

《SQLite 数据库 app》说明文档

软件名称：通讯录 KSJ

完成人：姜淑晶

学号：20152100061

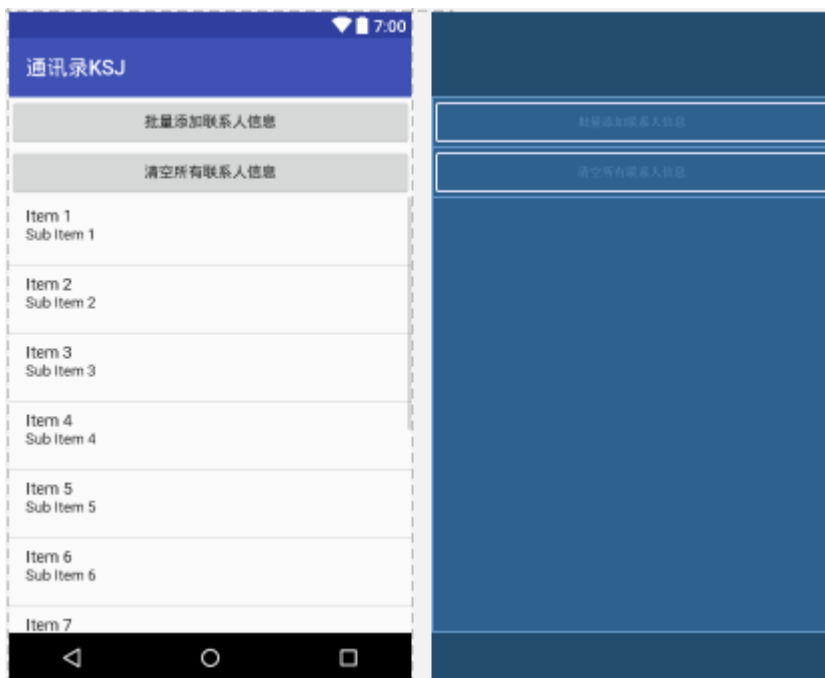
完成时间：2017-11-29

软件内容简介：

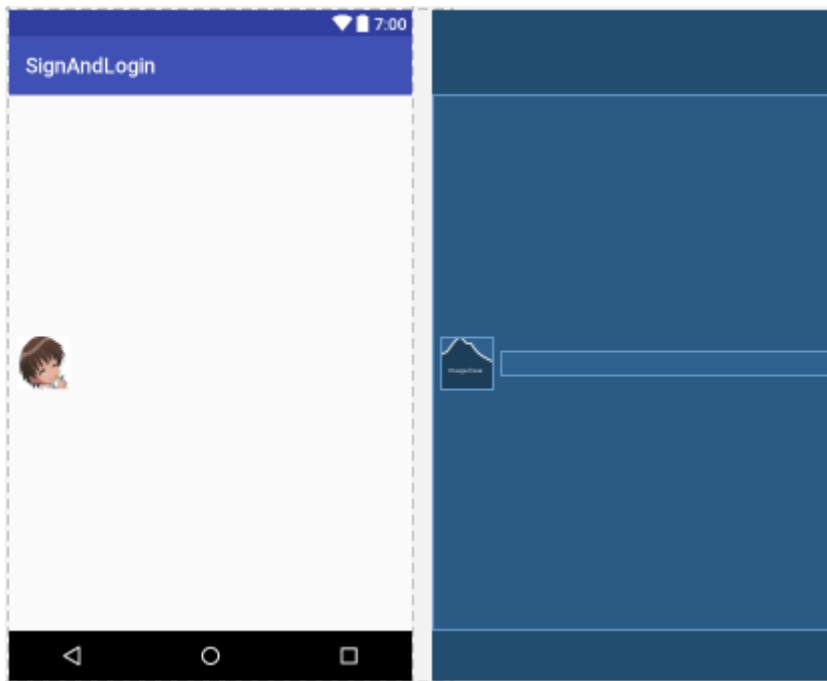
实现向 SQLite 数据库中批量添加数据，并通过 ListView 显示添加后的数据，以通讯录形式展示。

界面设计：

主界面布局：



列表项布局：



真机效果



初始界面，將數據表中原有的數據加載顯示。



点击“批量添加联系人”按钮，显示弹出框，由用户进行数据输入。



成功添加后，自动更新列表显示效果。



点击“清空所有联系人信息”按钮后，列表所示与后台数据表的所有内容都将清空。

代码设计：

详细代码说明：

页面布局 xml 文件代码：

主界面布局：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.kangsekjung.signandlogin.AddressBook">

    <Button
```

```

        android:id="@+id/addinfo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="批量添加联系人信息" />

