

## MODULE 3.1: DATA STORAGE

Nội dung kiến thức :

- Thao tác quản lý tập tin trên thiết bị bằng công cụ
- Lưu trạng thái ứng dụng với Shared Preferences
- Thiết lập cấu hình cho ứng dụng với Shared Preferences API
- Lập trình truy xuất tập tin được lưu ở các dạng:
  - o Static raw files
  - o Internal files
- External files (On SD Card)
- Truy xuất dữ liệu trên cơ sở dữ liệu quan hệ với SQLite
- Thao tác với Menu

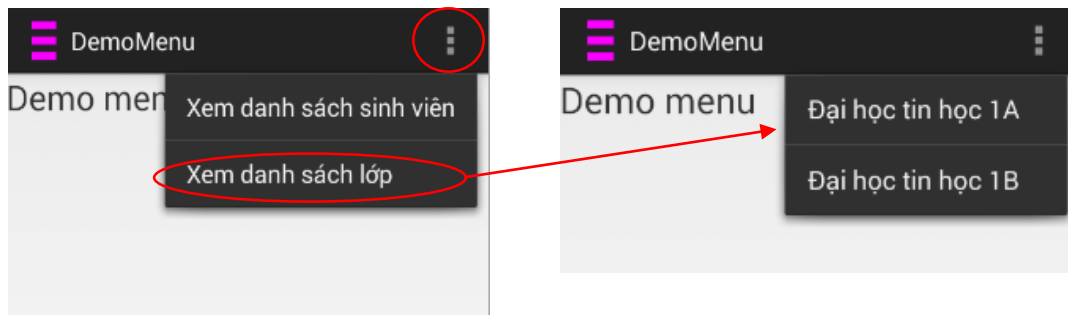
Bài tập 1 :

### Mục đích :

\_ Thực hành về Menu .

### Yêu cầu :

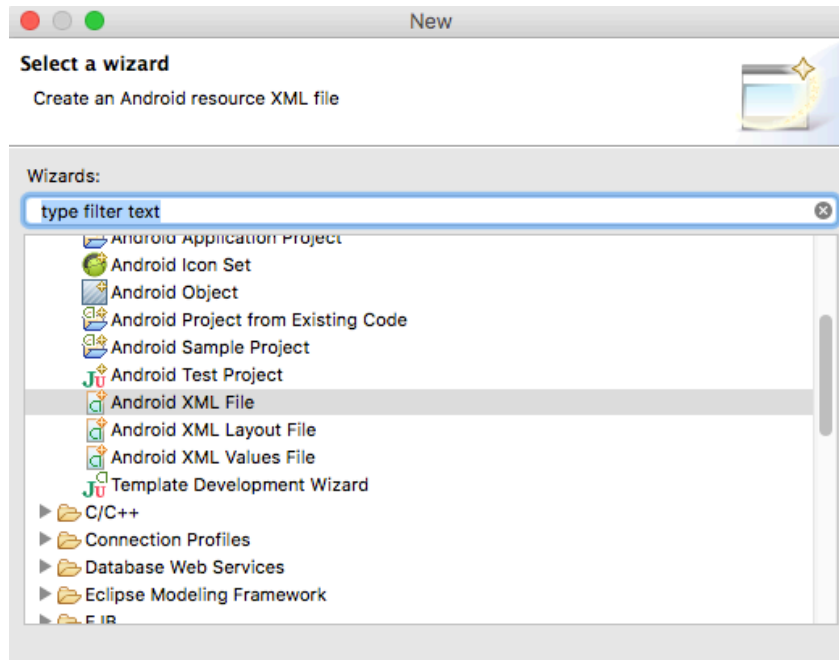
\_ Viết chương trình khi click vào Menu sẽ hiện thông tin như sau :



