配套说明文档

软件名称： SqliteDemo

完成人：吴思豪

学号：20152100100

完成时间：2017/12/1

软件内容简介：

一款可以添加用户姓名，学号，性别的数据库以及他的交互系统。

可以实现用户姓名，学号，性别的数据库添加，修改，查询，以及通过 RecyclerView 的拖曳，去移动 RecyclerView 的不同 item 的位置，还有通过 RecyclerView 的左侧或者右侧滑动，去删除 item 对应的数据库的内容。

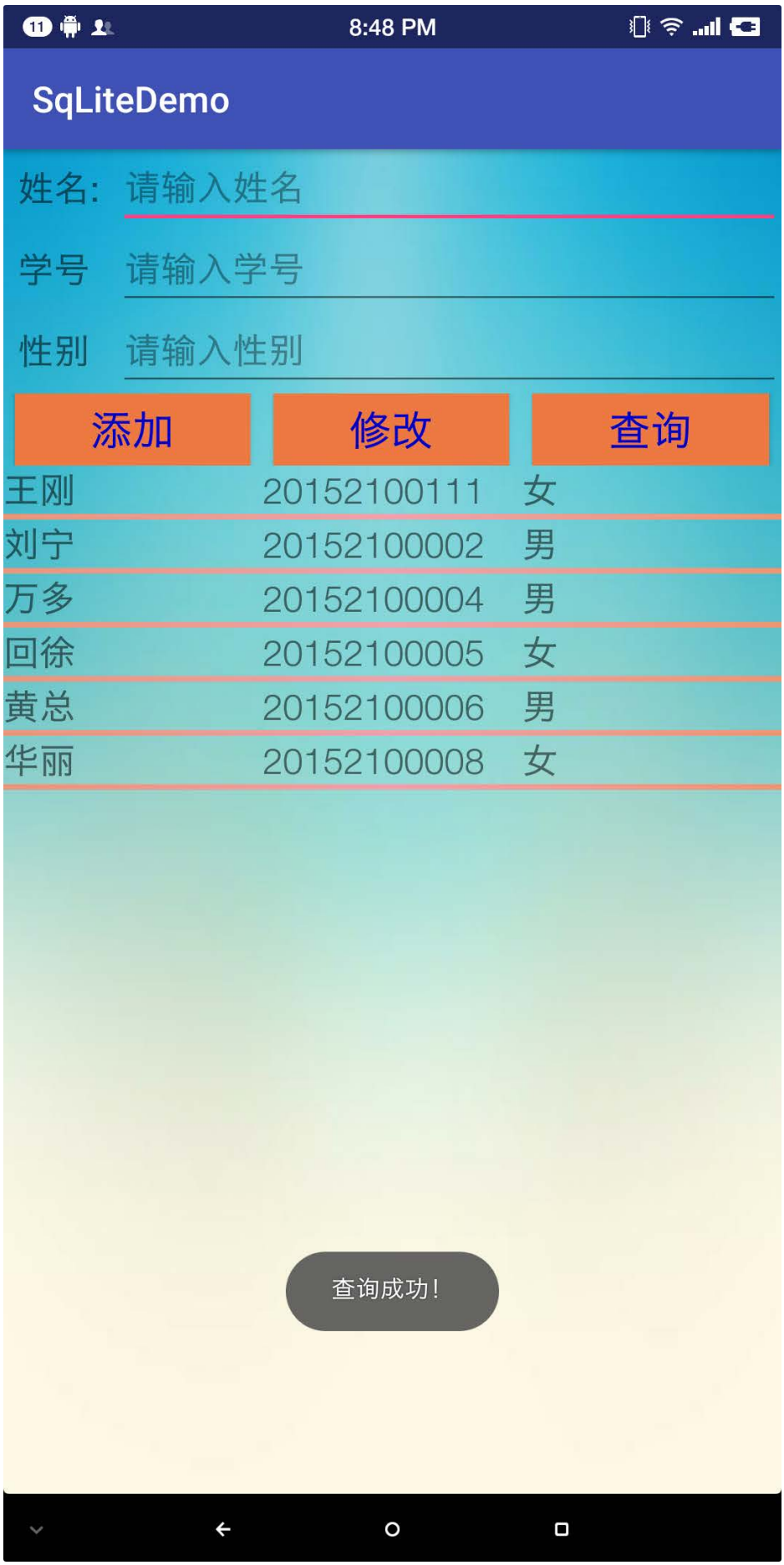
软件界面展示：

1

2


8:46 PM
SqLiteDemo
姓名:
请输入姓名
学号
请输入学号
性别
请输入性别
添加
修改
查询
王刚 20152100001 男
刘宁 20152100002 男
燕子 20152100003 女
万多 20152100004 男
回徐 20152100005 女
黄总 20152100006 男
华丽 20152100008 女


8:47 PM
SqLiteDemo
姓名: 王刚
学号 20152100111
性别 女
添加
修改
查询
王刚 20152100001 男
刘宁 20152100002 男
万多 20152100004 男
回徐 20152100005 女
黄总 20152100006 男
华丽 20152100008 女
修改成功!


