# SQLITE

## A. MỤC TIÊU

Trang bị cho sinh viên kiến thức về :

- Xây dựng ứng dụng lưu trữ dữ liệu với SQLite trong Android.

- Thực hiện được các truy vấn Insert, Update, Delete, Select trong Sqlite với Android.

## B. NỘI DUNG

- Tạo cơ sở dữ liệu SQLite lưu trữ dữ liệu

- Danh sách tất cả các toán tử trong SQLite

- Kiểu dữ liệu và lệnh trong SQLite

**-** Xây dựng các phương thức thực hiện truy vấn CSDL SQLite

## C. KẾT QUẢ SAU KHI HOÀN THÀNH

Hiểu được nguyên lý hoạt động của SQLite.

Hoàn thiện chương trình thêm, sửa, xóa, hiển thị SQLite.

## D. YÊU CẦU PHẦN CỨNG, PHẦN MỀM

- Hệ điều hành: Windows 7 32-64 bit trở lên./ MAC OS X 10.8.5 trở lên.

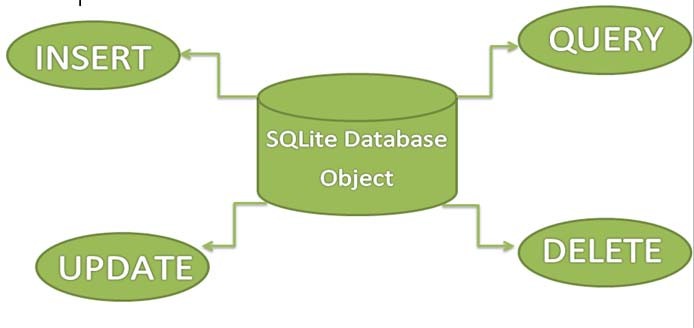
- Bộ nhớ trong: Tối thiểu 4 GB RAM, yêu cầu 8GB RAM.

- Ổ cứng: Tối thiêu 2GB trống cho ổ cứng, khuyến nghị 4GB (500MB cho IDE và 1.5GB cho Android SDK và image giả lập hệ thống).

Tỉ lệ màn hình**:** tối thiểu 1280x800.

## E. HƯỚNG DẪN CHI TIẾT

### 1. SQLite

QLite là cơ sở dữ liệu cơ sở truy vấn Cấu trúc, mã nguồn mở, trọng lượng nhẹ, không cần truy cập mạng và cơ sở dữ liệu độc lập. Nó hỗ trợ các tính năng cơ sở dữ liệu quan hệ nhúng.

Hình 8.1. Cấu trúc SQLite Database

Bất cứ khi nào một ứng dụng cần lưu trữ một lượng lớn dữ liệu thì việc sử dụng sqlite sẽ được ưu tiên hơn các hệ thống lưu trữ khác như SharedPreferences hoặc lưu dữ liệu trong ﬁles.

Android đã tích hợp triển khai cơ sở dữ liệu SQLite. Nó có sẵn cục bộ trên thiết bị (điện thoại di động và máy tính bảng) và chứa dữ liệu ở định dạng văn bản. Nó mang dữ liệu trọng lượng nhẹ và phù hợp với nhiều ngôn ngữ. Vì vậy, nó không yêu cầu bất kỳ quy trình quản trị hoặc thiết lập cơ sở dữ liệu nào.

Lưu ý: Cơ sở dữ liệu đã tạo được lưu trong thư mục:

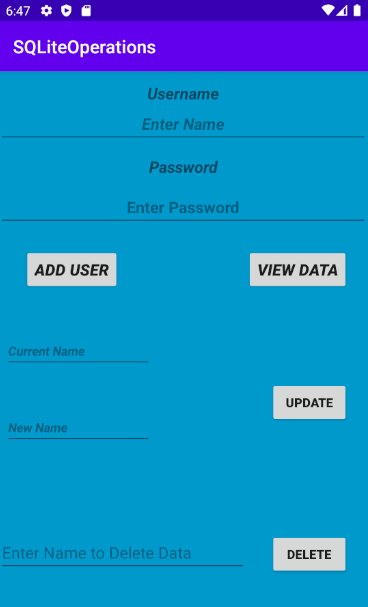
data/data/APP\_Name/databases/ DATABASE\_NAME.

