

中山大学数据科学与计算机学院本科生实验报告

(2017 年秋季学期)

课程名称：手机应用开发

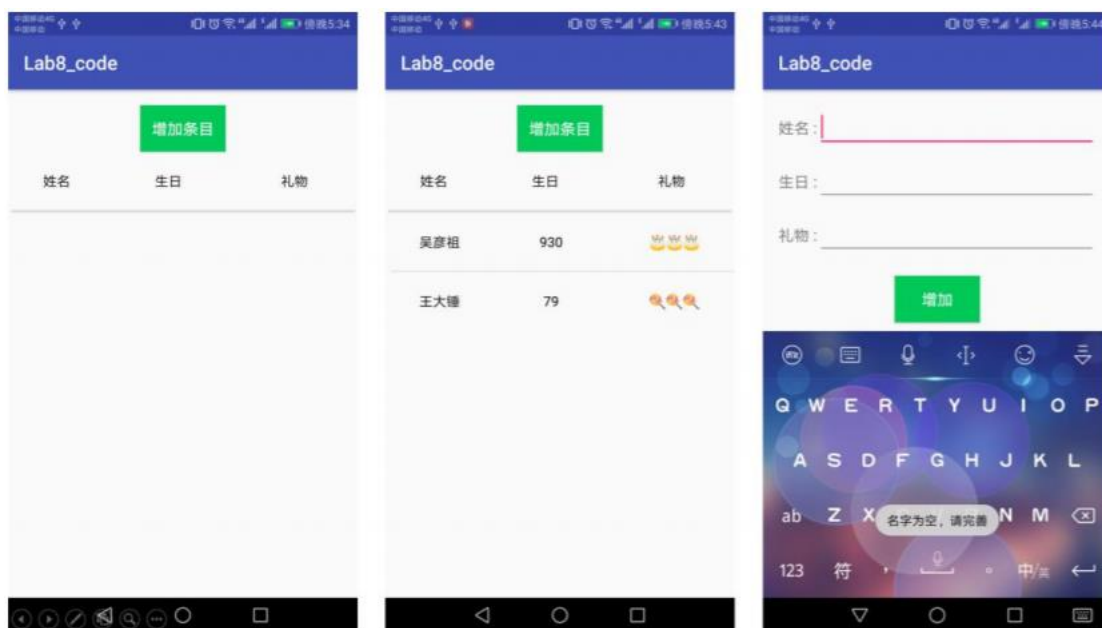
任课教师：刘宁

年级	2015	专业（方向）	计应
学号	14970011	姓名	宋思亭
电话	18826073511	Email	1056284551@qq.com
开始日期	2017.12.4	完成日期	2017.12.5

