aaa

**软工pk项目**



**A 题： Android版学生成绩管理系统**

**B 题： Android版学习任务管理工具**

**院 系： 软件学院 专 业： 软件工程**

**姓 名： 杨琳 学 号： 131110421**

**姓 名： 赵奇隆 学 号： 131110331**

摘 要

A题：现阶段是大数据阶段，传统的数据存储方式已经不够用了。但是需要进行数据存储和管理的地方越来越多。这其中，对学生成绩的管理尤为重要。为了方便对学生成绩进行存储和管理，开发出一个小型Andriod手机端系统——学生成绩管理系统。该系统面向的对象为小群体，比如说班级的学生成绩管理员。本系统能够方便成绩管理员对班级成绩进行管理，及时掌控班级成绩信息。该系统的总体模块为管理员模块，此模块下含有录入学生信息、查找学生信息、展示学生信息和删除学生信息四个子模块，展示学生信息模块下又包含基本信息展示和排名信息展示两个二级子模块。提供学生及成绩的详细信息——姓名，学号，C语言成绩，工数成绩，代几成绩的录入、计算、按学号查询、按排名展示、删除学生这几个功能。本系统基于Java程序设计语言，遵循软件项目开发流程，采用面向对象技术开发。此外，本系统采用了基于SQLite轻型数据库的数据库技术。经过功能测试与性能评估，本系统能基本实现其预期功能。

B题：当今，大学生的自制能力越来越不容乐观，但是大学生要完成的任务无论是从种类还是数量上却是越来越多。为提醒大学生及时完成自己的学习等任务，开发出一个小型Andriod手机端系统——学生学习任务管理系统。该系统面向的对象即为当代大学生。该系统的总体模块为学生模块，此模块下含有添加任务、删除任务、修改任务、查找任务和查看所有任务五个子模块。提供学习任务的详细信息——学习任务名称、具体内容、任务照片、任务类型、任务重要性，任务创建的时间的添加、修改、删除、查找和查看功能。本系统基于Java程序设计语言，遵循软件项目开发流程，采用面向对象技术开发。此外，本系统采用了基于SQLite轻型数据库的数据库技术。经过功能测试与性能评估，本系统能基本实现其预期功能。

关键词 学生学习任务管理系统；学生成绩管理系统；面向对象技术；Java程序设计语言；SQLite

Abstract

A：This stage is the big data stage, traditional way of data storage is not enough. But the need for data storage and management more and more. Among them, the management of student achievement is especially important. In order to facilitate the student achievement for storage and management, to develop a small android mobile phone system, student achievement management system. The system oriented object for small groups, such as the class's and grade's student achievement administrator. This system can facilitate the administrator to manage class performance, control class performance information in a timely manner. General module of the system as the administrator module, this module contains information into students, students to find students information, display information and delete student information four sub modules, again under student information module contains basic information display and ranking information display two secondary sub modules. The system provides the students and details - name, student id, C language result, Numbers of scores, generation of several grades of inputting, calculating, according to the student number query, show according to the rankings, delete students these functions. This system based on Java programming language, follow the software project development process, using object-oriented technology development. In addition, this system USES a database technology based on light SQLite database. After function test and performance evaluation, this system can realize the expected function.

B：Nowadays, college students' self-control ability is more and more is not optimistic, but college students to complete the task from the kinds and quantity is more and more. To remind students finish their learning task, such as to develop a small android mobile terminal system, students learning task management system. The system oriented object is the contemporary college students. General module of the system module for the students, under this module contains add tasks task, delete, modify tasks, find the tasks and view all task five modules. Provide detailed information learning tasks, learning task name, content, task, photo, task type, task significance, task creation time to add, modify, delete, search and view the function. This system based on Java programming language, follow the software project development process, using object-oriented technology development. In addition, this system USES a database technology based on light SQLite database. After function test and performance evaluation, this system can realize the expected function.

Key words：Students learning task management system; Java programming language; Object-oriented; technology,SQLite

目录

[摘 要 2](#_Toc420590496)

[Abstract 3](#_Toc420590497)

[目录 4](#_Toc420590498)

[第1章 需求分析(A)（赵奇隆编写） 9](#_Toc420590499)

[1.1 前言 9](#_Toc420590500)

[1.1.1 编写目的 9](#_Toc420590501)

[1.1.2 参考资料 9](#_Toc420590502)

[1.2 可行性分析 9](#_Toc420590503)

[1.2.1 功能要求 9](#_Toc420590504)

[1.2.2 性能要求 9](#_Toc420590505)

[1.2.3 目标 9](#_Toc420590506)

[1.2.4 人员 9](#_Toc420590507)

[1.3 数据描述 9](#_Toc420590508)

[1.3.1 静态数据 9](#_Toc420590509)

[1.3.2 动态数据 10](#_Toc420590510)

[1.3.3 数据库介绍 10](#_Toc420590511)

[1.3.4 数据采集 10](#_Toc420590512)

[1.4 功能需求 10](#_Toc420590513)

[1.4.1 功能介绍 10](#_Toc420590514)

[1.4.2 用例描述 11](#_Toc420590515)

[第2章 需求分析（B）（杨琳编写） 14](#_Toc420590516)

[2.1 前言 14](#_Toc420590517)

[2.1.1 编写目的 14](#_Toc420590518)

[2.1.2 参考资料 14](#_Toc420590519)

[2.2 可行性分析 14](#_Toc420590520)

[2.2.1 功能要求 14](#_Toc420590521)

[2.2.2 性能要求 14](#_Toc420590522)

[2.2.3 目标 14](#_Toc420590523)

[2.2.4 人员 14](#_Toc420590524)

[2.3 数据描述 14](#_Toc420590525)

[2.3.1 静态数据 14](#_Toc420590526)

[2.3.2 动态数据 15](#_Toc420590527)

[2.3.3 数据库介绍 15](#_Toc420590528)

[2.3.4 数据采集 15](#_Toc420590529)

[2.4 功能需求 15](#_Toc420590530)

[2.4.1 功能介绍 15](#_Toc420590531)

[2.4.2 用例描述 16](#_Toc420590532)

[第3章 概要设计（A）（杨琳编写） 21](#_Toc420590533)

[3.1 引言 21](#_Toc420590534)

[3.1.1 编写目的 21](#_Toc420590535)

[3.1.2 项目背景 21](#_Toc420590536)

[3.1.3 定义 21](#_Toc420590537)

[3.1.4 参考资料 21](#_Toc420590538)

[3.2 任务概述 21](#_Toc420590539)

[3.2.1 目标 21](#_Toc420590540)

[3.2.2 运行环境 21](#_Toc420590541)

[3.2.3 需求概述 21](#_Toc420590542)

[3.3 系统分析 21](#_Toc420590543)

[3.3.1 领域模型 21](#_Toc420590544)

[3.3.2 时序图 22](#_Toc420590545)

[3.4 系统概要设计 26](#_Toc420590546)

[3.4.1 整体包与类设计 26](#_Toc420590547)

[3.4.2 开发环境与开发工具 26](#_Toc420590548)

[3.5 数据库概念结构设计 27](#_Toc420590549)

[第4章 概要设计（B）（赵奇隆编写） 28](#_Toc420590550)

[4.1 引言 28](#_Toc420590551)

[4.1.1 编写目的 28](#_Toc420590552)

[4.1.2 项目背景 28](#_Toc420590553)

[4.1.3 定义 28](#_Toc420590554)

[4.1.4 参考资料 28](#_Toc420590555)

[4.2 任务概述 28](#_Toc420590556)

[4.2.1 目标 28](#_Toc420590557)

[4.2.2 运行环境 28](#_Toc420590558)

[4.2.3 需求概述 28](#_Toc420590559)

[4.3 系统分析 29](#_Toc420590560)

[4.3.1 领域模型 29](#_Toc420590561)

[4.3.2 时序图 30](#_Toc420590562)

[4.4 系统概要设计 35](#_Toc420590563)

[4.4.1 整体包与类设计 35](#_Toc420590564)

[4.4.2 开发环境与开发工具 36](#_Toc420590565)

[4.5 数据库概念结构设计 37](#_Toc420590566)

[第5章 详细设计（A）（赵奇隆编写） 38](#_Toc420590567)

[5.1 引言 38](#_Toc420590568)

[5.1.1 编写目的 38](#_Toc420590569)

[5.1.2 项目背景 38](#_Toc420590570)

[5.1.3 定义 38](#_Toc420590571)

[5.1.4 参考资料 38](#_Toc420590572)

[5.2 任务概述 38](#_Toc420590573)

[5.2.1 目标 38](#_Toc420590574)

[5.2.2 运行环境 38](#_Toc420590575)

[5.2.3 需求概述 39](#_Toc420590576)

[5.3 系统模块 39](#_Toc420590577)

[5.4 程序描述 39](#_Toc420590578)

[5.4.1 录入学生信息 39](#_Toc420590579)

[5.4.2 查找学生信息 42](#_Toc420590580)

[5.4.3 基本信息展示信息 44](#_Toc420590581)

[5.4.4 排名信息展示 45](#_Toc420590582)

[5.4.5 删除学生信息 46](#_Toc420590583)

[5.5 类图设计 48](#_Toc420590584)

[5.5.1 总类图关系 48](#_Toc420590585)

[5.5.2 类的详细设计 48](#_Toc420590586)

[5.6 数据库设计 52](#_Toc420590587)

[5.6.1 数据库结构设计 52](#_Toc420590588)

[5.6.2 数据字典设计 52](#_Toc420590589)

[第6章 详细设计（B）（杨琳编写） 54](#_Toc420590590)

[6.1 引言 54](#_Toc420590591)

[6.1.1 编写目的 54](#_Toc420590592)

[6.1.2 项目背景 54](#_Toc420590593)

[6.1.3 定义 54](#_Toc420590594)

[6.1.4 参考资料 54](#_Toc420590595)

[6.2 任务概述 54](#_Toc420590596)

[6.2.1 目标 54](#_Toc420590597)

[6.2.2 运行环境 54](#_Toc420590598)

[6.2.3 需求概述 55](#_Toc420590599)

[6.3 系统模块 56](#_Toc420590600)

[6.4 程序描述 57](#_Toc420590601)

[6.4.1 录入学习任务 57](#_Toc420590602)

[6.4.2 删除学习任务 59](#_Toc420590603)

[6.4.3 修改学习任务 61](#_Toc420590604)

[6.4.4 查找学习任务 63](#_Toc420590605)

[6.4.5 查看所有学习任务 65](#_Toc420590606)

[6.4.6 查看单个学习任务 66](#_Toc420590607)

[6.5 类图设计 68](#_Toc420590608)

[6.5.1 总类图关系 68](#_Toc420590609)

[6.5.2 类的详细设计 68](#_Toc420590610)

[6.6 数据库设计 72](#_Toc420590611)

[6.6.1 数据库结构设计 72](#_Toc420590612)

[6.6.2 数据字典设计 72](#_Toc420590613)

[第7章 实现（A）（杨琳实现） 74](#_Toc420590614)

[7.1 主界面 74](#_Toc420590615)

[7.2 录入信息界面 75](#_Toc420590616)

[7.3 学生列表界面 76](#_Toc420590617)

[7.4 学生排名界面 76](#_Toc420590618)

[7.5 查找学生界面 77](#_Toc420590619)

[第8章 实现（B）（赵奇隆实现） 78](#_Toc420590620)

[8.1 任务展示列表界面 78](#_Toc420590621)

[8.2 任务录入界面 78](#_Toc420590622)

[8.3 单个任务显示界面 79](#_Toc420590623)

[8.4 任务删除界面 79](#_Toc420590624)

[8.5 拍照功能 80](#_Toc420590625)

[8.6 任务搜索功能 80](#_Toc420590626)

[第9章 测试（A）（赵奇隆实现） 81](#_Toc420590627)