### 2. Ví dụ: Xây dựng Project quản lý thông tin đăng nhập gồm: Username và Password được lưu trữ trong CSDL SQLite. Xây dựng các chức năng ADDUSER, VIEWDATA, UPDATE và DELETE.

**Hướng dẫn:**

***Bước 1:*** Tạo một dự án mới và đặt tên là SQLiteOperations

***Bước 2:*** Mở res -> layout -> activity\_main.xml (hoặc) main.xml và tạo bố cục như hình.



Code tham khảo:

*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/activity\_main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.example.buoi1.MainActivity"  
 android:background="@android:color/holo\_blue\_dark">  
 <TextView  
 android:text="Username"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginTop="12dp"  
 android:id="@+id/textView"  
 android:textSize="18sp"  
 android:textStyle="bold|italic"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"  
 android:gravity="center" />  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="textPersonName"  
 android:ems="10"  
 android:id="@+id/editName"  
 android:textStyle="bold|italic"  
 android:layout\_below="@+id/textView"  
 android:layout\_alignParentRight="true"  
 android:layout\_alignParentEnd="true"  
 android:hint="Enter Name"  
 android:gravity="center\_vertical|center" />  
 <TextView  
 android:text="Password"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="13dp"  
 android:id="@+id/textView2"  
 android:textStyle="bold|italic"  
 android:textSize="18sp"  
 android:layout\_below="@+id/editName"  
 android:layout\_alignParentRight="true"  
 android:layout\_alignParentEnd="true"  
 android:gravity="center"  
 android:hint="Enter Password" />  
 <Button  
 android:text="View data"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/button2"  
 android:textSize="18sp"  
 android:onClick="viewdata"  
 android:textStyle="bold|italic"  
 android:layout\_alignBaseline="@+id/button"  
 android:layout\_alignBottom="@+id/button"  
 android:layout\_alignRight="@+id/button4"  
 android:layout\_alignEnd="@+id/button4" />  
 <Button  
 android:text="Add User"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/button"  
 android:textStyle="bold|italic"  
 android:textSize="18sp"  
 android:onClick="addUser"  
 android:layout\_marginLeft="28dp"  
 android:layout\_marginStart="28dp"  
 android:layout\_below="@+id/editPass"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginTop="23dp" />  
 <Button  
 android:text="Update"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/button3"  
 android:onClick="update"  
 android:textStyle="normal|bold"  
 android:layout\_below="@+id/editText3"  
 android:layout\_alignLeft="@+id/button4"  
 android:layout\_alignStart="@+id/button4"  
 android:layout\_marginTop="13dp" />  
 <EditText  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:inputType="textPersonName"  
 android:ems="10"  
 android:id="@+id/editText6"  
 android:layout\_alignTop="@+id/button4"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"  
 android:freezesText="false"  
 android:hint="Enter Name to Delete Data"  
 android:layout\_toLeftOf="@+id/button2"  
 android:layout\_toStartOf="@+id/button2" />  
 <Button  
 android:text="Delete"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginRight="21dp"  
 android:layout\_marginEnd="21dp"  
 android:id="@+id/button4"  
 android:onClick="delete"  
 android:textStyle="normal|bold"  
 tools:ignore="RelativeOverlap"  
 android:layout\_marginBottom="41dp"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignParentRight="true"  
 android:layout\_alignParentEnd="true" />  
 <EditText  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:inputType="textPersonName"  
 android:ems="10"  
 android:layout\_marginTop="47dp"  
 android:id="@+id/editText3"  
 android:textStyle="bold|italic"  
 android:textSize="14sp"  
 android:layout\_below="@+id/button"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginLeft="7dp"  
 android:layout\_marginStart="7dp"  
 android:hint="CurrentName"/>  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="textPassword"  
 android:ems="10"  
 android:layout\_marginTop="11dp"  
 android:id="@+id/editPass"  
 android:hint="Enter Password"  
 android:gravity="center\_vertical|center"  
 android:textSize="18sp"  
 android:layout\_below="@+id/textView2"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"  
 android:textAllCaps="false"  
 android:textStyle="normal|bold" />  
 <EditText  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:inputType="textPersonName"  
 android:ems="10"  
 android:id="@+id/editText5"  
 android:textStyle="bold|italic"  
 android:textSize="14sp"  
 android:hint="New Name"  
 android:layout\_alignTop="@+id/button3"  
 android:layout\_alignLeft="@+id/editText3"  
 android:layout\_alignStart="@+id/editText3"  
 android:layout\_marginTop="32dp" />  
</RelativeLayout>

