aaa

**软工pk项目**



**题 目： Android版学生成绩管理系统详细设计**

**院 系： 软件学院 专 业： 软件工程**

**姓 名： 赵奇隆 学 号： 131110331**

# 详细设计

## 引言

### 编写目的

### 项目背景

### 定义

1. 详细设计
2. 类图
3. 数据库设计

### 参考资料

1. 项目需求说明书；
2. 项目概要设计说明书。

## 任务概述

### 目标

### 运行环境

### 需求概述

## 系统模块



## 程序描述

### 录入学生信息

1 .**功能**：用户通过手机上的软键盘，进行学生学号，姓名，工数成绩，代几成绩，C语言成绩，照片，的录入同时包含了验证输入学号是否重复，输入的数字是否符合规格等功能。在输入成绩之后，要进行总成绩和平均成绩的计算

2. **性能**：要求点击确认后要对输入内容是否正确进行检测，检测的总时间不能超过500ms，同时在检测完成之后，如果出现错误提示输入错误，如果输入正确，则清空录入信息，进行下一次输入，更新处理时间不能超过200ms

3. **输入项目**：学生学号，学生姓名，学生工数成绩，学生代几成绩，学生C语言成绩，学生照片

4. **输出项目**: 录入成功则弹出Toast提示，提示录入成功，输入有错误时，提示具体错误，包括：学号输入非法字符，成绩输入非法字符，成绩小于0或者大于100，缺少某一部分的数据没有录入

5. **算法**：底层SQL语句，和字符串的检验

6. **程序逻辑**：



7. **限制条件**：输入学号不能重复，不能非法（包括负数），成绩录入必须在0—100之间，所有的输入项都不能为空。

8. **测试要点**：

|  |  |  |
| --- | --- | --- |
| 测试项目 | 测试点 | 结果和要求 |
| 拍照测试 | 进入拍照功能进行拍照 | 获取照片成功并进入自动裁剪界面 |
| 照片裁剪测试 | 拍完的照片能进行裁剪保存 | 裁剪成功，并能在界面上显示裁剪后的照片 |
| 结束输入 | 点击返回 | 返回到上一个界面，输入结束 |
| 学号输入 | 学号输入正确 | 最后点击确认提示录入成功 |
| 学号为空 | 弹出学号不能为空的提示 |
| 学号重复 | 弹出已经存在提示 |
| 学号输入非法字符 | 弹出学号输入错误提示 |
| 成绩输入 | 成绩为空 | 弹出来成绩信息不完整提示 |
| 成绩小于0或者大于100 | 弹出来成绩录入错误提示 |
| 成绩输入存在非法字符 | 弹出来成绩录入错误提示 |
| 成绩均录入正确 | 最后点击确认提示录入成功 |
| 姓名输入 | 姓名为空 | 弹出姓名为空的错误提示 |
| 姓名不为空 | 最后点击确认提示录入成功 |

### 查找学生信息

1 .**功能**：用户通过手机上的软键盘，进行学生学号输入，点击确认按钮，搜索到相应学号的学生，然后更新用户界面，显示搜索到的学生的学号，姓名，照片，成绩，如果搜索不到，则弹出搜索不到的相关提示

2. **性能**：要求点击确认后要对输入内容是否正确进行检测，检测的总时间不能超过500ms，同时在检测完成之后，进行学生搜索，搜索时间不超过200ms

3. **输入项目**：学生学号

4. **输出项目**: 搜索成功，找到学生，将学生的信息在界面上更新显示，搜索不到的话，在界面上显示没有搜索到相关学生

6. **程序逻辑**：



7. **限制条件**：学号要求必须是正整数

8. **测试要点**：

|  |  |  |
| --- | --- | --- |
| 测试项目 | 测试点 | 结果和要求 |
| 学号输入 | 学号正常 | 显示学生信息或者提示没有找到 |
| 学号为空 | 提示学号输入为空 |
| 学号输入非法字符 | 提示学号输入错误 |

### 基本信息展示信息

1 .**功能**：在进入学生列表时候调用系统数据库获取学生列表，然后在界面上显示学生的姓名，学号，总成绩，平均成绩，各科成绩，默认的排序是按照学号从小到大排列，在数据库中没有数据的时候，打开该页面显示没有数据提示，加强用户体验

2. **性能**：打开页面之后立刻显示列表，处理生成列表的时间不低于200ms

3. **输入项目**：无

4. **输出项目**: 按照学号大小排序的学生列表

5. **算法**：底层SQL语句，相关排序算法

6. **程序逻辑**：



7. **限制条件**：无

8. **测试要点**：

|  |  |  |
| --- | --- | --- |
| 测试项目 | 测试点 | 结果和要求 |
| 排序 | 数据库中乱序存入学生学号 | 显示的时候进行学号升序排序 |
| 数据读取显示 | 数据库存在数据 | 显示列表 |
| 数据库不存在数据 | 提示数据库中不存在数据 |