一、 实验目的

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

二、 实现内容



从左至右：初始界面，添加一部分条目，名字不能为空。



从左至右：名字不能重复，点击条目显示信息（可修改），长按删除条目。

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；

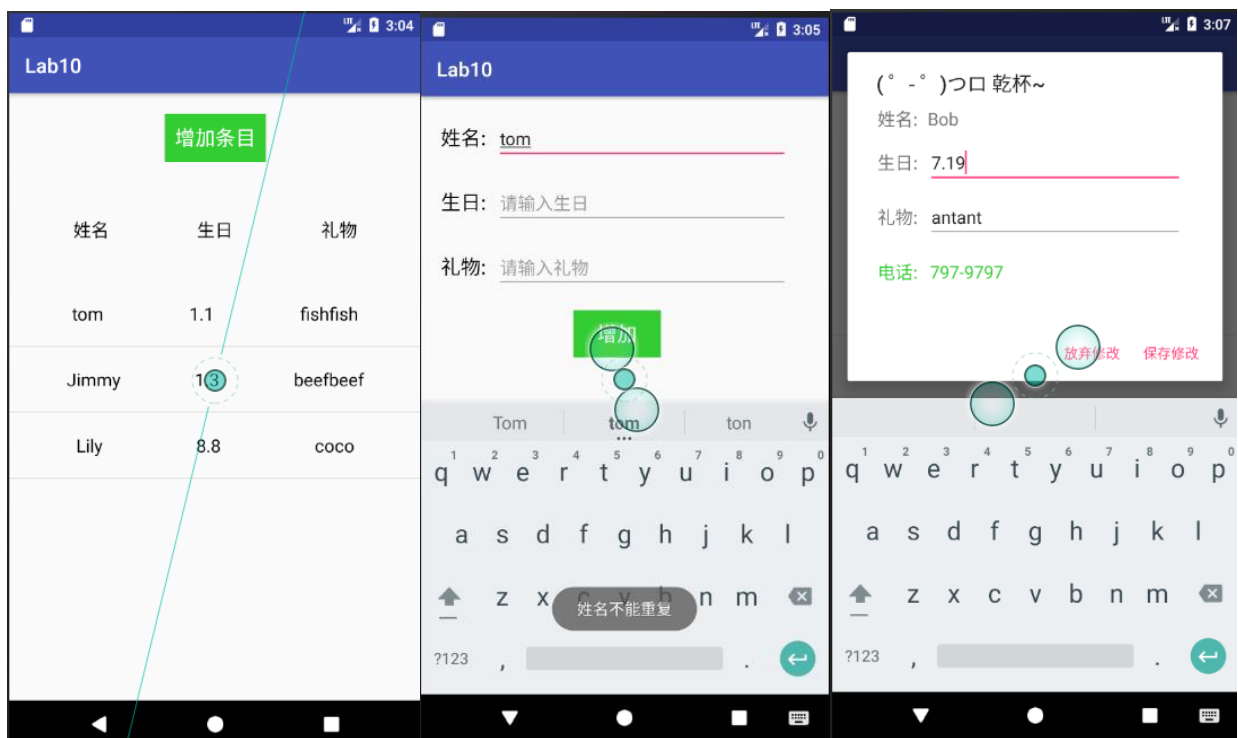
使用 ContentProvider 来获取手机通讯录中的电话号码。

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。

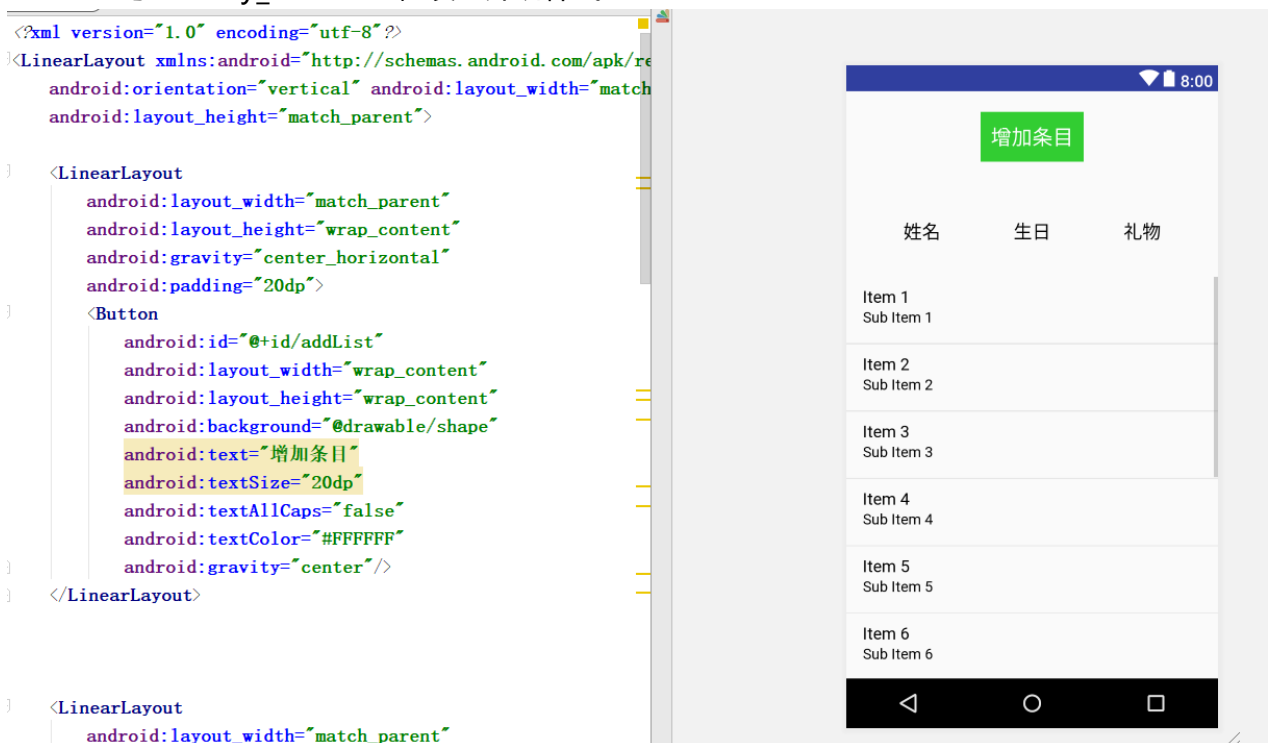
三、 课堂实验结果

（1） 实验截图



(2) 实验步骤以及关键代码

1. 写 activity_main.xml, 设置外观样式



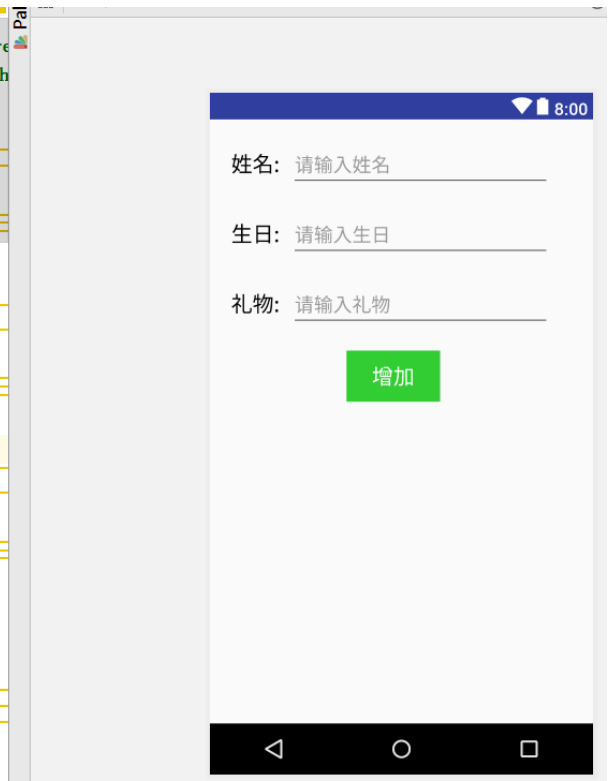
写 activity_additem.xml, 设置外观样式

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TableLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:stretchColumns="1"
        android:layout_marginRight="20dp"
        android:layout_marginLeft="20dp">
        <TableRow
            android:layout_marginTop="20dp">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="姓名:"
                android:textColor="#000000"
                android:textSize="20dp"
                android:gravity="center"/>
            <EditText
                android:id="@+id/nameET"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginLeft="10dp"
                android:layout_marginRight="20dp"
                android:hint="请输入姓名"

```



写 shape.xml，设置按钮样式

```

<solid android:color="#32CD32" />
<padding
    android:bottom="5dp"
    android:left="10dp"
    android:right="10dp"
    android:top="5dp" />

```

2. 写 MainActivity 类

打开应用时执行 dataUpdate，同步数据库中数据到列表

// 将数据库中的数据更新到UI中