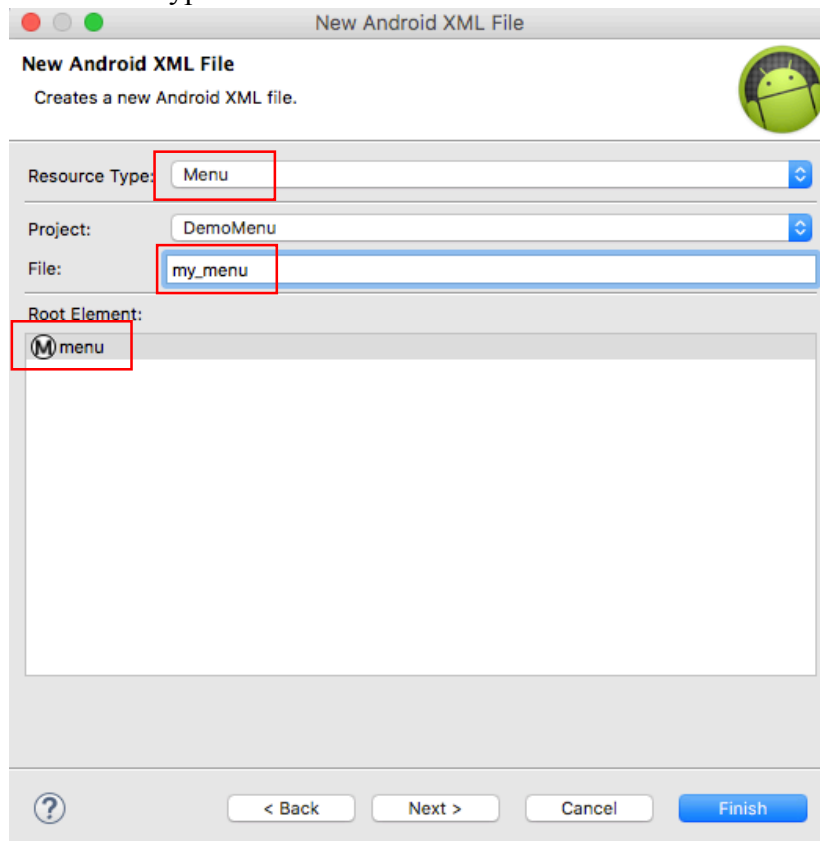
\_ Khi người dùng chọn xem danh sách sinh viên sẽ mở Activity của danh sách sinh viên .  
Khi chọn lớp thì sẽ mở activity của lớp tương ứng .

### Hướng dẫn :

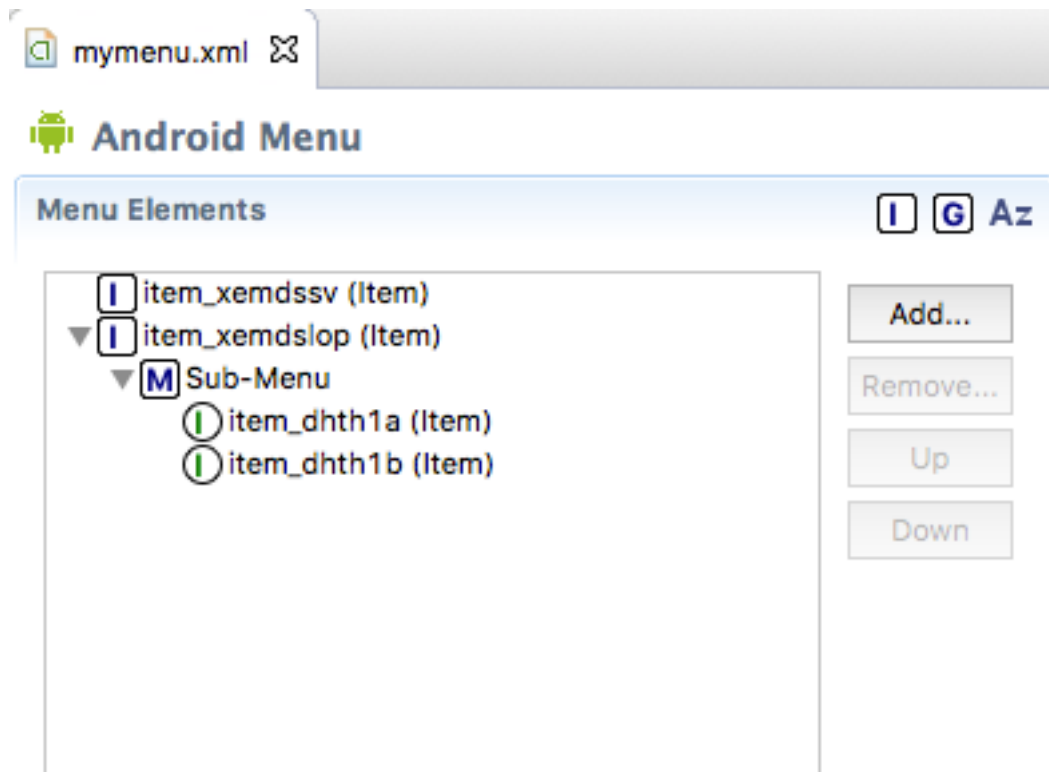
\_ Các tạo Menu, chọn File->New -> Other, chọn Android XML File



\_ Sau đó chọn Resource Type là Menu



\_ Để tạo menu item, ta mở file my\_menu.xml trong thư mục res->menu .



