

BÀI 10. PHIẾU BÀI TẬP

BÀI TẬP TRƯỚC KHI LÊN LỚP: Bài 10.1

BÀI TẬP THỰC HIỆN TRÊN LỚP: Bài 10.2

BÀI TẬP SAU KHI LÊN LỚP: Bài 10.3, bài 10.4, bài 10.5.

MỤC LỤC

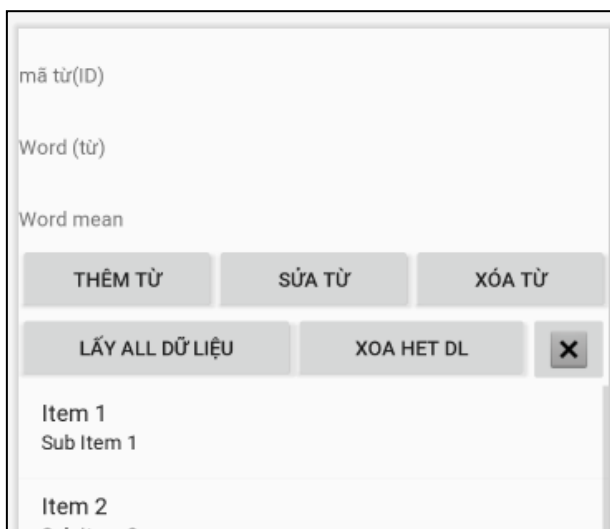
Bài tập 10.1 . Tạo ứng dụng học từ mới như hình dưới đây:	1
Bài tập 10.2. Xây dựng ứng dụng quản lý sách với SQLite.....	11
Bài tập 10.3. Xây dựng ứng dụng quản lý bán tạp hóa	27
Bài tập 10.4. Xây dựng ứng dụng sổ học bạ online	28
Bài tập 10.5. Xây dựng ứng dụng chat online thời gian thực.....	28

Bài tập 10.1 . Tạo ứng dụng học từ mới như hình dưới đây:

Yêu cầu:

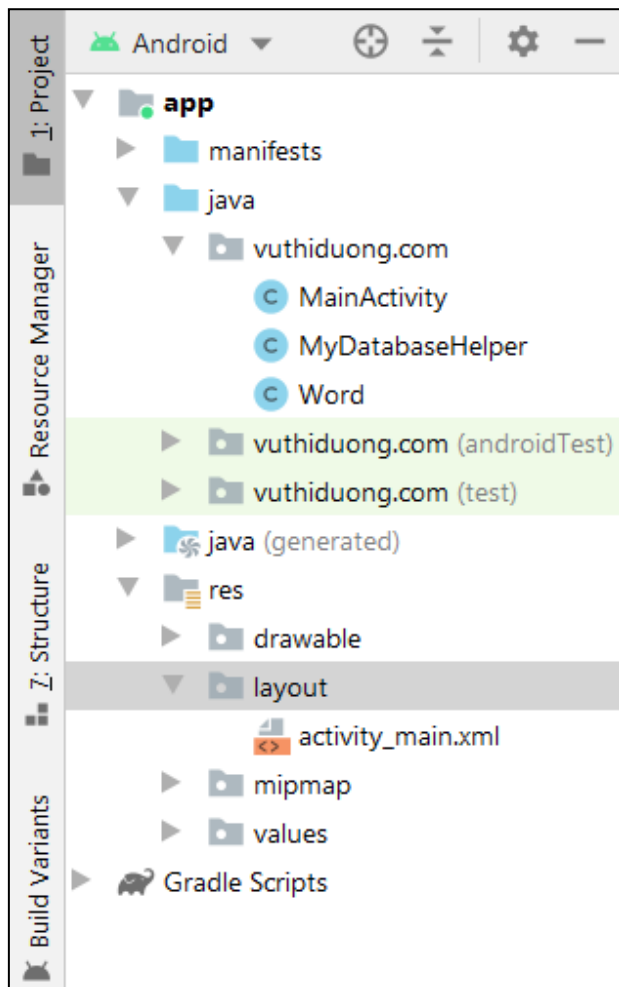
- Tạo CSDL SQLite : **DICTIONARY_DB**
- Tạo bảng Word gồm : **ID, Word, Mean**
- Xây dựng class Word để xử lý nghiệp vụ
- Xây dựng các chức năng như hình: Thêm từ, Sửa từ, xóa từ, lấy tất cả dữ liệu, xóa dữ liệu, hiển thị ở listView

Layout



The screenshot shows a mobile application interface for a dictionary. It has three input fields at the top: 'mã từ(ID)', 'Word (từ)', and 'Word mean'. Below these are three buttons: 'THÊM TỪ', 'SỬA TỪ', and 'XÓA TỪ'. Further down are two more buttons: 'LẤY ALL DỮ LIỆU' and 'XÓA HET DL', followed by a close button with an 'X' icon. At the bottom, there is a list view displaying 'Item 1' with 'Sub Item 1' and 'Item 2' with 'Sub Item 2'.

Cấu trúc ứng dụng



Activity_main.xml

```
<LinearLayout 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/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.com.MainActivity" >

    <TableRow
        android:id="@+id/tableRow1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="mã từ(ID)" />

        <EditText
            android:id="@+id/editID"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            >

            <requestFocus />
        </EditText>
```

```
</TableRow>

<TableRow
    android:id="@+id/tableRow2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Word (từ)" />

    <EditText
        android:id="@+id/editWord"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Từ word" />

</TableRow>

<TableRow
    android:id="@+id/tableRow3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Word mean" />

    <EditText
        android:id="@+id/editWordMean"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10" />

</TableRow>

<TableRow
    android:id="@+id/tableRow4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:weightSum="3"
    >

    <Button
        android:id="@+id/btnThemTu"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Thêm từ" />

    <Button
        android:id="@+id/btnSuaTu"
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:text="Sửa từ" />
```

```
<Button
    android:id="@+id/btnXoaTu"
    android:layout_width="0dp"
    android:layout_weight="1"
    android:layout_height="wrap_content"
    android:text="Xóa từ" />

</TableRow>

<TableRow
    android:id="@+id/tableRow5"
    android:layout_width="match_parent"
    android:weightSum="2"
    android:layout_height="wrap_content" >

    <Button
        android:id="@+id/btnLayAllDL"
        android:layout_weight="1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Lấy all Dữ liệu" />

    <Button
        android:id="@+id/btnXoaHetDL"
        android:layout_weight="1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Xoa het DL" />

    <ImageButton
        android:id="@+id/imgClose"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@android:drawable/btn_dialog" />

</TableRow>

<ListView
    android:id="@+id/lvWord"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
</ListView>

</LinearLayout>
```

