Friend 说明文档

完成人：詹萍
学号：20152100027
完成时间：2017 年 12 月 03 日

一、软件名称

Friend

二、软件内容简介

这是一款类似同学录的记录朋友信息的软件。

朋友是时常陪伴在我们身边的人，然而对于健忘症患者来说，常常难以记住朋友的生日、电话、喜好等基本信息。

为了方便用户记录和查看朋友信息，本软件主要实现了添加朋友信息、查看朋友信息列表、删除朋友信息和查询朋友信息的功能。

三、界面设计

1、主页——可进行功能选择：查看朋友列表或者添加新的朋友。如图 1。

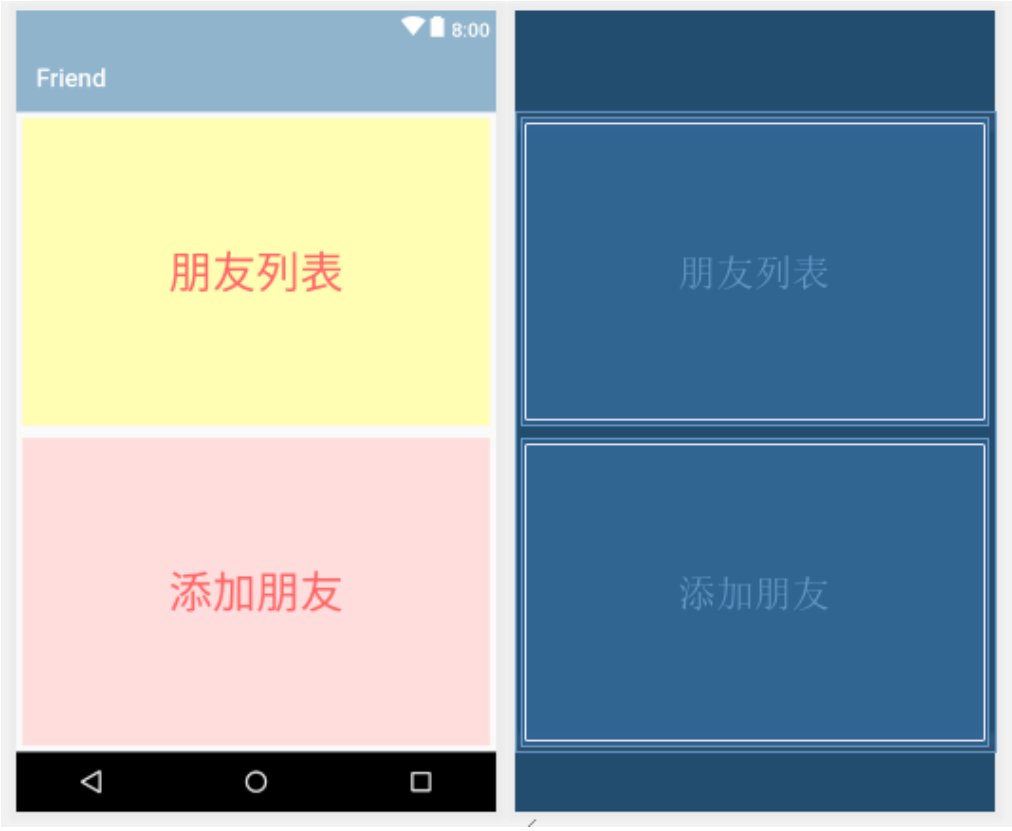


图 1

2、添加新朋友——通过主页的功能选择跳转到添加朋友的页面，输入朋友的信息，点击“完成”按钮实现添加朋友信息并返回首页。如图 2 所示。

图 2