8:48 PM
SqLiteDemo
姓名:
请输入姓名
学号
请输入学号
性别
请输入性别
添加
修改
查询
王刚 20152100111 女
刘宁 20152100002 男
万多 20152100004 男
回徐 20152100005 女
黄总 20152100006 男
华丽 20152100008 女
查询成功!


8:47 PM
SqLiteDemo
姓名:
请输入姓名
学号
请输入学号
性别
请输入性别
添加
修改
查询
刘宁 20152100002 男
王刚 20152100001 男
燕子 20152100003 女
万多 20152100004 男
回徐 20152100005 女
黄总 20152100006 男
华丽 20152100008 女


8:47 PM
SqLiteDemo
姓名:
请输入姓名
学号
请输入学号
性别
请输入性别
添加
修改
查询
刘宁 20152100002 男
王刚 20152100001 男
燕子 201521000
万多 20152100004 男
回徐 20152100005 女
黄总 20152100006 男
华丽 20152100008 女

代码设计：

1 增加修改数据库

```
147     private void insert(String name,String stu_num,String sex){
148
149         //实例化一个ContentValues用来装载待插入的数据
150         ContentValues values = new ContentValues();
151         values.put("name",name);
152         values.put("stuNum",stu_num);
153         values.put("sex",sex);
154         // 第一个参数为要插入的表名，
155         // 第二个为当values参数为空或者里面没有内容的时候，inser
156         // 第三个为插入的值
157         long id=db.insert( table: "Student", nullColumnHack: null,
158         //一定记得关闭数据库 避免内存泄漏
159         db.close();
160     }
161     //修改数据
162     public int update(String name,String stu_num,String sex){
163
164         //实例化一个ContentValues用来装载待修改的数据
165         ContentValues values = new ContentValues();
166         //这里不能改名字
167         //values.put("name",name);
168         values.put("stuNum",stu_num);
169         values.put("sex",sex);
170         //第一个为表名，第二个为修改的值，第三个为where 即修改的
171         //这里的参数也说明了，不能修改名字 因为传进来的name是存在
172         //要改名字得另起一个函数去根据别的参数去改
173         int number= db.update( table: "Student",values, whereClau
174         //一定记得关闭数据库 避免内存泄漏
175         db.close();
176         return number;
```

2 删除 查询某个数据

```
178         //删除数据
179         public int delete(String name){
180             //先打开数据库
181             SQLiteDatabase db = databaseHelper.getReadableDatabase()
182             ////第一个为表名，第二个为where即删除的位置在哪 第三个为
183             int number = db.delete( table: "Student", whereClause: "na
184             //一定记得关闭数据库 避免内存泄漏
185             db.close();
186             return number;
187         }
188
189         //查询根据name标识的某个数据
190         public boolean find(String name){
191             // 第一个为表名，第二个为要查询的列名，第三个是查询条件w
192             // 第五个为分组形式groupby，第六个为接受having条件，第七
193             // 利用游标遍历查询
194             Cursor cursor = db.query( table: "Student", columns: null,
195             boolean result = cursor.moveToNext();
196             // 记得关闭游标
197             cursor.close();
198             //一定记得关闭数据库 避免内存泄漏
199             db.close();
200             return result;
201         }
202
203         // 查询数据库中所有的数据
204         public void findAll(){
205             //先打开数据库
206             SQLiteDatabase db = databaseHelper.getReadableDatabase()
```