### 排名信息展示

1 .**功能**：在现实学生排名界面，从数据库中读取学生的信息，读取学生的总成绩等，然后进行排序，最后将排完序的列表显示在页面上，同时在前边标上序号，如果数据库中没有数据，那么界面提示，数据库中不存在数据

2. **性能**：打开页面之后立刻显示列表，处理生成排序列表的时间不低于200ms

3. **输入项目**：无

4. **输出项目**: 按照总成绩从高到低排序的学生列表

5. **算法**：底层SQL语句，排序算法

6. **程序逻辑**：



7. **限制条件**：无

8. **测试要点**：

|  |  |  |
| --- | --- | --- |
| 测试项目 | 测试点 | 结果和要求 |
| 排序 | 数据库中乱序存入学生成绩 | 显示的时候进行总成绩降序排序 |
| 数据读取显示 | 数据库存在数据 | 显示列表 |
| 数据库不存在数据 | 提示数据库中不存在数据 |

### 删除学生信息

1 .**功能**：在学生信息列表界面，再有数据的情况下，长按某一项学生的列表，弹出是否进行删除的提示框，点击取消，提示框消失，点击确认，调用系统数据库删除数据，同时更新列表，弹出Toast提示删除。

2. **性能**：要求点击确认删除之后，延迟总时间不能超过200ms，同时在删除完成之后，更新列表的时间不能超过200ms

3. **输入项目**：是否删除的的确认

4. **输出项目**: 删除之后更新的列表，和成功删除的Toast提示

6. **程序逻辑**：



7. **限制条件**：无

8. **测试要点**：

|  |  |  |
| --- | --- | --- |
| 测试项目 | 测试点 | 结果和要求 |
| 确认删除 | 点击确认删除 | 列表更新，弹出已删除的提示，对话框消失 |
| 取消删除 | 点击删除确认中的取消 | 对话框消失，其他不发生任何变化 |

## 类图设计

### 总类图关系



### 类的详细设计

MainActivity类描述

|  |
| --- |
| 属性 |
| -btnAddStudent：Button 界面的按钮控件，用来跳转到录入学生信息的界面 |
| -btnShowList：Button 界面的按钮控件，用来跳转到学生列表的界面 |
| -btnRankList：Button 界面的按钮控件，用来跳转到按照成绩排序后的列表的界面 |
| -btnSearchStudent：Button 界面的按钮控件，用来跳转到按照学号搜索的界面 |
| 方法 |
| #onCreate():void 当Android界面第一次创建的时候调用放置逻辑操作 |

ShowListActivity类描述

|  |
| --- |
| 属性： |
| -showListListView：ListView 列表类，用于显示学生列表 |
| -showListAdapter：ShowListAdapter 列表适配器，用于加载列表布局 |
| -dbManageHelper：DBManageHelper 数据库操作类，用于获取排序后学生列表 |
| 方法： |
| #onDestory():void 当界面销毁的时候进行调用，在里边关闭数据库 |
| #onCreate():void 当Android界面第一次创建的时候调用放置逻辑操作 |

RankActivity类的设计：

|  |
| --- |
| 属性： |
| -rankListView:ListView 列表类，用于显示学生排名列表 |
| -rankListAdapter：RankListAdapter 列表适配器，用于加载列表布局 |
| -dbManageHelper：DBManageHelper 数据库操作类，用于获取排序后学生列表 |
| 方法： |
| #onDestory():void 当界面销毁的时候进行调用，在里边关闭数据库 |
| #onCreate():void 当Android界面第一次创建的时候调用放置逻辑操作 |

InputActiviy类详细描述

|  |
| --- |
| 属性： |
| -photoHelper:PhotoHelper 拍照操作类 |
| -rankListAdapter：RankListAdapter 列表适配器，用于加载列表布局 |
| -dbManageHelper：DBManageHelper 数据库操作类，用于获取排序后学生列表 |
| -edtNameInput:EditText 输入框控件，用于输入学生姓名 |
| -edtIDInput:EditText 输入框控件，输入学生的ID |
| -edtCScoreInput:EditText 输入框控件，输入学生的C语言成绩 |
| -edtMathScoreInput:EditText 输入框控件，输入学生的工数成绩 |
| -edtAlgebraScoreInput:EditText 输入框控件，输入学生的代几成绩 |
| -btnConfirmInput：Button 确认录入按钮 |
| 方法： |
| #onDestory():void 当界面销毁的时候进行调用，在里边关闭数据库 |
| #onCreate():void 当Android界面第一次创建的时候调用放置逻辑操作 |

