**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

🙤🙧🟍🙥🙦



**PHÁT TRIỂN ỨNG DỤNG TRÊN THIẾT BỊ DI ĐỘNG**

**NT118.J12**

**BÀI THỰC HÀNH SỐ 3 – NHÓM 1**

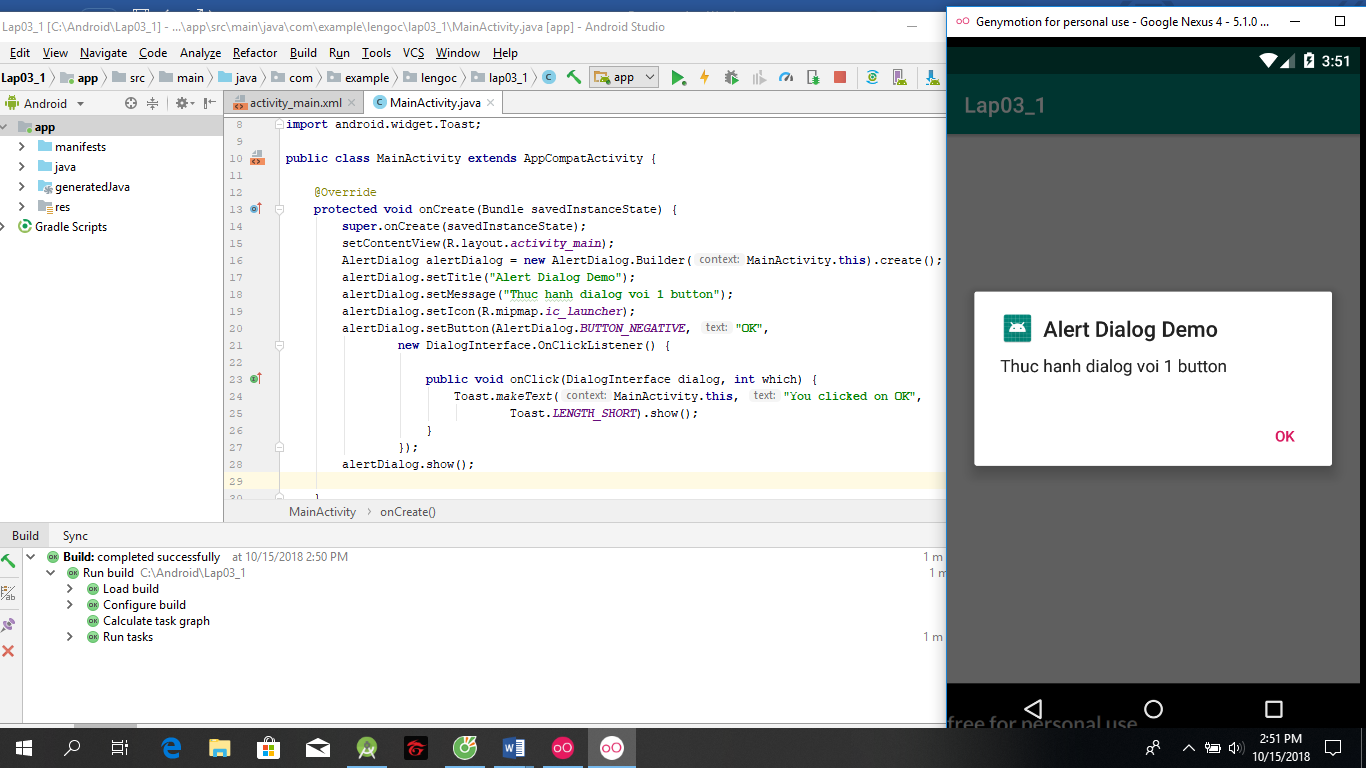
* **GV: Thái Huy Tân**

|  |  |  |
| --- | --- | --- |
| Nhóm 11 | | |
| **STT** | **Họ tên** | **MSSV** |
| 1 | Trương Thị Bích Chi | 15520062 |
| 2 | Lê Hồng Ngọc | 15520553 |
| 3 | Đỗ Trọng Nhân | 15520566 |

**TP. HỒ CHÍ MINH – 15/10/2018**

DIALOG

1. Alert Dialog hiển thị 1 Button



Trong file JAVA:

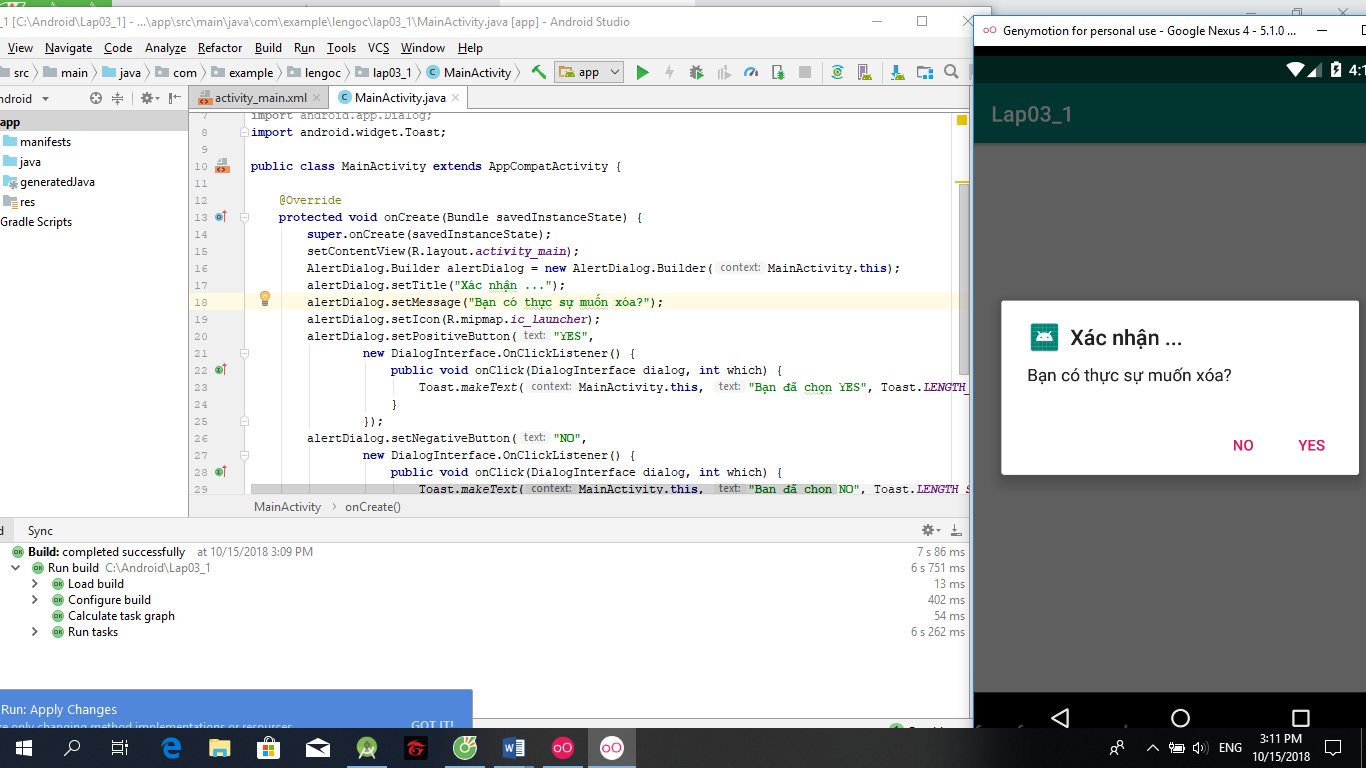
AlertDialog alertDialog = **new** AlertDialog.Builder(MainActivity.**this**).create();