3 查询所有数据和获得数据库的行数

```
204     public void findAll(){
205         //先打开数据库
206         SQLiteDatabase db = databaseHelper.getReadableDatabase()
207         // 利用游标遍历查询
208         Cursor cursor = db.rawQuery( sql: "select * from Student
209         while (cursor.moveToNext()){
210             Log.i( tag: "move to next", msg: "11111111111");
211             //这里获取的是从第二列开始，因为第一列为自增长的ID
212             //获取第二列的值,索引从0开始 并加入到ArrayStr1中
213             ArrayStr1.add(cursor.getString( i: 1));
214             //获取第三列的值,索引从0开始
215             ArrayStr2.add(cursor.getString( i: 2));
216             //获取第四列的值,索引从0开始
217             ArrayStr3.add(cursor.getString( i: 3));
218             Log.i( tag: "ArrayStr1", String.valueOf(ArrayStr1));
219             Log.i( tag: "ArrayStr2", String.valueOf(ArrayStr2));
220         }
221         // 记得关闭游标
222         cursor.close();
223         //一定记得关闭数据库 避免内存泄漏
224         db.close();
225     }
226
227     // 获取数据库的行数
228     public int allCaseNum( ){
229         //先打开数据库
230         SQLiteDatabase db = databaseHelper.getReadableDatabase()
231         Cursor cursor = db.rawQuery( sql: "select count(*) from S
232         cursor.moveToFirst();
233         count = cursor.getInt( i: 0);
```

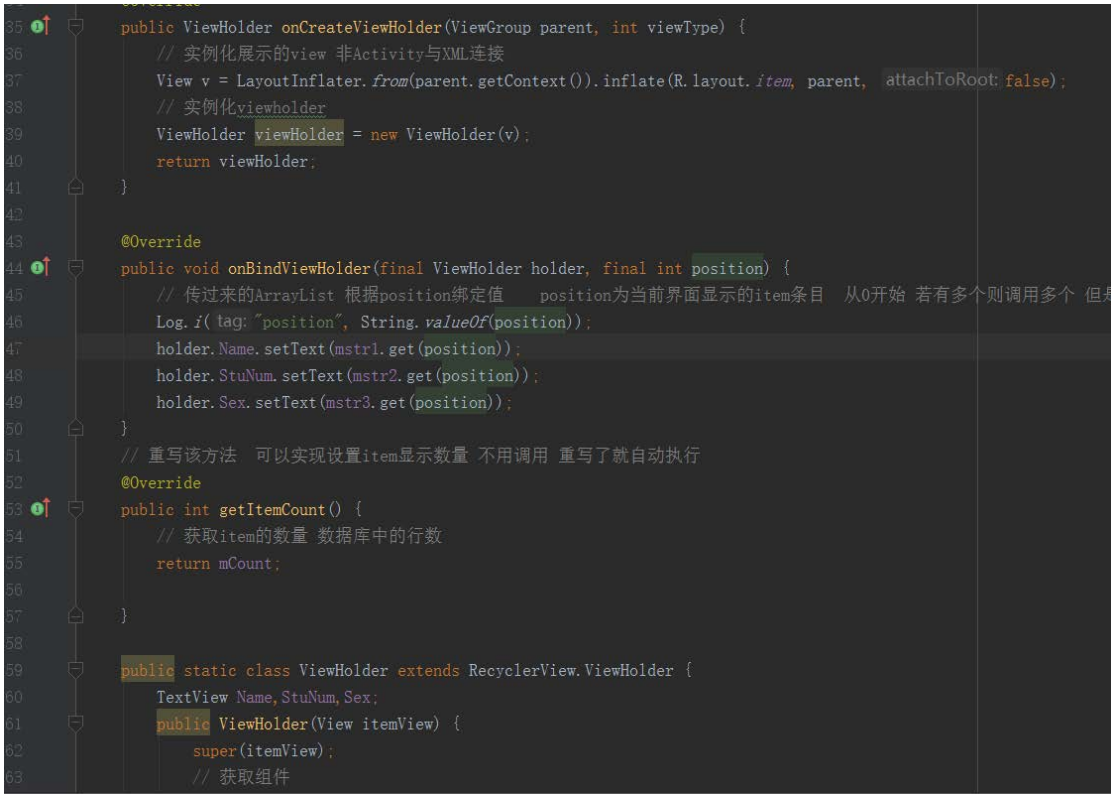
4 数据库帮助类

```
10
11      public class DataBaseHelper extends SQLiteOpenHelper {
12          private static final String DB_NAME = "mydata.db"; //数据库名
13          private static final int version = 1; //数据库版本
14
15          public DataBaseHelper(Context context) {
16              // SQLiteOpenHelper 的 参数 第一个为Context 其次是数据库名
17              super(context, DB_NAME,  factory: null, version);
18          }
19
20          // 创建数据库时被调用，通常在该方法中创建数据库表以及添加应用
21          @Override
22          public void onCreate(SQLiteDatabase sqLiteDatabase) {
23              //创建一个数据库表
24              sqLiteDatabase.execSQL("CREATE TABLE Student(id INTEGER
25                                      "name VARCHAR(10)," +
26                                      "stuNum VARCHAR(20)," +
27                                      "sex CHAR);");
28
29              //也可以写成:
30              /*String sql = "CREATE TABLE Student(id INTEGER PRIMARY K
31                      "sex CHAR);";
32              sqLiteDatabase.execSQL(sql);*/
33
```