***Bước 3****:* Trong MainActivity.java khởi tạo các hàm.

package com.example.SQLiteOperations;  
*//import…*public class MainActivity extends AppCompatActivity {  
 EditText Name, Pass, updateold, updatenew, delete;  
 myDbAdapter helper;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 Name = (EditText) findViewById(R.id.editName);  
 Pass = (EditText) findViewById(R.id.editPass);  
 updateold = (EditText) findViewById(R.id.editText3);  
 updatenew = (EditText) findViewById(R.id.editText5);  
 delete = (EditText) findViewById(R.id.editText6);  
 helper = new myDbAdapter(this);  
 }  
 public void addUser(View view) {  
 String t1 = Name.getText().toString();  
 String t2 = Pass.getText().toString();  
 if (t1.isEmpty() || t2.isEmpty()) {  
 Message.message(getApplicationContext(), "Enter Both Name and Password");  
 } else {  
 long id = helper.insertData(t1, t2);  
 if (id <= 0) {  
 Message.message(getApplicationContext(), "Insertion Unsuccessful");  
 Name.setText("");  
 Pass.setText("");  
 } else {  
 Message.message(getApplicationContext(), "Insertion Successful");  
 Name.setText("");  
 Pass.setText("");  
 }  
 }  
 }  
 public void viewdata(View view) {  
 String data = helper.getData();  
 Message.message(this, data);  
 }  
 public void update(View view) {  
 String u1 = updateold.getText().toString();  
 String u2 = updatenew.getText().toString();  
 if (u1.isEmpty() || u2.isEmpty()) {  
 Message.message(getApplicationContext(),"Enter Data");  
 } else {  
 Message.message(getApplicationContext(), "Enter Data");  
 int a = helper.updateName(u1, u2);  
 if (a <= 0) {  
 Message.message(getApplicationContext(), "Unsuccessful");  
 updateold.setText("");  
 updatenew.setText("");  
 } else {  
 Message.message(getApplicationContext(), "Updated");  
 updateold.setText("");  
 updatenew.setText("");  
 }  
 }  
 }  
  
 public void delete(View view) {  
 String uname = delete.getText().toString();  
 if (uname.isEmpty()) {  
 Message.message(getApplicationContext(), "Enter Data");  
 } else {  
 int a = helper.delete(uname);  
 if (a <= 0) {  
 Message.message(getApplicationContext(), "Unsuccessful");  
 delete.setText("");  
 } else {  
 Message.message(this, "DELETED");  
 delete.setText("");  
 }  
 }  
 }  
}

***Bước 4:*** Trong bước này, tạo một lớp java myDbAdapter.java.

Trong phần này, ta xác định các hàm được sử dụng để thực hiện các hoạt động chèn, cập nhật và xóa các hoạt động trong SQLite. Hơn nữa lớp này tạo một lớp khác sẽ mở rộng SQLiteOpenHelper. Mỗi hàm mang các phương thức tương đương để thực hiện các thao tác.

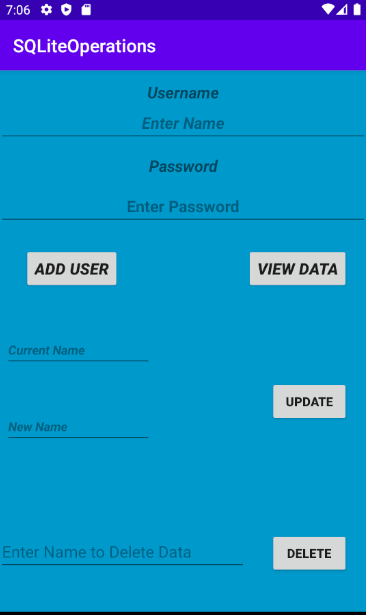
Code tham khảo:

package com.example.SQLiteOperations;  
*//import…*public class myDbAdapter {  
 myDbHelper myhelper;  
 public myDbAdapter(Context context) {  
 myhelper = new myDbHelper(context);  
 }  
 public long insertData(String name, String pass) {  
 SQLiteDatabase dbb = myhelper.getWritableDatabase();  
 ContentValues contentValues = new ContentValues();  
 contentValues.put(myDbHelper.*NAME*, name);  
 contentValues.put(myDbHelper.*MyPASSWORD*, pass);  
 long id = dbb.insert(myDbHelper.*TABLE\_NAME*, null, contentValues);  
 return id;  
 }  
 public String getData() {  
 SQLiteDatabase db = myhelper.getWritableDatabase();  
 String[] columns = {myDbHelper.*UID*, myDbHelper.*NAME*, myDbHelper.*MyPASSWORD*};  
 Cursor cursor = db.query(myDbHelper.*TABLE\_NAME*, columns, null, null, null, null, null);  
 StringBuffer buffer = new StringBuffer();  
 while (cursor.moveToNext()) {  
 int cid = cursor.getInt(cursor.getColumnIndex(myDbHelper.*UID*));  
 String name = cursor.getString(cursor.getColumnIndex(myDbHelper.*NAME*));  
 String password  
 = cursor.getString(cursor.getColumnIndex(myDbHelper.*MyPASSWORD*));  
 buffer.append(cid + " " + name + " " + password + " \n");  
 }  
 return buffer.toString();  
 }  
 public int delete(String uname) {  
 SQLiteDatabase db = myhelper.getWritableDatabase();  
 String[] whereArgs = {uname};  
 int count = db.delete(myDbHelper.*TABLE\_NAME*, myDbHelper.*NAME* + " = ?", whereArgs);  
 return count;  
 }  
 public int updateName(String oldName, String newName) {  
 SQLiteDatabase db = myhelper.getWritableDatabase();  
 ContentValues contentValues = new ContentValues();  
 contentValues.put(myDbHelper.*NAME*, newName);  
 String[] whereArgs = {oldName};  
 int count = db.update(myDbHelper.*TABLE\_NAME*, contentValues, myDbHelper.*NAME* + " = ? ",whereArgs );  
 return count;  
 }  
  
 static class myDbHelper extends SQLiteOpenHelper {  
 private static final String *DATABASE\_NAME* = "Android";  
 private static final String *TABLE\_NAME* = "Usertb";  
 private static final int *DATABASE\_Version* = 1;  
 private static final String *UID*="id";  
 private static final String *NAME* = "Name";  
 private static final String *MyPASSWORD*= "Password";  
 private static final String *CREATE\_TABLE* = "CREATE TABLE "+*TABLE\_NAME*+" ("+*UID*+" INTEGER PRIMARY KEY AUTOINCREMENT, "+*NAME*+" VARCHAR(255) ,"+ *MyPASSWORD*+" VARCHAR(225));";  
 private static final String *DROP\_TABLE* = "DROP TABLE IF EXISTS " + *TABLE\_NAME*;  
 private Context context;  
 public myDbHelper(Context context) {  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_Version*);  
 this.context = context;  
 }  
 public void onCreate(SQLiteDatabase db) {  
 try {  
 db.execSQL(*CREATE\_TABLE*);  
 } catch (Exception e) {  
 Message.message(context, "" + e);  
 }  
 }  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 try {  
 Message.message(context, "OnUpgrade");  
 db.execSQL(*DROP\_TABLE*);  
 onCreate(db);  
 } catch (Exception e) {  
 Message.message(context, "" + e);  
 }  
 }  
 }  
}