=>Đây là hàm khởi tạo một **AlertDialog.Builder**, thằng này có tác dụng vẽ ra giao diện cho AlertDialog, mặc định hàm trên nó sẽ sử dụng theme dialog mặc định khi show.

+Tạo các thiết lập cho dialog bên dưới:  
alertDialog.setTitle(**"Alert Dialog Demo"**);  
alertDialog.setMessage(**"Thuc hanh dialog voi 1 button"**);  
alertDialog.setIcon(R.mipmap.***ic\_launcher***);

+Tạo button, BUTTON\_NEGATIVE set button OK. Tạo sự kiện click vào nút OK, show lên 1 Toast thông báo trong 1s.  
alertDialog.setButton(AlertDialog.***BUTTON\_NEGATIVE***, **"OK"**,  
 **new** DialogInterface.OnClickListener() {  
  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 Toast.*makeText*(MainActivity.**this**, **"You clicked on OK"**,  
 Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
alertDialog.show();// **Hiển thị Dialog**

1. Alert Dialog hiển thị 2 Button



AlertDialog alertDialog = **new** AlertDialog.Builder(MainActivity.**this**).create();

=>Đây là hàm khởi tạo một **AlertDialog.Builder**, thằng này có tác dụng vẽ ra giao diện cho AlertDialog, mặc định hàm trên nó sẽ sử dụng theme dialog mặc định khi show.

+alertDialog.show(); Hiển thị Dialog.

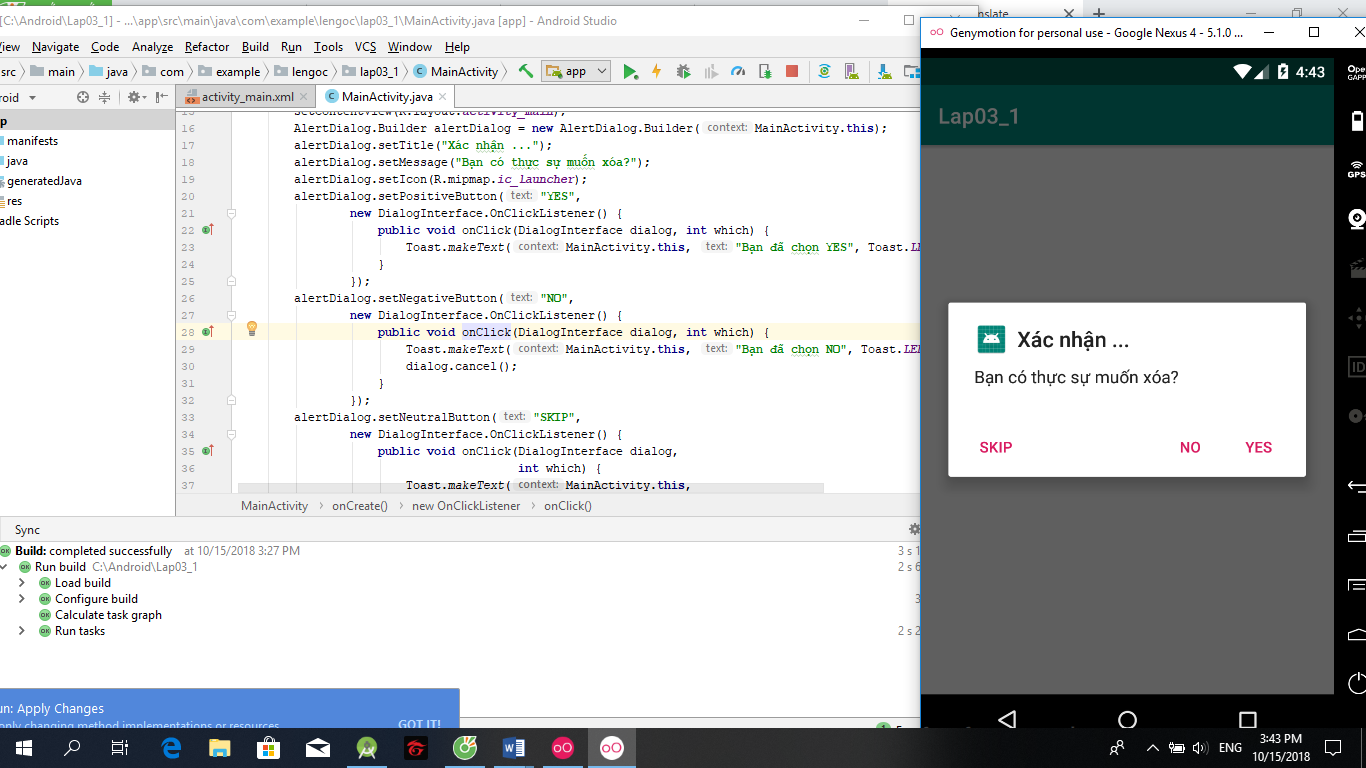
+Tạo button “YES” nằm bên phải với setPositiveButton và button “NO” với setPositiveButton

