

# 移动应用开发课程作业

## — 实现向 SQLite 数据库中批量添加数据

指导老师：曹阳

2017 年 12 月 5 日

华南师范大学计算机学院

计算机科学与技术（软件技术应用方向）

2015 级 5 班

王子彦

20142100012

[ziyan.wang@m.scnu.edu.cn](mailto:ziyan.wang@m.scnu.edu.cn)

**声明：**除特殊注明外，本次作业中所有的文档、源代码和相关资源均为本人原创。

该项目的源代码已在 Github 上以 Apache-2.0 协议发布：

<https://github.com/lonelyenvoy/ContactsDemo>

王子彦授予曹阳老师对本项目所有文件的查阅、拷贝、修改和使用的权利。

软件名称	ContactsDemo
完成人	王子彦
学号	20142100012
完成时间	2017 年 12 月 5 日

### 一、 软件内容简介

本次作业实现了一个通讯录 app，该通讯录遵循 Google Material Design 设计方法，UI 界面较美观。具备查看联系人、批量添加联系人、对联系人进行排序显示，和在数据表中执行用户输入的 SQL 命令功能。配合正则表达式对用户的输入进行验证和多种异常处理机制，app 具有良好的健壮性以及优越的用户体验。

您可在[此处](#)查阅本项目的源码。

## 二、界面设计

本次 app 的设计只使用了一个界面，遵循谷歌 Material Design 的设计方法，减少页面层级数，优化用户体验。用户可在该界面上完成通讯录所需的基本操作。

