

中山大学移动信息工程学院本科生实验报告

(2017 年秋季学期)

课程名称：移动应用开发

任课教师：郑贵锋

| | | | |
|------|-------------|--------|-----------------------|
| 年级 | 2015 | 专业（方向） | 互联网 |
| 学号 | 15352194 | 姓名 | 梁杰鑫 |
| 电话 | 15113959962 | Email | Alcanderian@gmail.com |
| 开始日期 | 2017.12.5 | 完成日期 | 2017.12.20 |

一、 实验题目

实现一个记录个人生日以及生日礼物的记录 APP，要求使用数据库，并在通讯录中查询个人联系电话。

二、 实现内容

实现一个生日备忘录，要求实现：

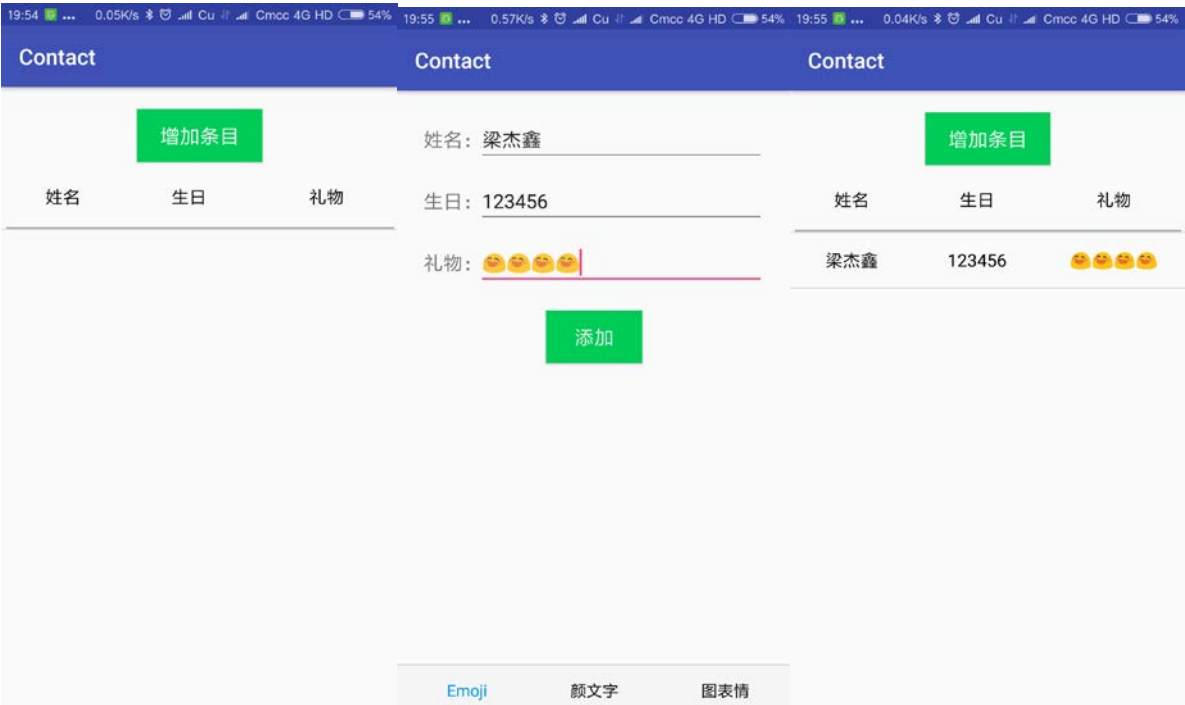
- 使用 SQLite 数据库保存生日的相关信息，并使得每一次运行程序都可以显示出已经存储在数据库
- 里的内容；
- 使用 ContentProvider 来获取手机通讯录中的电话号码。