[9.1 引言 81](#_Toc420590628)

[9.1.1 编写目的 81](#_Toc420590629)

[9.1.2 定义 81](#_Toc420590630)

[9.1.3 参考资料 82](#_Toc420590631)

[9.2 任务概述 82](#_Toc420590632)

[9.2.1 运行环境 82](#_Toc420590633)

[9.3 计划 82](#_Toc420590634)

[9.3.1 测试方案 82](#_Toc420590635)

[9.3.2 测试项目 83](#_Toc420590636)

[9.3.3 测试机构及人员 83](#_Toc420590637)

[9.3.4 测试条件 83](#_Toc420590638)

[9.4 黑盒测试 83](#_Toc420590639)

[9.4.1 录入学习任务： 83](#_Toc420590640)

[9.4.2 删除学习任务： 84](#_Toc420590641)

[9.4.3 修改学习任务： 85](#_Toc420590642)

[9.4.4 查找学习任务 86](#_Toc420590643)

[9.4.5 查看所有学习任务： 86](#_Toc420590644)

[9.4.6 查看单个学习任务 87](#_Toc420590645)

[9.5 白盒测试 88](#_Toc420590646)

[9.5.1 1.录入学生信息白盒测试 88](#_Toc420590647)

[9.5.2 2.查找学生信息白盒测试 90](#_Toc420590648)

[9.5.3 学生的基本列表信息展示白盒测试： 91](#_Toc420590649)

[9.5.4 排名信息展示白盒测试 92](#_Toc420590650)

[9.5.5 删除学生信息白盒测试： 93](#_Toc420590651)

[9.6 软件缺陷 93](#_Toc420590652)

[第10章 测试（B）（杨琳实现） 94](#_Toc420590653)

[10.1 引言 94](#_Toc420590654)

[10.1.1 编写目的 94](#_Toc420590655)

[10.1.2 定义 94](#_Toc420590656)

[10.1.3 参考资料 95](#_Toc420590657)

[10.2 任务概述 95](#_Toc420590658)

[10.2.1 运行环境 95](#_Toc420590659)

[10.3 计划 95](#_Toc420590660)

[10.3.1 测试方案 95](#_Toc420590661)

[10.3.2 测试项目 96](#_Toc420590662)

[10.3.3 测试机构及人员 96](#_Toc420590663)

[10.3.4 测试条件 96](#_Toc420590664)

[10.4 黑盒测试 96](#_Toc420590665)

[10.4.1 录入学生信息 96](#_Toc420590666)

[10.4.2 查找学生信息 97](#_Toc420590667)

[10.4.3 基本信息展示 98](#_Toc420590668)

[10.4.4 排名信息展示 98](#_Toc420590669)

[10.4.5 删除学生信息 99](#_Toc420590670)

[10.5 白盒测试 100](#_Toc420590671)

[10.5.1 录入学习任务白盒测试 100](#_Toc420590672)

[10.5.2 删除学习任务白盒测试 101](#_Toc420590673)

[10.5.3 修改学习任务白盒测试 102](#_Toc420590674)

[10.5.4 查找学习任务白盒测试 103](#_Toc420590675)

[10.5.5 查看所有学习任务白盒测试 104](#_Toc420590676)

[10.5.6 查看单个任务白盒测试 105](#_Toc420590677)

[10.5.7 完成任务白盒测试 106](#_Toc420590678)

[10.6 软件缺陷 106](#_Toc420590679)

[第11章 总结（A） 107](#_Toc420590680)

[第12章 总结（B） 108](#_Toc420590681)

[第13章 附录： 109](#_Toc420590682)

[第14章 互评 110](#_Toc420590683)

# 需求分析(A)（赵奇隆编写）

## 前言

### 编写目的

需求分析明确和定义了用户的需求，方便在之后的概要设计和详细设计中，更好的对系统进行设计，逐步开发强壮的系统框架，使设计适合于实施环境，提高系统的性能。通过需求分析，来明确用户的需求，安排项目规划与进度、组织软件开发与测试，方便后期工作的进行。

本文档面向的读者有经理、设计人员、开发人员。

### 参考资料

《软件工程——原理方法与应用》

《面向对象设计与UML》

## 可行性分析

### 功能要求

1) 管理人员能够添加学生基本信息，包括姓名，学号，三科成绩。

2) 管理人员能够查看所有学生的列表。

3) 管理人员能在查看学生列表的同时进行学生的删除。

4) 管理人员能按照学生总成绩输出学生的总排名表。

5) 管理人员能够按照学号查询学生。

### 性能要求

1) 响应时间：200ms之内。

2) 更新处理时间：200ms之内。

3) 数据的转换和传送时间：3000ms之内。

4) 操作系统：Android手机系统。

### 目标

该应用将成为一个普通的学生成绩管理系统的小Demo。

### 人员

一个人。

## 数据描述

### 静态数据

1) 相关标题栏。

2) 按钮上边的文字。

### 动态数据

学生信息，包括姓名，学号，学生照片，学生成绩，学生名次等。

### 数据库介绍

SQLite，是一款轻型的数据库，是遵守ACID的关系型数据库管理系统，它包含在一个相对小的C库中。它是D.RichardHipp建立的公有领域项目。它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，它占用资源非常的低，在嵌入式设备中，可能只需要几百K的内存就够了。

### 数据采集

学生信息由管理员手动输入，学生照片由管理员自行拍照录入，如果没有录入就使用默认照片头像。

## 功能需求

### 功能介绍

#### 功能划分

功能划分图如下：



#### 功能描述

1) 学生基本信息的录入，包括姓名，学号，C语言成绩，工数成绩，代几成绩。

2) 按照学号查询学生相关信息，并展示相关信息列表。

3) 系统数据库中所有的学生的列表展示，包括排名表的展示。

4) 从数据库中删除某个学生的基本信息。

### 用例描述



Figure 1系统用例图

用例图1：学生信息录入用例描述

|  |
| --- |
| **用例编号 ： User\_1 用例名称：学生信息录入** |
| **简要说明** |
| 用户已进入录入信息界面，通过输入框录入学生个人信息 |
| **执行者** |
| 用户 |
| **前置条件** |
| 用户进入录入学生信息界面 |
| **后置条件** |
| 录入的学生信息被存入数据库中，相关图片存入相关路径下的文件夹中，图片路径同时存如数据库 |
| **事件流** |
| 1. 基本事件流   1.1手指点击学号输入框，手机屏幕显示键盘，用户进行输入  1.2手指点击姓名输入框，手机屏幕显示键盘，用户进行输入  1.3手指点击工数成绩输入框，手机屏幕显示键盘，用户进行输入  1.4手指点击代几成绩输入框，手机屏幕显示键盘，用户进行输入  1.5手指点击C语言输入框，手机屏幕显示键盘，用户进行输入  1.6用户点击用户头像图片进入拍照功能  1.7 点击确认按钮，信息录入系统  1.7.1 信息录入系统数据库前进行学生平均成绩和总分的运算，并存入数据库  1.7.2 学生头像路径信息存入数据库，图片存入系统文件夹中   1. 备选事件流   2.1 用户输入成绩小于0或者大于100，提示重新输入  2.2 用户未录入完全信息，点击确认，进行相应提示 |
| 特殊需求：学号，成绩输入框中要求用系统输入框的强制输入数字功能 |

用例图2：学生信息显示用例描述

|  |
| --- |
| **用例编号 ： User\_2 用例名称：显示当前学生列表** |
| **简要说明** |
| 显示全部学生列表 |
| **执行者** |
| 用户 |
| **前置条件** |
| 用户进入录入学生信息列表界面 |
| **后置条件** |
| 显示学生列表 |
| **事件流** |
| 1 基本事件流  系统从数据库中读出所有学生的信息，包括姓名，学号，成绩，平均分，总分，头像信息，并在系统列表中显示 |
| 特殊需求：无 |

用例图3：删除学生信息用例描述

|  |
| --- |
| **用例编号 ： User\_3 用例名称：显示删除当前选中的学生** |
| **简要说明** |
| 显示在展示学生列表时候进行选中学生删除功能 |
| **执行者** |
| 用户 |
| **前置条件** |
| 用户进入学生信息列表界面，并长时间按相应学生item |
| **后置条件** |
| 学生信息从数据库中删除，学生头像从系统文件夹中删除 |
| **事件流** |
| 1 基本事件流  弹出确认删除记录的提示对话框  1.1 如果点击取消，提示对话框消失，系统不做任何删除操作  1.2 如果点击确认，提示对话框消失，系统删除相关信息 |
| 特殊需求：无 |

用例图4：输出学生成绩排名表用例描述

|  |
| --- |
| **用例编号 ： User\_4 用例名称：输出学生成绩排名表** |
| **简要说明** |
| 显示出学生成绩排名表 |
| **执行者** |
| 用户 |
| **前置条件** |
| 用户进入学生成绩排名表列表界面 |
| **后置条件** |
| 显示排名表 |
| **事件流** |
| 1 基本事件流  系统从数据库中读出所有学生的信息，包括姓名，学号，成绩，平均分，总分，头像信息，并在系统列表中显示，列表前边加入排名序号 |
| 特殊需求：无 |

用例图5：查询相关学生用例描述

|  |
| --- |
| **用例编号 ： User\_5 用例名称：查询相关学生** |
| **简要说明** |
| 通过学号查询相关学生信息 |
| **执行者** |
| 用户 |
| **前置条件** |
| 用户进入查询界面 |
| **后置条件** |
| 学生查询结果显示到界面上边 |
| **事件流** |
| 1 基本事件流  1.1 用户在输入框中输入学号  1.2 用户点击查找按钮  1.3 界面显示出查询结果  1.3.1 如果查到，那么显示学生信息  1.3.2 如果没有查到，显示未查到信息字样 |
| 特殊需求：输入框中要求用系统输入框的强制输入数字功能 |

# 需求分析（B）（杨琳编写）

## 前言

### 编写目的

本文档为学生学习任务管理系统需求分析说明书，主要从学生出发，对软件的学习任务的管理流程进行了分析说明，其主要目的是为系统的后期设计和开发提供依据，明确软件需求、安排项目规划与进度、组织软件开发与测试。

本文档供开发小组成员参考，以及为最终软件是否满足用户需求提供检验依据

### 参考资料

[1] 《软件工程——原理方法与应用》

[2] 《面向对象设计与UML》

## 可行性分析

### 功能要求

1. 学生能够设置学习任务的自定义名称、具体内容、任务照片、选择任务类型、选择任务的重要性；
2. 学生能够修改已经设定的任务的自定义名称、具体内容、任务照片、选择任务类型、修改任务的重要性；
3. 学生能够查看已经添加的学习任务；
4. 学生能够删除已有的学习任务；
5. 学生能够查看单个任务
6. 学生可以选择任务进行完成任务功能

### 性能要求

1. 响应时间：200ms之内；
2. 更新处理时间：200ms之内；
3. 数据的转换和传送时间：300ms之内；
4. 操作系统：Android手机系统。

### 目标

帮助学生实现学习任务的管理，促进学生落实学习任务的执行

### 人员

赵奇隆

## 数据描述

### 静态数据

1. 相关标题栏；
2. 按钮上文字。
3. 任务的种类描述。
4. 任务重要紧急程度的描述。

### 动态数据

1. 学习任务详细信息，包括学习任务名称、具体内容、任务照片、任务类型、任务重要性，任务创建的时间；
2. 学习任务列表。

### 数据库介绍

SQLite，是一款轻型的数据库，是遵守ACID的关系型数据库管理系统，它包含在一个相对小的C库中。它是D.RichardHipp建立的公有领域项目。它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，它占用资源非常的低，在嵌入式设备中，可能只需要几百K的内存就够了。

