import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import java.util.ArrayList;

import android.annotation.TargetApi;

import android.app.Activity;

import android.app.ProgressDialog;

import android.content.ContentResolver;

import android.content.pm.PackageManager;

import android.database.Cursor;

import android.net.Uri;

import android.os.Build;

import android.os.Bundle;

import android.os.Handler;

import android.provider.ContactsContract;

import android.util.Log;

import android.view.View;

import android.widget.AdapterView;

import android.widget.AdapterView.OnItemClickListener;

import android.widget.ArrayAdapter;

import android.widget.ListView;

import android.widget.Toast;

import static android.Manifest.permission.READ\_CONTACTS;

public class MainActivity extends AppCompatActivity {

private static final int REQUEST\_READ\_CONTACTS = 444;

private ListView mListView;

private ProgressDialog pDialog;

private Handler updateBarHandler;

ArrayList<String> contactList;

Cursor cursor;

int counter;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

pDialog = new ProgressDialog(MainActivity.this);

pDialog.setMessage("Reading contacts...");

pDialog.setCancelable(false);

pDialog.show();

mListView = (ListView) findViewById(R.id.list);

updateBarHandler = new Handler();

// Since reading contacts takes more time, let's run it on a separate thread.

new Thread(new Runnable() {

@Override

public void run() {

getContacts();

}

}).start();

// Set onclicklistener to the list item.

mListView.setOnItemClickListener(new OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> parent, View view,

int position, long id) {

//TODO Do whatever you want with the list data

Toast.makeText(getApplicationContext(), "item clicked : \n" + contactList.get(position), Toast.LENGTH\_SHORT).show();

}

});

}

private boolean mayRequestContacts() {

if (Build.VERSION.SDK\_INT < Build.VERSION\_CODES.M) {

return true;

}

if (checkSelfPermission(READ\_CONTACTS) == PackageManager.PERMISSION\_GRANTED) {

return true;

}

if (shouldShowRequestPermissionRationale(READ\_CONTACTS)) {

requestPermissions(new String[]{READ\_CONTACTS}, REQUEST\_READ\_CONTACTS);

} else {

requestPermissions(new String[]{READ\_CONTACTS}, REQUEST\_READ\_CONTACTS);

}

return false;

}

/\*\*

\* Callback received when a permissions request has been completed.

\*/

@Override

public void onRequestPermissionsResult(int requestCode, String[] permissions,

int[] grantResults) {

if (requestCode == REQUEST\_READ\_CONTACTS) {

if (grantResults.length == 1 && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {

getContacts();

}

}

}

public void getContacts() {

if (!mayRequestContacts()) {

return;

}

contactList = new ArrayList<String>();

String phoneNumber = null;

// String email = null;

Uri CONTENT\_URI = ContactsContract.Contacts.CONTENT\_URI;

String \_ID = ContactsContract.Contacts.\_ID;

String DISPLAY\_NAME = ContactsContract.Contacts.DISPLAY\_NAME;

String HAS\_PHONE\_NUMBER = ContactsContract.Contacts.HAS\_PHONE\_NUMBER;

Uri PhoneCONTENT\_URI = ContactsContract.CommonDataKinds.Phone.CONTENT\_URI;

String Phone\_CONTACT\_ID = ContactsContract.CommonDataKinds.Phone.CONTACT\_ID;

String NUMBER = ContactsContract.CommonDataKinds.Phone.NUMBER;

/\* Uri EmailCONTENT\_URI = ContactsContract.CommonDataKinds.Email.CONTENT\_URI;

String EmailCONTACT\_ID = ContactsContract.CommonDataKinds.Email.CONTACT\_ID;

String DATA = ContactsContract.CommonDataKinds.Email.DATA;\*/

StringBuffer output;

ContentResolver contentResolver = getContentResolver();

cursor = contentResolver.query(CONTENT\_URI, null, null, null, null);

// Iterate every contact in the phone

if (cursor.getCount() > 0) {

counter = 0;

while (cursor.moveToNext()) {

output = new StringBuffer();

// Update the progress message

updateBarHandler.post(new Runnable() {

public void run() {

pDialog.setMessage("Reading contacts : " + counter++ + "/" + cursor.getCount());

}

});

String contact\_id = cursor.getString(cursor.getColumnIndex(\_ID));

String name = cursor.getString(cursor.getColumnIndex(DISPLAY\_NAME));

int hasPhoneNumber = Integer.parseInt(cursor.getString(cursor.getColumnIndex(HAS\_PHONE\_NUMBER)));

if (hasPhoneNumber > 0) {

output.append("\n First Name:" + name);

//This is to read multiple phone numbers associated with the same contact

Cursor phoneCursor = contentResolver.query(PhoneCONTENT\_URI, null, Phone\_CONTACT\_ID + " = ?", new String[]{contact\_id}, null);

while (phoneCursor.moveToNext()) {

phoneNumber = phoneCursor.getString(phoneCursor.getColumnIndex(NUMBER));

output.append("\n Phone number:" + phoneNumber);

}

phoneCursor.close();

/\* // Read every email id associated with the contact

Cursor emailCursor = contentResolver.query(EmailCONTENT\_URI, null, EmailCONTACT\_ID + " = ?", new String[]{contact\_id}, null);

while (emailCursor.moveToNext()) {

email = emailCursor.getString(emailCursor.getColumnIndex(DATA));

output.append("\n Email:" + email);

}

emailCursor.close();

String columns[] = {

ContactsContract.CommonDataKinds.Event.START\_DATE,

ContactsContract.CommonDataKinds.Event.TYPE,

ContactsContract.CommonDataKinds.Event.MIMETYPE,

};\*/

String where = ContactsContract.CommonDataKinds.Event.TYPE + "=" + ContactsContract.CommonDataKinds.Event.TYPE\_BIRTHDAY +

" and " + ContactsContract.CommonDataKinds.Event.MIMETYPE + " = '" + ContactsContract.CommonDataKinds.Event.CONTENT\_ITEM\_TYPE + "' and " + ContactsContract.Data.CONTACT\_ID + " = " + contact\_id;

String[] selectionArgs = null;

String sortOrder = ContactsContract.Contacts.DISPLAY\_NAME;

/\*Cursor birthdayCur = contentResolver.query(ContactsContract.Data.CONTENT\_URI, columns, where, selectionArgs, sortOrder);

Log.d("BDAY", birthdayCur.getCount()+"");

if (birthdayCur.getCount() > 0) {

while (birthdayCur.moveToNext()) {

String birthday = birthdayCur.getString(birthdayCur.getColumnIndex(ContactsContract.CommonDataKinds.Event.START\_DATE));

output.append("Birthday :" + birthday);

Log.d("BDAY", birthday);

}

}

birthdayCur.close();

}\*/

// Add the contact to the ArrayList

contactList.add(output.toString());

}

// ListView has to be updated using a ui thread

runOnUiThread(new Runnable() {

@Override

public void run() {

ArrayAdapter<String> adapter = new ArrayAdapter<String>(getApplicationContext(), R.layout.contact\_text, R.id.text1, contactList);

mListView.setAdapter(adapter);

}

});

// Dismiss the progressbar after 500 millisecondds

updateBarHandler.postDelayed(new Runnable() {

@Override

public void run() {

pDialog.cancel();

}

}, 500);

}

}

}

}