<Button
    android:id="@+id/clearinfo"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="清空所有联系人信息"/>

<ListView
    android:id="@+id/lv"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:divider="#d9d9d9"
    android:dividerHeight="1dp">

</ListView>

</LinearLayout>

```

列表项布局:

```

<?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:gravity="center_vertical">

    <ImageView
        android:id="@+id/item_iv"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:layout_margin="8dp"
        android:background="@drawable/im1" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/item_tv"
        android:textSize="20sp" />

</LinearLayout>

```

软件功能实现 java 文件:

数据库创建类文件:

```

package com.kangsekjung.signandlogin;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

/**
 * 数据库的创建类
 * Created by jiangjing on 2017/11/27.
 */

```

```

public class TelNumberHelper extends SQLiteOpenHelper {

    public TelNumberHelper(Context context) {
        super(context, "TelNumber.db", null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        // db.execSQL("CREATE TABLE USERTELNUM(ID INTEGER PRIMARY KEY AUTOINCREMENT, NAME
        VARCHAR(20), TELNUM VARCHAR(15))");
        //创建表 USERTELNUM, 字段: 姓名 NAME, 电话号码 TELNUM
        db.execSQL("CREATE TABLE USERTELNUM(NAME VARCHAR(20), TELNUM VARCHAR(15))");
    }

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

    }

}

```

主界面 java 文件:

```

package com.kangsekjung.signandlogin;

import android.content.ContentValues;
import android.content.DialogInterface;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteStatement;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.ContentFrameLayout;
import android.util.Log;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import java.util.Vector;

public class AddressBook extends AppCompatActivity {

    Button add;
    Button clearinfo;
    ListView lv;
    Vector<String> AddrBook; //记录通讯录信息
    TelNumberHelper mHelper; //数据库变量

```



```

MyBaseAdapter mAdapterer; //ListView 适配器

private void Init() {
    add = (Button) findViewById(R.id.addinfo);
    lv = (ListView) findViewById(R.id.lv);

    AddrBook = new Vector<>();
    mHelper = new TelNumberHelper(this);
    clearinfo = (Button) findViewById(R.id.clearinfo);
}

private void setListener() {
    add.setOnClickListener(new View.OnClickListener() {
        EditText input;

        @Override
        public void onClick(View v) {
            //创建带输入框的弹出框
            input = new EditText(AddressBook.this);
            input.setFocusable(true);

            AlertDialog.Builder builder = new
AlertDialog.Builder(AddressBook.this);
            builder.setTitle("输入联系人信息")
                .setMessage("请输入联系人的姓名与手机号码并用空格隔开两
者。" +
                            "若想添加多个联系人,请用分号隔开各联系人的信息。
如: xxx 12345678910;yyy 12333333333"
                            + "注意: 均使用英文字符")
                .setIcon(android.R.drawable.ic_dialog_info).setView(nul
l).setView(input)
                .setNegativeButton("Cancel", null);

            builder.setPositiveButton("Ok", new
DialogInterface.OnClickListener() {
                //点击 ok 键的响应事件
                public void onClick(DialogInterface dialog, int which) {
                    //获取用户的输入并把数据添加到数据库的表中
                    insert(input.getText().toString());
                    //更新联系人信息列表
                    updateList();
                    //mAdapter.notifyDataSetChanged();
                    //更新 ListView 的显示
                    mAdapterer = new MyBaseAdapter();
                    lv.setAdapter(mAdapterer);
                }
            });
            builder.show();
        }
    });
}

```

```

    }
});

clearinfo.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //清空数据表
        SQLiteDatabase db = mHelper.getWritableDatabase();
        db.execSQL("DELETE FROM USERTELENUM");
        db.close();
        //更新显示联系人信息
        updateList();
        mAdapter = new MyBaseAdapter();
        lv.setAdapter(mAdapter);
    }
});
}

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

    Init();
    setListener();
    //初始化 ListView 内容
    updateList();
    mAdapter = new MyBaseAdapter();
    lv.setAdapter(mAdapter);
}

/**
 * 在数据表中数据更新到联系人信息存储结构中
 */
private void updateList() {
    AddrBook.clear();
    SQLiteDatabase db = mHelper.getReadableDatabase();
    //查询所有信息
    Cursor cursor = db.rawQuery("select * from USERTELENUM", null);
    while (cursor.moveToNext()) {
        //提取表中的第一个字段 (NAME)
        String name = cursor.getString(0);
        //提取表中的第一个字段 (TELENUM)
        String tel = cursor.getString(1);
        //把信息添加到存储列表中
        AddrBook.add(name + " : " + tel);
    }

    cursor.close();
}

```

```

        db.close();
    }

    /**
     * 将输入数据插入到数据表中
     *
     * @param str 用户输入的数据
     */
    private void insert(String str) {
        //输入非空则进行数据插入
        if (!str.equals("")) {
            String[] linkmans = str.split(";");//分割联系人