### 数据采集

学习任务详细内容由用户（学生）手动输入，任务照片由管用户（学生）自行拍照录入，如果没有录入就使用一张没有添加照片标识的图片。

## 功能需求

### 功能介绍

#### 功能划分

功能划分图如下：



#### 功能描述

1. 学生能够录入学习任务名称、具体内容、任务照片、任务类型、任务重要性；
2. 学生能够查询自己的学习任务列表；
3. 学生能够修改自己的学习任务列表，即删除学习任务或者讲学习任务标记为完成；
4. 学生能够修改学习任务列表中的每一项任务；

### 用例描述

Figure 2系统用例图



用例图1：添加学习任务用例描述

|  |
| --- |
| **用例编号 ：UseCase\_001 用例名称：** 录入任务 |
| **简要说明** |
| 学生进入学习任务管理系统界面，点击添加按钮进入添加界面 |
| **执行者** |
| 学生 |
| **前置条件** |
| 用户进入学习任务添加界面 |
| **后置条件** |
| 录入的学习任务信息被存入数据库中，相关图片存入相关路径下的文件夹中，图片路径同时存如数据库 |
| **事件流** |
| 1. 基本事件流   1.1点击进入学习任务添加界面  1.2输入学习任务名称、具体内容、选择学习任务类型、选择学习任务重要性  1.3 点击拍照按钮，进行学习任务的拍照  1.3点击拍照按钮，系统判断无重名任务  2 备选事件流：  1.1学生如果缺少任务名称，系统弹出提示：“请输入任务名称”  1.2如果任务名称和已有的任务名重复，则点击确认的时候，弹出提示重新输入 |
| 特殊需求：无 |

用例图2：查看已有学习任务用例描述

|  |
| --- |
| **用例编号 ： UseCase\_002 用例名称：查看已有学习任务列表** |
| **简要说明** |
| 显示学习任务列表 |
| **执行者** |
| 学生 |
| **前置条件** |
| 学生进入任务显示列表界面 |
| **后置条件** |
| 无 |
| **事件流** |
| 1. 基本事件流   1.1进入任务显示列表界面，可查看当前已有学习任务列表  1.2点击要查看的学习任务，进入具体的学习任务显示界面，查看具体内容（包括任务照片） |
| 特殊需求：无 |

用例图3：修改学习任务用例描述

|  |
| --- |
| **用例编号 ：UseCase\_003 用例名称： 修改学习任务** |
| **简要说明** |
| 修改任务列表中已存在的某一学习任务 |
| **执行者** |
| 学生 |
| **前置条件** |
| 进入学习任务列表界面 |
| **后置条件** |
| 数据库中修改学习任务表和对应照片 |
| **事件流** |
| 1. 基本事件流   1.1点击进入修改任务界面  1.2点击要修改的学习任务中的项（包括学习任务名称、具体内容、选择学习任务类型、选择学习任务重要性、对应照片），进行更改  1.4点击保存，任务保存入系统  2 备选事件流：  1.1学生如果缺少任务名称，系统弹出提示：“请输入任务名称”  1.2如果任务名称和已有的任务名重复，则点击确认的时候，弹出提示重新输入 |
| 特殊需求：无 |

用例图4：删除学习任务用例描述

|  |
| --- |
| **用例编号 ：UseCase\_004 用例名称：删除学习任务** |
| **简要说明** |
| 将想要删除的学习任务从系统中删除 |
| **执行者** |
| 学生 |
| **前置条件** |
| 进入学习任务列表界面 |
| **后置条件** |
| 从数据库中删除列表中删除的任务 |
| **事件流** |
| 1. 基本事件流   1.1进入学习任务列表界面  1.2长按相关任务，弹出删除确认的对话框  1.3 点击确认，删除任务，更新任务列表  1.4 点击取消，对话框消失 |
| 特殊需求：无 |

用例图5：查找学习任务用例描述

|  |
| --- |
| **用例编号 ：UseCase\_005 用例名称： 查找学习任务** |
| **简要说明** |
| 根据任务名称查找学习任务 |
| **执行者** |
| 学生 |
| **前置条件** |
| 进入学习任务显示列表 |
| **后置条件** |
| 学习任务显示列表的内容更新 |
| **事件流** |
| 1. 基本事件流   1.1 在界面上的输入框中输入要查询的任务名  1.2点击查找按钮  1.3 如果能够查找得到，在界面上更新显示  1.4 如果查找不到，在界面上提示没有查找到相关的任务 |
| 特殊需求：无 |

用例图6：查看单个学习任务用例描述

|  |
| --- |
| **用例编号 ：UseCase\_006 用例名称： 查看单个学习任务** |
| **简要说明** |
| 在主界面中进入单个任务查看界面，查看单个任务 |
| **执行者** |
| 学生 |
| **前置条件** |
| 进入学习任务显示列表 |
| **后置条件** |
| 单个任务信息显示在界面上 |
| **事件流** |
| 基本事件流  1.1 在主界面上的任务列表上点击任务列表中的某一项  1.2 页面进行跳转到单个任务查看界面  1.3 读取数据库中的信息，在界面上更新显示 |
| 特殊需求：无 |

用例图7：完成学习任务用例描述

|  |
| --- |
| **用例编号 ：UseCase\_007 用例名称：完成学习任务** |
| **简要说明** |
| 进入单个学习任务查看界面，点击完成任务，任务完成 |
| **执行者** |
| 学生 |
| **前置条件** |
| 进入单个学习任务查看界面 |
| **后置条件** |
| 从数据库中删除列表中删除的任务 |
| **事件流** |
| 1. 基本事件流   1.1 进入单个学习任务查看界面  1.2 点击完成任务，显示是否确认完成任务  1.3 点击确认，删除任务，更新任务列表  1.4 点击取消，对话框消失，界面不发生任何变化 |
| 特殊需求：无 |

# 概要设计（A）（杨琳编写）

## 引言

### 编写目的

概要设计说明书明确定义了任务和需求，能够使软件开发人员知道软件开发流程和软件大体架构，也能使软件测试时更有条理。

本文档由设计人员编写，供经理和开发小组成员参考。

### 项目背景

传统的纸质版学生成绩管理档案不仅需要大量的人力物力，还占据大量的地方，并且一般情况下没有备份，导致丢失不可恢复。电子版学生成绩管理系统为成绩管理提供了巨大的方便，大大减少了人力物力，并且可以随时轻松备份数据库里边的学生和成绩内容。

### 定义

1. 概要设计
2. 领域建模
3. 时序图

### 参考资料

[1] 《软件工程——原理方法与应用》

[2] 《面向对象设计与UML》

[3] 软件工程课堂课件

[4] UML课堂课件

## 任务概述

### 目标

该应用将成为一个普通的学生成绩管理系统的小Demo。

### 运行环境

[1] Andriod手机系统

[2] SQLite数据库

### 需求概述

[1]功能方面能够实现以下要求：

1) 管理人员能够添加学生基本信息，包括姓名，学号，三科成绩。

2) 管理人员能够查看所有学生的列表。

3) 管理人员能在查看学生列表的同时进行学生的删除。

4) 管理人员能按照学生总成绩输出学生的总排名表。

5) 管理人员能够按照学号查询学生。

[2]性能方面能够实现以下要求：

1) 响应时间：200ms之内。

2) 更新处理时间：200ms之内。

3) 数据的转换和传送时间：3000ms之内。

4) 操作系统：Android手机系统。

## 系统分析

### 领域模型



### 时序图

#### 添加学生时序图



#### 显示学生列表和删除学生时序图



#### 输出排名列表时序图



#### 学生信息查询时序图



## 系统概要设计

### 整体包与类设计



### 开发环境与开发工具

1. 开发环境

Android手机端

1. 开发工具

Eclipse Google ADT

## 数据库概念结构设计



# 概要设计（B）（赵奇隆编写）

## 引言

### 编写目的

本概要设计说明书编写的目的在于阐述学生学习任务管理系统在总体设计阶段的模块划分和功能分配，并明确系统的总体体系结构以及数据存储表的内容。本说明书的预期读者为软件设计人员。

### 项目背景

1. 项目名称：Andriod版学生学习成绩管理系统
2. 本系统的开发者：软件工程PK项目组赵奇隆
3. 项目委托方：软件工程PK项目同组人杨琳

### 定义

1. 概要设计
2. 领域建模
3. 时序图

### 参考资料

[1] 《软件工程——原理方法与应用》

[2] 《面向对象设计与UML》

## 任务概述

### 目标

帮助学生实现学习任务的管理，促进学生落实学习任务的执行

### 运行环境

[1] Andriod手机系统

[2] SQLite数据库

### 需求概述

[1]功能方面要实现一下条件：

1. 学生能够设置学习任务的自定义名称、具体内容、任务照片、选择任务类型、选择任务的重要性；
2. 学生能够修改已经设定的任务的自定义名称、具体内容、任务照片、选择任务类型、修改任务的重要性；
3. 学生能够查看已经添加的学习任务；
4. 学生能够删除已有的学习任务；

[2]性能方面要实现以下条件：

1. 响应时间：200ms之内；
2. 更新处理时间：200ms之内；
3. 数据的转换和传送时间：300ms之内；
4. 操作系统：Android手机系统。

## 系统分析

### 领域模型



### 时序图

#### 新建任务时序图



#### 删除任务时序图



#### 修改任务时序图



#### 查询任务时序图



#### 看单个任务时序图



#### 完成任务时序图



## 系统概要设计

### 整体包与类设计



### 开发环境与开发工具

1. 开发环境

Android手机端

1. 开发工具

Android studio

## 数据库概念结构设计



# 详细设计（A）（赵奇隆编写）

## 引言

### 编写目的

详细设计在概要设计的基础上进一步明确系统结构，详细地介绍系统的各个功能模块，详细描述各个类包括类中的属性、方法以及各个类之间的关系，同时进行数据库的详细设计。为后续的设计和完善提供方便。本详细设计文档的预期读者为本项目小组成员。

### 项目背景

传统的纸质版学生成绩管理档案不仅需要大量的人力物力，还占据大量的地方，并且一般情况下没有备份，导致丢失不可恢复。电子版学生成绩管理系统为成绩管理提供了巨大的方便，大大减少了人力物力，并且可以随时轻松备份数据库里边的学生和成绩内容。

### 定义

1. 详细设计

详细设计是软件工程中软件开发的一个步骤，就是对概要设计的一个细化，就是详细设计每个模块实现算法，所需的局部结构。详细设计的主要任务是设计每个模块的实现算法、所需的局部数据结构。

1. 类图

类别图是面向对象式的建模，一般都被用于概念建模的系统分类的应用程序，并可将模型建模转译成程式码。一般类图包括类别名称，类别的属性，类别的方法三个部分。类图显示了模型的静态结构，特别是模型中存在的类、类的内部结构以及它们与其他类的关系等。类图不显示暂时性信息。

1. 数据库设计

数据库设计是指根据用户的需求，在某一具体的数据库管理系统上，设计数据库的结构和建立数据库的过程。数据库系统需要操作系统的支持。数据库设计是建立数据库及其应用系统的技术，是信息系统开发和建设中的核心技术。由于数据库应用系统的复杂性，为了支持相关程序运行，数据库设计就变得异常复杂，因此最佳设计不可能一蹴而就，而只能是一种“反复探寻，逐步求精”的过程，也就是规划和结构化数据库中的数据对象以及这些数据对象之间关系的过程。