功能要求：

- A. 主界面包含增加生日条目按钮和生日信息列表；
- B. 点击“增加条目”按钮，跳转到下一个 Activity 界面，界面中包含三个信息输入框（姓名、生日、礼物）和一个“增加”按钮，姓名字段不能为空且不能重复；
- C. 在跳转到的界面中，输入生日的相关信息后，点击“增加”按钮返回到主界面，此时，主界面中应更新列表，增加相应的生日信息；
- D. 主界面列表点击事件：
 - 点击条目：
 - 弹出对话框，对话框中显示该条目的信息，并允许修改；
 - 对话框下方显示该寿星电话号码（如果手机通讯录中有的话，如果没有就显示“无”）
 - 点击“保存修改”按钮，更新主界面生日信息列表。
 - 长按条目：
 - 弹出对话框显示是否删除条目；
 - 点击“是”按钮，删除该条目，并更新主界面生日列表。

三、 课堂实验结果

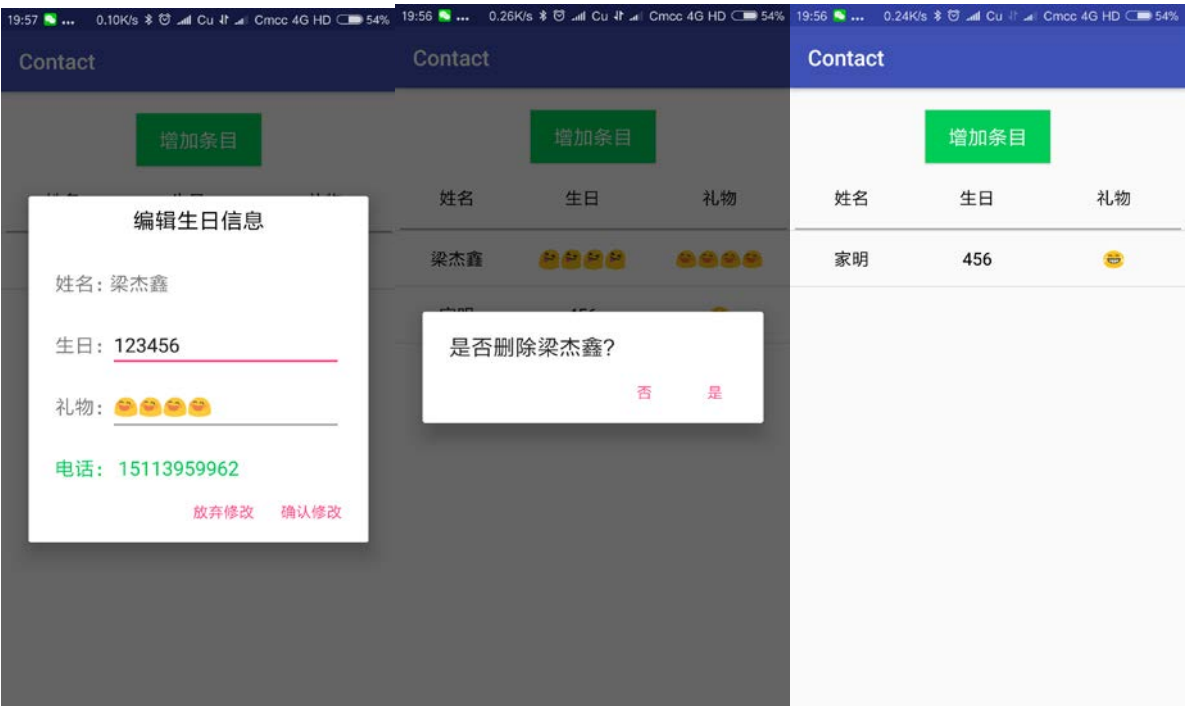
(1) 实验截图



初始状态

添加信息页面

添加之后



修改

删除 (可以看到修改结果)

删除之后

(2) 实验步骤以及关键代码

- SQLiteOpenHelper 以及数据逻辑的封装

继承 SQLiteOpenHelper，在构造函数中调用父类构造函数打开数据库，默认当没有数据库时会进行创建并调用 onCreate 函数如果数据库的版本发生变化则会调用 onUpgrade 或者 onDowngrade 函数，这个实验中用不到。