```
private void dataUpdate() {
    try {
        MyDataBase db = new MyDataBase(getBaseContext());
        SQLiteDatabase sqLiteDatabase = db.getWritableDatabase();
        Cursor cursor = sqLiteDatabase.rawQuery("select * from " + TABLE_NAME, selectionArgs: null);
        datas = new ArrayList<Map<String, String>>();
        if (cursor != null) {
            while (cursor.moveToNext()) {
                String name1 = cursor.getString(columnIndex: 0);
                String birth1 = cursor.getString(columnIndex: 1);
                String gift1 = cursor.getString(columnIndex: 2);
                Map<String, String> map = new HashMap<String, String>();
                map.put("name", name1);
                map.put("birth", birth1);
                map.put("gift", gift1);
                datas.add(map);
            }
            adapter = new SimpleAdapter(context: MainActivity.this, datas, R.layout.item,
                new String[]{"name", "birth", "gift"}, new int[]{R.id.nameLV, R.id.birthLV, R.id.giftLV});
            LV.setAdapter(adapter);
        }
    } catch (SQLException e) {
        Log.i("tag: DataBaseLog", e.toString());
    }
}
```

写点击列表弹出对话框事件

为了实现自定义对话框，新建 dialoglayout.xml 文件

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

    <TableLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:stretchColumns="1"
        android:layout_marginRight="20dp"
        android:layout_marginLeft="20dp">

        <TableRow
            android:layout_marginTop="10dp">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="10dp"
                android:gravity="center"
                android:text="姓名:"
                android:textSize="18sp" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="10dp"
                android:gravity="center"
                android:text="生日:"
                android:textSize="18sp" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="10dp"
                android:gravity="center"
                android:text="礼物:"
                android:textSize="18sp" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="10dp"
                android:gravity="center"
                android:text="电话:"
                android:textSize="18sp" />

        </TableRow>

    </TableLayout>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:gravity="center"
        android:text="无"
        android:textSize="18sp" />

</LinearLayout>
```

在.java 文件中使用 LayoutInflater 自定义对话框

// 使用LayoutInflater实现自定义对话框

```
LayoutInflater factory = LayoutInflater.from(MainActivity.this);  
View newView = factory.inflate(R.layout.dialoglayout, root: null);  
AlertDialog.Builder builder = new AlertDialog.Builder(context: MainActivity.this);
```

```
nameTV = (TextView)newView.findViewById(R.id.nameDL);  
birthET = (EditText)newView.findViewById(R.id.birthDL);  
giftET = (EditText)newView.findViewById(R.id.giftDL);  
phone = (TextView)newView.findViewById(R.id.phoneDL);
```

```
nameTV.setText(datas.get(position).get("name"));  
birthET.setText(datas.get(position).get("birth"));  
giftET.setText(datas.get(position).get("gift"));
```

为了读取联系人列表，首先申请动态权限

在 AndroidManifest 中注册

```
<uses-permission android:name="android.permission.READ_CONTACTS"/>
```

```
<uses-permission-sdk-25 android:name="android.permission.READ_CONTACTS"/>
```

// 动态权限申请

```
if (ContextCompat.checkSelfPermission(context: MainActivity.this,  
    Manifest.permission.READ_CONTACTS) != PackageManager.PERMISSION_GRANTED) {  
    ActivityCompat.requestPermissions(activity: MainActivity.this,  
        new String[] {Manifest.permission.READ_CONTACTS}, requestCode: 0);  
}
```

使用 Cursor 读取联系人列表

```

// 读取联系人列表
String str1 = "";
Cursor cursor1 = getResolver().query(ContactsContract.Contacts.CONTENT_URI,
    projection: null, selection: null, selectionArgs: null, sortOrder: null);
while (cursor1.moveToNext()) {
    String str2 = cursor1.getString(cursor1.getColumnIndex(columnName: "_id"));
    if (cursor1.getString(cursor1.getColumnIndex(columnName: "display_name"))
        .equals(nameTV.getText().toString())) {
        // 判断某条联系人的信息中, 是否有电话号码
        if (Integer.parseInt(cursor1.getString(cursor1
            .getColumnIndex(columnName: "has_phone_number"))) > 0) {
            // 取出该条联系人信息中的电话号码
            Cursor cursor2 = getResolver().query(
                ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
                projection: null,
                selection: "contact_id = " + str2,
                selectionArgs: null,
                sortOrder: null
            );
            while (cursor2.moveToNext()) {
                str1 = str1 + cursor2.getString(cursor2.getColumnIndex(columnName: "data1")) + "\n";
            }
            cursor2.close();
        }
    }
}
cursor1.close();