### 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

该应用将成为一个普通的学生成绩管理系统的小Demo。

### 运行环境

[1] Andriod手机系统

[2] SQLite数据库

### 需求概述

[1]功能方面能够实现以下要求：

1) 管理人员能够添加学生基本信息，包括姓名，学号，三科成绩。

2) 管理人员能够查看所有学生的列表。

3) 管理人员能在查看学生列表的同时进行学生的删除。

4) 管理人员能按照学生总成绩输出学生的总排名表。

5) 管理人员能够按照学号查询学生。

[2]性能方面能够实现以下要求：

1) 响应时间：200ms之内。

2) 更新处理时间：200ms之内。

3) 数据的转换和传送时间：3000ms之内。

4) 操作系统：Android手机系统。

## 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 |  |

# 详细设计（B）（杨琳编写）

## 引言

### 编写目的

详细设计在概要设计的基础上进一步明确系统结构，详细地介绍系统的各个功能模块，详细描述各个类包括类中的属性、方法以及各个类之间的关系，同时进行数据库的详细设计。为后续的设计和完善提供方便。本详细设计文档的预期读者为本项目小组成员。

### 项目背景

当代大学生学习任务繁重，应该能有一个能够管理当前任务并且能够管理自己的任务。这个应用就是一个基于Android手机端的学习任务管理小工具。

### 定义

1. 详细设计

详细设计是软件工程中软件开发的一个步骤，就是对概要设计的一个细化，就是详细设计每个模块实现算法，所需的局部结构。详细设计的主要任务是设计每个模块的实现算法、所需的局部数据结构。

1. 类图

类别图是面向对象式的建模，一般都被用于概念建模的系统分类的应用程序，并可将模型建模转译成程式码。一般类图包括类别名称，类别的属性，类别的方法三个部分。类图显示了模型的静态结构，特别是模型中存在的类、类的内部结构以及它们与其他类的关系等。类图不显示暂时性信息。

1. 数据库设计

数据库设计是指根据用户的需求，在某一具体的数据库管理系统上，设计数据库的结构和建立数据库的过程。数据库系统需要操作系统的支持。数据库设计是建立数据库及其应用系统的技术，是信息系统开发和建设中的核心技术。由于数据库应用系统的复杂性，为了支持相关程序运行，数据库设计就变得异常复杂，因此最佳设计不可能一蹴而就，而只能是一种“反复探寻，逐步求精”的过程，也就是规划和结构化数据库中的数据对象以及这些数据对象之间关系的过程。

### 参考资料

1. 项目计划任务书；
2. 项目需求说明书；
3. 项目概要设计说明书。

## 任务概述

### 目标

能够管理学生日常的学习任务，并能够实现学习任务的增删改查。

### 运行环境

[1] Andriod手机系统

[2] SQLite数据库

### 需求概述

[1]功能方面能够实现以下要求：

1) 学生能够实现学习任务的录入

2) 学生能够实现学习任务的修改

3) 学生能实现学习任务的删除

4) 学生能够实现学习任务的查找

5) 学生能够浏览所有的学习任务

[2]性能方面能够实现以下要求：

1) 响应时间：200ms之内。

2) 更新处理时间：200ms之内。

3) 数据的转换和传送时间：3000ms之内。

4) 操作系统：Android手机系统。

## 系统模块



## 程序描述

### 录入学习任务

1. **功能**：学生输入学习任务并保存，学习任务包括：任务名，任务描述（可选），任务类型（默认的是单词复习任务，除此之外还有课堂复习任务，考试复习任务，任务的选择通过下拉菜单选择），任务紧急程度（默认的是紧急且重要，除此之外，还包括：紧急不重要，重要不紧急，不重要不紧急），任务照片（可选，如果没有拍照的话，系统自动为该任务生成一张默认的图片,该照片显示在输入界面上），点击确认保存后进行错误判断，然后如果无错的话就能够保存入数据库

**2. 性能：** 数据存入数据库中的时间不能多于200ms

**3. 输入项目：**任务名称，任务描述（可不填），任务种类，任务紧急重要程度，任务照片（可不选）

**4. 输出项目：**无

**5. 算法：**SQL语句操作

**6. 程序逻辑：**



**7. 限制条件：**任务名称不能出现重复

**8. 测试要点：**

|  |  |  |
| --- | --- | --- |
| 测试项目 | 测试点 | 结果和要求 |
| 输入测试 | 进行信息输入 | 能够正常输入 |
| 任务名重复 | 输入重复的额任务名 | 弹出任务名重复的通知 |
| 任务名不重复 | 输入不存在重复的任务名 | 能正常保存 |
| 拍照测试 | 进行任务拍照 | 能调用系统相机进行拍照 |
| 照片显示设置 | 拍完照片之后查看界面 | 拍完之后的照片正确显示在界面上 |
| 任务类型选择测试 | 点击任务选择的选项框 | 弹出各种任务 |
| 任务紧急程度测试 | 点击任务紧急程度选项框 | 弹出四种任务紧急程度 |
| 任务描述不输入测试 | 不输入任务描述，其他正常输入 | 正常保存 |
| 不输入照片测试 | 不进行拍照，保存 | 正常保存，照片为系统默认的照片 |

### 删除学习任务

**1. 功能:** 在主页面上有每个任务的现实列表，然后长按列表中的某一项，弹出一个确认删除的对话框，点击确认删除，进入数据库中删除该任务的记录，然后更新主界面

**2. 性能：**删除和显示的总时间不能超过200ms

**3. 输入项目：**是否删除的选择

**4. 输出项目：**删除之后的更新列表

**5. 算法：**SQL操作

**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 |
| 长按任务 | 长按任务 | 显示确认删除任务的对话框 |
| 点击确认删除 | 点击确认删除 | 弹出已经删除的提示，对话框消失，列表更新，该任务消失 |
| 点击取消 | 点击对话框的取消删除按钮 | 对话框消失，列表界面不发生任何变化 |

### 修改学习任务

**1. 功能：**在主界面上点击任意一个任务列表项，界面跳转到任务显示界面，再点击任务修改按钮，跳转到任务修改（和任务输入界面差不多）界面上边显示任务名称，任务描述，任务类型，任务紧急程度，任务照片，点击相关输入框可以修改任务，点击任务类型和任务紧急程度，修改任务类型和紧急程度，点击拍照按钮，任务和拍照，修改完成之后点击确认之后进行数据库的修改，然后弹出修改完成通知。

**2. 性能：**点击完成修改之后，系统操作数据库时间不能多于200ms

**3. 输入项目：**修改内容，包括任务名，任务描述，任务类型，任务紧急程度，任务照片

**4. 输出项目：**修改完成后输出修改完成的提示

**5. 算法：**SQL操作

**6. 程序逻辑：**



**7. 限制条件：任务不能重名**

**8. 测试要点**

|  |  |  |
| --- | --- | --- |
| 测试项目 | 测试点 | 结果和要求 |
| 输入测试 | 进行信息输入 | 能够正常输入 |
| 任务名重复 | 输入重复的额任务名 | 弹出任务名重复的通知 |
| 任务名不重复 | 输入不存在重复的任务名 | 能正常保存 |
| 拍照测试 | 进行任务拍照 | 能调用系统相机进行拍照 |
| 照片显示设置 | 拍完照片之后查看界面 | 拍完之后的照片正确显示在界面上 |
| 任务类型选择测试 | 点击任务选 择的选项框 | 弹出各种任务 |
| 任务紧急程度测试 | 点击任务紧急程度选项框 | 弹出四种任务紧急程度 |
| 任务描述不输入测试 | 不输入任务描述，其他正常输入 | 正常保存 |
| 不输入照片测试 | 不进行拍照，保存 | 正常保存，照片为系统默认的照片 |

### 查找学习任务

**1. 功能：**用户在主界面上边的输入框中输入自己要搜搜索的任务，点击搜索按钮，系统从数据库中搜索到相应的任务，然后在搜索之后将任务显示列表中，如果任务不存在，则显示，任务不存在

**2. 性能：**搜索的时间和更新的时间不能超过200ms

**3. 输入项目：**任务名

**4. 输出项目：**搜索到的任务

**5. 算法：**SQL查询语句

**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 |
| 任务输入 | 任务输入为空 | 没有任何翻反应 |
| 输入不存在的任务名 | 弹出Toast提示，不存在任务 |
| 输入存在的任务名 | 界面更新，显示当前的任务 |

### 查看所有学习任务

**1. 功能：**进入系统主界面，系统调用数据库，获取到所有的数据信息，然后在界面上显示，其中不同的任务显示不同的图标（共三种图标，包括考试任务，课堂任务，单词任务），根据任务不同的紧急程度，任务名称显示不同的颜色，其中紧急重要显示红色，紧急不重要显示黄色，不紧急且重要显示蓝色，不紧急不重要显示绿色。其中界面的上任务列表的排列顺序是从时间早到晚排列，任务录入早的放在前边，和任务紧急程度排列任务越紧急排在越前边

**2. 性能：**从系统内调用数据库，并获取到所有的数据，并按照时间先后任务紧急程度顺序显示在界面上

**3. 输入项目：**无

**4. 输出项目：**任务列表

**5. 算法：**SQL语句，任务按照种类排序的算法

**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 |
| 界面显示 | 数据库中存入数据 | 数据列表正确显示 |
| 界面显示（任务类型） | 录入不同类型的任务 | 不同的任务分别对应自己的任务 |
| 界面显示(任务紧急程度) | 录入四组不同程度的任务 | 在不同的任务列表中显示不同的颜色，分别对应不同的任务种类 |

### 查看单个学习任务

**1. 功能：**主界面上边点击任务，跳转到单个任务界面，界面显示当前点击的任务的任务名，任务描述，任务类型，任务重要程度，任务照片，右上角显示修改任务的按钮，同时下边有完成任务按钮，点击按钮，任务完成（等同与删除），跳转到主界面，主界面更新任务列表

2. 性能：要求加载界面的时间不能超过200ms，完成任务的时间不能超过200ms

**3. 输入项目：**无

**4. 输出项目：**任务信息的显示

**5. 算法：**SQL语句

**6. 程序逻辑：**





**7. 限制条件：**无

**8. 测试要点：**

|  |  |  |
| --- | --- | --- |
| 测试项目 | 测试要点 | 测试结果 |
| 任务显示 | 进入界面 | 是否能够正常显示 |
| 完成任务 | 点击完成任务 | 弹出任务完成提示，进行界面跳转，主界面该任务不再显示 |
| 修改任务跳转 | 点击修改任务按钮 | 进行界面跳转到修改任务界面 |