+Xử lý sự kiện onClick ở 2 button:

YES: show Toast thông báo “**Bạn đã chọn YES”**

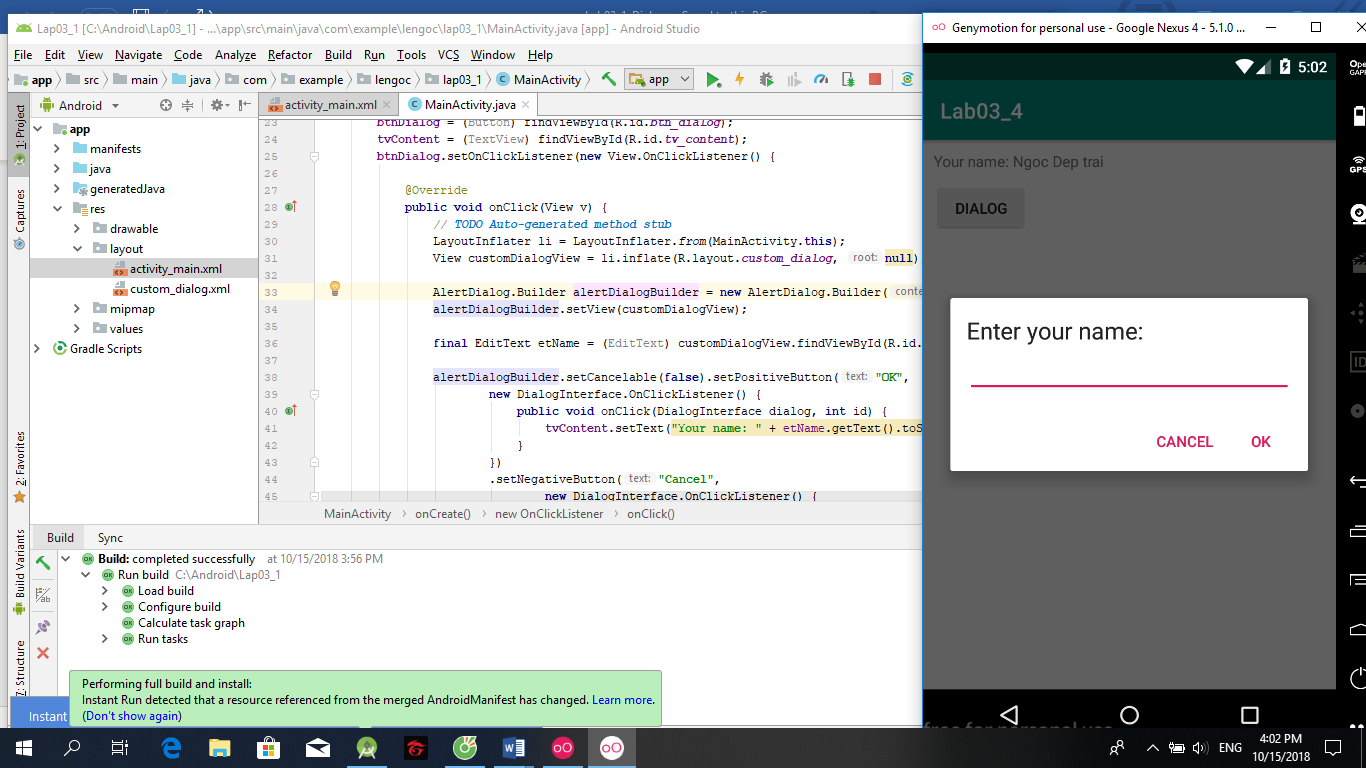
NO: show Toast thông báo “**Bạn đã chọn NO”,** dialog.cancel(); Hủy bỏ hộp thoại Dialog

1. Alert Dialog hiển thị 3 Button



+Nút Neutral luôn được đặt ở cực bên trái trong khi các nút Negative và Positive được đóng gói cùng nhau về phía bên phải.

1. Custom Alert Dialog:



+Thay đổi nội dung các file strings.xml, dimens.xml để tạo ra giao diện mình muốn.

+Tạo file layout custom\_dialog với TextView và EditText để nhập tên của bạn.

+activity\_main với TextView hiển thị tên của bạn và Button Dialog.

+ MainActivity.java:

.Tạo 2 biến Button btnDialog và TextView tvContent để sử dụng

.Tạo sự kiện onClick trên btnDialog:

+Dùng lớp LayoutInflater để định View “customDialogView” từ custom\_dialog cho giao diện dialog được hiển thị.

**+**Vẽ giao diện cho Dialog

AlertDialog.Builder alertDialogBuilder = **new** AlertDialog.Builder(MainActivity.**this**);  
 alertDialogBuilder.setView(customDialogView);

+Tạo giao diện EditText

+ alertDialogBuilder.setCancelable(false) => đây là thuộc tính nếu set false thì khi show dialog lên người dùng click ra bên ngoài dialog thì nó vẫn không bị mất, nếu set true thì sẽ mất khi click vào bất kì đâu ngoài dialog.

.setPositiveButton("OK",new DialogInterface.OnClickListener()