\_ Khi tạo Menu Item thì có 2 thuộc tính chính là Id và Title .

\_ Muốn tạo Sub-Menu, ta chọn vào menu muốn tạo và add 1 Sub-Menu, sau đó add Menu Item như bình thường .

\_ Để chạy mymenu, ta mở file MainActivity.java và thêm code :

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    //getMenuInflater().inflate(R.menu.main, menu);
    getMenuInflater().inflate(R.menu.mymenu, menu);
    return true;
}
```

\_ Để viết sự kiện cho Menu Item ta cần Override phương thức onOptionsItemSelected, trong phương thức này ta dựa vào Id của các Menu Item để xử lý :

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    switch (id) {
        case R.id.item_xemdssv:
            //Xử lý xem danh sách sinh viên
            break;
        case R.id.item_dhth1a:
            //Xử lý xem thông tin lớp 1A
            break;
        case R.id.item_dhth1b:
            break;
        default:
            break;
    }
    return super.onOptionsItemSelected(item);
}
```

\_ Để viết Menu Runtime, ta viết như sau :

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Đối số 1 là nhóm
    // Đối số 2 là Id cho Menu Item
    // Đối số 3 là thứ tự xuất hiện của Menu Item
    // Đối số 4 là tiêu đề cho Menu Item
    int itemId = 113;
    menu.add(0, itemId, 0, "Menu 1");
    itemId = 114;
    menu.add(0, itemId, 1, "Menu 2");
    SubMenu sub3 = menu.addSubMenu("Menu 3");
    itemId = 115;
    sub3.add(0, itemId, 0, "File 1 Menu 3");
    itemId = 116;
    sub3.add(0, itemId, 1, "File 2 Menu 3");
    itemId = 117;
    sub3.add(0, itemId, 2, "File 3 Menu 3");
    return true;
}
```

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    switch (id) {
        case 113:
            // Xử lý Menu 1
            break;
        case 114:
            // Xử lý Menu 2
            break;
        case 115:
            // Xử lý File 1 của Menu 3
            break;
        case 116:
            // Xử lý File 2 của Menu 3
            break;
        case 117:
            // Xử lý File 3 của Menu 3
            break;
    }
    return super.onOptionsItemSelected(item);
}
```

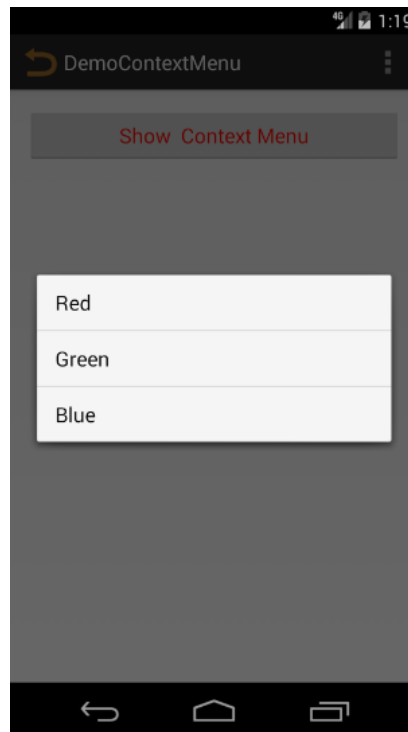
Bài tập 2 :

**Mục đích :**

\_ Thực hành về Context Menu.

**Yêu cầu :**

\_ Viết chương trình như sau :



\_ Khi nhấn vào nút Show Context Menu, thì context menu sẽ xuất hiện cho phép đổi màu chữ : đỏ, xanh lá và xanh dương .

### Hướng dẫn :

\_ Tạo Menu layout như sau :

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item android:id="@+id/itemRed" android:title="Red" />
    <item android:id="@+id/itemGreen" android:title="Green" />
    <item android:id="@+id/itemBlue" android:title="Blue" />
</menu>
```

\_ Trong tập tin strings.xml, thêm các định nghĩa màu :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">DemoContextMenu</string>
    <string name="hello_world">Hello world!</string>
    <string name="action_settings">Settings</string>

    <color name="clrred">#F00</color>
    <color name="clrgreen">#0F0</color>
    <color name="clrbblue">#00F</color>

</resources>
```

\_ Trong lớp MainActivity.java, ta phải đăng ký ContextMenu với đối tượng muốn dùng bằng phương thức registerForContextMenu .

```
private Button btnShowContext;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    btnShowContext = (Button)findViewById(R.id.btnshowcontext);
    registerForContextMenu(btnShowContext);
}
```