## 类图设计

### 总类图关系



### 类的详细设计

|  |  |
| --- | --- |
| MainActivity类描述(主界面类) | |
| 属性 | 功能 |
| -dataManegeHelper：DataManageHelper | 任务数据管理工具 |
| -taskList:ListView | 任务列表 |
| -taskAdapter:TaskAdapter | 任务列表适配器 |
| -search:EditText | 搜索框 |
|  | |
| 方法 | 功能 |
| #onDestory():void | 当界面销毁的时进行调用，在里边关闭数据库 |
| #onCreate():void | 界面第一次创建的时候调用放置逻辑操作 |

|  |  |
| --- | --- |
| AddTaskActivity类描述（添加任务界面类） | |
| 属性 | 功能 |
| -dataManegeHelper：DataManageHelper | 任务数据管理工具 |
| -taskPhoto:Button | 拍照按钮 |
| -taskName:EditText | 任务名称输入修改框 |
| -taskDesc:EditText | 任务描述输入修改框 |
| -taskImportance:Spinner | 任务重要性选择控件 |
| -taskType：Spinner | 任务类型选择控件 |
|  | |
| 方法 | 功能 |
| #onDestory():void | 当界面销毁的时进行调用，在里边关闭数据库 |
| #onCreate():void | 界面第一次创建的时候调用放置逻辑操作 |

|  |  |
| --- | --- |
| ShowActivity类描述（显示单个任务详细信息的界面类） | |
| 属性 | 功能 |
| -dataManegeHelper：DataManageHelper | 任务数据管理工具 |
| -taskDesc:TextView | 任务描述显示控件 |
| -taskImportance:TextView | 任务重要性显示控件 |
| -taskType:TextView | 任务类型选择控件 |
| -taskComplete：Button | 完成任务按钮 |
| -taskName:TextView | 任务名称 |
|  | |
| 方法 | 功能 |
| #onDestory():void | 当界面销毁的时进行调用，在里边关闭数据库 |
| #onCreate():void | 界面第一次创建的时候调用放置逻辑操作 |

|  |  |
| --- | --- |
| TaskAdapter 类描述（任务列表的适配器类，用于加载布局） | |
| 属性 | 功能 |
| -taskArrayList:ArrayList<Task> | 任务列表 |
| -context:Context | 上下文 |
|  | |
| 方法 | 功能 |
| +TaskAdapter(context:Context，taskArrayList:ArrayList<Task>) | 构造函数 |
| +getView() | 获取界面view |

|  |  |
| --- | --- |
| DBopenHelper类描述（动态生成数据库的类） | |
| 属性 | 功能 |
| -context:Context | 上下文 |
|  | |
| 方法 | 功能 |
| +DBOpenHelper(context:Context) | 构造函数 |
| +onCreate() | 第一次动态生成数据库的时候调用 |
| +onUpgrade() | 数据库升级的时候使用 |

|  |  |
| --- | --- |
| DBManageHelper类描述（数据库原子操作基类） | |
| 属性 | 功能 |
| -db:SQLiteDatabase | 数据库实例 |
| -dbOpenHelper:DBopenHelper | 动态生成数据库的基类 |
| -context:Context | 上下文 |
|  | |
| 方法 | 功能 |
| -getCursor() | 获取数据库的游标（数据库中的第一个游标） |
| -getItem():Cursor | 获取到某确定一项的游标 |
| -madifyItem(name:String,sesc:String,time:long,taskType:int,taskImp:int,photo:String) | 修改数据库某一行 |
| -deleteItem(name:String,sesc:String,time:long,taskType:int,taskImp:int,photo:String) | 删除数据库某一行 |
| -close() | 关闭数据库 |

|  |  |
| --- | --- |
| DataManageHelper类描述（管理学生类的操作） | |
| 属性 | 功能 |
| -context:Context | 上下文 |
| adManageHelper:DBManageHelper | 数据库原子操作类 |
| -cursor:Cursor | 游标 |
|  | |
| 方法 | 功能 |
| +DataManagerHelper(context:Context) | 构造函数 |
| +getAllTaskArray():ArrayList<Task> | 获取所有的任务列表 |
| +getItemFromDB(name :String):Task | 查询某个名字的任务，查询不到返回null |
| +modifyTask(task:Task) | 修改任务 |
| +deleteTask(task:TAsk) | 删除任务 |
| +close() | 关闭数据库 |
| +saveTask(task:Task) | 存储任务 |

|  |  |
| --- | --- |
| Task类描述（动态生成数据库的类） | |
| 属性 | 功能 |
| -describe:String | 任务描述 |
| -picture:String | 图片路径 |
| -taskType:int | 任务类型 |
| -taskWeight:int | 任务重要程度 |
| -name:String | 任务名称 |
|  | |
| 方法 | 功能 |
| +Task(name:String,desc:String,pic:String,weight:int,type:int) | 构造函数 |
| +getter.... | Getter |
| +setter.... | Setter |

## 数据库设计

### 数据库结构设计



### 数据字典设计

1．Task(学习任务表)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 类型 | 是否为空 | 默认值 | 备注 |
| Task\_id | 任务序号 | Varchar(MAX) | 否 | Null | 主键、外键 |
| Task\_name | 任务名称 | Varchar(MAX) | 否 | Null |  |
| Task\_des | 任务描述 | Varchar(MAX) | 是 | Null |  |
| Task\_photo\_directory | 任务关联的照片路径 | list | 否 | Null |  |
| Creat\_time | 创建时间 | Date | 否 | Null |  |

2．TaskType(任务类型表)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 类型 | 是否为空 | 默认值 | 备注 |
| Task\_id | 任务序号 | Varchar(MAX) | 否 | Null | 主键、外键 |
| Task\_name | 任务名称 | Varchar(MAX) | 否 | Null |  |
| Task\_des | 任务描述 | Varchar(MAX) | 是 | Null |  |

3．Task\_importance(任务重要性表)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 含义 | 类型 | 是否为空 | 默认值 | 备注 |
| Task\_id | 任务序号 | Varchar(MAX) | 否 | Null | 主键、外键 |
| Importance\_type | 任务重要性 | Varchar(MAX) | 否 | Null |  |