Word.java

```
package vuthiduong.com;
public class Word {
    private int mId;
    private String mWord;
    private String mMean;

    public Word() {

    }

    public Word(int id, String word, String mean) {
        this.mId = id;
        this.mWord = word;
        this.mMean = mean;
    }
}
```

```
}

public Word(String word, String mean) {
    this.mWord = word;
    this.mMean = mean;
}

public void setMean(String mean) {
    this.mMean = mean;
}

public void setWord(String word) {
    this.mWord = word;
}

public int getId() {
    return mId;
}

public String getWord() {
    return mWord;
}

public String getMean() {
    return mMean;
}

@Override
public String toString() {
    return mId + "-" + mWord + ":" + mMean;
}

@Override
public int hashCode() {
    final int prime = 31;
    int result = 1;
    result = prime * result + mId;
    return result;
}

@Override
public boolean equals(Object obj) {
    if (this == obj)
        return true;
    if (obj == null)
        return false;
    if (getClass() != obj.getClass())
        return false;
    Word other = (Word) obj;
    if (mId != other.mId)
        return false;
    return true;
}
}
```

MyDatabaseHelper.java

```
package vuthiduong.com;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
```

```
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

import java.util.ArrayList;
import java.util.List;

public class MyDatabaseHelper extends SQLiteOpenHelper {
    private static final String TAG = "MyDatabaseHelper";
    private static final String DATABASE_NAME = "DICTIONARY_DB";
    private static final int DATABASE_VERSION = 1;
    private static final String TABLE_WORD = "WORD";
    private static final String ID_COLUMN = "id";
    private static final String WORD_COLUMN = "word";
    private static final String MEAN_COLUMN = "mean";
    private static final String CREATE_WORD_TABLE_SQL =
        "CREATE TABLE IF NOT EXISTS " + TABLE_WORD + " (" +
        ID_COLUMN + " INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT," +
        WORD_COLUMN + " TEXT NOT NULL," +
        MEAN_COLUMN + " TEXT NOT NULL" +
        ")";
    private static MyDatabaseHelper sInstance;

    public static MyDatabaseHelper getInstance(Context context) {
        if (sInstance == null) {
            sInstance = new MyDatabaseHelper(context.getApplicationContext());
        }
        return sInstance;
    }
    private MyDatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        Log.e(TAG, "onCreate: ");
        try{
            // db.execSQL("DROP TABLE IF EXISTS " + TABLE_WORD);
            db.execSQL(CREATE_WORD_TABLE_SQL);
        } catch (Exception e){
            Log.e(TAG,e.toString());
        }
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        Log.e(TAG, "onUpgrade: ");
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_WORD);
        onCreate(db);
    }

    public boolean insertWord(Word word) {
        /**
         * Long insert(String table, String nullColumnHack, ContentValues values)
         * chèn một bản ghi trên các cơ sở dữ liệu. Bảng chỉ định tên bảng,
         * nullColumnHack không cho phép các giá trị hoàn toàn vô giá trị.
         * Nếu số thứ hai là null, android sẽ lưu trữ các giá trị null nếu giá
         * trị
         *
         * Là trống rỗng. Đối số thứ ba xác định các giá trị được lưu trữ.
         */
        SQLiteDatabase db = getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(WORD_COLUMN, word.getWord());
        values.put(MEAN_COLUMN, word.getMean());
    }
}
```

```
        long rowId = db.insert(TABLE_WORD, null, values);
        db.close();
        if (rowId != -1)
            return true;
        return false;
    }
    /**
     * Cursor query(String table, String[] columns, String selection, String[]
     * selectionArgs,
     * String groupBy, String having, String orderBy)
     */
    public Word getWord(String id) {
        SQLiteDatabase db = getReadableDatabase();
        Word word = null;
        Cursor cursor = db.query(TABLE_WORD, new String[]{ID_COLUMN,
WORD_COLUMN, MEAN_COLUMN},
            WORD_COLUMN + " = ?",
            new String[]{String.valueOf(id)}, null, null, null);
        if (cursor != null && cursor.moveToFirst()) {
            word = new Word(cursor.getInt(0), cursor.getString(1),
cursor.getString(2));
            cursor.close();
        }
        db.close();
        return word;
    }
    /**
     * thực hiện 1 câu truy vấn trực tiếp
     */
    public List<Word> getAllWord() {
        SQLiteDatabase db = getReadableDatabase();
        List<Word> words = new ArrayList<Word>();
        String sql = "SELECT * FROM " + TABLE_WORD;
        Cursor cursor = db.rawQuery(sql, null);
        if (cursor != null && cursor.moveToFirst()) {
            do {
                words.add(new
Word(cursor.getInt(cursor.getColumnIndex(ID_COLUMN)),
                    cursor.getString(cursor.getColumnIndex(WORD_COLUMN)),
                    cursor.getString(cursor.getColumnIndex(MEAN_COLUMN))));
            } while (cursor.moveToNext());
            cursor.close();
        }
        db.close();
        return words;
    }
    //Đếm tổng số dòng trong database
    public int getTotalWord() {
        SQLiteDatabase db = getReadableDatabase();
        String sql = "SELECT * FROM " + TABLE_WORD;
        Cursor cursor = db.rawQuery(sql, null);
        int totalRows = cursor.getCount();
        cursor.close();
        return totalRows;
    }
    /**
     * int update(String table, ContentValues values, String whereClause,
     * String[] whereArgs)
     * @param word
     * @return
     */
}
```

```
public int updateWord(Word word) {
    SQLiteDatabase db = getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(WORD_COLUMN, word.getWord());
    values.put(MEAN_COLUMN, word.getMean());
    int rowEffect = db.update(TABLE_WORD, values, ID_COLUMN + " = ?",
        new String[]{String.valueOf(word.getId())});
    db.close();
    return rowEffect;
}

public int deleteAllWord() {
    SQLiteDatabase db = getReadableDatabase();
    int rowEffect = db.delete(TABLE_WORD, null, null);
    db.close();
    return rowEffect;
}

public int deleteWord(Word word) {
    SQLiteDatabase db = getReadableDatabase();
    int rowEffect = db.delete(TABLE_WORD, ID_COLUMN + " = ?", new
String[]{String.valueOf(word.getId())});
    db.close();
    return rowEffect;
}
}
```