```

使用 builder 自定义对话框

// 自定义对话框

```

builder.setView(new View());
builder.setTitle(" ( °- °)つロ 乾杯~");
builder.setPositiveButton(text: "保存修改", (dialog, which) -> {
    if (birthET.length() != 0) {
        MyDatabase db = new MyDatabase(getBaseContext());
        SQLiteDatabase sqLiteDatabase = db.getWritableDatabase();
        sqLiteDatabase.execSQL(sql: "update " + TABLE_NAME + " set birth = ? where name = ?",
            new Object[] {birthET.getText().toString(), nameTV.getText().toString()});
        sqLiteDatabase.close();
    }
    if (giftET.length() != 0) {
        MyDatabase db = new MyDatabase(getBaseContext());
        SQLiteDatabase sqLiteDatabase = db.getWritableDatabase();
        sqLiteDatabase.execSQL(sql: "update " + TABLE_NAME + " set gift = ? where name = ?",
            new Object[] {giftET.getText().toString(), nameTV.getText().toString()});
        sqLiteDatabase.close();
    }
    dataUpdate();
});
builder.setNegativeButton(text: "放弃修改", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        //To do
    }
});
builder.show();

```

写增加条目的按钮点击事件, 点击跳转到编辑界面

// 点击增加条目的按钮跳转到编辑界面

```
add.setOnClickListener((v) -> {  
    Intent intent = new Intent( packageContext: MainActivity.this, AddActivity.class);  
    startActivityForResult(intent, requestCode: 9);  
});
```

写长按 listview 询问删除条目事件

// 长按listview弹出询问删除对话框

```
LV.setOnItemClickListener((parent, view, position, id) -> {  
    AlertDialog.Builder message = new AlertDialog.Builder( context: MainActivity.this);  
    message.setTitle("是否删除?");  
    message.setPositiveButton( text: "是", (dialogInterface, i) -> {  
        // 删除数据库数据  
        MyDataBase db = new MyDataBase(getBaseContext());  
        SQLiteDatabase sqLiteDatabase = db.getWritableDatabase();  
        sqLiteDatabase.execSQL( sql: "delete from " + TABLE_NAME + " where name = ?",  
            new String[]{datas.get(position).get("name")});  
        sqLiteDatabase.close();  
        // 删除列表中的数据  
        datas.remove(position);  
        adapter.notifyDataSetChanged();  
    });  
    message.setNegativeButton( text: "否", new DialogInterface.OnClickListener() {  
        public void onClick(DialogInterface dialogInterface, int i) {  
            // 不执行操作  
        }  
    });  
    message.create().show();  
    return true;  
});
```

重写 onActivityResult 函数，每次返回该页面时更新列表数据

// 从其它页面返回时更新数据

@Override

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    super.onActivityResult(requestCode, resultCode, data);  
    if (requestCode == 9 && resultCode == 99) {  
        dataUpdate();  
    }  
}
```

3. 写 AddActivity 类

写增加按钮点击监听事件


```

addBtn.setOnClickListener((v) -> {
    findViewById();
    if (addET.length() == 0) {
        Toast.makeText(context: AddActivity.this, text: "姓名不能为空", Toast.LENGTH_SHORT).show();
    } else {
        findViewById();

        String namet = addET.getText().toString();
        String birtht = birthET.getText().toString();
        String giftt = giftET.getText().toString();