# 实现（A）（杨琳实现）

## 主界面



图3-9 首页

描述：此界面功能是显示所有功能模块并为所有功能模块提供入口。

## 录入信息界面



图3-10 录入学生信息界面

描述：此界面用于用户添加学生信息并保存到数据库

## 学生列表界面



图3-12 展示学生成绩列表界面

描述：该界面的功能是显示学生列表，能够从数据库中读取学生成绩信息并展示到列表中

## 学生排名界面



图3-13 展示学生排名列表界面

描述：该界面功能是显示学生成绩排名，能够从数据库中将学生排名信息读取出来并显示

## 查找学生界面

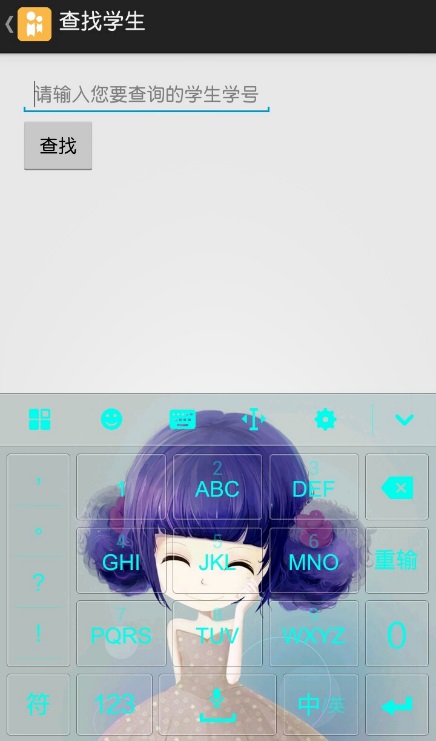


图3-14 查找学生界面

描述：该界面功能是为用户查询学生成绩信息提供接口

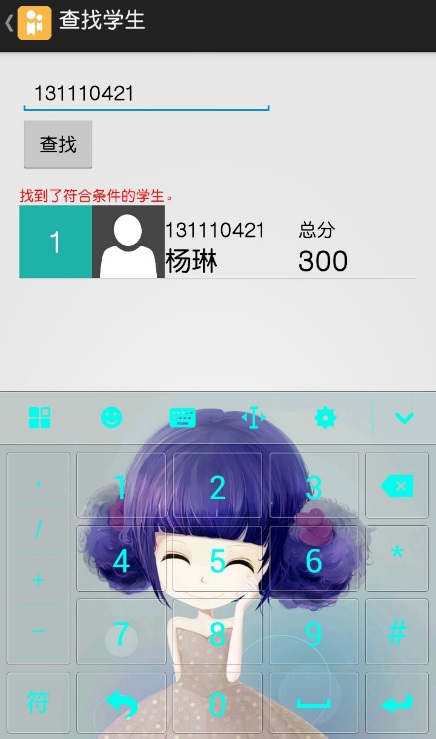


图3-15 被查找学生成绩信息展示界面

描述：该界面继图3-14后从数据库中读取被查找的学生成绩信息

# 实现（B）（赵奇隆实现）

## 任务展示列表界面

描述：该页面仅起展示作用



图3-10 任务列表显示界面

描述：该界面能够展示当前用户所添加的任务列表，实现查询功能

## 任务录入界面



图3-11 添加任务界面

描述： 在此界面中能够添加学习任务相关信息包括任务所对应照片，并点击保存将信息保存到数据库中，将照片添加到相应路径

## 单个任务显示界面



图3-12 查看已添加任务详细信息

描述：该界面功能即为展示用户所要查看的任务的具体信息，并可以点击完成任务从而将任务从数据库中删除

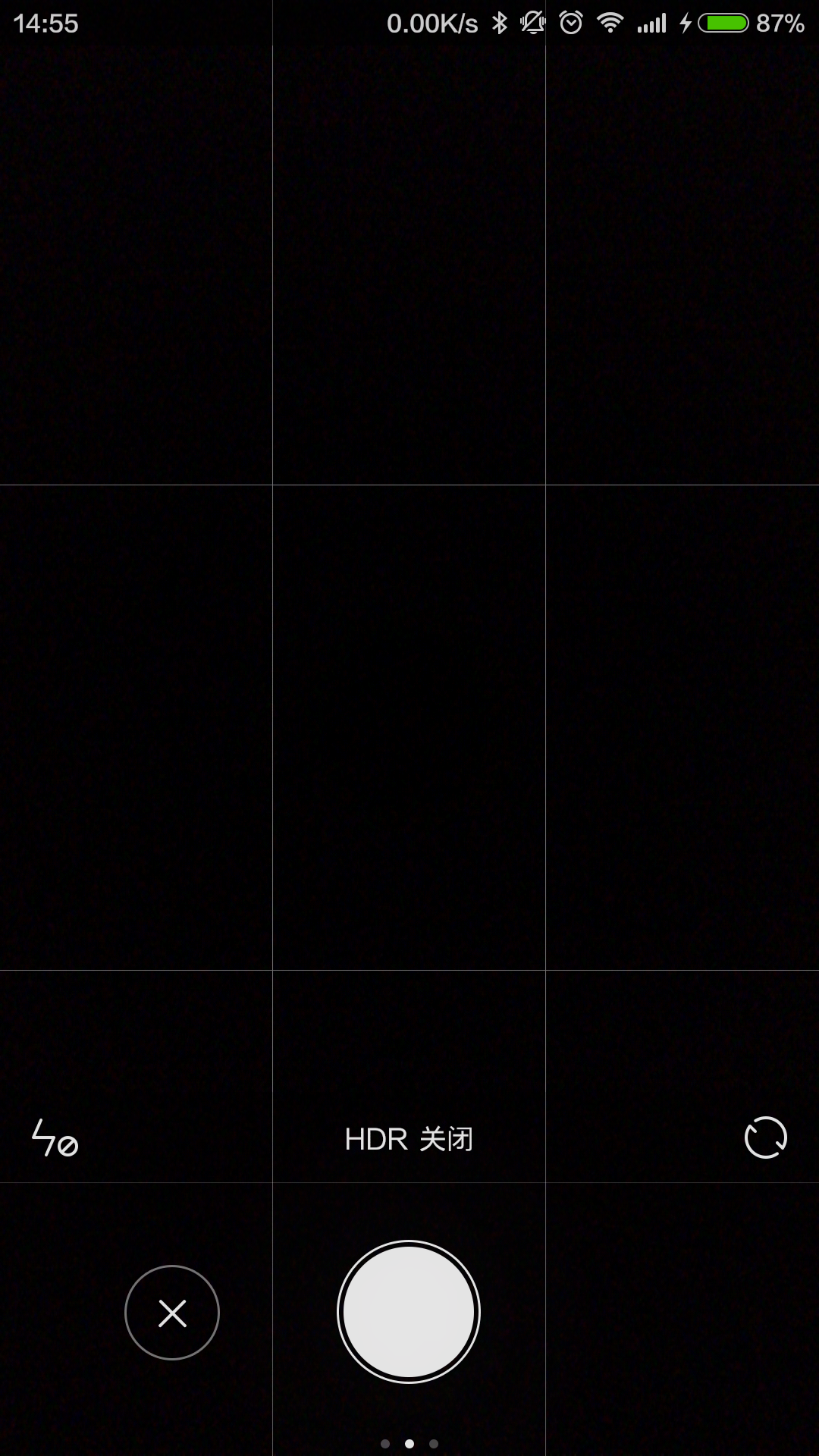
## 任务删除界面



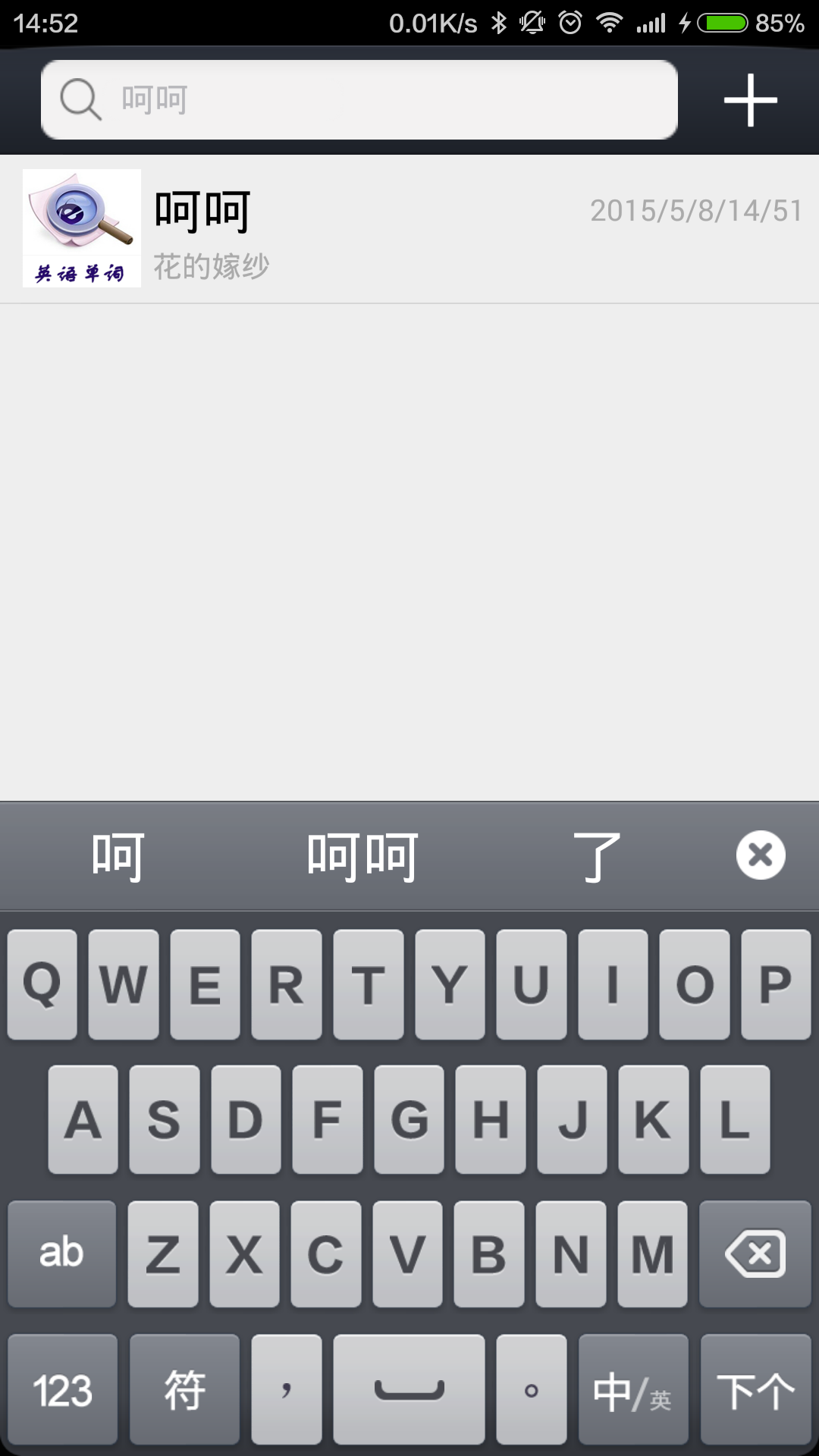
图3-13 删除任务界面

描述： 在该界面中用户可以删除想要删除的任务。即在想要删除的任务行的右侧向左滑动，出现删除后可以点击，从而从数据库中删除任务。

## 拍照功能



## 任务搜索功能



描述：描述进行任务的搜索，搜索到就在列表上边显示

# 测试（A）（赵奇隆实现）

## 引言

### 编写目的

软件测试的定义是在规定的条件下对程序进行操作，以发现程序错误，衡量软件质量，并对其是否能满足设计要求进行评估的过程。软件测试利用测试工具按照测试方法和流程对产品进行功能和性能的测试，根据不同需要编写不同的设计维护测试系统，对测试方案出现的问题进行评估。测试计划是描述测试目的、范围、方法和软件测试等的文档。详细的测试计划可以帮助测试项目组之外的人了解为什么和怎样验证产品。如何规划整个项目周期的测试工作，如何将测试工作上升到测试管理的高度都依赖于测试计划的制定，测试计划是测试工作顺利进行的基础。

测试报告编写的目的：

1. 检验项目各个功能以及查找错误；
2. 给出详细的测试范围；
3. 测试数据；
4. 测试内容；
5. 便于软件制作完成后的测试工作。

读者范围：

软件测试人员以及使用该软件的用户。

### 定义

1. 黑盒测试

也称功能测试，它是通过测试来检测每个功能是否都能正常使用。在测试中，把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试着眼于程序外部结构，不考虑内部逻辑结构，主要针对软件界面和软件功能进行测试。

1. 白盒测试

也称结构测试或逻辑驱动测试，它是按照程序内部的结构测试程序，通过测试来检测产品内部动作是否按照设计规格说明书的规定正常进行，检验程序中的每条通路是否都能按预定要求正确工作。 这一方法是把测试对象看作一个打开的盒子，测试人员依据程序内部逻辑结构相关信息，设计或选择测试用例，对程序所有逻辑路径进行测试，通过在不同点检查程序的状态，确定实际的状态是否与预期的状态一致。

1. 软件缺陷
2. 软件未达到产品说明表明的功能。
3. 软件出现说明书指明不会出现的错误。
4. 软件功能超出产品说明书指明的范围。

### 参考资料

1. 项目需求说明书；
2. 项目概要设计说明书；
3. 项目详细设计说明书；
4. 《软件工程——原理、方法与应用》 史济民、顾春华、郑红著。

## 任务概述

本章节的目标是进行软件测试，并编写测试说明书。测试包括黑盒测试和白盒测试。黑盒测试是对系统整体进行功能性测试，在测试中，把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。白盒测试是按照程序内部的结构测试程序，通过测试来检测产品内部动作是否按照设计规格说明书的规定正常进行，检验程序中的每条通路是否都能按预定要求正确工作。这一方法是把测试对象看作一个打开的盒子，测试人员依据程序内部逻辑结构相关信息，设计或选择测试用例，对程序所有逻辑路径进行测试，通过在不同点检查程序的状态，确定实际的状态是否与预期的状态一致。在完成测试后，还需要将测试用例编写成测试说明书，详细说明每一个测试样例的测试内容、输入、期望输出与实际输出，分析测试结果，以确保程序可以实现正确实现所有需求的功能。

### 运行环境

Android系统

## 计划

### 测试方案

1. 测试方法说明如下
2. 从考虑软件内部构造和具体实现角度采用白盒测试和黑盒测试方法。
3. 从软件开发的过程按阶段采取单元测试、集成测试、确认测试、系统测试、验收测试。
4. 选取测试用例的原则
5. 测试用例功能点

每个测试用例都要有测点，找准测点，不要同时覆盖多个功能。

1. 测试用例的易读

从执行者的角度去写测试用例，不要有包含太多的术语。

1. 测试用例的执行粒度

粒度越小越好。

1. 步骤清晰

一个测试用例多个步骤，可一个重点，步骤指明怎么操作。

1. 总体设计

先正常，后异常。

### 测试项目

录入学生信息，查找学生信息，学生的基本列表信息展示，排名信息展示，删除学生信息

### 测试机构及人员

赵奇隆

### 测试条件

Android系统4.0以上的手机一部

## 黑盒测试

### 录入学习任务：

#### 录入学习任务等价类表

|  |  |  |
| --- | --- | --- |
| 输入数据 | 有效等价类 | 无效等价类 |
| 学习任务名称 | 无重复学习任务名称 | 与已存在的任务名称重复 |
| 任务类型 | 点击任务选择选项框能够弹出各种任务 | 点击任务选择选项框不能弹出各种任务 |
| 任务紧急程度 | 点击任务紧急程度选项框能够弹出四种任务紧急程度 | 点击任务紧急程度选项框不能弹出四种任务紧急程度 |
| 任务描述 | 输入或者不输入任务描述都能正常保存任务 | 无 |
| 拍照 | 进行或者不进行任务拍照都能正常保存任务 | 无 |
| 照片显示 | 查看任务时能够正常显示所拍照片 | 查看任务时不能正常显示所拍照片 |

#### 录入学习任务测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项目 | 测试点 | 期望结果和要求 | 测试数据 | 是否通过 |
| 任务名重复 | 输入重复的额任务名 | 弹出任务名重复的通知 | 第一次和第二次录入不同任务名 | 是 |
| 第一次和第二次录入相同任务名 | 否 |
| 拍照测试 | 进行任务拍照 | 能调用系统相机进行拍照 | 点击相机图标进行拍照 | 是 |
| 照片显示设置 | 拍完照片之后查看界面 | 拍完之后的照片正确显示在界面上 | 拍照后查看是否显示 | 是 |
| 任务类型选择测试 | 点击任务选择的选项框 | 弹出各种任务 | 点击选择框看能否弹出各种任务 | 是 |
| 任务紧急程度测试 | 点击任务紧急程度选项框 | 弹出四种任务紧急程度 | 点击紧急程度选项框看能否弹出各种任务 | 是 |
| 任务描述不输入测试 | 不输入任务描述，其他正常输入 | 正常保存 | 不输入任务描述看能否支持保存 | 是 |
| 不输入照片测试 | 不进行拍照，保存 | 正常保存，照片为系统默认的照片 | 不拍照看能否支持保存 | 是 |

### 删除学习任务：

#### 删除学习任务等价类表

|  |  |  |
| --- | --- | --- |
| 输入数据 | 有效等价类 | 无效等价类 |
| 长按任务 | 弹出确认删除选择框 | 1. 不弹出任何框 2. 弹出其他非任务确认删除选择框 |
| 点击确认 | 删除成功，列表中没有所删除的项 | 1. 点击确认后列表中依然有以确认删除的任务 2. 点击确认后屏幕没有反应 |
| 点击取消 | 退出确认选择弹窗，列表中依然存在被长按过的学习任务 | 1. 不能够退出确认选择弹窗 2. 退出后依旧删除放弃删除的任务 |

#### 删除学习任务测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项目 | 测试点 | 结果和要求 | 测试数据 | 是否通过 |
| 长按任务 | 长按任务 | 显示确认删除任务的对话框 | 长按 | 是 |
| 短按 | 否 |
| 点击确认删除 | 点击确认删除 | 弹出已经删除的提示，对话框消失，列表更新，该任务消失 | 点击删除 | 是 |
| 点击取消 | 点击对话框的取消删除按钮 | 对话框消失，列表界面不发生任何变化 | 点击取消 | 是 |