MainActivity.java

```
package vuthiduong.com;

import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.ListView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.List;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity implements
View.OnClickListener{
    private static final String TAG = "MainActivity";
    EditText editWord, editWordMean, editID;
    Button btnXoaHetDL, btnThemTu, btnXoaTu, btnLayDL, btnSuaTu;
    ImageButton imgClose;
    ListView lvDsTu;
    ArrayAdapter<Word> adapter=null;
    List<Word> dsTu=new ArrayList<Word>();

    MyDatabaseHelper db=null;
    Word wordSelected=new Word();
    // OnClickListner myClick=new OnClickListner() {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```



```
//Lấy điều khiển
getFromWidget();
//get Instance.: csLL, tạo bảng
db = MyDatabaseHelper.getInstance(this);
//Toast.makeText(MainActivity.this, "Tạo csdl, tạo bảng",
Toast.LENGTH_LONG).show();
//insert data
if (db.getTotalWord()==0) fakeData();
//chuẩn bị dữ liệu
//gán dl cho listview

refreshDataForListView();

Toast.makeText(MainActivity.this, "so bg:"+dsTu.size(),
Toast.LENGTH_LONG).show();
// Button btnXoaHetDL, btnThemTu, btnXoaTu, btnLayHetTu, btnSuaTu;
btnXoaHetDL.setOnClickListener(MainActivity.this);
btnThemTu.setOnClickListener(MainActivity.this);
btnXoaTu.setOnClickListener(MainActivity.this);
btnLayDL.setOnClickListener(MainActivity.this);
btnSuaTu.setOnClickListener(MainActivity.this);
imgClose.setOnClickListener(MainActivity.this);
lvDsTu.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
        long arg3) {
        wordSelected=dsTu.get(arg2);
        editID.setText(wordSelected.getId()+"");
        editWord.setText(wordSelected.getWord()+"");
        editWordMean.setText(wordSelected.getMean()+"");
    }
});
}
public void refreshDataForListView(){
    try {
        dsTu=db.getAllWord();
        adapter=new ArrayAdapter<Word>(
            MainActivity.this,
            android.R.layout.simple_list_item_1,
            dsTu);
        lvDsTu.setAdapter(adapter);
        adapter.notifyDataSetChanged();
    }catch (Exception e){
        System.out.println(" có lỗi: "+e.toString());
    }
}

public void fakeData(){
    //chèn ví dụ 3 từ
    db.deleteAllWord();
    db.insertWord(new Word("book", "Sách(n), đặt chỗ(v)"));
    db.insertWord(new Word("table", "bàn(n)"));
    db.insertWord(new Word("action movie", "Phim hành động "));
}
public void getFromWidget(){

    editID=(EditText) findViewById(R.id.editID);
    editWord=(EditText) findViewById(R.id.editWord);
    editWordMean=(EditText) findViewById(R.id.editWordMean);
    //Lấy các nút lệnh
    //btnTaoBang=(Button) findViewById(R.id.btnCreateTable);
    btnThemTu= findViewById(R.id.btnThemTu);
```

```
btnLayDL= findViewById(R.id.btnLayALLDL);
btnXoaTu= findViewById(R.id.btnXoaTu);
btnSuaTu= findViewById(R.id.btnSuaTu);
btnXoaHetDL= findViewById(R.id.btnXoaHetDL);
imgClose=findViewById(R.id.imgClose);
//lấy ListView
lvDsTu=(ListView) findViewById(R.id.lvWord);
//tự sinh mã số
editID.setEnabled(false);
}

@Override
public void onClick(View view) {
    switch(view.getId())
    {
        case R.id.btnXoaHetDL:
            db.deleteAllWord();
            refreshDataForListView();
            break;
        case R.id.btnThemTu:
            them();
            break;
        case R.id.btnXoaTu:
            xoaTu();
            break;
        case R.id.btnLayALLDL:
            refreshDataForListView();
            break;
        case R.id.btnSuaTu:
            capNhatTu();
            break;
        case R.id.imgClose:
            finish();
            break;
    } //end of switch
}

public void them(){
    Word w=new Word();
    w.setWord(editWord.getText()+"");
    w.setMean(editWordMean.getText()+"");
    boolean t=db.insertWord(w);
    //laays lai trang thai s mowis nhats tuwf csdl
    Toast.makeText(this, " chèn dòng + "+ t+":"+w.toString(),
    Toast.LENGTH_SHORT).show();
    //dsTu.clear();
    refreshDataForListView();
}

public void capNhatTu(){
    Word w = wordSelected;
    w.setWord(editWord.getText()+"");
    w.setMean(editWordMean.getText()+"");
    db.updateWord(w);
    Toast.makeText(this, " sửa : "+w.toString(), Toast.LENGTH_SHORT).show();
    refreshDataForListView();
}

public void laySoDong(){
    //loge total rows in database
    Log.e(TAG, "onCreate: " + db.getTotalWord());
}