\_ Đồng thời ta phải override thêm 2 phương thức để gọi ContextMenu và sự kiện của nó

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo menuInfo) {
    // TODO Auto-generated method stub
    super.onCreateContextMenu(menu, v, menuInfo);
    getMenuInflater().inflate(R.menu.my_context_menu, menu);
}

@Override
public boolean onContextItemSelected(MenuItem item) {
    // TODO Auto-generated method stub
    int id = item.getItemId();
    switch (id) {
        case R.id.itemRed:
            btnShowContext.setTextColor(
                getResources().getColor(R.color.clrred));
            break;
        case R.id.itemGreen:
            btnShowContext.setTextColor(
                getResources().getColor(R.color.clrgreen));
            break;
        case R.id.itemBlue:
            btnShowContext.setTextColor(
                getResources().getColor(R.color.clrblue));
            break;
        default:
            btnShowContext.setTextColor(
                getResources().getColor(R.color.clrred));
            break;
    }
    return super.onContextItemSelected(item);
}
```

**Bài tập 3 :****Mục đích :**

\_ Thực hành về SQLite, custom layout, intent, menu ... .

**Yêu cầu :**

\_ Viết chương trình quản lý sách như sau :



\_ Khi chọn chức năng thêm tác giả (chú ý mở dưới dạng Dialog) :



\_ Chọn Xóa trắng để xóa dữ liệu và focus tới Mã tác giả, chọn Lưu tác giả để lưu vào trong CSDL .

\_ Khi chọn Xem danh sách Tác giả, chương trình sẽ mở một màn hình mới để hiển thị danh sách như sau :



STT	Mã tác giả	Tên tác giả
1	tg01	Nguyen Nhat Anh
2	tg02	Hoang Trung Thong
3	tg03	Han Mac Tu
4	tg04	Ho Xuan Huong

\_ Khi chọn một tác giả trong ListView, sẽ xuất hiện giao diện :

STT	Mã tác giả	Tên tác giả
1	tg01	Nguyen Nhat Anh
2	tg02	Hoang Trung Thong
3	tg03	Han Mac Tu
4	tg04	Ho Xuan Huong

**View Detail**

Mã tác giả:

Tên tác giả:

Xóa trắng      Update

\_ Chọn Update để cập nhật lại tác giả .

\_ Khi nhấn thật lâu vào từng tác giả trong danh sách, chương trình sẽ hiện Alert Dialog hỏi muốn xóa Tác giả này hay không ? Chọn có để xóa, chọn không để tắt thông báo .

STT	Mã tác giả	Tên tác giả
1	tg01	Nguyen Nhat Anh
2	tg02	Hoang Trung Thong
3	tg03	Han Mac Tu
4	tg04	Ho Xuan Huong

**Remove**

Xóa [tg04 - Ho Xuan Huong] hả?

Không      Có

\_ Khi chọn chức năng Quản lý sách, màn hình chương trình sẽ như sau :

Id	Tên sách	Ngày XB
1	Mot Thoi De Nho	22-02-2007
2	Thang Quy Nho	23-01-2008
3	Ba Lo Mau Xanh	21-04-2005

\_ Load danh sách tác giả vào trong Spinner .

\_ Sử dụng DatePickerDialog để chọn ngày xuất bản .

\_ Chọn thêm sách lưu vào CSDL đúng với tác giả chọn trong Spinner, đồng thời cập nhật lại ListView bên dưới .

### Hướng dẫn :

\_ Giao diện chính :

```
<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" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        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="wrap_content"
        android:text="Thêm Tác Giả" />
```

```

<Button
    android:id="@+id/buttonShowAuthorList"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Xem danh sách Tác Giả" />

<Button
    android:id="@+id/buttonInsertBook"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="14dp"
    android:text="Quản lý Sách" />

</LinearLayout>

```

\_ Nội dung hàm MainActivity :

```

public class MainActivity extends Activity {

    Button btnCreateDatabase = null;
    Button btnInsertAuthor = null;
    Button btnShowAuthorList = null;
    Button btnShowAuthorList2 = null;
    Button btnTransaction = null;
    Button btnShowDetail = null;
    Button btnInsertBook = null;
    public static final int OPEN_AUTHOR_DIALOG = 1;
    public static final int SEND_DATA_FROM_AUTHOR_ACTIVITY = 2;
    SQLiteDatabase database = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnInsertAuthor = (Button)
findViewById(R.id.btnInsertAuthor);
        btnInsertAuthor.setOnClickListener(new MyEvent());
        btnShowAuthorList = (Button)
findViewById(R.id.buttonShowAuthorList);
        btnShowAuthorList.setOnClickListener(new MyEvent());
        btnInsertBook = (Button)
findViewById(R.id.buttonInsertBook);
        btnInsertBook.setOnClickListener(new MyEvent());
        getDatabase();
    }
}

```