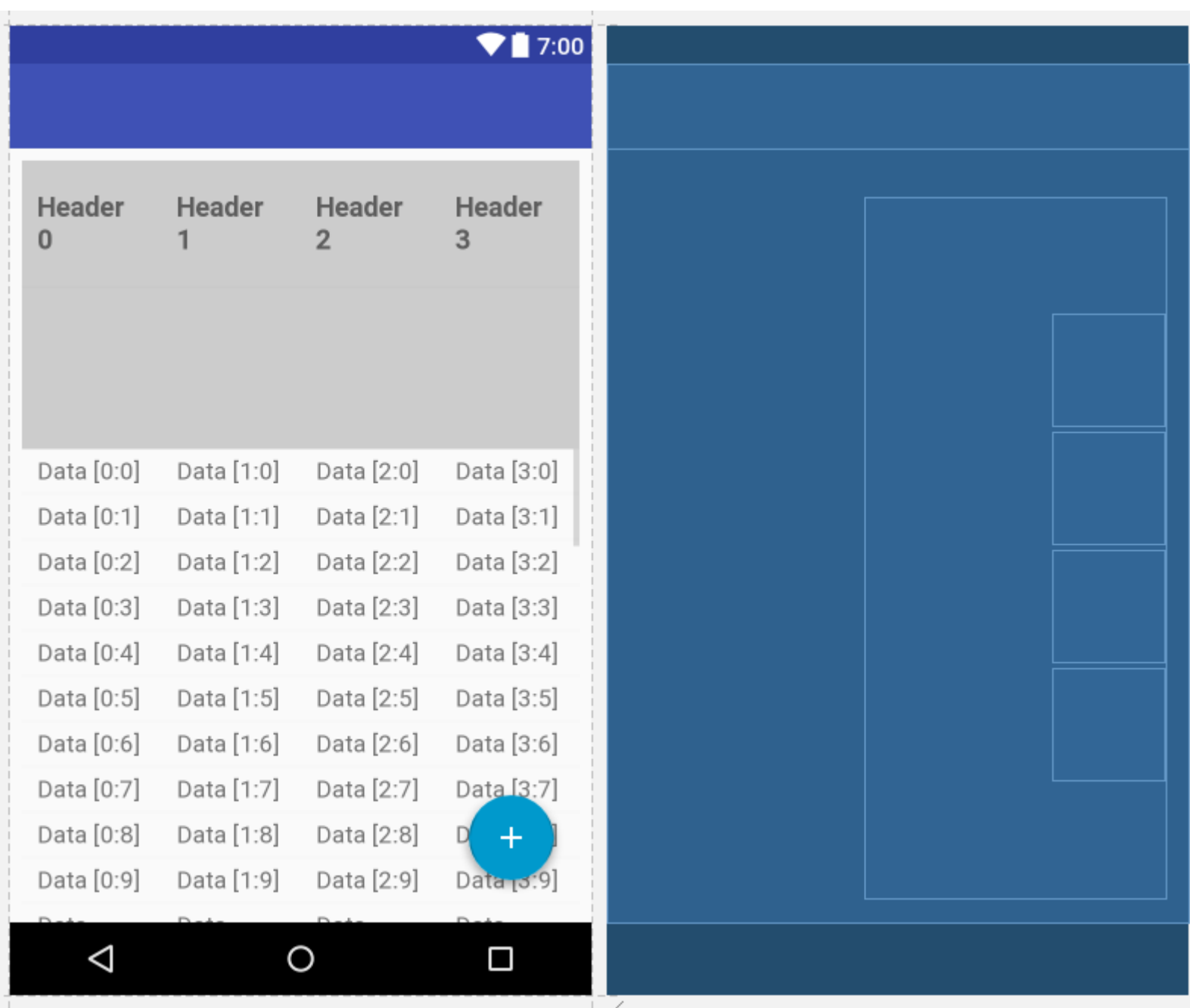


图 2-1 Android Studio 界面设计原型图



图 2-2 App 真机运行截图

为了您能更直观地了解本次作业的成果，请查看  
同目录下的“[app 完整功能演示图](#)”动画

(约 60 秒)

### 三、项目设计中的亮点

#### 3.1 通讯录列表可按任意项排序

仅需点击通讯录的表头，即可以某一项为基准对列表进行升序或降序排列。如图为按邮箱升序排列的结果。



#	姓名	手机	邮箱
1	子彦	13609046213	admin@envoy.ink
3	小林	18026342612	lin@126.com
2	张华	15361527951	zhang@qq.com

图 3-1 通讯录列表排序显示

### 3.2 基于 Snackbar 的友好的用户提示信息

在 app 中的大部分操作完成后，都会在屏幕底部显示 Snackbar 给用户良好的反馈信息。同时，右下角的 Material Design “+”按钮会隐藏，动画效果良好，提高了用户体验。



图 3-2 在屏幕底部用 Snackbar 显示提示信息

### 3.3 批量添加联系人

App 制定了一个完善的语法来引导用户输入批量添加的联系人信息，用正则表达式进行正确性校验，用户能够轻松输入大量的信息，同时在用户输入有误时也能给出良好的反馈。



图 3-3 批量添加联系人

### 3.4 用户能够直接执行 SQL 语句

在本次 app 的设计中加入了一个特别板块，用户可以手工输入 SQL 语句并执行，具有很高的自由度。同时，在完善的 SQL 检查与异常处理机制下，用户无须担心输错命令，对数据库中已存储的数据造成负面影响。



图 3-8 SQL 语句输入界面



### 3.5 右下角圆点按钮的细节优化

在屏幕显示大量数据时，屏幕右下角的“+”按钮会遮盖用户需要的数据，这正是 Material Design 的一个被诟病的问题，但开发者已经解决。当屏幕上的数据超出一个屏幕能够显示的范围时，用户如果向下滑动屏幕，则圆形按钮会自动隐藏；当用户需要这个按钮的时候，只需要向上反方向滑动，按钮就会重新展示。（具体实现过程详见代码）

## 四、代码设计

由于代码量较大，下面选择几项重要的部分展示。如有兴趣，请在 Android Studio 中 clone 开发者的 Github 代码仓库，打开项目文件进行详细查阅。

```

<ink.envoy.contactsdemo.view.HiddenFloatingActionsMenu
    android:id="@+id/actionsMenu"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_marginBottom="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginRight="16dp"
    android:layout_gravity="bottom|end"
    android:layout_margin="16dp"
    app:fab_labelStyle="@style/menu_labels_style"
    app:fab_labelsPosition="left">

    <com.getbase.floatingactionbutton.FloatingActionButton
        android:id="@+id/sqlActionButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:fab_icon="@mipmap/ic_sql"
        app:fab_colorNormal="@color/colorAccent"
        app:fab_colorPressed="@color/colorAccentLight"
        app:fab_title="执行SQL命令" />

    <com.getbase.floatingactionbutton.FloatingActionButton
        android:id="@+id/clearActionButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:fab_icon="@mipmap/ic_delete"
        app:fab_title="清空" />

    <com.getbase.floatingactionbutton.FloatingActionButton
        android:id="@+id/multiAddActionButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:fab_icon="@mipmap/ic_multi_add"
        app:fab_title="批量添加" />

    <com.getbase.floatingactionbutton.FloatingActionButton
        android:id="@+id/addActionButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:fab_icon="@mipmap/ic_add"
        app:fab_title="添加" />

</ink.envoy.contactsdemo.view.HiddenFloatingActionsMenu>