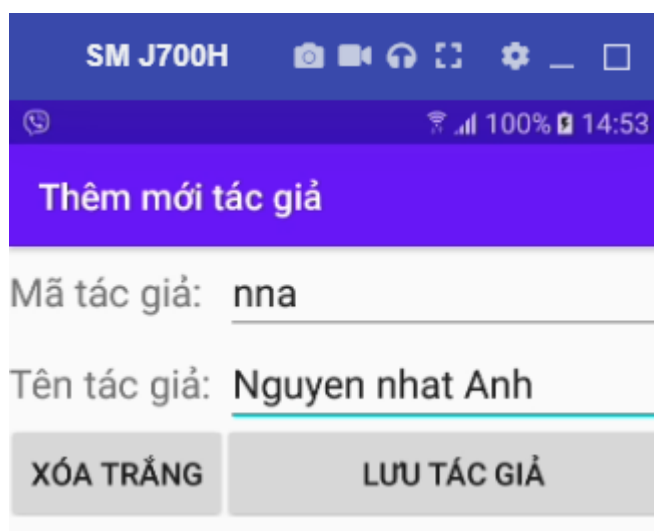
public void xoaTu(){
    //delete word
    db.deleteWord(wordSelected);
}
```

```
refreshDataForListView();  
}  
}
```

Bài tập 10.2. Xây dựng ứng dụng quản lý sách với SQLite



Chọn thêm tác giả



Chọn xem danh sách tác giả

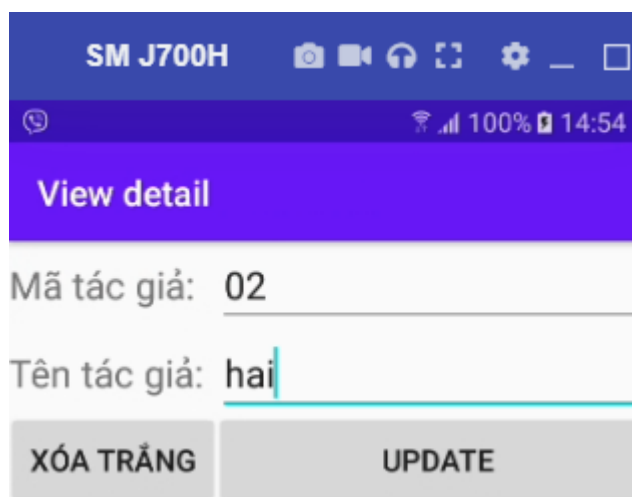


SM J700H 100% 14:54

Bai28_QLlite_QLSach

STT	Mã tác giả.	Tên tác giả.
1	01	mot
2	02	002
3	003	ba
4	nna	Nguyen nhat Anh

Chọn 1 dòng cho listview cho phép sửa tác giả



SM J700H 100% 14:54

View detail

Mã tác giả: 02

Tên tác giả: hai

XÓA TRẮNG UPDATE

Xem kết quả cập nhật

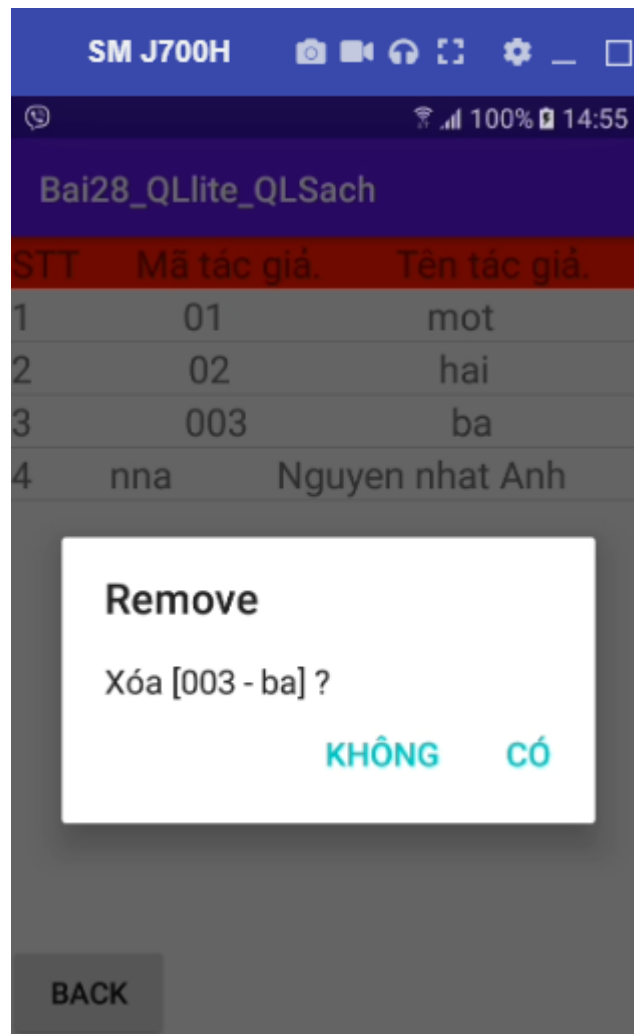


SM J700H 100% 14:55

Bai28_QLlite_QLSach

STT	Mã tác giả.	Tên tác giả.
1	01	mot
2	02	hai
3	003	ba
4	nna	Nguyen nhat Anh

Chọn lâu 1 dòng listview danh sách tác giả



Chọ có sẽ xóa tác giả vừa chọn.

Thực hiện chương trình

1. cấu trúc ứng dụng và Android mainifesh

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="vuthiduong.com">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:label="Thêm mới tác giả"
            android:name=".CreateAuthorActivity"> </activity>
        <activity
            android:name=".ShowListAuthorActivity" android:label="Danh sách tác
```

```
giả"> </activity>
    <activity
        android:label="Quản lý sách" android:name=".InsertBookActivity">
    </activity>
</application>
</manifest>
```

Activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    tools:context=".MainActivity" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="#004000"
        android:gravity="center"
        android:text="Quản Lý Sách"
        android:textColor="#FFFFFF"
        android:textSize="20sp" />