```

/**
 * 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;
}

/**
 * hàm tạo CSDL và các bảng liên quan
 *
 * @return
 */
public SQLiteDatabase getDatabase() {
    try {
        database = openOrCreateDatabase("mydata.db",
SQLiteDatabase.CREATE_IF_NECESSARY, null);
        if (database != null) {
            if (isTableExists(database, "tblAuthors"))
                return database;
            database.setLocale(Locale.getDefault());
            database.setVersion(1);
            String sqlAuthor = "create table tblAuthors (" +
                "id integer primary key autoincrement,"
                + "firstname text, " + "lastname
text)";
            database.execSQL(sqlAuthor);
            String sqlBook = "create table tblBooks (" + "id

```

```

integer primary key autoincrement," + "title text, "
                                + "dateadded date,"
                                + "authorid integer not null
constraint authorid references tblAuthors(id) on delete cascade)";
database.execSQL(sqlBook);

Toast.makeText(MainActivity.this, "OK OK",
Toast.LENGTH_LONG).show();
    }
    } catch (Exception e) {
        Toast.makeText(this, e.toString(),
Toast.LENGTH_LONG).show();
    }
    return database;
}

public void createDatabaseAndTrigger() {
    if (database == null) {
        getDatabase();
        Toast.makeText(MainActivity.this, "OK OK",
Toast.LENGTH_LONG).show();
    }
}

/**
 * hàm mở màn hình nhập Tác giả
 */
public void showInsertAuthorDialog() {
    Intent intent = new Intent(MainActivity.this,
CreateAuthorActivity.class);
    startActivityForResult(intent, OPEN_AUTHOR_DIALOG);
}

/**
 * hàm xem danh sách tác giả dùng Activity Tôi làm 2 cách để các  

bạn ôn tập
 * lại ListView bạn gọi hàm nào thì gọi 1 thôi showAuthorList1  

hoặc
 * showAuthorList2
 */
public void showAuthorList1() {
    Intent intent = new Intent(MainActivity.this,
ShowListAuthorActivity.class);
    startActivity(intent);
}

```

```

    }

    /**
     * hàm xem danh sách tác giả dùng ListActivity Tôi làm 2 cách để
các bạn ôn
     * tập lại ListView bạn gọi hàm nào thì gọi 1 thôi
    showAuthorList1 hoặc
     * showAuthorList2
     */
    public void showAuthorList2() {
        Intent intent = new Intent(MainActivity.this,
        ShowListAuthorActivity2.class);
        startActivity(intent);
    }

    /**
     * Tôi cung cấp thêm hàm này để các bạn nghiên cứu thêm về
    transaction
     */
    public void interactDBWithTransaction() {
        if (database != null) {
            database.beginTransaction();
            try {
                // làm cái gì đó tùm lum ở đây,
                // chỉ cần có lỗi xảy ra thì sẽ kết thúc
                transaction

                ContentValues values = new ContentValues();
                values.put("firstname", "xx");
                values.put("lastname", "yyy");
                database.insert("tblAuthors", null, values);
                database.delete("tblAuthors", "ma=?", new
                String[] { "x" });

                // Khi nào hàm này được gọi thì các thao tác bên
trên mới thực

                // hiện được
                // Nếu nó không được gọi thì mọi thao tác bên
trên đều bị hủy

                database.setTransactionSuccessful();
            } catch (Exception ex) {
                Toast.makeText(MainActivity.this,
                ex.getMessage(), Toast.LENGTH_LONG).show();
            } finally {
                database.endTransaction();
            }
        }
    }

```

```

    }
}

/**
 * hàm xử lý kết quả trả về
 */
@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    // TODO Auto-generated method stub
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == SEND_DATA_FROM_AUTHOR_ACTIVITY) {
        Bundle bundle = data.getBundleExtra("DATA_AUTHOR");
        String firstname = bundle.getString("firstname");
        String lastname = bundle.getString("lastname");
        ContentValues content = new ContentValues();
        content.put("firstname", firstname);
        content.put("lastname", lastname);
        if (database != null) {
            long authorid = database.insert("tblAuthors",
null, content);

            if (authorid == -1) {
                Toast.makeText(MainActivity.this,
                    authorid + " - " + firstname + "
- " + lastname + " ==> insert error!", Toast.LENGTH_LONG)
                    .show();
            } else {
                Toast.makeText(MainActivity.this,
                    authorid + " - " + firstname + "
- " + lastname + " ==>insert OK!", Toast.LENGTH_LONG)
                    .show();
            }
        }
    }
}