5 间隔绘制

```
70          * 绘制间隔
71          */
72      private void drawVertical(Canvas c, RecyclerView parent) {
73          final int left = parent.getPaddingLeft();
74          final int right = parent.getWidth() - parent.getPaddingR
75          final int childCount = parent.getChildCount();
76          for (int i = 0; i < childCount; i++) {
77              final View child = parent.getChildAt(i);
78              final RecyclerView.LayoutParams params = (RecyclerVie
79                      .getLayoutParams();
80              final int top = child.getBottom() + params.bottomMarg
81                      Math.round(ViewCompat.getTranslationY(child)
82              final int bottom = top + mDivider.getIntrinsicHeight
83              mDivider.setBounds(left, top, right, bottom);
84              mDivider.draw(c);
85          }
86      }
87
88      /**
89       * 绘制间隔
90       */
91      private void drawHorizontal(Canvas c, RecyclerView parent) {
92          final int top = parent.getPaddingTop();
93          final int bottom = parent.getHeight() - parent.getPaddin
94          final int childCount = parent.getChildCount();
95          for (int i = 0; i < childCount; i++) {
96              final View child = parent.getChildAt(i);
97              final RecyclerView.LayoutParams params = (RecyclerVie
98                      .getLayoutParams();
```

6 适配器



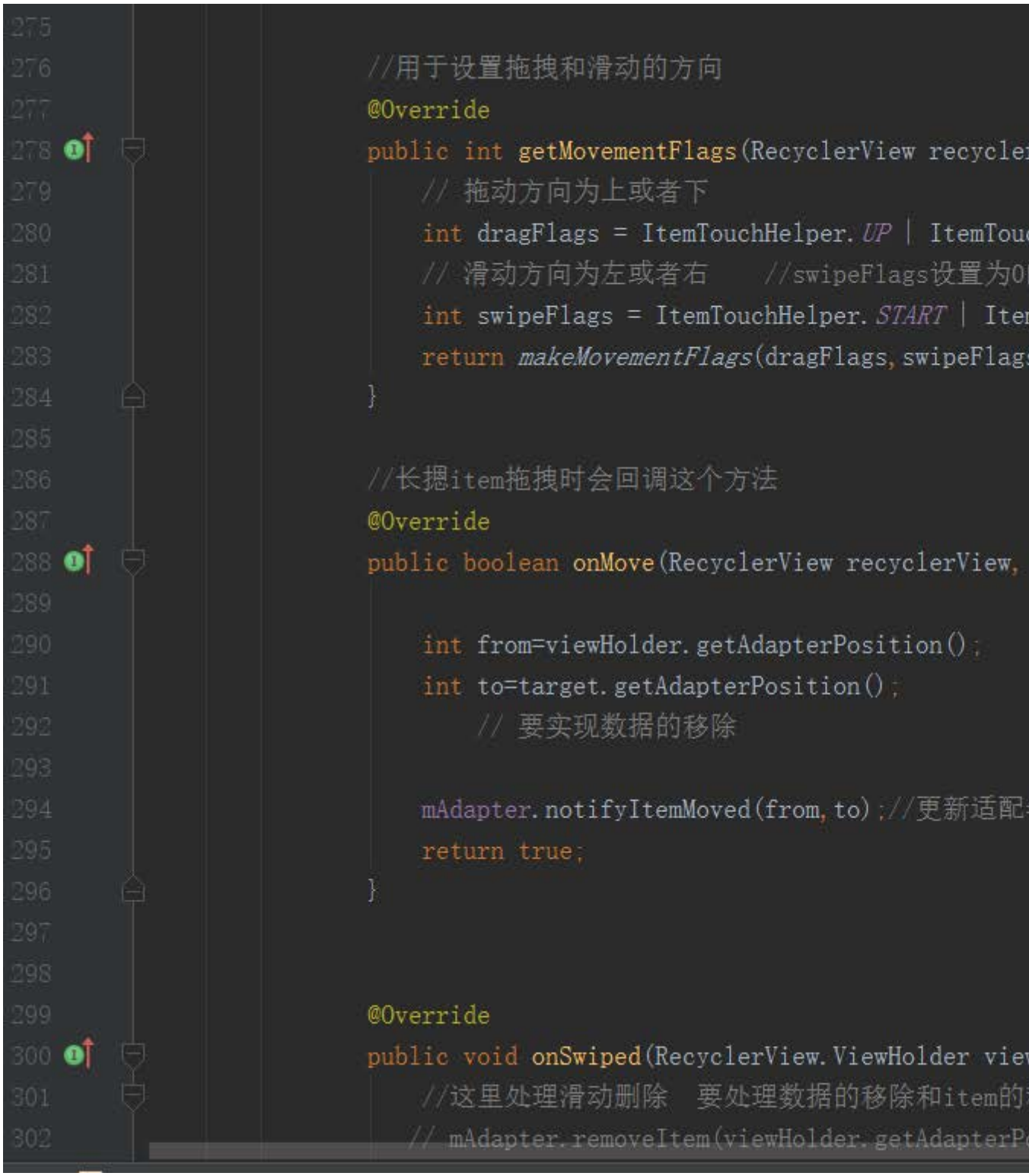
```
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        // 实例化展示的view 非Activity与XML连接
        View v = LayoutInflater.from(parent.getContext()).inflate(R.layout.item, parent,  attachToRoot: false);
        // 实例化viewholder
        ViewHolder viewHolder = new ViewHolder(v);
        return viewHolder;
    }

    @Override
    public void onBindViewHolder(final ViewHolder holder, final int position) {
        // 传过来的ArrayList 根据position绑定值    position为当前界面显示的item条目  从0开始 若有多个则调用多个 但是
        Log.i( tag: "position", String.valueOf(position));
        holder.Name.setText(mstr1.get(position));
        holder.StuNum.setText(mstr2.get(position));
        holder.Sex.setText(mstr3.get(position));
    }
    // 重写该方法  可以实现设置item显示数量 不用调用 重写了就自动执行
    @Override
    public int getItemCount() {
        // 获取item的数量 数据库中的行数
        return mCount;

    }

    public static class ViewHolder extends RecyclerView.ViewHolder {
        TextView Name,StuNum,Sex;
        public ViewHolder(View itemView) {
            super(itemView);
            // 获取组件
```