    <Button
        android:id="@+id/btnInsertAuthor"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="center"
        android:text="Thêm Tác Giả" />

    <Button
        android:id="@+id/btnShowAuthorList"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="center"
        android:text="Xem danh sách Tác Giả" />

    <Button
        android:id="@+id/btnShowBookActivity"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="center"
        android:text="Quản lý Sách" />

</LinearLayout>
```

activity_insert_book.xml

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/TableLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:stretchColumns="*"
    >
```

```
>
<TableRow
    android:id="@+id/tableRow1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tác giả:" />
    <Spinner
        android:id="@+id/spinner1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_span="2" />
</TableRow>

<TableRow
    android:id="@+id/tableRow2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tên sách:" />

    <EditText
        android:id="@+id/editTextTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_span="2"
        android:ems="10" >
        <requestFocus />
    </EditText>
</TableRow>

<TableRow
    android:id="@+id/tableRow3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Ngày XB:" />

    <EditText
        android:id="@+id/editTextDate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="date" />

    <Button
        android:id="@+id/btnDate"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="..." />
```

```
</TableRow>

<TableRow
    android:id="@+id/tableRow4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <Button
        android:id="@+id/btnInsertBook"
        android:layout_width="wrap_content"
        android:layout_height="27dp"
        android:layout_span="3"
        android:text="Thêm Sách" />

</TableRow>

<TableRow
    android:id="@+id/tableRow5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >
</TableRow>

<ListView
    android:id="@+id/listViewBook"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
</ListView>
```

```
</TableLayout>
```

activity_create_author.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

<TableLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
    <TableRow
        android:id="@+id/tableRow1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >
        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Mã tác giả:"
            android:textAppearance="?android:attr/textAppearanceMedium" />
        <EditText
            android:id="@+id/editFirstName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:ems="10" >
            <requestFocus />
        </EditText>
    </TableRow>

    <TableRow
        android:id="@+id/tableRow2"
```



```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >

        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Tên tác giả:"
            android:textAppearance="?android:attr/textAppearanceMedium" />
        <EditText
            android:id="@+id/editLastName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:ems="10" />
    </TableRow>

    <TableRow
        android:id="@+id/tableRow3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >
        <Button
            android:id="@+id/btnClear"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Xóa trắng" />
        <Button
            android:id="@+id/btnSaveAuthor"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Lưu tác giả" />
    </TableRow>
    <TableRow
        android:id="@+id/tableRow4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >
    </TableRow>
</TableLayout>
</LinearLayout>
```

layout_show_data.xml

```
<?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="match_parent"
    android:orientation="vertical" >
    <ListView
        android:id="@+id/listViewShowData"
        android:layout_width="match_parent"
        android:layout_height="405dp" >
    </ListView>
    <Button
        android:id="@+id/buttonBack"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Back" />
</LinearLayout>
```

my_layout_for_show_list_data.xml

```
<?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="match_parent"
    android:orientation="horizontal" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="0.3"
        android:gravity="left"
        android:text="Medium Text"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="left"
        android:layout_weight="0.4"
        android:gravity="left"
        android:text="Medium Text"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="left"
        android:layout_weight="0.3"
        android:text="Medium Text"
        android:gravity="left"
        android:textAppearance="?android:attr/textAppearanceMedium" />

</LinearLayout>
```

InforData.java

```
public class InforData {

    private Object field1;
    private Object field2;
    private Object field3;
    public InforData(Object field1, Object field2, Object field3) {
        this.field1 = field1;
        this.field2 = field2;
        this.field3 = field3;
    }
    public InforData() { }
    public Object getField1() {
        return field1;
    }
    public Object getField2() {
        return field2;
    }
    public Object getField3() {
        return field3;
    }
}
```

```
public void setField1(Object field1) {
    this.field1 = field1;
}
public void setField2(Object field2) {
    this.field2 = field2;
}
public void setField3(Object field3) {
    this.field3 = field3;
}
@Override
public String toString() {
    return this.field1 + " - " + this.field2 + " - " + this.field3;
}
}
```

MyDatabaseHelper.java

```
public class MyDatabaseHelper extends SQLiteOpenHelper {
    private static final String TAG = "MyDatabaseHelper";
    private static final String DATABASE_NAME = "qlsach";
    private static final int DATABASE_VERSION = 1;
    private static final String TABLE_AUTHOR = "tblAuthors";
    private static final String AUTHORS_ID = "id";
    private static final String AUTHORS_FIRT_NAME = "firstName";
    private static final String AUTHORS_LAST_NAME = "lastName";
    private static final String CREATE_AUTHOR_TABLE_SQL =
        "CREATE TABLE IF NOT EXISTS " + TABLE_AUTHOR + " (" +
            AUTHORS_ID + " INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT," +
            AUTHORS_FIRT_NAME + " TEXT NOT NULL," +
            AUTHORS_LAST_NAME + " TEXT NOT NULL" +
        ")";
    // String sqlAuthor="create table tblAuthors (" +
    //     " id integer primary key autoincrement," +
    //     "firstNameetName text, LastName text)";
    //-----
    private static final String TABLE_BOOK = " tblBooks";
    private static final String BOOK_ID = " id ";
    private static final String BOOK_TITLE = " title ";
    private static final String BOOK_DATE_ADD = "dateAdd ";
    private static final String BOOK_AUTHOR_ID_REF = "authorId";
    //-----
    String CREATE_BOOK_TABLE_SQL="create table IF NOT EXISTS "+TABLE_BOOK+" (" +
        BOOK_ID + " integer primary key autoincrement," +
        BOOK_TITLE+ " text, " +
        BOOK_DATE_ADD + " date, " +
        BOOK_AUTHOR_ID_REF+" integer not null " +")";
    //các em viết nốt giúp cố ràng buộc
    // String sqlBook="create table tblBooks (" +
    //     " id integer primary key autoincrement," +
    //     " title text, dateAdd date , " +
    //     " authorId integer not null constraint authorId " +
    //     " references tblAuthors(id) on delete cascade)";
    private static MyDatabaseHelper sInstance;

    public static MyDatabaseHelper getInstance(Context context) {
        if (sInstance == null) {
            sInstance = new MyDatabaseHelper(context.getApplicationContext());
        }
        return sInstance;
    }
}
```