/**
 * class xử lý sự kiện
 *
 * @author drthanh
 */
private class MyEvent implements OnClickListener {

```

```

@Override
public void onClick(View v) {
    // TODO Auto-generated method stub
    if (v.getId() == R.id.btnInsertAuthor) {
        showInsertAuthorDialog();
    } else if (v.getId() == R.id.buttonShowAuthorList) {
        showAuthorList1();
    }

    else if (v.getId() == R.id.buttonInsertBook) {
        Intent intent = new Intent(MainActivity.this,
InsertBookActivity.class);
        startActivity(intent);
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {

    getMenuInflater().inflate(R.menu.activity_simple_database_main,
menu);

    return true;
}
}

```

\_ Giao diện thêm tác giả 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/editTextFirstName"
            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/editTextLastName"
            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/buttonClear"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Xóa trắng" />

    <Button
        android:id="@+id/buttonInsert"
        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>

```

\_ Code xử lý thêm tác giả :

```

public class CreateAuthorActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_create_author);
        final Button btnInsert =(Button)
        findViewById(R.id.buttonInsert);
        final EditText txtFirstname=(EditText)
        findViewById(R.id.editTextFirstName);
        final EditText txtLastname=(EditText)
        findViewById(R.id.editTextLastName);
    }
}

```

```
        final Intent intent= getIntent();
        btnInsert.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                Intent intent=new Intent();
                Bundle bundle=new Bundle();
                bundle.putString("firstname",
txtFirstname.getText().toString());
                bundle.putString("lastname",
txtLastname.getText().toString());
                intent.putExtra("DATA_AUTHOR", bundle);

                setResult(MainActivity.SEND_DATA_FROM_AUTHOR_ACTIVITY, intent);
                CreateAuthorActivity.this.finish();
            }
        });
        final Button btnClear=(Button) findViewById(R.id.buttonClear);
        btnClear.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                txtFirstname.setText("");
                txtLastname.setText("");
                txtFirstname.requestFocus();
            }
        });

        Bundle bundle= intent.getBundleExtra("DATA");
        if(bundle!=null && bundle.getInt("KEY")==1)
        {
            String f2=bundle.getString("getField2");
            String f3=bundle.getString("getField3");
            txtFirstname.setText(f2);
            txtLastname.setText(f3);
            btnInsert.setText("Update");
            this.setTitle("View Detail");
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
```

```
        getMenuInflater().inflate(R.menu.activity_create_author,  
menu);  
        return true;  
    }  
}
```

\_ Giao diện thêm sách :

```
<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/buttonDate"
            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/buttonInsertBook"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            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>

```

\_ Code xử lý giao diện thêm sách :

```

public class InsertBookActivity extends Activity {

    SQLiteDatabase database=null;
    List<InforData>listBook=null;
    List<InforData>listAuthor=null;
    InforData authorData=null;
    MySimpleArrayAdapter adapter=null;
    int day,month,year;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_insert_book);
        Spinner pinner=(Spinner) findViewById(R.id.spinner1);
        listAuthor=new ArrayList<InforData>();
        InforData d1=new InforData();
        d1.setField1("_");
    }
}

```

```

        d1.setField2("Show All");
        d1.setField3("_");
        listAuthor.add(d1);
        //Lệnh xử lý đưa dữ liệu là Tác giả và Spinner
        database=openOrCreateDatabase("mydata.db",
SQLiteDatabase.CREATE_IF_NECESSARY, null);
        if(database!=null)
        {
            Cursor cursor=database.query("tblAuthors", null, null,
null, null, null, null);
            cursor.moveToFirst();
            while(cursor.isAfterLast()==false)
            {
                InforData d=new InforData();
                d.setField1(cursor.getInt(0));
                d.setField2(cursor.getString(1));
                d.setField3(cursor.getString(2));
                listAuthor.add(d);
                cursor.moveToNext();
            }
            cursor.close();
        }
        adapter=new MySimpleArrayAdapter(InsertBookActivity.this,
R.layout.my_layout_for_show_list_data,listAuthor);

        adapter.setDropDownViewResource(android.R.layout.simple_spinner_d
ropdown_item);
        pinner.setAdapter(adapter);
        //Xử lý sự kiện chọn trong Spinner
        //chọn tác giả nào thì hiển thị toàn bộ sách của tác giả đó
mà thôi
        //Nếu chọn All thì hiển thị toàn bộ không phân hiệt tác giả
        pinner.setOnItemClickListener(new
OnItemSelectedListener() {
            @Override
            public void onItemClick(AdapterView<?> arg0, View
arg1,
                int arg2, long arg3) {
                // TODO Auto-generated method stub
                if(arg2==0)
                {
                    //Hiển thị mọi sách trong CSDL
                    authorData=null;