### 修改学习任务：

#### 修改学习任务等价类

|  |  |  |
| --- | --- | --- |
| 输入数据 | 有效等价类 | 无效等价类 |
| 学习任务名称 | 无重复学习任务名称 | 与已存在的任务名称重复 |
| 任务类型 | 点击任务选择选项框能够弹出各种任务 | 点击任务选择选项框不能弹出各种任务 |
| 任务紧急程度 | 点击任务紧急程度选项框能够弹出四种任务紧急程度 | 点击任务紧急程度选项框不能弹出四种任务紧急程度 |
| 任务描述 | 输入或者不输入任务描述都能正常保存任务 | 无 |
| 拍照 | 进行或者不进行任务拍照都能正常保存任务 | 无 |
| 照片显示 | 查看任务时能够正常显示所拍照片 | 查看任务时不能正常显示所拍照片 |
| 输入数据 | 有效等价类 | 无效等价类 |
|  |  |  |

#### 修改学习任务测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项目 | 测试点 | 期望结果和要求 | 测试数据 | 是否通过 |
| 任务名重复 | 输入重复的额任务名 | 弹出任务名重复的通知 | 第一次和第二次录入不同任务名 | 是 |
| 第一次和第二次录入相同任务名 | 否 |
| 拍照测试 | 进行任务拍照 | 能调用系统相机进行拍照 | 点击相机图标进行拍照 | 是 |
| 照片显示设置 | 拍完照片之后查看界面 | 拍完之后的照片正确显示在界面上 | 拍照后查看是否显示 | 是 |
| 任务类型选择测试 | 点击任务选择的选项框 | 弹出各种任务 | 点击选择框看能否弹出各种任务 | 是 |
| 任务紧急程度测试 | 点击任务紧急程度选项框 | 弹出四种任务紧急程度 | 点击紧急程度选项框看能否弹出各种任务 | 是 |
| 任务描述不输入测试 | 不输入任务描述，其他正常输入 | 正常保存 | 不输入任务描述看能否支持保存 | 是 |
| 不输入照片测试 | 不进行拍照，保存 | 正常保存，照片为系统默认的照片 | 不拍照看能否支持保存 | 是 |

### 查找学习任务

#### 查找学习任务等价类表

|  |  |  |
| --- | --- | --- |
| 输入数据 | 有效等价类 | 无效等价类 |
| 任务名称 | 1. 任务名称不为空 2. 存在的任务名 3. 不存在的任务名 | 任务名为空 |

#### 查找学习任务测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项目 | 测试点 | 期望结果和要求 | 测试数据 | 是否通过 |
| 任务输入 | 任务输入为空 | 没有任何翻反应 | 不输入 | 否 |
| 输入不存在的任务名 | 弹出Toast提示，不存在任务 | 输入一个不存在的任务名 | 否 |
| 输入存在的任务名 | 界面更新，显示当前的任务 | 输入存在任务名 | 是 |

### 查看所有学习任务：

#### 查看所有学习任务等价类表

|  |  |  |
| --- | --- | --- |
| 输入数据 | 有效等价类 | 无效等价类 |
| 界面显示 | 数据库中输入信息界面能够正确显示数据列表 | 1. 界面不能正确显示 2. 界面不能正确显示数据库中数据列表 |
| 界面显示（任务类型） | 录入每行任务类型不同的多行任务信息，界面能正确显示出不同任务对应不同任务类型 | 1. 界面显示的任务与所输入的任务类型不对应 2. 界面不能正确显示录入的任务 |
| 界面显示(任务紧急程度) | 录入每行任务紧急程度不同的多行任务信息，界面能正确显示出不同任务对应不同任务紧急程度 | 1. 界面显示的任务与所输入的任务紧急程度不对应 2. 界面不能正确显示录入的任务 |

#### 查看所有学习任务测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项目 | 测试点 | 结果和要求 | 测试数据 | 是否通过 |
| 界面显示 | 数据库中存入数据 | 数据列表正确显示 | 在数据库中存入数据看界面是否显示 | 是 |
| 界面显示（任务类型） | 录入不同类型的任务 | 不同的任务分别对应自己的任务 | 录入不同的任务类型任务看界面是否能显示正确 | 是 |
| 界面显示(任务紧急程度) | 录入四组不同程度的任务 | 在不同的任务列表中显示不同的颜色，分别对应不同的任务种类 | 录入不同紧急程度的任务看界面是否能显示 | 是 |

### 查看单个学习任务

#### 查看单个学习任务等价类表

|  |  |  |
| --- | --- | --- |
| 输入数据 | 有效等价类 | 无效等价类 |
| 点击进入单个任务显示界面 | 能够正常并且正确显示 | 1. 不能正常显示 2. 不能正确显示 |
| 点击完成任务 | 能够弹出任务完成提示，进行页面跳转并且主界面不再显示该任务 | 1. 不能弹出任务完成提示 2. 不能进行页面跳转 3. 主界面中该任务依旧存在 |
| 点击修改任务按钮 | 界面跳转到任务修改界面 | 1. 界面不跳转 2. 界面跳转到其他非任务修改界面 |

#### 查看单个学习任务测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项目 | 测试要点 | 测试结果 | 测试数据 | 是否通过 |
| 任务显示 | 进入界面 | 是否能够正常显示 | 点击进入界面看能否正确显示信息 | 是 |
| 完成任务 | 点击完成任务 | 弹出任务完成提示，进行界面跳转，主界面该任务不再显示 | 点击完成任务看能否完成界面跳转并主界面中不再显示该任务 | 是 |
| 修改任务跳转 | 点击修改任务按钮 | 进行界面跳转到修改任务界面 | 点击修改任务按钮看界面能否正确跳转 | 是 |

## 白盒测试

### 1.录入学生信息白盒测试



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试路径 | 覆盖节点 | 输入数据 | 预期输出 | 实际输出 |
| Acefhjln | 1,2,4,6,7,9,11,13,15 | 正确输入所有数据 | 任务保存成功，弹出任务保存成功的提示 | 任务保存成功，弹出任务保存成功的提示 |
| Bp | 1，3，15 | 输入空学号 | 弹出学号不能为空提示 | 弹出学号不能为空提示 |
| Adq | 1，2，5，15 | 学号输入非法数字 | 弹出提示，学号输入错误 | 弹出提示，学号输入错误 |
| Acegr | 1，2，4，6，8，15 | 数据库中存在输入的学号 | 弹出学号已存在的提示 | 弹出学号已存在的提示 |
| Acefis | 1，2，4，6，7，10，15 | 不输入学生姓名 | 弹出学生姓名为空的提示 | 弹出学生姓名为空的提示 |
| Acefhkt | 1，2，4，6，7，9，12，15 | 成绩为空 | 弹出学生成绩不能为空的提示 | 弹出学生成绩不能为空的提示 |
| Acefhjmu | 1,2,4,6,7,9,11,13,15 | 结束输入 | 结束输入，界面跳转 | 结束输入，界面跳转 |
| Acefhjlna | 1,2,4,6,7,9,11,13,15，1 | 不结束输入 | 清空当前界面的输入，重新输入下一个数据 | 清空当前界面的输入，重新输入下一个数据 |

### 2.查找学生信息白盒测试



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试路径 | 覆盖节点 | 输入数据 | 预期输出 | 实际输出 |
| Abdg | 1，2，3，5，7 | 正常且存在于系统内的学号 | 正常查询到相关数据 | 正常查询到相关数据 |
| Abeh | 1，2，3，6，7 | 输入正确的学号但是和系统内的重复 | 提示没有找到相关学生 | 提示没有找到相关学生 |
| Acf | 1，2，4，7 | 输入非法的字符串 | 提示输入错误 | 提示输入错误 |

### 学生的基本列表信息展示白盒测试：



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试路径 | 覆盖节点 | 输入数据 | 预期输出 | 实际输出 |
| Ab | 1，2，3，4 | 数据库中存在数据 | 显示出任务列表 | 显示出任务列表 |
| Qcd | 1，2，3，4 | 数据库中不存在数据 | 弹出没有任务的提示 | 弹出没有任务的提示 |

### 排名信息展示白盒测试



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试路径 | 覆盖节点 | 输入数据 | 预期输出 | 实际输出 |
| Ab | 1，2，3，4 | 数据库中存在数据 | 显示出任务列表 | 显示出任务列表 |
| acd | 1，2，3，4 | 数据库中不存在数据 | 弹出没有任务的提示 | 弹出没有任务的提示 |

### 删除学生信息白盒测试：



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试路径 | 覆盖节点 | 输入数据 | 预期输出 | 实际输出 |
| Ab | 1，2，3，4 | 点击确认删除 | 显示完成删除的提示，列表更新 | 显示完成删除提示，列表更新 |
| acd | 1，2，3，4 | 点击取消 | 列表不发生任何变化 | 列表不发生任何变化 |

## 软件缺陷

达到任务书和需求的标准

# 测试（B）（杨琳实现）

## 引言

### 编写目的

软件测试的定义是在规定的条件下对程序进行操作，以发现程序错误，衡量软件质量，并对其是否能满足设计要求进行评估的过程。软件测试利用测试工具按照测试方法和流程对产品进行功能和性能的测试，根据不同需要编写不同的设计维护测试系统，对测试方案出现的问题进行评估。测试计划是描述测试目的、范围、方法和软件测试等的文档。详细的测试计划可以帮助测试项目组之外的人了解为什么和怎样验证产品。如何规划整个项目周期的测试工作，如何将测试工作上升到测试管理的高度都依赖于测试计划的制定，测试计划是测试工作顺利进行的基础。

测试报告编写的目的：

1. 检验项目各个功能以及查找错误；
2. 给出详细的测试范围；
3. 测试数据；
4. 测试内容；
5. 便于软件制作完成后的测试工作。

读者范围：

软件测试人员以及使用该软件的用户。

### 定义

1. 黑盒测试

也称功能测试，它是通过测试来检测每个功能是否都能正常使用。在测试中，把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试着眼于程序外部结构，不考虑内部逻辑结构，主要针对软件界面和软件功能进行测试。

1. 白盒测试

也称结构测试或逻辑驱动测试，它是按照程序内部的结构测试程序，通过测试来检测产品内部动作是否按照设计规格说明书的规定正常进行，检验程序中的每条通路是否都能按预定要求正确工作。 这一方法是把测试对象看作一个打开的盒子，测试人员依据程序内部逻辑结构相关信息，设计或选择测试用例，对程序所有逻辑路径进行测试，通过在不同点检查程序的状态，确定实际的状态是否与预期的状态一致。

1. 软件缺陷
2. 软件未达到产品说明表明的功能。
3. 软件出现说明书指明不会出现的错误。
4. 软件功能超出产品说明书指明的范围。

### 参考资料

1. 项目需求说明书；
2. 项目概要设计说明书；
3. 项目详细设计说明书；
4. 《软件工程——原理、方法与应用》 史济民、顾春华、郑红著。

## 任务概述

本章节的目标是进行软件测试，并编写测试说明书。测试包括黑盒测试和白盒测试。黑盒测试是对系统整体进行功能性测试，在测试中，把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。白盒测试是按照程序内部的结构测试程序，通过测试来检测产品内部动作是否按照设计规格说明书的规定正常进行，检验程序中的每条通路是否都能按预定要求正确工作。这一方法是把测试对象看作一个打开的盒子，测试人员依据程序内部逻辑结构相关信息，设计或选择测试用例，对程序所有逻辑路径进行测试，通过在不同点检查程序的状态，确定实际的状态是否与预期的状态一致。在完成测试后，还需要将测试用例编写成测试说明书，详细说明每一个测试样例的测试内容、输入、期望输出与实际输出，分析测试结果，以确保程序可以实现正确实现所有需求的功能。

### 运行环境

Android系统

## 计划

### 测试方案

1. 测试方法说明如下
2. 从考虑软件内部构造和具体实现角度采用白盒测试和黑盒测试方法。
3. 从软件开发的过程按阶段采取单元测试、集成测试、确认测试、系统测试、验收测试。
4. 选取测试用例的原则
5. 测试用例功能点