```
private MyDatabaseHelper(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
}
@Override
public void onCreate(SQLiteDatabase db) {
    Log.e(TAG, "onCreate: ");
    try{
//        db.execSQL("DROP TABLE IF EXISTS " + TABLE_WORD);
        db.setLocale(Locale.getDefault());
        db.setVersion(1);
        db.execSQL(CREATE_AUTHOR_TABLE_SQL);
        db.execSQL(CREATE_BOOK_TABLE_SQL);
    }
    catch (Exception e){
        Log.e(TAG,e.toString());
    }
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    Log.e(TAG, "onUpgrade: ");
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_BOOK);
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_AUTHOR);
    onCreate(db);
}

/**
 * hàm kiểm tra xem bảng có tồn tại trong CSDL hay chưa
 * @param database - cơ sở dữ liệu
 * @param tableName - tên bảng cần kiểm tra
 * @return trả về true nếu tồn tại
 */
public boolean isTableExists(SQLiteDatabase database,String tableName)
{
    Cursor cursor = database.rawQuery(
        "select DISTINCT tbl_name " +
        "from sqlite_master " +
        "where tbl_name = '"+tableName+"'", null);
    if(cursor!=null) {
        if(cursor.getCount()>0) {
            cursor.close();
            return true;
        }
        cursor.close();
    }
    return false;
}

/**
 * Long insert(String table, String nullColumnHack, ContentValues values)
 * chèn một bản ghi trên các cơ sở dữ liệu. Bảng chỉ định tên bảng,
 * nullColumnHack không cho phép các giá trị hoàn toàn vô giá trị.
 * Nếu số thứ hai là null, android sẽ lưu trữ các giá trị null nếu giá trị
 * là trống rỗng. Đối số thứ ba xác định các giá trị được lưu trữ.
 */
public long insertInfor(String table_name, ContentValues values) {
    SQLiteDatabase db = getWritableDatabase();
    long rowId = db.insert(table_name, null, values);
    db.close();
    return rowId;
}

/**
 * Cursor query(String table, String[] columns, String selection, String[]
```

```
selectionArgs,
    * String groupBy, String having, String orderBy)
    */
    public List<InforData> getInforByID(String tableName,String fieldName,String id)
{
    List<InforData> lists = new ArrayList<InforData>();
    SQLiteDatabase db = getReadableDatabase();
    Cursor cursor = db.query(tableName, null, fieldName+"=?",
new String[]{id}, null, null, null);
    if (cursor != null && cursor.moveToFirst()) {
        do {
            lists.add(new InforData(cursor.getInt(0),
                cursor.getString(1),
                cursor.getString(2)));
        } while (cursor.moveToNext());
        cursor.close();
    }
    return lists;
}
/**
 * thực hiện 1 câu truy vấn trực tiếp
 *
 */
    public List<InforData> getAllInfor(String TABLE_NAME) {
        SQLiteDatabase db = getReadableDatabase();
        List<InforData> lists = new ArrayList<InforData>();
        String sql = "SELECT * FROM " + TABLE_NAME;
        Cursor cursor = db.rawQuery(sql, null);
        if (cursor != null && cursor.moveToFirst()) {
            do {
                lists.add(new InforData(cursor.getInt(0),
                    cursor.getString(1),
                    cursor.getString(2)));
            } while (cursor.moveToNext());
            cursor.close();
        }
        db.close();
        return lists;
    }
    /**
    * int update(String table, ContentValues values, String whereClause, String[]
whereArgs)
    */
    public int updateInforByID(String tableName, ContentValues values, String field,
        String[] fieldValue) {
        SQLiteDatabase db = getWritableDatabase();
        int rowEffect = db.update(tableName, values, field,fieldValue);
        db.close();
        return rowEffect;
    }

    public int deleteAllTable(String tableName) {
        SQLiteDatabase db = getReadableDatabase();
        int rowEffect = db.delete(tableName, null,null);
        db.close();
        return rowEffect;
    }
    public int deleteInforByID(String tableName, String field ,String[] fieldValue) {
        SQLiteDatabase db = getReadableDatabase();
        int rowEffect = db.delete(tableName, field,fieldValue);
        db.close();
        return rowEffect;
    }
}
```

```
}  
}
```

MainActivity

```
public class MainActivity extends AppCompatActivity {  
    //sinh viên tự làm.... Có hướng dẫn trong giờ học trên lớp.  
}
```

ShowListAuthorActivity

```
public class ShowListAuthorActivity extends AppCompatActivity {  
    List<InforData> list=new ArrayList<InforData>();  
    InforData dataClick=null;  
    // SQLiteDatabase database=null;  
    MyDatabaseHelper database=null;  
    MySimpleArrayAdapter adapter=null;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        // TODO Auto-generated method stub  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.layout_show_data);  
        updateUI();  
        Button btn=(Button)findViewById(R.id.buttonBack);  
        btn.setOnClickListener(new OnClickListener() {  
  
            @Override  
            public void onClick(View v) {  
                // TODO Auto-generated method stub  
                ShowListAuthorActivity.this.finish();  
            }  
        });  
    }  
    public void updateUI()  
    {  
        List<InforData> listTmp=new ArrayList<InforData>();  
        database= MyDatabaseHelper.getInstance(this);  
        if (database!=null)  
        {  
            InforData header=new InforData();  
            header.setField1("STT");  
            header.setField2("Mã tác giả.");  
            header.setField3("Tên tác giả.");  
            list.add(header);  
            listTmp=database.getAllInfor(????????);  
            list.addAll(listTmp);  
            adapter =new MySimpleArrayAdapter>ShowListAuthorActivity.this,  
                R.layout.my_layout_for_show_list_data,  
                list);  
            final ListView lv=(ListView) findViewById(R.id.ListViewShowData);  
            lv.setAdapter(adapter);  
            lv.setOnItemClickListener(new OnItemClickListener() {  
  
