd"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="addStudent"  
 android:text="Add"  
 android:layout\_margin="4dp"/>  
  
 <Button  
 android:id="@+id/btnLoad"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="loadStudents"  
 android:text="Load"  
 android:layout\_margin="4dp"/>  
  
 <Button  
 android:id="@+id/btnDelete"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="deleteStudent"  
 android:text="Delete"  
 android:layout\_margin="4dp"/>  
  
 <Button  
 android:id="@+id/btnUpdate"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="updateStudent"  
 android:text="Update"  
 android:layout\_margin="4dp"/>  
  
 <Button  
 android:id="@+id/btnFind"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="findFirstStudent"  
 android:text="Find"  
 android:layout\_margin="4dp"/>  
 </LinearLayout>  
  
 <TextView  
 android:id="@+id/txtData"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:layout\_marginTop="16dp"  
 android:background="#e0e0e0"  
 android:padding="8dp"  
 android:text="Data List"  
 android:textSize="18sp"  
 android:scrollbars="vertical"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/button\_layout" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**Mã code activity\_sqlite\_demo\_application.xml**

package com.example.ketnoi\_csdl;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import java.util.List;  
  
public class SQLiteDemoApplicationActivity extends AppCompatActivity {  
  
 EditText studentid, studentname;  
 TextView datalist;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_sqlite\_demo\_application);  
  
 studentid = findViewById(R.id.studentid);  
 studentname = findViewById(R.id.studentname);  
 datalist = findViewById(R.id.txtData);  
 }  
  
 public void addStudent(View view) {  
 DataHandler dbHandler = new DataHandler(this, null, null, 1);  
 try {  
 int id = Integer.parseInt(studentid.getText().toString());  
 String name = studentname.getText().toString();  
 Student student = new Student(id, name);  
 dbHandler.addDataHandler(student);  
 Toast.makeText(this, "Sinh viên đã được thêm!", Toast.LENGTH\_SHORT).show();  
 studentid.setText("");  
 studentname.setText("");  
 loadStudents(view);  
 } catch (NumberFormatException e) {  
 Toast.makeText(this, "ID phải là số!", Toast.LENGTH\_SHORT).show();  
 }  
 }  
  
 public void loadStudents(View view) {  
 DataHandler dbHandler = new DataHandler(this, null, null, 1);  
 datalist.setText(dbHandler.loadDataHandler());  
 }  
  
 public void deleteStudent(View view) {  
 DataHandler dbHandler = new DataHandler(this, null, null, 1);  
 try {  
 int id = Integer.parseInt(studentid.getText().toString());  
 dbHandler.deleteDataHandler(id);  
 Toast.makeText(this, "Sinh viên đã được xóa!", Toast.LENGTH\_SHORT).show();  
 studentid.setText("");  
 studentname.setText("");  
 loadStudents(view);  
 } catch (NumberFormatException e) {  
 Toast.makeText(this, "ID phải là số!", Toast.LENGTH\_SHORT).show();  
 }  
 }  
  
 public void updateStudent(View view) {  
 DataHandler dbHandler = new DataHandler(this, null, null, 1);  
 try {  
 int id = Integer.parseInt(studentid.getText().toString());  
 String name = studentname.getText().toString();  
 dbHandler.updateDataHandler(id, name);  
 Toast.makeText(this, "Sinh viên đã được cập nhật!", Toast.LENGTH\_SHORT).show();  
 studentid.setText("");  
 studentname.setText("");  
 loadStudents(view);  
 } catch (NumberFormatException e) {  
 Toast.makeText(this, "ID phải là số!", Toast.LENGTH\_SHORT).show();  
 }  
 }  
  
 public void findFirstStudent(View view) {  
 DataHandler dbHandler = new DataHandler(this, null, null, 1);  
 String name = studentname.getText().toString();  
 Student student = dbHandler.findFirstDataHandler(name);  
 if (student != null) {  
 datalist.setText("ID: " + student.getStudentID() + "\nName: " + student.getStudentName());  
 } else {  
 datalist.setText("Không tìm thấy sinh viên có tên " + name);  
 }  
 }  
  
 public void findAllStudent(View view) {  
 DataHandler dbHandler = new DataHandler(this, null, null, 1);  
 String name = studentname.getText().toString();  
 List<Student> lst = dbHandler.findAllDataHandler(name);  
 String studentsList = "";  
 if (!lst.isEmpty()) {  
 for (Student st:lst) {  
 studentsList += "ID: " + String.valueOf(st.getStudentID())  
 + ", Tên: " + st.getStudentName()  
 + System.getProperty("line.separator");  
 }  
 datalist.setText(studentsList);  
 } else {  
 datalist.setText("Không tìm thấy sinh viên có tên " + name);  
 }  
 }  
}