3、查看朋友列表——通过主页的功能选择跳转到查看朋友列表的页面。如图 3 所示。

①可以上下滑动查看朋友列表信息。

②可以左滑删除某个好友的信息。

③通过好友名字可以在搜索栏搜索好友的信息。

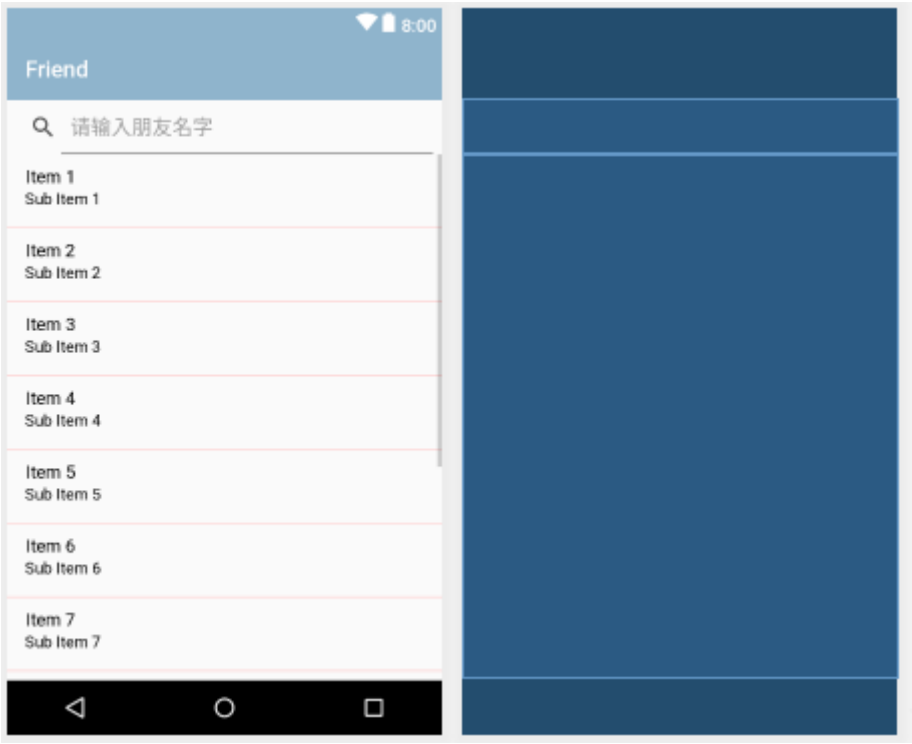


图 3

4、朋友信息列表的条目，布局采用如图 4 所示布局。
包括朋友的基本信息、头像、姓名、生日、电话、以及用户对朋友的描述（爱好、个人看法等信息）。

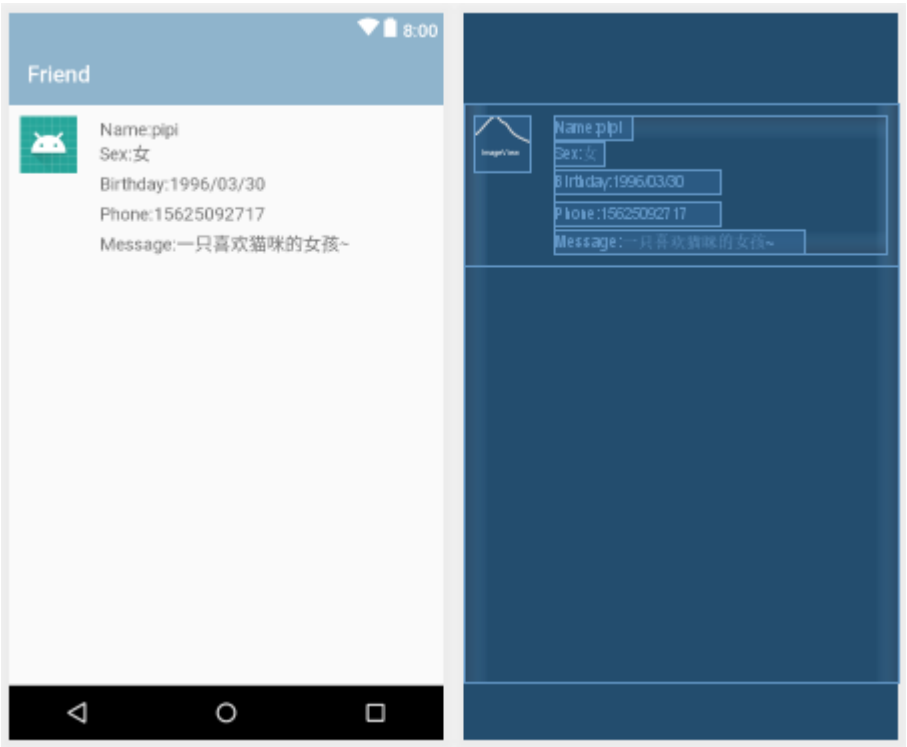


图 4

四、代码设计

1、代码目录有条理、划分明确。如图 5 所示。

①将 MyHelper 类（继承自 SQLitOpeneHelper 类）独立写成一个类文件，保证了该类良好的可重用性。在类中建立本软件的数据库 friend.db 以及朋友信息的表格 information。