                @Override  
                public void onItemClick(AdapterView<?> arg0, View arg1,  
                    int arg2, long arg3) {  
                    // TODO Auto-generated method stub  
                    Toast.makeText>ShowListAuthorActivity.this,"View -->" +  
list.get(arg2).toString(), Toast.LENGTH_LONG).show();  
  
                    Intent intent=new Intent>ShowListAuthorActivity.this,  
CreateAuthorActivity.class);
```

```
Bundle bundle=new Bundle();
bundle.putInt("KEY", 1);
bundle.putString("getField1", list.get(arg2).getField1().toString());
bundle.putString("getField2", list.get(arg2).getField2().toString());
bundle.putString("getField3", list.get(arg2).getField3().toString());
intent.putExtra("DATA", bundle);
dataClick=list.get(arg2);
startActivityResult(intent, MainActivity.OPEN_AUTHOR_DIALOG);
}
});
lv.setOnItemClickListener(new OnItemClickListener() {

    @Override
    public boolean onItemClick(AdapterView<?> arg0, View arg1,
        int arg2, long arg3) {
        // TODO Auto-generated method stub
        final InforData data=list.get(arg2);
        final int pos=arg2;
        Toast.makeText>ShowListAuthorActivity.this, "Edit-->" +data.toString(),
        Toast.LENGTH_LONG).show();
        Builder b=new Builder>ShowListAuthorActivity.this);
        b.setTitle("Remove");
        b.setMessage("Xóa [" +data.getField2() + " - " +data.getField3() +"] ?");
        b.setPositiveButton("Có", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                // Gọi hàm xóa thông tin theo ID. Sinh viên tự làm

                int n=database.deleteInforByID(????????);
                if(n>0)
                {
                    Toast.makeText>ShowListAuthorActivity.this, "Remove ok",
                    Toast.LENGTH_LONG).show();
                    list.remove(pos);
                    adapter.notifyDataSetChanged();
                }
                else
                {
                    Toast.makeText>ShowListAuthorActivity.this, "Remove not ok",
                    Toast.LENGTH_LONG).show();
                }
            }
        });
        b.setNegativeButton("Không", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which)
            {
                // TODO Auto-generated method stub
                dialog.cancel();
            }
        });
        b.show();
        return false;
    }

});
}
```

@Override

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    // TODO Auto-generated method stub
    super.onActivityResult(requestCode, resultCode, data);
    //database =openOrCreateDatabase("qlsach.db",
    SQLiteDatabase.CREATE_IF_NECESSARY, null);
    if(resultCode==MainActivity.SEND_DATA_FROM_AUTHOR_ACTIVITY)
    {
        Bundle bundle=data.getBundleExtra("DATA_AUTHOR");
        String f2=bundle.getString("firstName");
        String f3=bundle.getString("lastName");
        String f1=dataClick.getField1().toString();
        ContentValues values=new ContentValues();
        values.put("firstName", f2);
        values.put("lastName", f3);
        if(database!=null) {
            //Cập nhật thông tin infor theo mã id, sv tự làm
            int n=database.updateInforByID(????????);
            if (n>0) {
                Toast.makeText>ShowListAuthorActivity.this, "update ok ",
                Toast.LENGTH_LONG).show();
                dataClick.setField2(f2);
                dataClick.setField3(f3);
                if (adapter != null) adapter.notifyDataSetChanged();
            }else
                Toast.makeText>ShowListAuthorActivity.this, "update not ok ",
                Toast.LENGTH_LONG).show();
        }
    }
}
```

MySimpleArrayAdapter

```
/**
 * class dùng để custom layout
 * dùng chung cho hiển thị Sách và Tác gia
 */
public class MySimpleArrayAdapter extends ArrayAdapter<InforData> {

    private Activity context;
    private int layout;
    private List<InforData>list;
    public MySimpleArrayAdapter(Context context, int textViewResourceId,
        List<InforData> objects) {
        super(context, textViewResourceId, objects);
        this.context=(Activity) context;
        this.layout=textViewResourceId;
        this.list=objects;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {

        LayoutInflater flater=context.getLayoutInflater();
        View row=flater.inflate(layout, parent, false);
        TextView txt1=(TextView) row.findViewById(R.id.textView1);
        TextView txt2=(TextView) row.findViewById(R.id.textView2);
        TextView txt3=(TextView) row.findViewById(R.id.textView3);
        try {
            InforData data = list.get(position);
```



```
txt1.setText(data.getField1() == null ? "-" : data.getField1().toString());
txt2.setText(data.getField2() == null ? "-" : data.getField2().toString());
txt3.setText(data.getField3() == null ? "-" : data.getField3().toString());
}catch (Exception e){
    Log.e("custom MySimpleArrayadt",e.toString());
}
if(position==0)
{
    row.setBackgroundColor(Color.RED);
}

return row;
}
}
```