        MyDataBase db = new MyDataBase(getBaseContext());
        SQLiteDatabase sqLiteDatabase = db.getWritableDatabase();
        Cursor cursor = sqLiteDatabase.rawQuery(sql: "select * from " +
            TABLE_NAME + " where name like ?", new String[]{namet});

        if (cursor.moveToFirst() == true) {
            Toast.makeText(context: AddActivity.this, text: "姓名不能重复", Toast.LENGTH_SHORT).show();
        } else {
            ContentValues contentValues = new ContentValues();
            contentValues.put("name", namet);
            contentValues.put("birth", birtht);
            contentValues.put("gift", giftt);
            sqLiteDatabase.insert(TABLE_NAME, nullColumnHack: null, contentValues);
            sqLiteDatabase.close();
            setResult(resultCode: 99, new Intent());
            finish();
        }
    }
}

```

4. 写 MyDataBase 类

```

public MyDataBase(Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {
    super(context, name, factory, version);
}

public MyDataBase(Context context) { this(context, DB_NAME, factory: null, DB_VERSION); }

// 创建数据库: 可直接执行创建数据库的 SQL 语句
@Override
public void onCreate(SQLiteDatabase sqLiteDatabase) {
    String CREATE_TABLE = "CREATE TABLE if not exists " + TABLE_NAME
        + "(name TEXT PRIMARY KEY," + "birth TEXT," + "gift TEXT)";
    try {
        sqLiteDatabase.execSQL(CREATE_TABLE);
    } catch (SQLException e) {
        Log.i(tag: "DataBaseLog", e.toString());
    }
}