②将 MyAdapter 类（继承自 BaseAdapter 类）独立写成一个类文件，保证了该类良好的可重用性。在类中实现对 ListView 的适配，每一个 item 的数据通过构造函数的参数传进本类中。

③将朋友的信息封装成一个类 Friend。方便信息的读取和存储。

④将滑动类 SwipeLayout 独立封装成一个类，在 ListItem 中需要使用滑动功能时只需要引用即可，可以不知道滑动类的细节

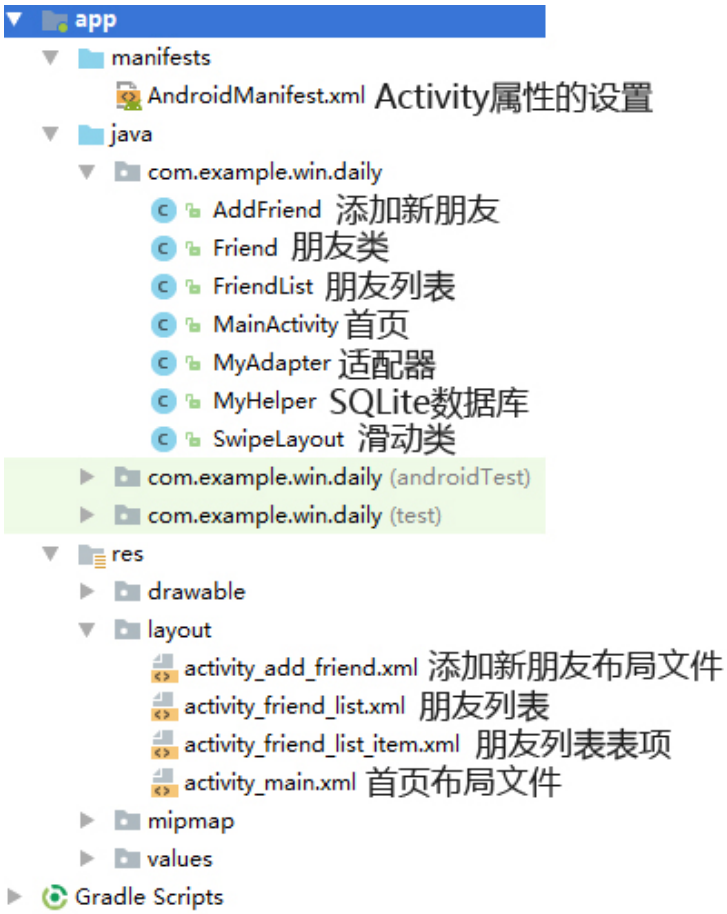


图 5

2、搜索亮点：搜索栏实现搜索功能。

使用 Android 官网推荐的控件 SearchView 来实现搜索功能。

①.xml 布局文件代码如下：

```
<SearchView
        android:id="@+id/sv_query"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:iconifiedByDefault="false"
```

```
android:queryHint="请输入朋友名字" />
```

②在.java 文件中实现点击搜索的功能。

判断输入是否为空：

```
//查询语句
String querytext = query.getQuery().toString().trim();
if(TextUtils.isEmpty(querytext)){
    Toast.makeText(FriendList.this,"输入为空!
",Toast.LENGTH_SHORT).show();
 }
```

更新 Adapter、ListView 上面的信息，只显示搜索结果：

```
List<Friend> queryFriendsList = new ArrayList<Friend>();
//请空列表
queryFriendsList.clear();
getData(querytext,queryFriendsList);
//创建一个 Adapter 的实例,设置适配器
mlistview.setAdapter(new
MyAdapter(FriendList.this,queryFriendsList));
```

3、删除亮点：左滑删除。

①在 activity_friend_list_item 文件中**引用已有的滑动文件：**

```
<!--引用 SwipeLayout.java 类-->
<com.example.win.daily.SwipeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
</com.example.win.daily.SwipeLayout>
```