```

图 4-1 主界面 FloatingActionButton 布局

```

public class HiddenFloatingActionsMenu extends FloatingActionsMenu {

    private boolean isShown = true;
    private int ANIM_DURATION = 300;
    private boolean mVisible = false;

    public HiddenFloatingActionsMenu(Context context) { super(context); }

    public HiddenFloatingActionsMenu(Context context, AttributeSet attrs) { super(context, attrs); }

    public HiddenFloatingActionsMenu(Context context, AttributeSet attrs, int defStyleAttr) {
        super(context, attrs, defStyleAttr);
    }

    public void show(boolean isVisible) {
        mVisible = isVisible;
        int translationX = isVisible ? 0 : (getWidth()/2) + getMarginRight();
        this.animate().translationX(translationX).setDuration(ANIM_DURATION).start();
    }

    public boolean isShown() { return isShown; }

    private int getMarginRight() {
        int marginBottom = 0;
        final ViewGroup.LayoutParams layoutParams = getLayoutParams();
        if (layoutParams instanceof ViewGroup.MarginLayoutParams) {
            marginBottom = ((MarginLayoutParams) layoutParams).rightMargin;
        }
        return marginBottom;
    }

    public boolean getVisible() { return mVisible; }

}

```

图 4-2 开发者设计的可隐藏式 ActionsMenu，用于解决按钮遮盖主界面内容的问题

```

// parse user input into readable data
String rawContactInfos = dialog.getInputEditText().getText().toString();
final Pattern pattern = Pattern.compile("\\s?(\\S+)\\s(\\S+)\\s([^;\\s]+)\\s?;?");
Matcher matcher = pattern.matcher(rawContactInfos);

// retrieve contact inf
boolean ok = true;
List<String[]> contactInfos = new LinkedList<>();
while (matcher.find()) {
    if (matcher.groupCount() != 3) {
        ok = false;
        break;
    }
    contactInfos.add(new String[] {matcher.group(1), matcher.group(2), matcher.group(3)});
}
if (ok && !contactInfos.isEmpty()) {
    // save to DB
    new ContactsDataAccessor(getApplicationContext()).put(contactInfos);
    // refresh
    loadData();
    // feedback
    showTextOnSnackbar(view, actionsMenu, "添加成功");
} else {
    showTextOnSnackbar(view, actionsMenu, "输入有误, 添加失败");
}

```

图 4-3 批量添加联系人的核心算法

```

private void showTextOnSnackbar(View view, final HiddenFloatingActionsMenu actionsMenu,
                                String text) {
    // toggle actions menu button
    actionsMenu.show(false);
    new Handler().postDelayed(new Runnable() {
        @Override
        public void run() {
            actionsMenu.show(true);
        }
    }, SNACKBAR_SHOW_TIME + 500);

    // show successful feedback
    Snackbar.make(view, text, Snackbar.LENGTH_INDEFINITE).setDuration(SNACKBAR_SHOW_TIME)
        .setAction(R.string.ok, new View.OnClickListener() {
            @Override
            public void onClick(View view) {}
        }).show();
}

```

图 4-4 在屏幕底部用 Snackbar 显示提示的工具函数

## 五、软件操作流程

软件启动后，用户可以看到在 app 内已存储的联系人信息，长按某条信息可以将其删除。点击右下角的圆形“+”按钮，可以看到四个功能，分别是添加单个联系人、批量添加联系人、清空列表和执行 SQL 命令。用户可按提示操作，大部分操作完成后都会在屏幕底部显示提示信息。

## 六、难点和解决方案

本次项目引用了较多第三方库，如何解决不同的包带来的兼容性问题是一大难点。Gradle 自动化编译已经为我们解决了大部分依赖和兼容性问题，但由于每个第三方包对安卓系统的 Target Version 不可能完全一致，在不一致时会出现 Gradle 无法正常执行的问题。经过分析，得出以下解决方案：

在 App Module 下的 build.gradle 文件加入以下代码，其中 25.3.0 位项目第三方包中最低版本的 Target Version，即可正常编译运行。



```
configurations.all {
    resolutionStrategy.eachDependency { DependencyResolveDetails details ->
        def requested = details.requested
        if (requested.group == 'com.android.support') {
            if (!requested.name.startsWith("multidex")) {
                details.useVersion '25.3.0' // the lowest version of third-party packages
            }
        }
    }
}
```

图 6 解决兼容性问题的 Gradle 源码

## 七、不足之处和今后的设想

本次项目历时较短，未能投入足够多的时间设计和开发 app，虽然界面相对美观，但功能较简陋，只完成了通讯录所需的基本功能，且不符合 I18N 的规范。  
将来可在此 app 的基础上进一步增加功能，开发有实用价值的产品。

## 附录

### 参考文献

《Android 移动开发基础案例教程》，黑马程序员编著，2017，人民邮电出版社