

手机平台应用开发实验报告

学号: 14331098

班级: 周五下午班

姓名: 黄建武

实验名称: Lab8

一、参考资料

1. [Android官方开发者文档](#)
2. [DatePicker只显示月日不显示年](#)
3. [解决AlertDialog造成程序崩溃的问题](#)
4. [通过名字获取联系人号码](#)

二、实验环境

- 系统: Windows 10 64位
- 工具: Android Studio 2.1.3
- JDK版本: 1.8
- minSdk: 19
- targetSdk: 24

三、实验目的

1. 学习 SQLite 数据库的使用；
2. 学习 ContentProvider 的使用；
3. 复习 Android 界面编程。

四、实验步骤

1. 阅读实验要求文档以及对应的课件PPT。
2. 运行Demo程序，了解实验要求细节。
3. 首先实现主界面布局，页面整体使用一个垂直布局LinearLayout，上方按钮水平居中，按钮下方是一个水平布局的LinearLayout，三个TextView的宽度权重为1:1:2，中间的分割线是一个ImageView控件，最下面是一个ListView，高度权重为1。
4. 实现添加生日信息页面，页面整体使用一个垂直布局LinearLayout，中间每一行都是一个水平布局的LinearLayout，包含一个TextView和一个EditView，生日那里做了改进，使用了DatePicker，并且隐藏年份只显示月日。
5. 实现主页面和次页面的页面数据传递，给ListView增加适配器，点击“增加条目”按钮跳转到次页面，编辑生日信息后回传数据给主页面，主页面接受数据后更新ListView。
6. 为ListView添加点击的事件监听器，实现点击条目弹出Dialog，一开始使用的是自定义AlertDialog，但是取消和确定按钮默认样式不好看，改用了自定义的DatePickerDialog，实现弹出的页面也能进行日期选择。
7. 添加数据库操作，将信息存入数据库，使用名字作为主键，当点击增加时就往数据库插入数据，如果产生冲突证明名字重复，插入成功则将信息返回给首页，省去查询名字是否重复的步骤。主页面在每次打开应用时都从数据库中读取整个表，更新ListView，实现数据持久化。单击ListView后更新修改的信息。长按删除对应的条目。
8. 添加读取通讯录的操作，每次点击条目时候，通过名字查询通讯录对应的号码，如果游标不为空，显示电话，否则显示“无”。
9. 之前每一步完成后都有进行测试，程序运行正常才进入下一步。进行最后的调试和完善。

主要代码部分如下：

```
// MyDB.java
public class MyDB extends SQLiteOpenHelper {
```

```

private static final String TABLENAME = "lab8";
private SQLiteDatabase db;

public MyDB(Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {
    super(context, name, factory, version);
    db = getWritableDatabase();
}

@Override
public void onCreate(SQLiteDatabase db) {
    String sql = "create table if not exists " + TABLENAME +
        " (name TEXT PRIMARY KEY, birthMon INTEGER, birthDay INTEGER, gift TEXT)";
    db.execSQL(sql);
}

public long insert2DB(String name, int month, int day, String gift) {
    ContentValues cv = new ContentValues();
    cv.put("name", name);
    cv.put("birthMon", month);
    cv.put("birthDay", day);
    cv.put("gift", gift);
    return db.insert(TABLENAME, null, cv);
}

public void delete(String name) {
    db.delete(TABLENAME, "name = ?", new String[]{name});
}

public void update(String name, int month, int day, String gift) {
    ContentValues cv = new ContentValues();
    cv.put("name", name);
    cv.put("birthMon", month);
    cv.put("birthDay", day);
    cv.put("gift", gift);
    db.update(TABLENAME, cv, "name = ?", new String[]{name});
}

public Cursor getAll() {
    return db.rawQuery("select * from lab8", null);
}

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

```

```

// 进入应用读取数据库信息
listItems = new ArrayList<>();
Cursor cursor = mydb.getAll();
while (cursor.moveToNext()) {
    String name = cursor.getString(0),
        birthMon = String.valueOf(cursor.getInt(1)),
        birthDay = String.valueOf(cursor.getInt(2)),
        gift = cursor.getString(3);
    Vector<String> v = new Vector<>(4);
    v.add(name);
    v.add(birthMon);
    v.add(birthDay);
    v.add(gift);
    listItems.add(v);
}
cursor.close();

```

```

// 隐藏DatePicker的年份
DatePickerDialog dialog = new DatePickerDialog(this, null, 2016, 11, 19);
dialog.setTitle(R.string.dialogtitle);

```

```
View dialogview = getLayoutInflater().inflate(R.layout.dialog, null);
dialog.setView(dialogview);
final DatePicker dp = (DatePicker) dialogview.findViewById(R.id.birthDate);
dp.updateDate(2016, Integer.parseInt(month) - 1, Integer.parseInt(day));
((ViewGroup) ((ViewGroup) dp.getChildAt(0)).getChildAt(0)).getChildAt(0).setVisibility(View.GONE);
```

```
// 根据名字查询联系人号码
Cursor cursor = getContentResolver().query(
    Phone.CONTENT_URI,
    new String[]{Phone.NUMBER},
    Phone.DISPLAY_NAME + "= ?",
    new String[]{name}, null);
String phonenum = "无";
if (cursor != null && cursor.moveToNext()) {
    phonenum = cursor.getString(0);
    cursor.close();
}
phoneTv.setText(phonenum);
```