②在 activity_friend_list_item 布局文件中**设置滑动后显示的删除图标**。

```
<View
     android:layout_centerInParent="true"
     android:layout_width="50dp"
     android:layout_height="50dp"
     android:background="@drawable/ic_action_delete"/>
```

③在 MyAdapter 文件中**监听滑动删除事件**。

```
//点击删除的事件监听响应
LinearLayout delete = (LinearLayout)
view.findViewById(R.id.ll_deleteButton);
delete.setOnClickListener(new View.OnClickListener(){
    public void onClick(View v){
        Toast.makeText(context, "click delete button",
Toast.LENGTH_SHORT).show();
```

```
            //从数据库删除
            MyHelper myHelper = new MyHelper(context);
            //获得可以读取数据的 SQLiteDataBase 对象
            SQLiteDatabase db = myHelper.getWritableDatabase();
            db.delete("information","_id=?",
                  new String[]{friendsList.get(p).getId()});
            //从朋友列表删除
            friendsList.remove(p);
            notifyDataSetChanged();
        }
});
```

五、软件操作流程

在真机上操作：

1、添加好友的操作：

① 点击按钮“添加朋友”从主页面跳转到填写朋友信息的页面。如图 6 所示。

② 根据 EditText 的提示信息输入朋友的姓名、性别、生日、电话号码和对朋友的描述，完成之后点击按钮“确定”将朋友信息保存起来并返回首页。如图 7 所示。

图 6

图 7

2、打开朋友列表：

① 点击按钮“朋友列表”从主页面跳转到朋友页面。如图 8 所示。

② 为了展示下拉效果，我事先在数据库中添加了几个朋友的数据，下拉列表。如图 9 所示。可以看到我们刚才添加的新朋友的信息。

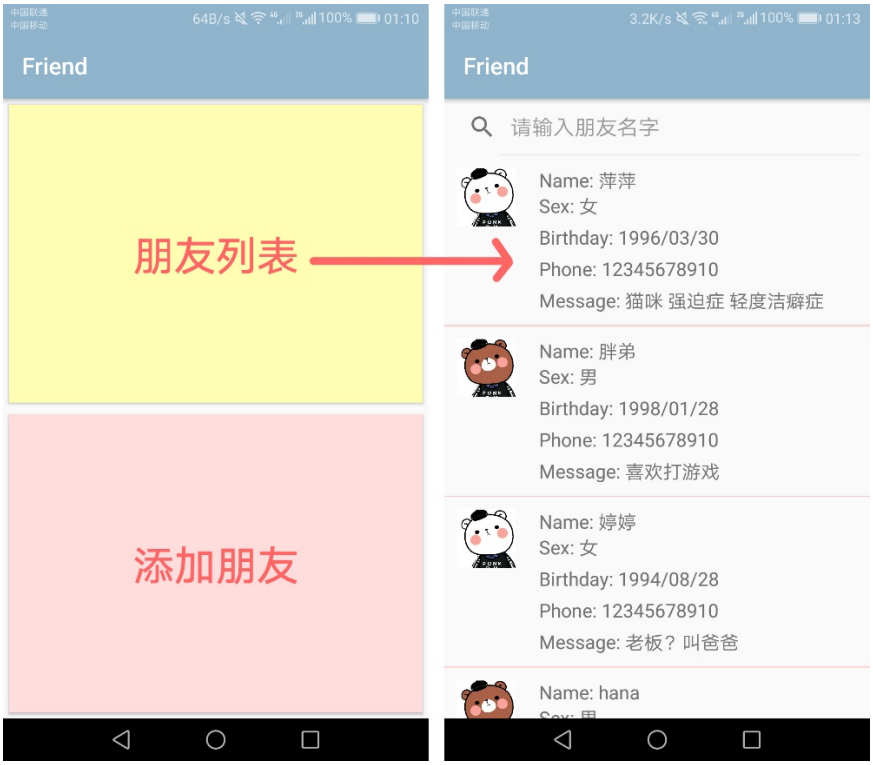


图 8

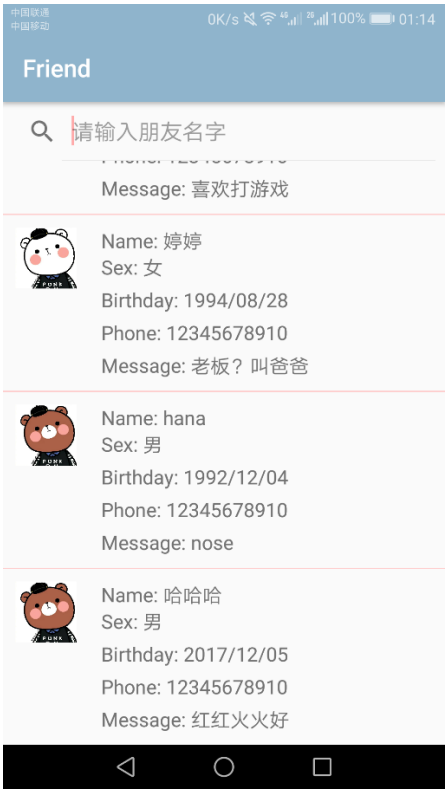


图 9

3、搜索朋友信息。

根据搜索栏的提示输入朋友的名字，点击搜索 icon 搜索朋友的信息。如图 10 所示。

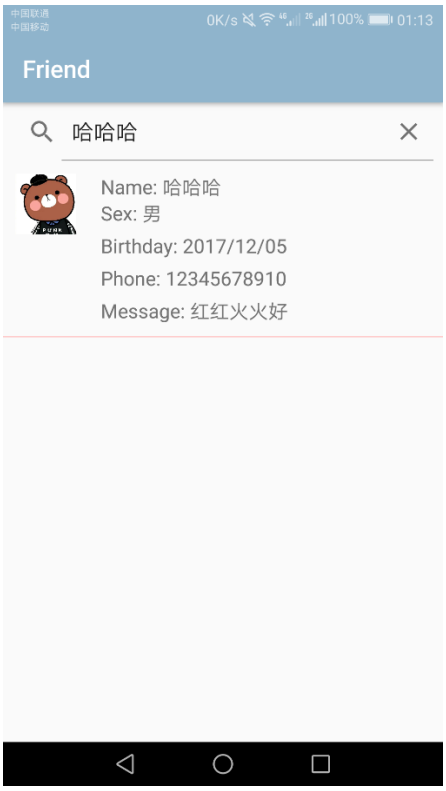


图 10

4、左滑删除朋友信息。

当发现不再需要某个朋友的信息时可以左滑删除该朋友的信息。图 11 删除前，图 12 为删除后。

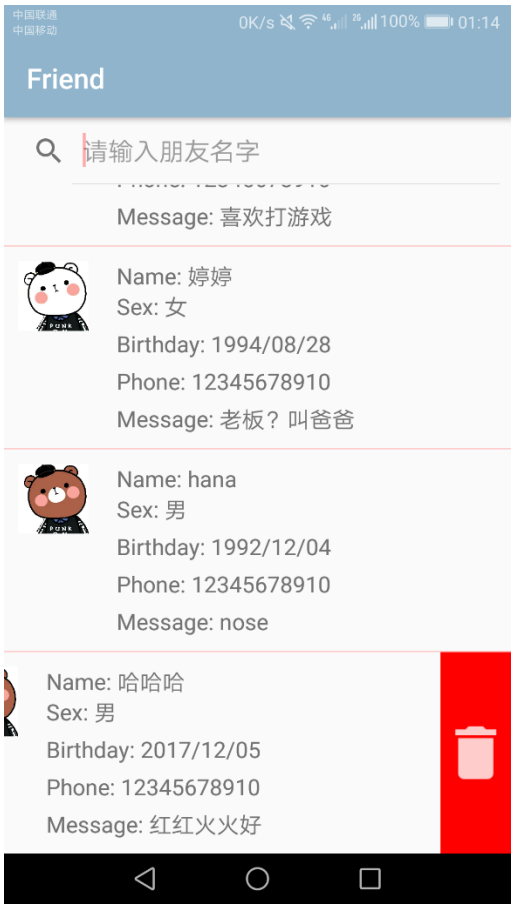


图 11

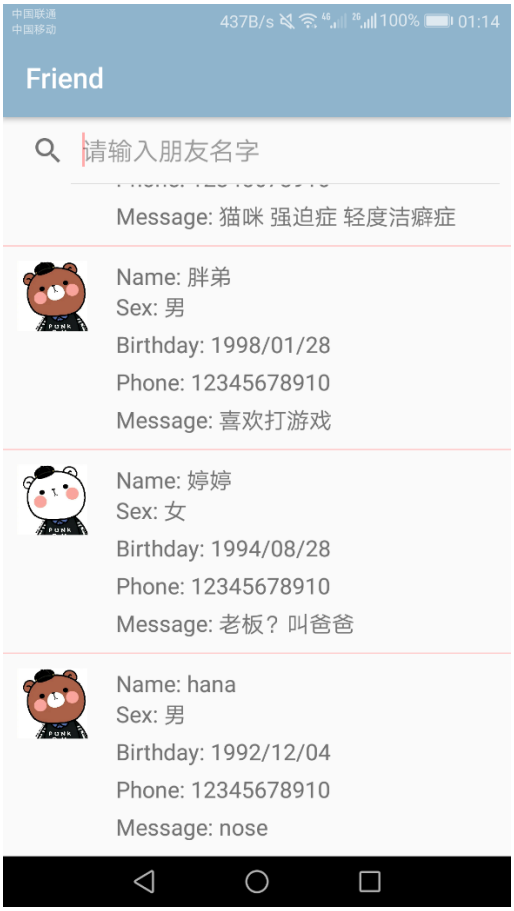


图 12

六、难点和解决方案

1、搜索之后更新 ListView 的显示

① 难点：

搜索之后更新 ListView 的显示，让其只显示搜索返回的结果。

② 解决方案：

通过新建一个只用于保存搜索结果的 List 来将每次搜索的结果保存起来，并用这个 List 的数据去作为适配显示的内容，更新 ListView 的显示。

2、滑动删除功能的设置

① 难点：

第一点是实现向左滑动显示删除的图标。

第二点是点击删除事件存放的位置。

② 解决方案：

在网上百度之后采用了别人封装好的滑动类来实现滑动功能，只需要在 item 相应的.xml 文件引用即可。引用代码如下：

```
<!--引用 SwipeLayout.java 类-->
<com.example.win.daily.SwipeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