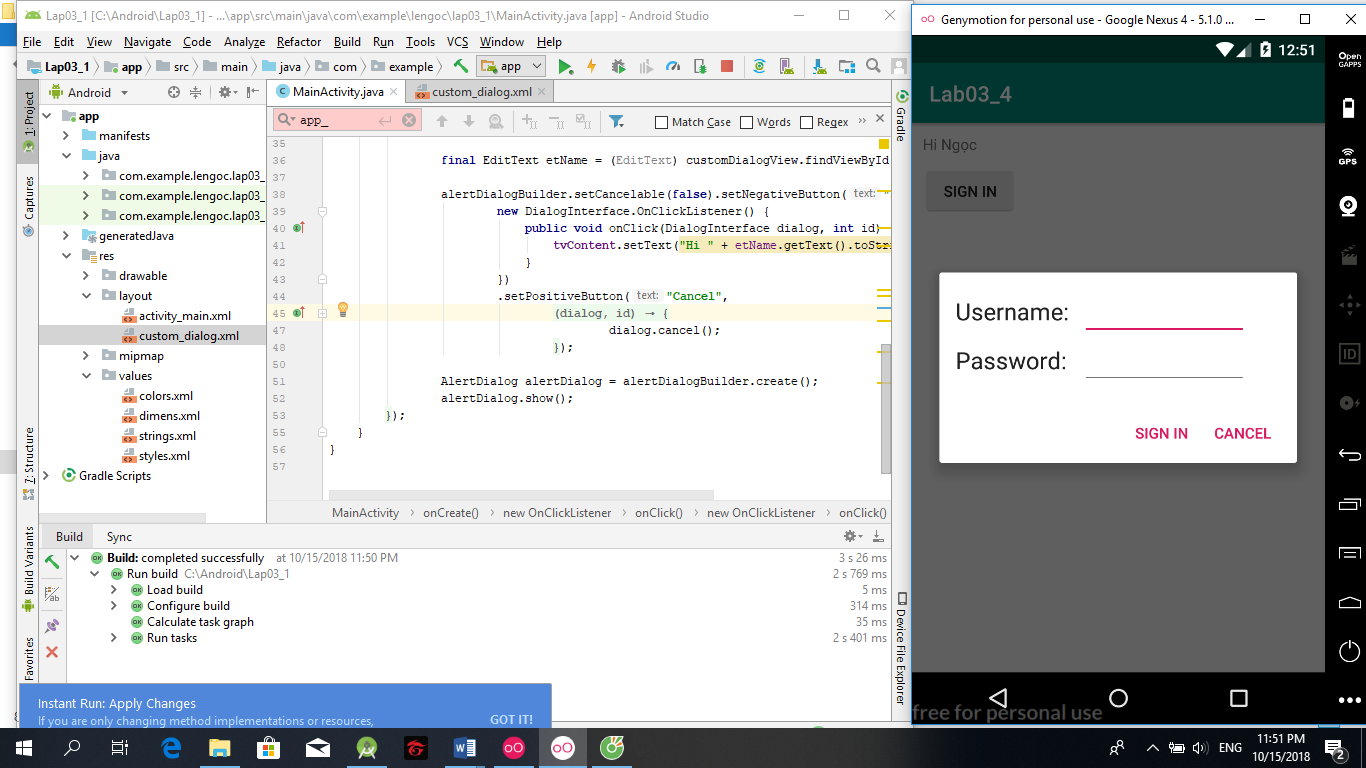
{

public void onClick(DialogInterface dialog, int id) {  
 tvContent.setText("Your name: " + etName.getText().toString());  
 }  
})

=>> Tạo sự kiện nhấn vào nut Ok khi nhập vào editText trong dialog, lấy nội dung từ editText hiện thị ra giao diện custom\_dialog.

+Sự kiện “Cancel” hủy hộp thoại dialog

1. Tạo 1 customdialog như sau:



+Từ câu 4 trên chỉnh sữa code

\*Trong strings.xml sữa thành:

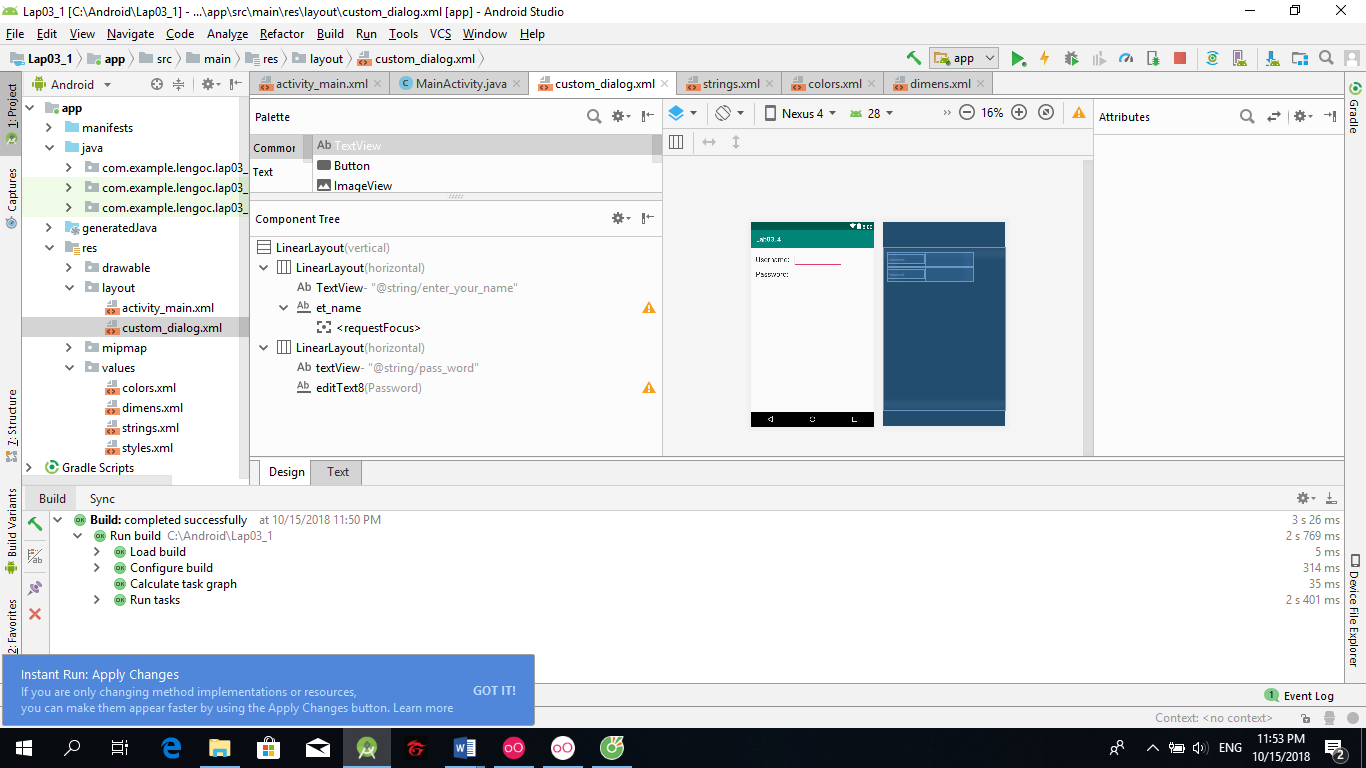
<string name="dialog">Sign in</string>

<**string name="pass\_word"**>Password:</**string**>

\*Trong ManActivity:

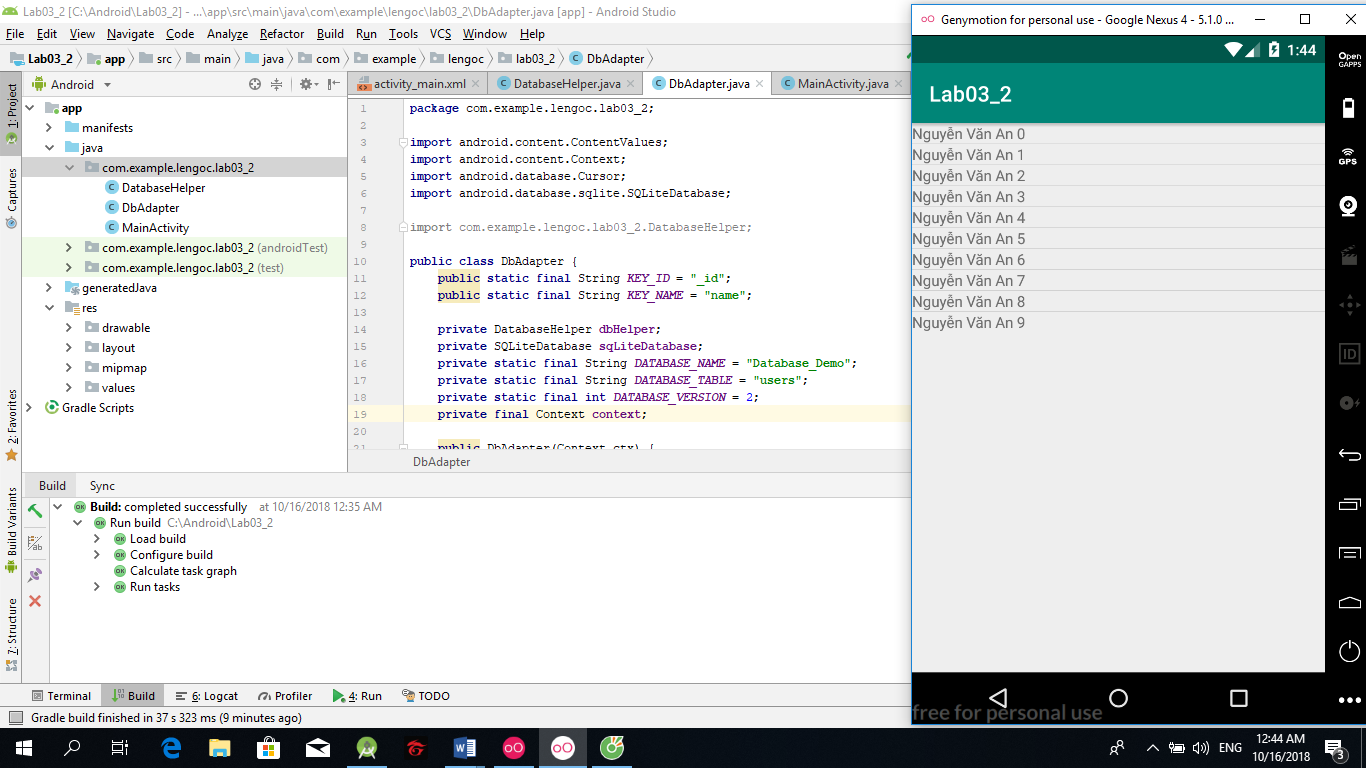
alertDialogBuilder.setCancelable(**false**).setNegativeButton(**"Sign in"**,  
 **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** id) {  
 **tvContent**.setText(**"Hi "** + etName.getText().toString());  
 }  
 })  
 .setPositiveButton(**"Cancel"**,  
 **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** id) {  
 dialog.cancel();  
 }  
 });