在这个时候就可以用 getWritableDatabase 获取数据库，免去在数据库操作的时候每次都要调用 getWritableDatabase 函数。

```
public class ContactDb extends SQLiteOpenHelper {
    public static final String tb_contact = "contact";
    public static final String c_cid = "cid";
    public static final String c_name = "name";
    public static final String c_birth = "birth";
    public static final String c_gift = "gift";
    private static SQLiteDatabase db;

    public ContactDb(Context context) {
        super(context, context.getString(R.string.db_name), null, 1);
        if (db == null)
            db = getWritableDatabase();
    }
}
```

ContactDb 的 onCreate 函数，不需要判断表是否存在，因为是新文件

```
@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL("CREATE TABLE contact (cid INTEGER PRIMARY KEY, name TEXT, birth TEXT, gift TEXT)");
}

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

接下来是增删改查的函数，insert 的返回值是插入的行号，在我们的主键为 INTEGER 的情况下，其实就是 cid。

```
public int insert(String name, String birth, String gift) {
    ContentValues cv = new ContentValues();
    cv.put(c_name, name);
    cv.put(c_birth, birth);
    cv.put(c_gift, gift);
    return (int) db.insert(tb_contact, null, cv);
}
```

Update, delete

```
public void update(Integer cid, String name, String birth, String gift) {
    String where = "cid = ?";
    String[] args = {cid.toString()};
    ContentValues cv = new ContentValues();
    cv.put(c_name, name);
    cv.put(c_birth, birth);
    cv.put(c_gift, gift);
    db.update(tb_contact, cv, where, args);
}

public void delete(Integer cid) {
    String where = "cid = ?";
    String[] args = {cid.toString()};
    db.delete(tb_contact, where, args);
}
```

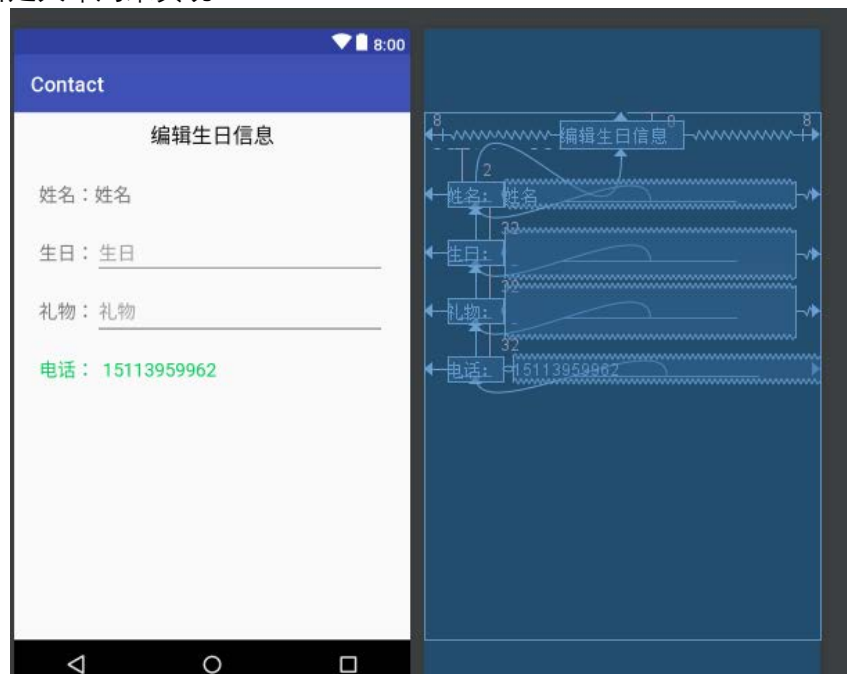
按照 cid 查询人物信息或者取出所有人物信息