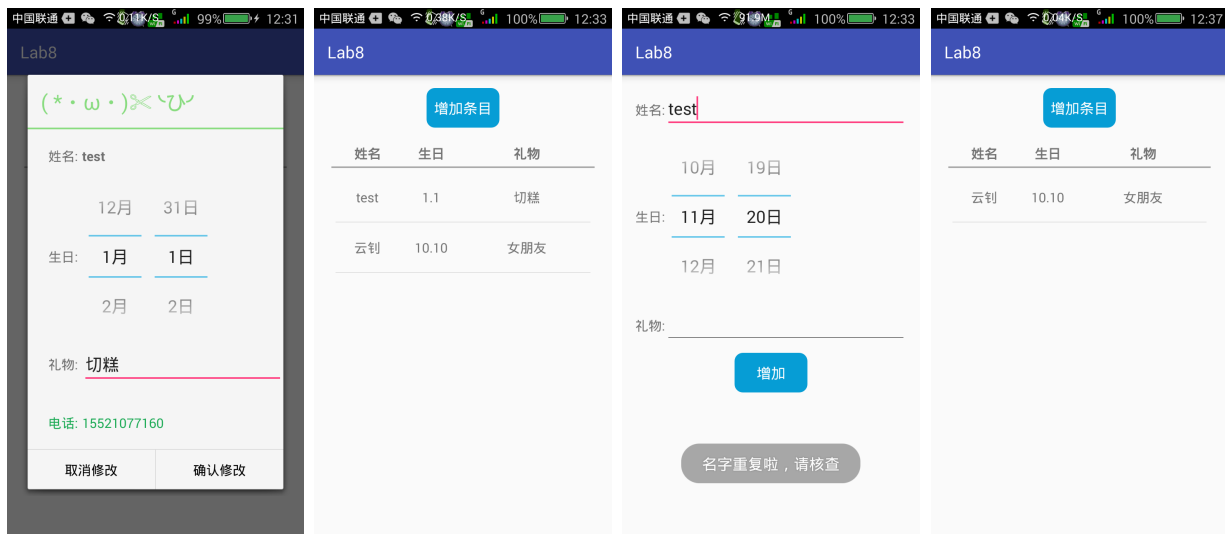
```
// 增加条目
@Override
public void onClick(View v) {
    String name = nameEdit.getText().toString(), gift = giftEdit.getText().toString();
    int month = dp.getMonth() + 1, day = dp.getDayOfMonth();

    if (name.isEmpty()) {
        showToast(R.string.emptyname);
    }
    // 插入失败, 名字重复
    else if (MainActivity.mydb.insert2DB(name, month, day, gift) == -1) {
        showToast(R.string.duplicatename);
    } else {
        Intent intent = new Intent();
        intent.putExtra("name", name)
            .putExtra("birthMon", Integer.toString(month))
            .putExtra("birthDay", Integer.toString(day))
            .putExtra("gift", gift);

        AddItemActivity.this.setResult(RESULT_OK, intent);
        AddItemActivity.this.finish();
    }
}
```

五、实验结果





编辑修改条目

修改成功

名字重复

删除条目

六、实验问题

1. 增加条目后没有使用notifyDataSetChanged()更新ListView。

2. Android DatePicker控件获取时间月份少1

查阅资料后才知道，使用DatePicker的getMonth()获取到的月份是0到11。

3. 按后退按钮程序崩溃

在此页面如果不增加条目，直接按后退按钮返回主页面，程序会崩溃退出，进行调试后发现主页面的onActivityResult函数没有判断返回结果是否为RESULT_OK，直接调用getExtras会空指针异常。

4. The specified child already has a parent. You must call removeView() on the child's parent first.

使用自定义Dialog时，第一次点击条目能正常弹出Dialog，但第二次点击时程序崩溃，报出这个错误，在stackoverflow查到，每次弹出Dialog时，setView函数的参数的View都要重新声明，不能使用同一个View。

5. 删除条目时出现错误

长按某个条目后能进行删除，ListView也能正常更新，但是退出后重新打开应用发现真正被删除的是原本要删除的下一个条目，有点懵逼，仔细检查后发现我记录了长按的条目在列表中的位置，删除列表中对应该条目后，再根据该位置的名字去删除数据库中的行，而此时该位置已经是下一个条目，于是更改操作顺序，先删除数据库中的行后再删除列表条目。

七、实验总结

本次实验有一定难度，首先界面布局就花了很多时间，其中大部分花在了DatePicker以及在自定义的Dialog中显示DatePicker等。至于时间逻辑方面，用到了之前的ListView添加适配器，最重要的还是对数据库进行增删改查，以及使用Content Provider读取通讯录信息。在对数据库的操作中，使用了SQLiteOpenHelper能更方便地进行操作，可以自动编译成sql语句，不过调用函数的参数表也会显得特别冗长，与执行原生的sql语句各有优缺点吧。比如在获取全表的数据时，使用rawQuery("select * from lab8", null)比调用query()传递一长串的参数要简便很多。