\*Trong Custom\_Dialog

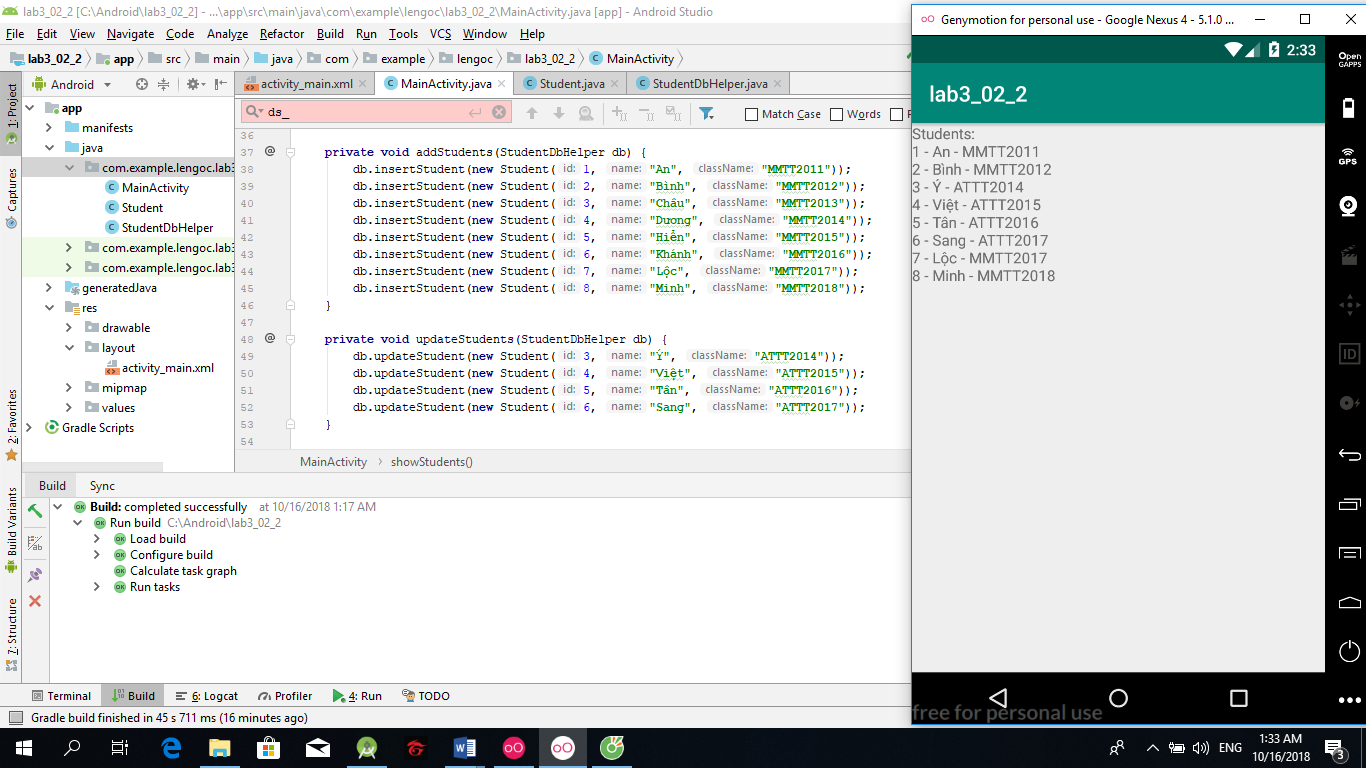


SQLITE

Bài tập 1: tạo 1 CSDL SQLite, thêm vào CSDL danh sách các user, sau đó lấy dữ liệu từ CSDL hiện lên listview:



**Bài tập 2: Quản lý sinh viên**



Giải thích tổng quát:

+ Tạo class Object Student gồm id, name, className cùng với các phương thức cần thiết.

+ Trong hàm main: Gọi hàm xử lý thêm Student, sau đó truyền các thông tin Object Student qua StudentDbHelper (class tự định nghĩa) và sử lý thêm dữ liệu, sau đó truyền dữ liệu về lại và gọi hàm hiển thị dữ liệu lên view.

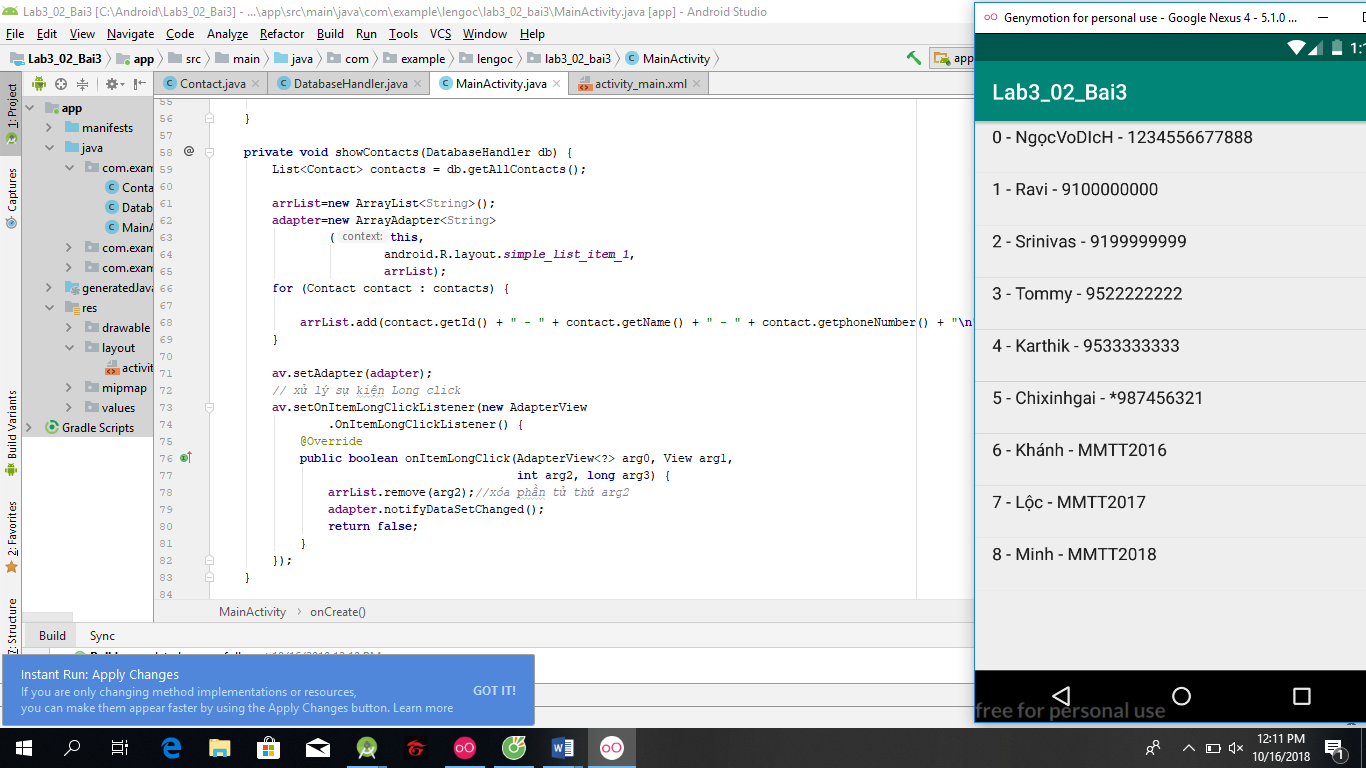
+ Update dữ liệu Student: Gọi hàm updateStudent trong main và truyền dữ liệu mới muốn thay đổi (id của Student muốn update, name và className là thông tin muốn thay đổi).