```

```

        loadAllListBook();
    }
    else
    {
        //Hiển thị sách theo tác giả chọn
        trong Spinner
        authorData=listAuthor.get(arg2);

        loadListBookByAuthor(authorData.getField1().toString());
    }
}
@Override
public void onNothingSelected(AdapterView<?> arg0) {
    // TODO Auto-generated method stub
    authorData=null;
}
});

setCurrentDateOnView();
//lệnh xử lý DatePickerDialog
Button bChangeDate=(Button) findViewById(R.id.buttonDate);
bChangeDate.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        showDialog(113);
    }
});
//Lệnh xử lý thêm mới một sản phẩm theo tác giả đang chọn
Button btnInsertBook =(Button)
findViewById(R.id.buttonInsertBook);
btnInsertBook.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        if(authorData==null)
        {
            Toast.makeText(InsertBookActivity.this,
            "Please choose an author to insert", Toast.LENGTH_LONG).show();
            return;
        }
        EditText txtTitle=(EditText)

```



```

findViewById(R.id.editTextTitle);
        ContentValues values=new ContentValues();
        values.put("title",
txtTitle.getText().toString());
        Calendar c=Calendar.getInstance();
        c.set(year, month, day);
        SimpleDateFormat dfmt=new SimpleDateFormat("dd-
MM-yyyy");
        values.put("dateadded",dfmt.format(c.getTime()));
        values.put("authorid",
authorData.getField1().toString());
        long bId=database.insert("tblBooks", null,
values);
        if(bId>0)
        {
            Toast.makeText(InsertBookActivity.this,
"Insert Book OK", Toast.LENGTH_LONG).show();

            loadListBookByAuthor(authorData.getField1().toString());
        }
        else
        {
            Toast.makeText(InsertBookActivity.this,
"Insert Book Failed", Toast.LENGTH_LONG).show();
        }
    }
});
}
/*
 * Hàm hiển thị mọi sách trong CSDL
 */
public void loadAllListBook()
{
    Cursor cur=database.query("tblBooks", null, null, null,
null, null, null);
    cur.moveToFirst();
    listBook=new ArrayList<InforData>();
    while(cur.isAfterLast()==false)
    {
        InforData d=new InforData();
        d.setField1(cur.getInt(0));
        d.setField2(cur.getString(1));
        d.setField3(cur.getString(2));
        listBook.add(d);
    }
}

```

```

        cur.moveToNext();
    }
    cur.close();
    adapter=new MySimpleArrayAdapter(InsertBookActivity.this,
R.layout.my_layout_for_show_list_data, listBook);
    ListView lv=(ListView) findViewById(R.id.listViewBook);
    lv.setAdapter(adapter);
}
/**
 * hàm hiển thị sách theo tác giả
 * @param authorid
 */
public void loadListBookByAuthor(String authorid)
{
    Cursor cur=database.query("tblBooks", null, "authorid=?",
new String[]{authorid}, null, null, null);
    cur.moveToFirst();
    listBook=new ArrayList<InforData>();
    while(cur.isAfterLast()==false)
    {
        InforData d=new InforData();
        d.setField1(cur.getInt(0));
        d.setField2(cur.getString(1));
        d.setField3(cur.getString(2));
        listBook.add(d);
        cur.moveToNext();
    }
    cur.close();
    adapter=new MySimpleArrayAdapter(InsertBookActivity.this,
R.layout.my_layout_for_show_list_data, listBook);
    ListView lv=(ListView) findViewById(R.id.listViewBook);
    lv.setAdapter(adapter);
}
@Override
protected Dialog onCreateDialog(int id) {
    // TODO Auto-generated method stub
    if(id==113)
    {
        return new DatePickerDialog(this, dateChange, year,
month, day);
    }
    return null;
}
/**