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

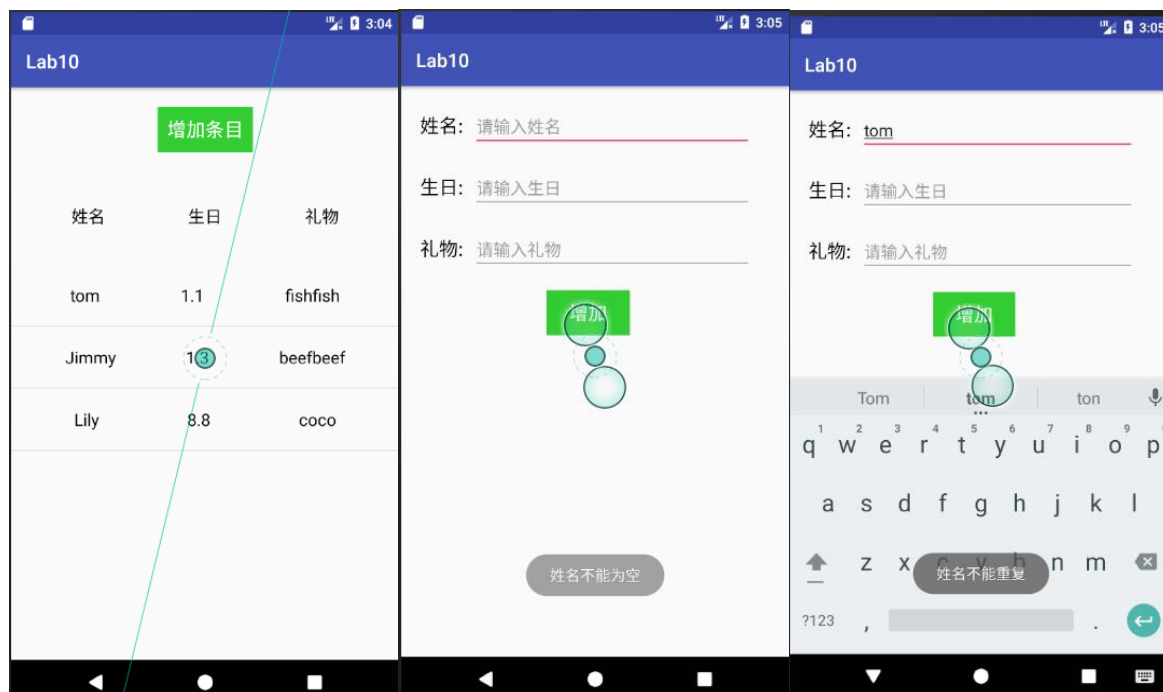
```

(3) 实验遇到困难以及解决思路

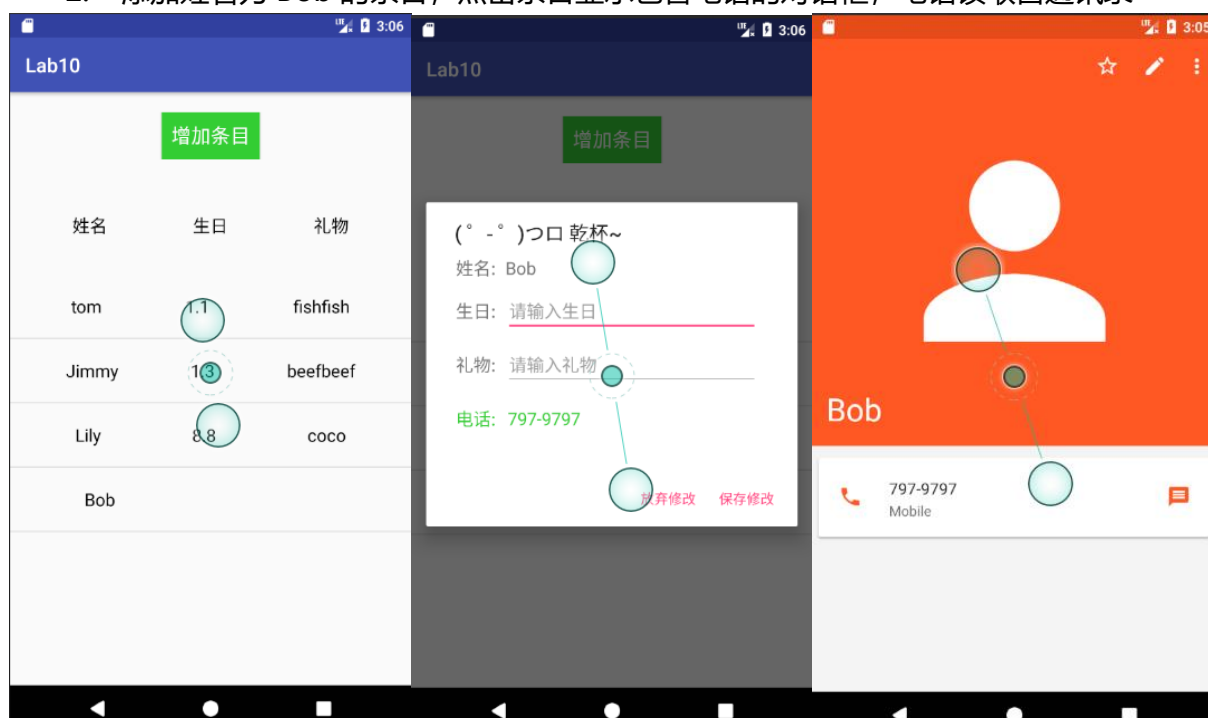
1. 点击列表跳出，获取系统权限后解决

四、 课后实验结果

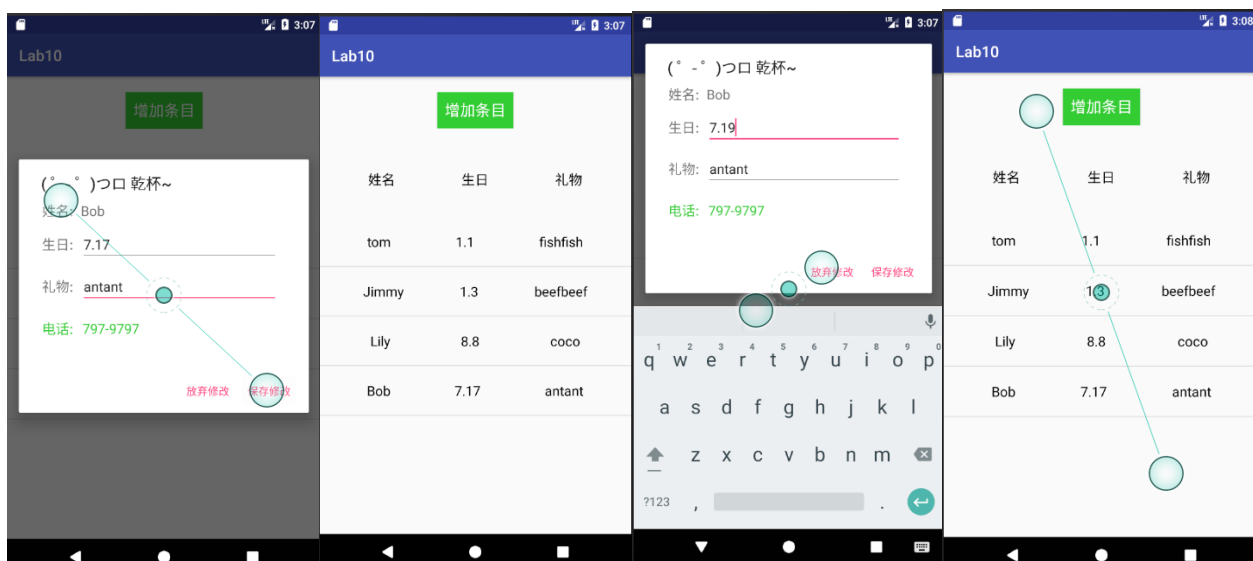