**Bài tập 3: Quản lý Contact**

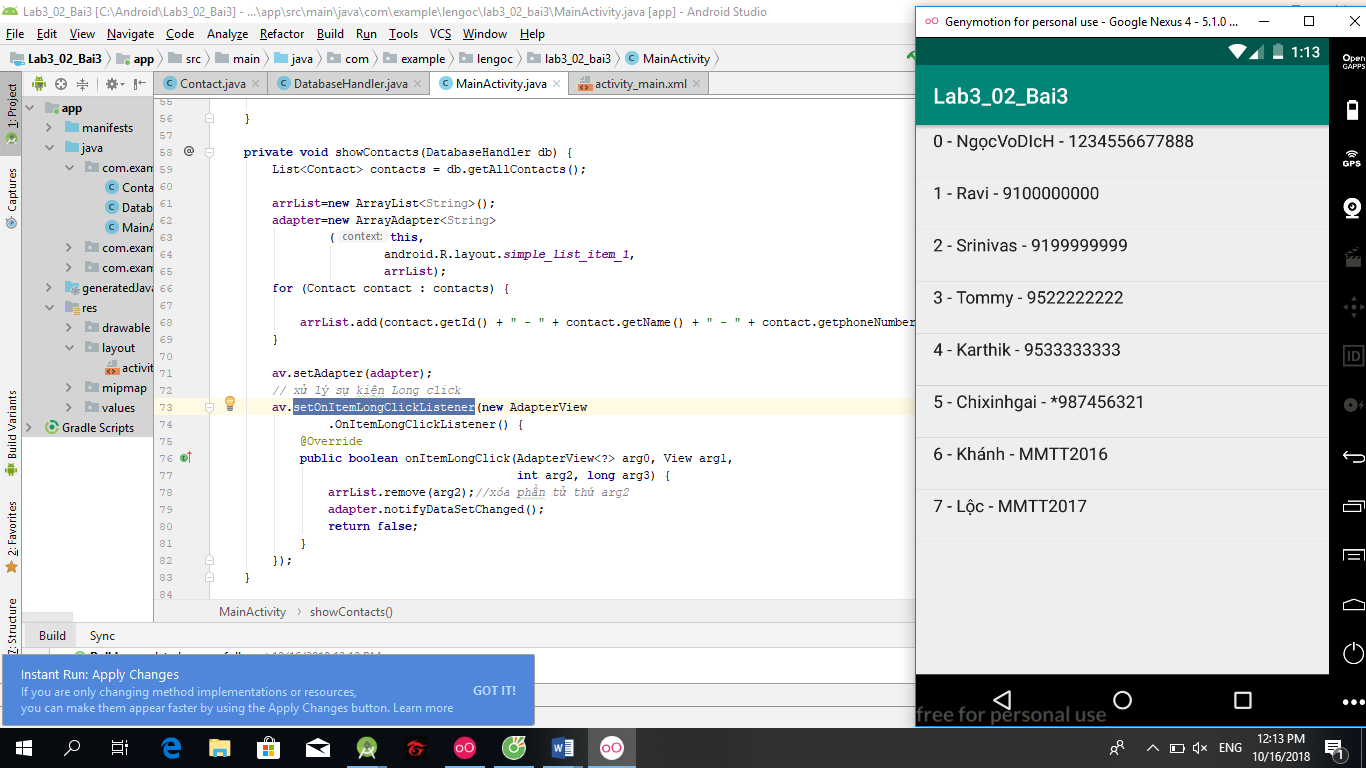
1. CSDL dữ liệu có cấu trúc như sau:

* **Database Name:** contactsManager
* **Table Name:** contacts

|  |  |  |
| --- | --- | --- |
| **Columns** | **DataType** | **Key** |
| **id** | INTEGER | Primary key |
| **name** | TEXT |  |
| **phone\_number** | TEXT |  |



+Sự kiện setOnItemLongClickListener xóa phần tử thứ 8:



+Trong Contact:

**package** com.example.lengoc.lab3\_02\_bai3;  
  
**public class** Contact {  
 **private int id** ;  
 **private** String **name** = **""**;  
 **private** String **phoneNumber** = **""**;  
  
 **public** Contact() { }  
 **public** Contact(String name, String phoneNumber) {  
 **this**.**name** = name;  
 **this**.**phoneNumber** = phoneNumber;  
 }  
  
 **public** Contact(**int** id, String name, String phoneNumber) {  
 **this**.**id** = id;  
 **this**.**name** = name;  
 **this**.**phoneNumber** = phoneNumber;  
 }  
  
 **public int** getId() {  
 **return id**;  
 }  
  
 **public void** setId(**int** id) {  
 **this**.**id** = id;  
 }  
  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public void** setName(String name) {  
 **this**.**name** = name;  
 }  
  
 **public** String getphoneNumber() {  
 **return phoneNumber**;  
 }  
  
 **public void** setphoneNumber(String phoneNumber) {  
 **this**.**phoneNumber** = phoneNumber;  
 }  
}

+Trong DatabaseHandler:

**package** com.example.lengoc.lab3\_02\_bai3;  
**import** com.example.lengoc.lab3\_02\_bai3.Contact;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**import** android.content.ContentValues;  
**import** android.content.Context;  
**import** android.database.Cursor;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.database.sqlite.SQLiteOpenHelper;  
  
**public class** DatabaseHandler **extends** SQLiteOpenHelper {  
  
  
 **private static final int *DATABASE\_VERSION*** = 1;  
 **private static final** String ***DATABASE\_NAME*** = **"contactsManager"**;  
 **private static final** String ***TABLE\_CONTACTS*** = **"contacts"**;  
  
 *// Contacts Table Columns names* **private static final** String ***KEY\_ID*** = **"id"**;  
 **private static final** String ***KEY\_NAME*** = **"name"**;  
 **private static final** String ***KEY\_PH\_NO*** = **"phone\_number"**;  
  
 **public** DatabaseHandler(Context context) {  
 **super**(context, ***DATABASE\_NAME***, **null**, ***DATABASE\_VERSION***);  
 }  
  
 *// Creating Tables* @Override  
 **public void** onCreate(SQLiteDatabase db) {  
 String CREATE\_CONTACTS\_TABLE = **"CREATE TABLE "** + ***TABLE\_CONTACTS*** + **"("** + ***KEY\_ID*** + **" INTEGER PRIMARY KEY AUTOINCREMENT,"** + ***KEY\_NAME*** + **" TEXT,"** + ***KEY\_PH\_NO*** + **" TEXT"** + **")"**;  
 db.execSQL(CREATE\_CONTACTS\_TABLE);  
 }  
  
 *// Upgrading database* @Override  
 **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
 *// Drop older table if existed* db.execSQL(**"DROP TABLE IF EXISTS "** + ***TABLE\_CONTACTS***);  
  