            SQLiteDatabase db = mHelper.getWritableDatabase();

            ContentValues values;
            for (String s : linkmans) {
                String[] info = s.split(" "); //分割具体信息

                //插入信息
                values = new ContentValues();
                values.put("NAME", info[0]);
                values.put("TELNUM", info[1]);
                db.insert("USERTELENUM", null, values);
            }
        }
    }

    class MyBaseAdapter extends BaseAdapter {
        @Override
        public int getCount() {
            return AddrBook.size();
        }

        @Override
        public Object getItem(int pos) {
            return AddrBook.get(pos);
        }

        @Override
        public long getItemId(int pos) {
            return pos;
        }

        @Override
        public View getView(int pos, View convertView, ViewGroup parent) {
            //将设计好的列表项布局转换成 view 对象
            View view = View.inflate(AddressBook.this, R.layout.list_item, null);
            //显示联系人信息

```

```

        TextView tv = (TextView) view.findViewById(R.id.item_tv);
        tv.setText(AddrBook.get(pos));
        //显示头像
        ImageView iv = (ImageView) view.findViewById(R.id.item_iv);
        iv.setBackgroundResource(R.drawable.im1);

        return view;
    }
}

```

软件操作流程：

中國移動 中國聯通 65B/s 2G 4G 85% 16:32

通訊錄KSJ

批量添加聯系人信息

清空所有聯系人信息

sek : 15625049399

jung : 13560233333

路人甲 : 15625042333

路人乙 : 13523333333

路人丙 : 15623333333

路人丁 : 12345678910

点击按钮将弹出输入框, 进行数据录入

点击按钮将清空所有数据(包括数据库表中的数据)

列表区域, 显示数据表里的所有数据

难点（或遇到的问题）：

1. 调用 `ListView.setAdapter()` 造成页面崩溃；

2. 利用带输入框的弹出框做信息输入的接口，第一次点击可正常使用，但第二次点击按钮后，发生页面崩溃。

解决方案:

1. 在定义 BaseAdapter 子类时,重载 getView() 中使用了 findViewById(R.id.xxxx) 获取控件,使得没正确获取到对应控件。原因: 在一个 mainactivity 活动里有一个 Gridview,在 gridView 里有我们的控件,那么我们应当使用 gridView.findViewById(R.id.xxxx);来查找 id,因为这个布局文件加载后返回给了 gridView,而不是 MainActivity。而在使用自定义的 BaseAdapter 子类时,列表项的控件在另一个 xml 文件中,而不是主活动所对应的 xml 文件,因此在用 inflate 函数将列表项的 xml 文件转换成 view 对象后,应利用该对象进行控件获取。应该为:

```
TextView tv = (TextView) view.findViewById(R.id.item_tv);
ImageView iv = (ImageView) view.findViewById(R.id.item_iv);
```

2. 原因: 一开始将输入框 (view) 设置为 ‘全局变量’ (在 onClick 函数外创建对象), 在第一次创建对话框的时候, 已经绑定了一个 builder, 所以在第二次点击对话框的时候, 再次绑定 builder 的时候, 新的 builder 就不会接收我们自定义的 view, 因为它认为你已经绑定过了, 此时就算你将之前的 builder 在对话框消失的同时销毁掉也是没用的, 因为我们的 view 仍然会有绑定的过往记录。因此, 需要将 view (输入框) 变成局部变量。先定义一个未赋值的 EditText 变量, 在 onClick 函数中才进行新建对象与赋值。

从

```
EditText input=new EditText(AddressBook.this);
public void onClick(View v) {.....}
```

改为:

```
EditText input;
public void onClick(View v) {
    input = new EditText(AddressBook.this);.....}
```

不足之处:

1. 没有对用户的输入做格式检查与错误处理, 如果用户没有按照提示正确输入数据, 在进行插入操作时将发生错误, 使页面崩溃。
2. 没有为删除按钮提供二次确认的机会, 只要点击就直接连同数据表中的所有数据一并删除。

今后的设想:

1. 添加格式检查与错误处理;
2. 为删除按钮提供二次确认的机会;
3. 实现点击列表项就可以向对应记录拨打出电话。

参考资料:

1. setAdapter 函数问题: <https://www.cnblogs.com/lj-103/p/5531800.html>
2. 第二次点击崩溃问题: <https://www.2cto.com/kf/201406/312052.html>