每个测试用例都要有测点，找准测点，不要同时覆盖多个功能。

1. 测试用例的易读

从执行者的角度去写测试用例，不要有包含太多的术语。

1. 测试用例的执行粒度

粒度越小越好。

1. 步骤清晰

一个测试用例多个步骤，可一个重点，步骤指明怎么操作。

1. 总体设计

先正常，后异常。

### 测试项目

录入学习任务，删除学习任务，修改学习任务，查找学习任务，查看所有学习任务，查看单个任务，完成任务

### 测试机构及人员

杨琳

### 测试条件

Android系统4.0以上的手机一部

## 黑盒测试

### 录入学生信息

录入学生信息等价表

|  |  |  |
| --- | --- | --- |
| 输入数据 | 有效等价类 | 无效等价类 |
| 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 | 1. 无法完成拍照 2. 不能自动进入裁剪界面 |
| 对裁剪完成的照片进行保存 | 能够保存裁剪完成的照片并能在界面上显示 | 1. 不能够完成裁剪 2. 裁剪完成的照片不能再界面上显示 |
| 点击返回 | 结束输入，能够返回到上一个界面 | 不能返回到上一个界面 |
| 学号输入 | 数字字符 | 1. 学号为空 2. 学号重复 3. 学号输入非数字字符 |
| 成绩输入 | 0-100之间的数字字符 | 1. 成绩为空 2. 成绩小于0或大于100 3. 成绩存在非数字字符 |
| 姓名输入 | 姓名不为空 | 姓名为空 |

录入学生信息测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项目 | 测试点 | 结果和要求 | 测试数据 | 是否通过 |
| 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 | 进入拍照功能看能不能成功获取照片并进入自动裁剪界面 | 是 |
| 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 | 裁剪并保存看能否成功 | 是 |
| 结束输入 | 点击返回 | 返回到上一个界面，输入结束 | 点击返回看能够返回到上一界面 | 是 |
| 学号输入 | 学号输入正确 | 最后点击确认提示录入成功 | 输入纯数字学号 | 是 |
| 学号为空 | 弹出学号不能为空的提示 | 输入空学号 | 否 |
| 学号重复 | 弹出已经存在提示 | 输入重复学号 | 否 |
| 学号输入非法字符 | 弹出学号输入错误提示 | 输入非数字字符学号 | 否 |
| 成绩输入 | 成绩为空 | 弹出来成绩信息不完整提示 | 不输入成绩 | 否 |
| 成绩小于0或者大于100 | 弹出来成绩录入错误提示 | 1. 输入小于0成绩（-42） 2. 输入大于100成绩（139） | 否 |
| 成绩输入存在非法字符 | 弹出来成绩录入错误提示 | 输入非数字字符成绩（a1） | 否 |
| 成绩均录入正确 | 最后点击确认提示录入成功 | 输入合法成绩（98） | 是 |
| 姓名输入 | 姓名为空 | 弹出姓名为空的错误提示 | 不输入姓名 | 否 |
| 姓名不为空 | 最后点击确认提示录入成功 | 输入姓名（张三、uu、123） | 是 |

### 查找学生信息

查找学生信息等价表

|  |  |  |
| --- | --- | --- |
| 输入数据 | 有效等价类 | 无效等价类 |
| 输入学号 | 正常学号（数字字符） | 1. 学号为空（会有学号为空提示） 2. 学号不存在（会有学号不存在提示） 3. 学号输入非法（会有输入错误提示） |

查找学生信息测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项目 | 测试点 | 结果和要求 | 测试数据 | 是否通过 |
| 学号输入 | 学号正常 | 显示学生信息或者提示没有找到 | 输入正确学号（1324363） | 是 |
| 学号为空 | 提示学号输入为空 | 不输入学号 | 否 |
| 学号输入非法字符 | 提示学号输入错误 | 输入非法学号（1ha27251、hhdhss） | 否 |

### 基本信息展示

基本信息展示等价类表

|  |  |  |
| --- | --- | --- |
| 输入数据 | 有效等价类 | 无效等价类 |
| 排序 | 在数据库中乱序存入学生学号能够按照血厚升序顺序进行排序 | 在数据库中乱序存入学生学号不能按照血厚升序顺序进行排序 |
| 数据读取显示 | 1. 数据库中存在数据时正确显示列表 2. 数据库中不存在数据时显示列表为空 | 1. 数据库中存在数据时显示列表与输入不对应 2. 数据库中存在数据时显示列表为空 3. 数据库中不存在数据时不显示列表为空的界面 |

基本信息展示测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项目 | 测试点 | 结果和要求 | 测试数据 | 是否通过 |
| 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 | 按（163775、123465、132421顺序输入学号）看能否正确排序 | 是 |
| 数据读取显示 | 数据库存在数据 | 显示列表 | 在数据库中存储多行数据看能否在界面正确显示 | 是 |
| 数据库不存在数据 | 提示数据库中不存在数据 | 将数据库清空看界面是否提示数据库中不存在信息 | 是 |

### 排名信息展示

排名信息展示等价类表

|  |  |  |
| --- | --- | --- |
| 输入数据 | 有效等价类 | 无效等价类 |
| 排序 | 在数据库中乱序存入学生学号能够按照血厚升序顺序进行排序 | 在数据库中乱序存入学生学号不能按照血厚升序顺序进行排序 |
| 数据读取显示 | 1. 数据库中存在数据时正确显示列表 2. 数据库中不存在数据时显示列表为空 | 1. 数据库中存在数据时显示列表与输入不对应 2. 数据库中存在数据时显示列表为空 3. 数据库中不存在数据时不显示列表为空的界面 |

排名信息展示测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项目 | 测试点 | 结果和要求 | 测试数据 | 是否通过 |
| 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 | 输入（“86、89、90”“94、98、79”“88、80、90”）三组学生成绩看能否正确按总成绩名 | 是 |
| 数据读取显示 | 数据库存在数据 | 显示列表 | 在数据库中存储多行数据看能否在界面正确显示 | 是 |
| 数据库不存在数据 | 提示数据库中不存在数据 | 将数据库清空看界面是否提示数据库中不存在信息 | 是 |

### 删除学生信息

删除学生信息等价类表

|  |  |  |
| --- | --- | --- |
| 输入数据 | 有效等价类 | 无效等价类 |
| 点击确认删除 | 列表更新，弹出已删除的提示，对话框消失 | 1. 列表不更新 2. 不弹出已删除对话框提示 |
| 点击取消删除 | 对话框消失，其他不发生任何变化 | 列表更新，仍旧删除学生信息 |

删除学生信息测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项目 | 测试点 | 结果和要求 | 测试数据 | 是否通过 |
| 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 | 点击确认看是否删除 | 是 |
| 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 | 点击取消看是否仍然保留 | 是 |

## 白盒测试

### 录入学习任务白盒测试



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试路径 | 覆盖节点 | 输入数据 | 预期输出 | 实际输出 |
| Ab | 1，2，3，4 | 输入数据库中不存在的任务 | 弹出存入成功的Toast，界面跳转到主界面 | 弹出存入成功的Toast，界面跳转到主界面 |
| Qcd | 1，2，3，4 | 输入数据库中存在的数据 | 弹出存在同名任务的提示 | 弹出存在同名任务的提示 |

### 删除学习任务白盒测试



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试路径 | 覆盖节点 | 输入数据 | 预期输出 | 实际输出 |
| Ab | 1，2，3，4 | 确认删除 | 提示删除成功，然后更新界面 | 提示删除成功，然后更新界面 |
| Qcd | 1，2，3，4 | 取消删除 | 关闭提示框，界面不存在任何变化 | 关闭提示框，界面不存在任何变化 |

### 修改学习任务白盒测试



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试路径 | 覆盖节点 | 输入数据 | 预期输出 | 实际输出 |
| Ab | 1，2，3，4 | 输入数据库中不存在的任务 | 弹出存入成功的Toast，界面跳转到主界面 | 弹出存入成功的Toast，界面跳转到主界面 |
| Qcd | 1，2，3，4 | 输入数据库中除了当前任务之外存在的任务名 | 弹出存在同名任务的提示 | 弹出存在同名任务的提示 |

### 查找学习任务白盒测试



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试路径 | 覆盖节点 | 输入数据 | 预期输出 | 实际输出 |
| Ab | 1，2，3，4 | 输入数据库中存在的任务 | 查询结果显示在界面上 | 查询结果显示在界面上 |
| Qcd | 1，2，3，4 | 输入数据库中不存在的任务名 | 显示出没有查询到结果 | 显示出没有查询到结果 |

### 查看所有学习任务白盒测试



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试路径 | 覆盖节点 | 输入数据 | 预期输出 | 实际输出 |
| Ab | 1，2，3，4 | 数据库中存在数据 | 任务列表显示在界面上 | 任务列表显示在界面上 |
| Qcd | 1，2，3，4 | 数据库中不存在数据 | 任务列表没有显示任何数据 | 任务列表没有显示任何数据 |

### 查看单个任务白盒测试



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试路径 | 覆盖节点 | 输入数据 | 预期输出 | 实际输出 |
| Ab | 1，2，3 | 任务的id | 任务详细信息显示在界面上 | 任务详细信息显示在界面上 |

### 完成任务白盒测试



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试路径 | 覆盖节点 | 输入数据 | 预期输出 | 实际输出 |
| Ab | 1，2，3，4 | 确认完成 | 提示删除成功，然后更新界面 | 提示删除成功，然后更新界面 |
| Qcd | 1，2，3，4 | 取消完成 | 关闭提示框，界面不存在任何变化 | 关闭提示框，界面不存在任何变化 |

## 软件缺陷

达到任务书和需求的标准

# 总结（A）

项目总结（杨琳）

经过一个月的设计与开发，学生学习成绩管理系统项目大致完成。经过一系列测试，功能和结构上基本实现了最初的要求。经过这么长时间的合作，我发现两个人合作真心不容易，其中问题百出，由此也对团队工作有了进一步的认识，同时也对软件开发中文档的重要性有了比较深刻的认识。

在开发流程中，我所遇到的两个比较大的问题，一个是团队合作问题，一个是Github使用问题。

团队合作方面，因为缺乏经验，水平不高，并且写文档过程中双方各不理解。最后我们决定采用完全相同的形式完成文档。包括使用相同形式的各种图表。并且在项目过程中随时保持联系。

Github方面。Git是非常流行的代码管理与版本控制工具，Git比一般SVN的优势在于Git采用了分布式与集中式并行的版本管理方式，我们既可以使用GitHub服务器，也可以进行分布式管理。但是Git的学习曲线非常陡峭，在学习理解Git的过程中，遇到了一系列问题，初期上传的东西都很杂乱，使用命令行传的。后期全程使用Github的可视化界面进行上传，方便快捷。

# 总结（B）

项目总结（赵奇隆）

经过一个月的设计与开发，学生学习任务管理系统项目大致完成。经过一系列测试，其功能基本能够满足学生学习任务的记录提醒操作。同时还有拍照功能。与此同时，对于此次项目，我们PK项目组从技术、管理、测试等方面学到了之前在课堂上并没有特别注意到的但是十分重要的技巧，从而得到了真正的锻炼。

过程中出现的问题是：前期因为忙UML进度比较慢，并且前期文档也出现了许多问题，合作方面双方都缺乏精力和经验。不过经过后期的通力合作，终于完成了文档和工程，同时也明白了写文档并不是作用很小或者说很容易的。文档写不好，就说明没有透彻了解整个项目的需求和设计。在后期编码的时候就会出现很多难以解决的问题。

# 附录：

GitHub地址：

A: <https://github.com/Lynn-yqjykn/softWareProPK>

B: https://github.com/zhaoqilong/TaskManageSys

# 互评

杨琳：需求详细，概要设计合理，详细设计到位，编码实现完整，测试到位

赵奇隆：需求写的很充分，没有遗漏的地方，概要设计满足了自己的要求，详细设计很详尽，测试准确，代码可维护性强，容易测试