7 拖曳和滑动设置



```
275
276             //用于设置拖拽和滑动的方向
277             @Override
278             public int getMovementFlags(RecyclerView recycle
279                 // 拖动方向为上或者下
280                 int dragFlags = ItemTouchHelper.UP | ItemTou
281                 // 滑动方向为左或者右     //swipeFlags设置为0
282                 int swipeFlags = ItemTouchHelper.START | Ite
283                 return makeMovementFlags(dragFlags,swipeFlag
284             }
285
286             //长摁item拖拽时会回调这个方法
287             @Override
288             public boolean onMove(RecyclerView recyclerView,
289
290                 int from=viewHolder.getAdapterPosition();
291                 int to=target.getAdapterPosition();
292                     // 要实现数据的移除
293
294                 mAdapter.notifyItemMoved(from,to);//更新适配
295                 return true;
296             }
297
298
299             @Override
300             public void onSwiped(RecyclerView.ViewHolder vie
301                 //这里处理滑动删除  要处理数据的移除和item的
302                 // mAdapter.removeItem(viewHolder.getAdapterP
```

软件操作流程：

打开软件，输入用户的姓名，学号，性别，就可将数据保存到数据库中，点击修改，可以修改对应的值，

点击查询，会出现一个列表显示各项数据

难点及解决方案：

1 绘制 RecyclerView 的间距宽度和样式：

通过建立一个样式类：ItemDividerLine 去继承 RecyclerView.ItemDecoration

然后重写方法，再新建一个 XML 文件，设置分隔线的宽度和样式，在主题.xml 中

加上

```
<item name="android:listDivider">@drawable/divider_bg</item>
```

即可

2 设置 RecyclerView 可拖曳移动和滑动删除

通过 ItemTouchHelper 类，重写方法，即可设置，具体参考代码

3 滑动删除要删除数据库的内容

先获取当前的 recyclerView 的 position，再通过该 position 去获得对应的组件的 name 值，再传入 name 值给删除数据库方法即可

不足之处及今后设想：

加上滑动删除时会有删除的按钮，点击该按钮才删除