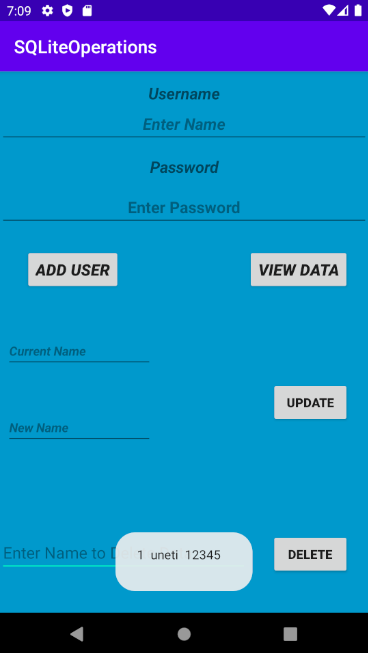
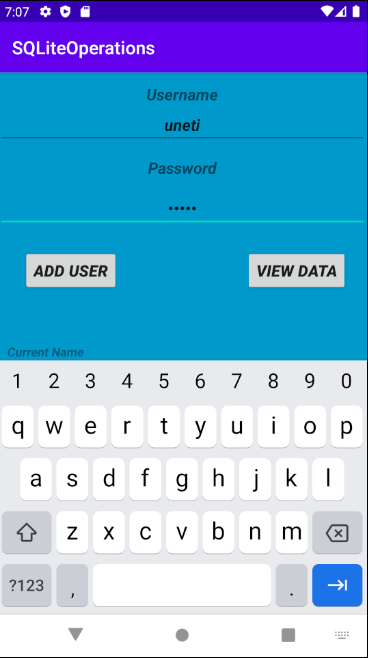
***Bước 5:*** Trong bước này, tạo một lớp java khác Message.class

**package** com.example.SQLiteOperations;  
  
*// Phần import;*  
  
public class Message {  
 public static void message(Context context, String message) {  
 Toast.*makeText*(context, message, Toast.*LENGTH\_LONG*).show();  
 }  
}

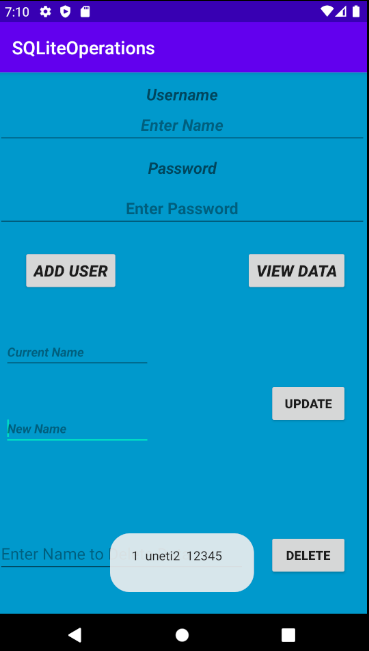
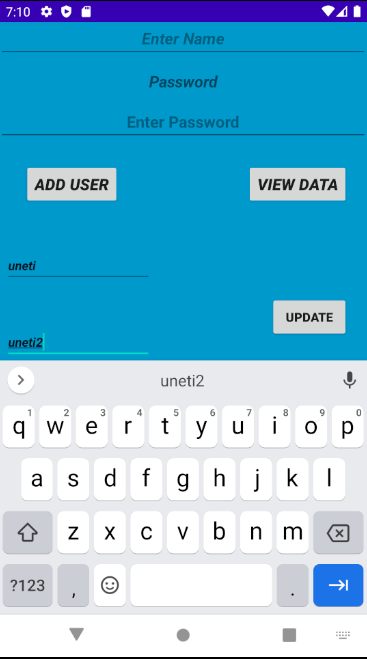
Kết quả đạt được:



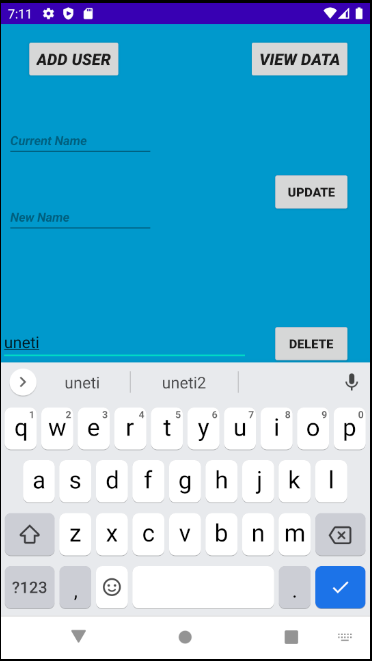
Trước và sau khi thêm user:



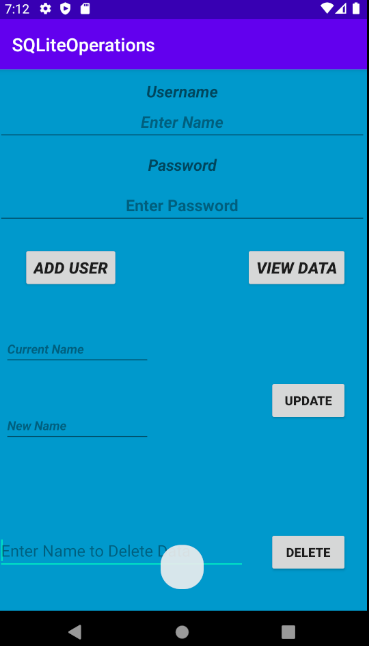
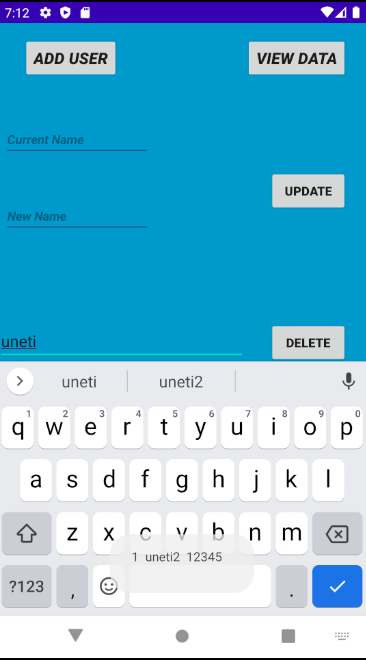
Chức năng sửa tên user:



Chức năng xóa:



Trước và sau xóa:



## F. BÀI TẬP ÔN TẬP

**Bài tập 1.** Tạo một dự án quản lý thông tin hàng hóa của 1 cửa hàng gồm các thông tin: mã hàng hóa, tên hàng hóa, số lượng, đơn giá, đơn vị tính. Thiết kế giao diện và có các chức năng sau:

- Thêm hàng hóa.

- Sửa thông tin của hàng hóa theo mã hàng hóa.  
 - Xóa hàng hóa theo mã hàng hóa.

- Hiển thị toàn hộ hàng hóa.

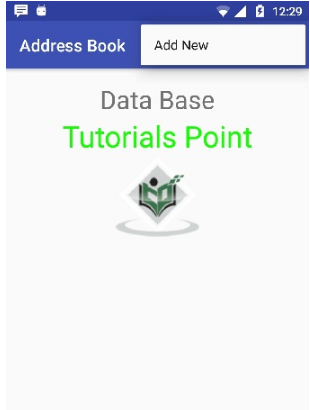
**Hướng dẫn:**

- Thiết kế giao diện.

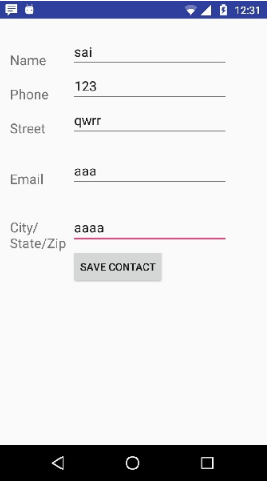
- Tương tự ví dụ trên ta khởi tạo class myDbAdapter để quản lý thông tin của hàng hóa của cửa hàng.

- Tạo các hàm xử lý thêm, sửa, xóa và hiển thị thông tin của cửa hàng (sử dụng SQLiteOpenHelper để xử lý).

**Bài tập 2.**

a. Sử dụng SQLite viết chương trình sau:

b. Khi bấm vào button Add New sẽ hiện ra màn hình:

c. Button SAVE CONTACT sẽ lưu thông tin vào CSDL.

**Bài tập 3.** Viết chương trình quản lý sách được mô tả như sau:  
o Một tác giả sẽ có nhiều cuốn sách, thông tin tác giả gồm: mã , tên  
o Mỗi cuốn sách thuộc về một nhà tác giả nào đó, thông tin mỗi cuốn sách gồm: mã  
sách, tên sách, ngày xuất bản  
o Hãy chọn kiểu dữ liệu hợp lý để tạo cơ sở dữ liệu (sử dụng SQLite) cho đúng với  
mô tả trên  
- Thiết kế giao diện như hình bên dưới:



Chú ý:

- Phải tạo Tác giả trước khi vào chức năng quản lý Sách

- Load danh sách Tác giả vào Spinner

- Sử dụng DatePickerDialog để chọ ngày xuất bản

- Chọn “Thêm sách” lưu vào CSDL đúng với tác giả chọn trong Spinner, đồng thời cập nhập vào ListView bên dưới.