```

```

    * xử lý DatePickerDialog
    */
    private DatePickerDialog.OnDateSetListener dateChange= new
    OnDateSetListener() {

        @Override
        public void onDateSet(DatePicker view, int year1, int
        monthOfYear,
            int dayOfMonth) {
            // TODO Auto-generated method stub
            year=year1;
            month=monthOfYear;
            day=dayOfMonth;
            EditText eDate=(EditText)
            findViewById(R.id.editTextDate);
            eDate.setText(day+"-"+(month+1)+"-"+year);
        }
    };
    /**
    * thiết lập ngày tháng năm hiện tại
    */
    public void setCurrentDateOnView()
    {
        EditText eDate=(EditText) findViewById(R.id.editTextDate);
        Calendar cal=Calendar.getInstance();
        day=cal.get(Calendar.DAY_OF_MONTH);
        month=cal.get(Calendar.MONTH);
        year=cal.get(Calendar.YEAR);
        eDate.setText(day+"-"+(month+1)+"-"+year);
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_insert_book,
        menu);

        return true;
    }
}

```

\_ Giao diện custom layout :

```

<?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"
android:gravity="left"
>

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="0.2"
    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="0.4"
    android:text="Medium Text"
    android:gravity="left"
    android:textAppearance="?android:attr/textAppearanceMedium" />

</LinearLayout>
```

\_ Tạo lớp đối tượng InforData :

```
public class InforData {
    private Object field1;
    private Object field2;
    private Object field3;
```

```

    public Object getField1() {
        return field1;
    }
    public void setField1(Object field1) {
        this.field1 = field1;
    }
    public Object getField2() {
        return field2;
    }
    public void setField2(Object field2) {
        this.field2 = field2;
    }
    public Object getField3() {
        return field3;
    }
    public void setField3(Object field3) {
        this.field3 = field3;
    }
    @Override
    public String toString() {
        // TODO Auto-generated method stub
        return this.field1 + " - " +this.field2 + " - "+this.field3;
    }
}

```

\_ Code xử lý custom layout :

```

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);
        // TODO Auto-generated constructor stub
        this.context=(Activity) context;
        this.layout=textViewResourceId;
        this.list=objects;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup
parent) {

```

```

        // TODO Auto-generated method stub
        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);

        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(
));

        if(position==0)
        {
            row.setBackgroundColor(Color.RED);
        }
        return row;
    }
}

```

\_ Giao diện xem danh sách tác giả :

```

<?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" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical" >

        <ListView
            android:id="@+id/listViewShowData"
            android:layout_width="match_parent"
            android:layout_height="405dp" >
        </ListView>
    </LinearLayout>
</LinearLayout>

```

```
</LinearLayout>
```

```
<Button
```

```
    android:id="@+id/buttonBack"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Back" />
```

```
</LinearLayout>
```

\_ Code xử lý giao diện xem danh sách tác giả :

```
public class ShowListAuthorActivity extends Activity{
    @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();
            }
        });
    }
    List<InforData>list=new ArrayList<InforData>();
    InforData dataClick=null;
    SQLiteDatabase database=null;
    MySimpleArrayAdapter adapter=null;
    public void updateUI()
    {
        database=openOrCreateDatabase("mydata.db",
        SQLiteDatabase.CREATE_IF_NECESSARY, null);
        if(database!=null)
        {

            Cursor cursor=database.query("tblAuthors", null, null,
            null, null, null, null);
            startManagingCursor(cursor);
            InforData header=new InforData();
```

```

        header.setField1("STT");
        header.setField2("Mã tác giả");
        header.setField3("Tên tác giả");
        list.add(header);
        cursor.moveToFirst();
        while(!cursor.isAfterLast())
        {
            InforData data=new InforData();
            data.setField1(cursor.getInt(0));
            data.setField2(cursor.getString(1));
            data.setField3(cursor.getString(2));
            list.add(data);
            cursor.moveToNext();
        }
        cursor.close();
        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();
        AlertDialog.Builder b=new
Builder>ShowListAuthorActivity.this);
        b.setTitle("Remove");
        b.setMessage("Xóa [" +data.getField2() + " -
"+data.getField3() +"] hả?");
        b.setPositiveButton("Có", new
DialogInterface.OnClickListener() {

            @Override
            public void onClick(DialogInterface
dialog, int which) {

                // TODO Auto-generated method
stub

                int
n=database.delete("tblAuthors", "id=?", new
String[]{data.getField1().toString()});
                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.update("tblAuthors", values,
"id=?", new String[]{f1});
            if(n>0)
            {
                Toast.makeText(ShowListAuthorActivity.this,
"update ok ok ok ", Toast.LENGTH_LONG).show();
                dataClick.setField2(f2);
                dataClick.setField3(f3);
                if(adapter!=null)

```

```
adapter.notifyDataSetChanged();  
    }  
    }  
    }  
}
```