 *// Create tables again* onCreate(db);  
 }  
  
 *// Các hàm còn lại: để insert, select, update và delete contact sv tự viết  
// Adding new contact* **public void** addContact(Contact contact) {  
 SQLiteDatabase db = **this**.getWritableDatabase();  
 String nullColumnHack = **null**; *// Allow null value* ContentValues values = **new** ContentValues();  
 **if** (contact.getId() != -1) {  
 values.put(***KEY\_ID***, contact.getId());  
 values.put(***KEY\_NAME***, contact.getName());  
 values.put(***KEY\_PH\_NO***, contact.getphoneNumber());  
 db.insert(***TABLE\_CONTACTS***, nullColumnHack, values);  
 }  
 db.close();  
 }  
  
  
 *// Getting single contact* **public** Contact getContact(**int** id) {  
  
 SQLiteDatabase db = **this**.getReadableDatabase();  
 String[] fields = {***KEY\_ID***, ***KEY\_NAME***, ***KEY\_PH\_NO***};  
 String criterials = ***KEY\_ID*** + **"=?"**;  
 String[] parameters = {String.*valueOf*(id)};  
 String groupby = **null**;  
 String having = **null**;  
 String orderby = **null**;  
 Cursor cursor = db.query(***TABLE\_CONTACTS***, fields, criterials,  
 parameters, groupby, having, orderby);  
 **if** (cursor != **null**)  
 cursor.moveToFirst();  
 Contact contact = **new** Contact(Integer.*parseInt*(cursor.getString(0)),  
 cursor.getString(1), cursor.getString(2));  
 **return** contact;  
 }  
  
 *// Getting All Contacts* **public** List<Contact> getAllContacts() {  
 List<Contact> contacts = **new** ArrayList<Contact>();  
 String[] criterial = **null**;  
 String selectQuery = **"SELECT \* FROM "** + ***TABLE\_CONTACTS***;  
 SQLiteDatabase db = **this**.getReadableDatabase();  
 Cursor cursor = db.rawQuery(selectQuery, criterial);  
 **if** (cursor.moveToFirst()) {  
 **do** {  
 Contact contact = **new** Contact();  
 contact.setId(Integer.*parseInt*(cursor.getString(0)));  
 contact.setName(cursor.getString(1));  
 contact.setphoneNumber(cursor.getString(2));  
 *// Add student to list* contacts.add(contact);  
 } **while** (cursor.moveToNext());  
 }  
 **return** contacts;  
 }  
  
 *// Updating single contact* **public int** updateContact(Contact contact) {  
 SQLiteDatabase db = **this**.getWritableDatabase();  
 ContentValues values = **new** ContentValues();  
 values.put(***KEY\_NAME***, contact.getName());  
 values.put(***KEY\_PH\_NO***, contact.getphoneNumber());  
 String whereClause = ***KEY\_ID*** + **"=?"**;  
 String[] whereArgs = {String.*valueOf*(contact.getId())};  
 **return** db.update(***TABLE\_CONTACTS***, values, whereClause, whereArgs);  
 }  
  
 *// Deleting single contact* **public void** deleteContact(Contact contact) {  
 SQLiteDatabase db = **this**.getWritableDatabase();  
 String whereClause = **"\_id=?"**;  
 String[] whereArgs = {String.*valueOf*(contact.getId())};  
 db.delete(***TABLE\_CONTACTS***, whereClause, whereArgs);  
 }  
}

Trong MainActivity.java

**package** com.example.lengoc.lab3\_02\_bai3;  
  
**import** android.app.Activity;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.AdapterView;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.Button;  
**import** android.widget.ListView;  
**import** android.widget.TextView;  
**import** java.util.ArrayList;  
**import** java.util.List;  
**import** com.example.lengoc.lab3\_02\_bai3.DatabaseHandler;  
**import** com.example.lengoc.lab3\_02\_bai3.Contact;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 ArrayList<String>**arrList**=**null**;  
 ArrayAdapter<String> **adapter**=**null**;  
 ListView **av**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 *// View by Id* **av** = (ListView) findViewById(R.id.***lv\_contacts***);  
*// Init database* DatabaseHandler db = **new** DatabaseHandler(**this**);  
  
 *// Add students* addContacts(db);  
  
 *//Update Students* updateContacts(db);  
  
 *// Show students* showContacts(db);  
 }  
  
 **private void** addContacts(DatabaseHandler db) {  
 db.addContact(**new** Contact(1, **"An"**, **"MMTT2011"**));  
 db.addContact(**new** Contact(2, **"Bình"**, **"MMTT2012"**));  
 db.addContact(**new** Contact(3, **"Châu"**, **"MMTT2013"**));  
 db.addContact(**new** Contact(4, **"Dương"**, **"MMTT2014"**));  
  
 }  
  
 **private void** updateContacts(DatabaseHandler db) {  
 db.updateContact(**new** Contact(1,**"Ravi"**, **"9100000000"**));  
 db.updateContact(**new** Contact(2,**"Srinivas"**, **"9199999999"**));  
 db.updateContact(**new** Contact(3,**"Tommy"**, **"9522222222"**));  
 db.updateContact(**new** Contact(4,**"Karthik"**, **"9533333333"**));  
  
 }  
  
 **private void** showContacts(DatabaseHandler db) {  
 List<Contact> contacts = db.getAllContacts();  
  
 **arrList**=**new** ArrayList<String>();  
 **adapter**=**new** ArrayAdapter<String>  
 (**this**,  
 android.R.layout.***simple\_list\_item\_1***,  
 **arrList**);  
 **for** (Contact contact : contacts) {  
  
 **arrList**.add(contact.getId() + **" - "** + contact.getName() + **" - "** + contact.getphoneNumber() + **"\n"**) ;  
 }  
  
 **av**.setAdapter(**adapter**);

***// xử lý sự kiện Long click*****av**.setOnItemLongClickListener(**new** AdapterView  
 .OnItemLongClickListener() {  
 @Override  
 **public boolean** onItemLongClick(AdapterView<?> arg0, View arg1,  
 **int** arg2, **long** arg3) {  
 **arrList**.remove(arg2);*//xóa phần tử thứ arg2* **adapter**.notifyDataSetChanged();  
 **return false**;  
 }  
 });  
 }  
  
  
  
}