SearchActivity类的描述

|  |
| --- |
| 属性： |
| -searchListView：ListView 列表类，用于显示搜索到的学生列表 |
| -showListAdapter：ShowListAdapter 列表适配器，用于加载列表布局 |
| -dbManageHelper：DBManageHelper 数据库操作类，用于获取排序后学生列表 |
| -btnSearch:Button 确认搜索按钮 |
| 方法： |
| #onDestory():void 当界面销毁的时候进行调用，在里边关闭数据库 |
| #onCreate():void 当Android界面第一次创建的时候调用放置逻辑操作 |

Student类描述

|  |
| --- |
| 属性 |
| -id:int 学号 |
| -name:String 姓名 |
| -MathScore:int 工数成绩 |
| -algebraScore:int 代几成绩 |
| -CScore:int C语言成绩 |
| -totalScore:int 总成绩 |
| -averScore:double 平均成绩 |
| -photo:String 照片存放路径 |
| 方法： |
| +Student(id:int,name:String,MathAcore:int,algebraScore:int,CScore:int) 构造方法 |
| +getId()：int |
| +setId(id:int):void |
| +getName():String |
| +setName(name:String):void |
| +getMathScore():int |
| +setMathScore(score:int):void |
| +getAlgebraScore():int |
| +getAlgebraScore(score:int):void |
| +setCScore(score:int):void |
| +getPhoto():String |
| +getCScore():int |
| +setPhoto(derictory:String):void |
| +getTotalScore():int |
| +getAverScore():double |

ListAdapter类描述

|  |
| --- |
| 属性： |
| 方法： |
| +ListAdapter（context:Context,layout:int,cursor:cursor,flag:int）构造函数 |
| +bindView() 加载布局view |

RandListAdapter类描述

|  |
| --- |
| 属性： |
| 方法： |
| +ListAdapter（context:Context,layout:int,cursor:cursor,flag:int）构造函数 |
| +bindView() 加载布局view |

PhotoHelper类描述

|  |
| --- |
| 属性： |
| -context ：Context 上下文 |
| 方法： |
| +takePhoto():void 拍照 |
| +savePhoto():String 保存照片，并且返回照片保存路径 |
| +cutPhoto():void 照片裁剪，裁剪部分照片当头像 |
| +PhotoHelper(context : Context) 构造函数 |

DBHelper类描述

|  |
| --- |
| 属性： |
| -context:Context 上下文 |
| -sqliteDataBase:SQLiteDataBase 数据库实例 |
| 方法： |
| +DBHelper(context:Context,name :String,factory:CursorFactory,version:int) 构造函数，参数分别是：上下文，数据库名，游标工厂，数据库版本 |
| +onCreate():void 当创建数据库的时候调用 |
| +onUpgrade(arg0 :SQLiteDatabase,arf1:int,arg2:int):void 数据库升级调用，参数分别是数据库实例，以前的版本和要升级的版本 |
| +saveStudent(student:Student):void 将一个学生保存进数据库中 |
| +getStudentList():ArrayList<Student> 获取系统数据库中所有的学生的信息（按照学号排序） |
| +deleteStudentByID(id:int):void 按照学号删除学生 |
| +getStudentByID(id:int) : Student 根据学号从数据库中去数据然后获取到一个学生的实例 |

## 数据库设计

### 数据库结构设计

数据库表关系图：



### 数据字典设计

1．Student(学生表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 类型 | 是否为空 | 默认值 | 备注 |
| Student\_id | 学号 | Int(255) | 否 | Null | 主键、外键 |
| StudentName | 学生姓名 | Varchar(MAX) | 否 | Null |  |
| PhotoDirectory | 照片所在目录 | Text | 是 | Null |  |

2．StudentScore(学生成绩表)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 类型 | 是否为空 | 默认值 | 备注 |
| Student\_id | 学号 | Int(255) | 否 | Null | 主键、外键 |
| MathScore | 工数成绩 | Int(3) | 否 | Null |  |
| AlgebraScore | 线代成绩 | Int(3) | 否 | Null |  |
| CScore | C语言成绩 | Int(3) | 否 | Null |  |

3．TotalAverageScore(总成绩\_平均成绩表)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 类型 | 是否为空 | 默认值 | 备注 |
| Student\_id | 学号 | Int(255) | 否 | Null | 主键、外键 |
| TotalScore | 总成绩 | Int(3) | 否 | Null |  |
| AverageScore | 平均成绩 | Int(3) | 否 | Null |  |