1. 打开应用读取数据库数据，点击增加条目跳转到添页面，包含姓名字段不能为空或重复的提示



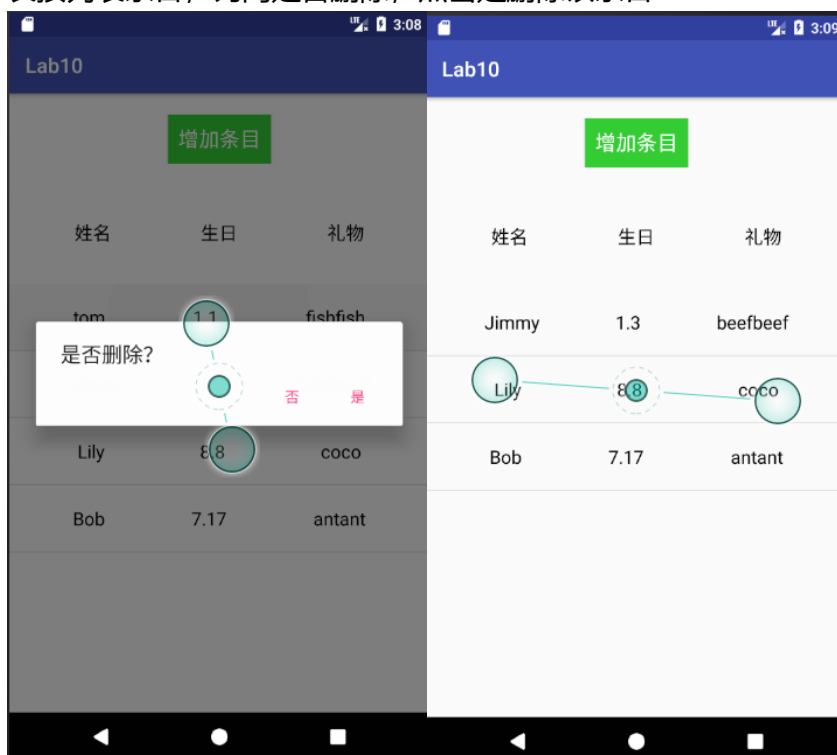
2. 添加姓名为 Bob 的条目，点击条目显示包含电话的对话框，电话读取自通讯录



3. 点击修改条目，修改内容保存，点击取消修改，修改内容不保存



4. 长按列表条目，询问是否删除，点击是删除该条目



五、 实验思考及感想

1. 安卓开发过程中有很多坑等我们去跳，需要多用搜索引擎多看文档
2. 遇到奇怪的问题不知道怎么解决时，可以重启试试