</com.example.win.daily.SwipeLayout>
```

最终确定将点击删除的事件写在适配器的 getView 方法中，因为对于每个不同的 item，他们在适配器中对应不同的 view，需要明确点击的是某个具体的 item 的删除事件，因此将对应的删除事件写在适配器中。

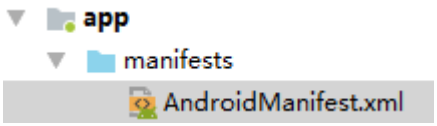
3、Activity 重复生成的问题

① 难点：

从主页面跳转到功能页面，在从功能页面跳转回到主页面；重复上述步骤，会发现任务栈中保留了很多主页面和功能页面的 Activity。这样在不断点击返回时系统会从任务栈中不断调出之前打开的压在栈底的 Activity，但我们希望每个页面只在栈中保留一份。

② 解决方案：

在如下目录的 AndroidManifest.xml 文件中对每个 Activity 的启动模式进行设置。



MainActivity 启动模式设置：

```
<activity android:name=".MainActivity"
            android:launchMode="singleTask">
```

AddFriend 和 FriendList 启动模式设置：

```
<activity android:name=".AddFriend"
            android:launchMode="singleTask" />
<activity android:name=".FriendList"
            android:launchMode="singleTask" />
```

七、不足之处

1、暂且没有实现“假”删除的功能。当用户点击删除时真的删除了好友的数据，可能会出现用户误删的情况。

2、只对应朋友性别设置了两种头像，尚未实现用户选择头像的功能。

3、搜索之后需要返回首页再进入朋友列表才能重新查看全部好友的信息。

八、今后的设想

1、实现假删除的功能，在数据库表项设置一个删除标志，当用户点击删除之后将标志的值设置为删除状态，实际不删除用户的好友信息。
2、实现用户选择头像的功能，可以从手机相册选择或者打开相机来拍照获取。
3、实现模糊搜索，ListView 根据搜索的文本自动匹配、动态返回符合查询结果的表项。