package org.intracode.contactmanager;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.net.Uri;

import java.util.ArrayList;

import java.util.List;

/\*\*

\* Created by Johnny Manson on 19.01.14.

\*/

public class DatabaseHandler extends SQLiteOpenHelper {

private static final int DATABASE\_VERSION = 1;

private static final String DATABASE\_NAME = "contactManager",

TABLE\_CONTACTS = "contacts",

KEY\_ID = "id",

KEY\_NAME = "name",

KEY\_PHONE = "phone",

KEY\_EMAIL = "email",

KEY\_ADDRESS = "address",

KEY\_IMAGEURI = "imageUri";

public DatabaseHandler(Context context) {

super(context, DATABASE\_NAME, null, DATABASE\_VERSION);

}

@Override

public void onCreate(SQLiteDatabase db) {

db.execSQL("CREATE TABLE " + TABLE\_CONTACTS + "(" + KEY\_ID + " INTEGER PRIMARY KEY AUTOINCREMENT," + KEY\_NAME + " TEXT," + KEY\_PHONE + " TEXT," + KEY\_EMAIL + " TEXT," + KEY\_ADDRESS + " TEXT," + KEY\_IMAGEURI + " TEXT)");

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

db.execSQL("DROP TABLE IF EXISTS " + TABLE\_CONTACTS);

onCreate(db);

}

public void createContact(Contact contact) {

SQLiteDatabase db = getWritableDatabase();

ContentValues values = new ContentValues();

values.put(KEY\_NAME, contact.getName());

values.put(KEY\_PHONE, contact.getPhone());

values.put(KEY\_EMAIL, contact.getEmail());

values.put(KEY\_ADDRESS, contact.getAddress());

values.put(KEY\_IMAGEURI, contact.getImageURI().toString());

db.insert(TABLE\_CONTACTS, null, values);

db.close();

}

public Contact getContact(int id) {

SQLiteDatabase db = getReadableDatabase();

Cursor cursor = db.query(TABLE\_CONTACTS, new String[] { KEY\_ID, KEY\_NAME, KEY\_PHONE, KEY\_EMAIL, KEY\_ADDRESS, KEY\_IMAGEURI }, KEY\_ID + "=?", new String[] { String.valueOf(id) }, null, null, nul

if (cursor != null)

cursor.moveToFirst();

Contact contact = new Contact(Integer.parseInt(cursor.getString(0)), cursor.getString(1), cursor.getString(2), cursor.getString(3), cursor.getString(4), Uri.parse(cursor.getString(5)));

db.close();

cursor.close();

return contact;

}

public void deleteContact(Contact contact) {

SQLiteDatabase db = getWritableDatabase();

db.delete(TABLE\_CONTACTS, KEY\_ID + "=?", new String[] { String.valueOf(contact.getId()) });

db.close();

}

public int getContactsCount() {

SQLiteDatabase db = getReadableDatabase();

Cursor cursor = db.rawQuery("SELECT \* FROM " + TABLE\_CONTACTS, null);

int count = cursor.getCount();

db.close();

cursor.close();

return count;

}

public int updateContact(Contact contact) {

SQLiteDatabase db = getWritableDatabase();

ContentValues values = new ContentValues();

values.put(KEY\_NAME, contact.getName());

values.put(KEY\_PHONE, contact.getPhone());

values.put(KEY\_EMAIL, contact.getEmail());

values.put(KEY\_ADDRESS, contact.getAddress());

values.put(KEY\_IMAGEURI, contact.getImageURI().toString());

int rowsAffected = db.update(TABLE\_CONTACTS, values, KEY\_ID + "=?", new String[] { String.valueOf(contact.getId()) });

db.close();

return rowsAffected;

}

public List<Contact> getAllContacts() {

List<Contact> contacts = new ArrayList<Contact>();

SQLiteDatabase db = getWritableDatabase();

Cursor cursor = db.rawQuery("SELECT \* FROM " + TABLE\_CONTACTS, null);

if (cursor.moveToFirst()) {

do {

contacts.add(new Contact(Integer.parseInt(cursor.getString(0)), cursor.getString(1), cursor.getString(2), cursor.getString(3), cursor.getString(4), Uri.parse(cursor.getString(5))));

}

while (cursor.moveToNext());

}

cursor.close();

db.close();

return contacts;

}

}