```
public ContactItem select(Integer cid) {
    String where = "cid = ?";
    String[] args = {cid.toString()};
    Cursor cursor = db.query(
        tb_contact,
        new String[]{c_cid, c_name, c_birth, c_gift},
        where,
        args, null, null, null);
    ContactItem ret = null;
    if (cursor.moveToNext())
        ret = curToItem(cursor);
    cursor.close();
    return ret;
}

public ArrayList<ContactItem> queryAll() {
    ArrayList<ContactItem> ret = new ArrayList<>();
    Cursor cursor = db.query(
        tb_contact,
        new String[]{c_cid, c_name, c_birth, c_gift},
        null, null, null, null, null);
    while (cursor.moveToNext())
        ret.add(curToItem(cursor));
    cursor.close();
    return ret;
}
```

- 主界面

主界面的布局很简单，主要展示一下 alertDialog 的布局，因为需要在 alertDialog 中进行输入，就需要自定义布局来实现



更新数据

```
AlertDialog.Builder builder = new AlertDialog.Builder(ListActivity.this);
builder.setView(ad_edit)
    .setPositiveButton("确认修改", (dialog, which) -> {
        item.birth = ((TextView) ad_edit.findViewById(R.id.et_birth)).getText().toString();
        item.gift = ((TextView) ad_edit.findViewById(R.id.et_gift)).getText().toString();
        dbo.update(item.cid, item.name, item.birth, item.gift);
        adapter.notifyDataSetChanged();
    })
    .setNegativeButton("放弃修改", null)
    .show();
}
```

增加数据的时候，利用 `startActivityForResult` 传递 request。并重写 `onActivityResult` 来接受返回的数据。返回的数据中只需要包含新插入数据的 cid，在通过数据库查询得到。

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == 1000 && data.hasExtra("cid")) {
        adapter.data.add(dbo.select(data.getIntExtra("cid", -1)));
        adapter.notifyDataSetChanged();
    }
}
```

● 添加人物页面

```
btn_add.setOnClickListener((v) -> {
    String name = et_name.getText().toString();
    if (name.trim().isEmpty()) {
        Toast.makeText(AddActivity.this, "名字不能为空", Toast.LENGTH_SHORT).show();
    } else {
        Integer cid = dbo.insert(name, et_birth.getText().toString(), et_gift.getText().t
        Intent intent = new Intent(AddActivity.this, ListActivity.class);
        intent.putExtra("cid", cid);
        setResult(1000, intent);
        finish();
    }
});
```

插入数据后取得 cid，并回传给主界面

● ContactProvider

这个类用于查询通讯录中的数据，其查询的方式和数据库是一样的。不同的电话号码之间最好用换行相隔，不然在 `alertDialog` 中会超出显示范围。另外，在查询选项中必须用“”将要查询的人名包起来，不认识查询不到纯数字以外的人名的。

```
private String getTelByName(String name) {
    try {
        Cursor cursor = getContentResolver().query(
            ContactsContract.CommonDataKinds.Phone.CONTENT_URI, null,
            String.format("%s = '%s'", ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME, name),
            null, null
        );
        StringBuilder ret = new StringBuilder();
        while (cursor.moveToNext()) {
            ret.append(cursor.getString(cursor.getColumnIndex(
                ContactsContract.CommonDataKinds.Phone.NUMBER
            )));
            if (!cursor.isLast())
                ret.append('\n');
        }
        cursor.close();
        String s = ret.toString();
        return s.isEmpty() ? "无" : s;
    } catch (Exception e) {
        e.printStackTrace();
        return "无";
    }
}
```

四、实验思考

数据库是处理事务的时候的必须手段，实现数据的持久化并且结构化的存储。以方便增删改查，以及其他复杂的业务逻辑。同时，在手机 APP 中也需要连接用户的通讯录，这时候就要用到 `ContactProvider`。这次实验主要熟悉了这两个常用工具的使用方法。