ShowListAuthorActivity

```
public class ShowListAuthorActivity extends AppCompatActivity {
    List<InforData> list=new ArrayList<InforData>();
    InforData dataClick=null;
    // SQLiteDatabase database=null;
    MyDatabaseHelper database=null;
    MySimpleArrayAdapter adapter=null;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout_show_data);
        updateUI();
        Button btn=(Button)findViewById(R.id.buttonBack);
        btn.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                ShowListAuthorActivity.this.finish();
            }
        });
    }
    public void updateUI()
    {
        List<InforData> listTmp=new ArrayList<InforData>();
        database= MyDatabaseHelper.getInstance(this);
        if (database!=null)
        {
            InforData header=new InforData();
            header.setField1("STT");
            header.setField2("Mã tác giả.");
            header.setField3("Tên tác giả.");
            list.add(header);
            listTmp=database.getAllInfor("tblAuthors");
            list.addAll(listTmp);
            adapter =new
MySimpleArrayAdapter(ShowListAuthorActivity.this,R.layout.my_layout_for_show_list_data,list);
            final ListView lv=(ListView) findViewById(R.id.ListViewShowData);
            lv.setAdapter(adapter);
            lv.setOnItemClickListener(new OnItemClickListener() {

                @Override
                public void onItemClick(AdapterView<?> arg0, View arg1,
                    int arg2, long arg3) {
```

```
// TODO Auto-generated method stub
Toast.makeText>ShowListAuthorActivity.this, "View -->" +
list.get(arg2).toString(), Toast.LENGTH_LONG).show();

Intent intent=new Intent>ShowListAuthorActivity.this,
CreateAuthorActivity.class;
Bundle bundle=new Bundle();
bundle.putInt("KEY", 1);
bundle.putString("getField1", list.get(arg2).getField1().toString());
bundle.putString("getField2", list.get(arg2).getField2().toString());
bundle.putString("getField3", list.get(arg2).getField3().toString());
intent.putExtra("DATA", bundle);
dataClick=list.get(arg2);
startActivityForResult(intent, MainActivity.OPEN_AUTHOR_DIALOG);
}
});
lv.setOnItemClickListener(new OnItemClickListener() {

@Override
public boolean onItemClick(AdapterView<?> arg0, View arg1,
int arg2, long arg3) {
// TODO Auto-generated method stub
final InforData data=list.get(arg2);
final int pos=arg2;
Toast.makeText>ShowListAuthorActivity.this, "Edit-->" +data.toString(),
Toast.LENGTH_LONG).show();
Builder b=new Builder>ShowListAuthorActivity.this;
b.setTitle("Remove");
b.setMessage("Xóa [" +data.getField2() + " - " +data.getField3() +"] ?");
b.setPositiveButton("Có", new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int which) {
// TODO Auto-generated method stub
int n=database.deleteInforByID(.....);
if(n>0)
{
Toast.makeText>ShowListAuthorActivity.this, "Remove ok",
Toast.LENGTH_LONG).show();
list.remove(pos);
adapter.notifyDataSetChanged();
}
else
{
Toast.makeText>ShowListAuthorActivity.this, "Remove not ok",
Toast.LENGTH_LONG).show();
}
}
});
b.setNegativeButton("Không", new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int which)
{
// TODO Auto-generated method stub
dialog.cancel();
}
});
b.show();
return false;
}

});
}
```

```
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    // TODO Auto-generated method stub
    super.onActivityResult(requestCode, resultCode, data);
    if(resultCode==MainActivity.SEND_DATA_FROM_AUTHOR_ACTIVITY)
    {
        Bundle bundle=data.getBundleExtra("DATA_AUTHOR");
        String f2=bundle.getString("firstName");
        String f3=bundle.getString("lastName");
        String f1=dataClick.getField1().toString();
        ContentValues values=new ContentValues();
        values.put("firstName", f2);
        values.put("lastName", f3);
        if(database!=null) {
            int n=database.updateInforByID(????????);
            if (n>0) {
                Toast.makeText(ShowListAuthorActivity.this, "update ok ",
                Toast.LENGTH_LONG).show();
                dataClick.setField2(f2);
                dataClick.setField3(f3);
                if (adapter != null) adapter.notifyDataSetChanged();
            }else
                Toast.makeText(ShowListAuthorActivity.this, "update not ok ",
                Toast.LENGTH_LONG).show();
        }
    }
}
```

Bài tập 10.3. Xây dựng ứng dụng quản lý bán tạp hóa

Xây dựng ứng dụng quản lý bán café gồm các chức năng sau :

- Với nhân viên bán hàng
 - o Đăng nhập
 - o Tạo các hóa đơn bán hàng (café, nước uống, bánh) dựa trên khách mua
- Với quản lý của hàng:
 - o Đăng nhập
 - o Thống kê doanh thu bán hàng
 - o Kiểm tra danh mục hàng tồn (ví dụ : tổng có 100 gói bim bim đã bán 55 còn 45 gói)
- Yêu cầu: Thiết kế DB trên SQLite hợp lý, xây dựng giao diện phù hợp với các tính năng.
- Có các thông báo khi thực hiện thành công hay có lỗi.

Bài tập 10.4. Xây dựng ứng dụng số học bạ online

Tại trường THCS Minh Khai, ban giám hiệu nhà trường muốn tạo ra một ứng dụng CNTT nhằm thực hiện mục tiêu chuyển đổi số cho phép :

- Ban giám hiệu có chức năng
 - Login
 - Duyệt kết quả
 - Thống kê tình trạng, tỷ lệ học sinh theo học lực
- Giáo viên
 - Login
 - Nhập điểm học sinh
 - Nhập nhận xét cho học sinh
- Phụ huynh / học sinh
 - Xem thời khóa biểu/ lịch học
 - Xem bảng điểm

Yêu cầu : Thực hiện trên CSDL SQLite và lập trình để thực hiện ứng dụng trên

Nâng cao: Trong thực tế , hệ thống cần xây dựng server để cung cấp API để ứng dụng lấy data (json) hiển thị về phía client. (Khuyến khích các sinh viên làm phát triển thành bài tập lớn, đồ án môn học).

Bài tập 10.5. Xây dựng ứng dụng chat online thời gian thực

Trong thực tế , ngoài cơ sở dữ liệu SQLite dùng cho android được tích hợp cùng trong ứng dụng thì còn nhiều cơ sở dữ liệu khác có thể được dùng tùy thuộc vào tính chất dự án.

Một trong các CSDL thông dụng dùng để có thể thực hiện thời gian thực (realtime) đó chính là firebase.

Trong bài tập này yêu cầu các em sử dụng android kết hợp firebase cho phép người sử dụng thực hiện chat online.

Để làm bài tập này các em tham khảo nội dung chương 5 (trang từ 136- 144) giáo trình [1].

[1] . Nguyễn Bá Nghiễn, Nguyễn Thái Cường, Vũ Thị Dương, Nguyễn Văn Tĩnh, *Phát triển ứng dụng trên thiết bị di động*, 2021