删除了所有敏感信息

目录

[1. 概 述 1](#_Toc516652520)

[1.1、项目开发目的 1](#_Toc516652521)

[1.2、项目开发内容 1](#_Toc516652522)

[1.3、技术和方法 1](#_Toc516652523)

[1.4、功能 2](#_Toc516652524)

[1.4、总结 2](#_Toc516652525)

[1. 项 目 详 情 2](#_Toc516652526)

[2.1、项目背景 2](#_Toc516652527)

[2.2、技术 3](#_Toc516652528)

[2.3、功能需求 3](#_Toc516652529)

[2.4、界面需求 3](#_Toc516652530)

[2.5、项目概要设计 4](#_Toc516652531)

[2.5.1、类设计 4](#_Toc516652532)

[2.5.2、数据库设计 13](#_Toc516652533)

[2.6、项目详细设计 14](#_Toc516652534)

[2.6.1、Main类 14](#_Toc516652535)

[2.6.2、JdbcUtils类 14](#_Toc516652536)

[2.6.3、Jdbc类 14](#_Toc516652537)

[2.6.4、GUISystem类 15](#_Toc516652538)

[2.6.5、Login类 15](#_Toc516652539)

[2.6.6、Register类 18](#_Toc516652540)

[2.6.7、RegisterWrite类 19](#_Toc516652541)

[2.6.8、RegisterWriteT类 20](#_Toc516652542)

[2.6.9、RegisterWriteS类 20](#_Toc516652543)

[2.6.10、Center类 21](#_Toc516652544)

[2.6.11、BaseS类 22](#_Toc516652545)

[2.6.12、SInfo类 23](#_Toc516652546)

[2.6.13、SReport类 23](#_Toc516652547)

[2.6.14、PwdCh类 23](#_Toc516652548)

[2.6.15、BaseT类 24](#_Toc516652549)

[2.6.16、TInfo类 24](#_Toc516652550)

[2.6.17、TSInfo类 24](#_Toc516652551)

[2.6.18、TScore类 25](#_Toc516652552)

[2.6.19、TScoreCount类 25](#_Toc516652553)

[2.7、项目测试 26](#_Toc516652554)

[2.8、项目总结 28](#_Toc516652555)

[2.9、参考文献 28](#_Toc516652556)

# 概 述

## 1.1、项目开发目的

随着社会的进步，越来越多的工作被程序所简化。

本程序——学生管理系统，包含了学生和教师的登入功能、学生对个人信息的修改功能、老师对学生的增删查改和老师对学生成绩的管理等多种功能。该程序能够方便教师对学生的管理。

## 1.2、项目开发内容

设计并开发了 GUI 界面的学生管理系统，教师可以管理学生，学生可以在该系统中查看自己的信息。

## 1.3、技术和方法

自动验证 : 使用多种验证的方法，能够判断是老师还是学生，还可以准确地判

断登陆错误地原因并告知使用者。使用了多种验证方法进行注册，自动判断注册者是学生还是老师，并向相应地数据表中注入数据。注册模块还会对账号和密码是否合法做出判断，并将不合法的信息告知用户。

JDBC : 使用符合 JDBC 规范的 MySQL 驱动来连接数据库（5.0以后的版本会自

动加载驱动）。

Junit4 : 使用 Junit4 对方法进行测试。

## 1.4、功能

登入 : 教师或学生可以登入进入系统。

注册 : 教师或学生可以进行注册。

个人信息(学生面板) : 学生可以进行查看和修改。

个人成绩(学生面板) : 学生可以进行查看。

学生信息(老师面板) : 老师可以进行增删改查(包括学号、密码)。

学生成绩(老师面板) : 老师可以进行增删改查。

学生成绩统计(老师面板) : 老师可以查看学生的各种成绩信息

个人信息(老师面板) : 老师可以对自己的信息进行查看或修改

## 1.4、总结

通过这次的项目开发，我熟练了很多java的函数。这次实验中，我使用了面向对象的方法，使得各个功能彼此分开。当代码写了 500 多行的时候，就已经很吃力了，我深刻地体会到了面向对象方法的优点，如果一开始使用面向过程，在修改的时候都会花费很多时间。

这个项目比较简单，所以使用地算法并不多。

# 项 目 详 情

## 2.1、项目背景

由于学校的学生众多，传统的管理方式太耗费时间，所以需要一个计算机系统来优化操作，节省时间。

## 2.2、技术

自动验证 : 使用多种验证的方法，能够判断是老师还是学生，还可以准确地判

断登陆错误地原因并告知使用者。使用了多种验证方法进行注册，自动判断注册者是学生还是老师，并向相应地数据表中注入数据。注册模块还会对账号和密码是否合法做出判断，并将不合法的信息告知用户。

JDBC : 使用符合 JDBC 规范的 MySQL 驱动来连接数据库（5.0以后的版本会自

动加载驱动）。

Junit4 : 使用 Junit4 对方法进行测试。

关联组合 : 使用事件来关联某些组件。

## 2.3、功能需求

1．学生登入/注册

2. 教师登入/注册

3. 学生登入后个人信息的修改

4. 学生登入后个人信息的查看(至查看，不能修改)

5. 教师登入后个人密码的修改

6. 教师登入后对学生信息的添加/查看/修改/删除

7. 教师登入后对学生成绩的添加/查看/修改/删除

8. 教师登入后对学生成绩的统计(高低分排序、指定课程平均分统计等)。

## 2.4、界面需求

具备友好的用户操作界面即可。

## 2.5、项目概要设计

### 2.5.1、类设计

图1.1 教师登入功能和界面1

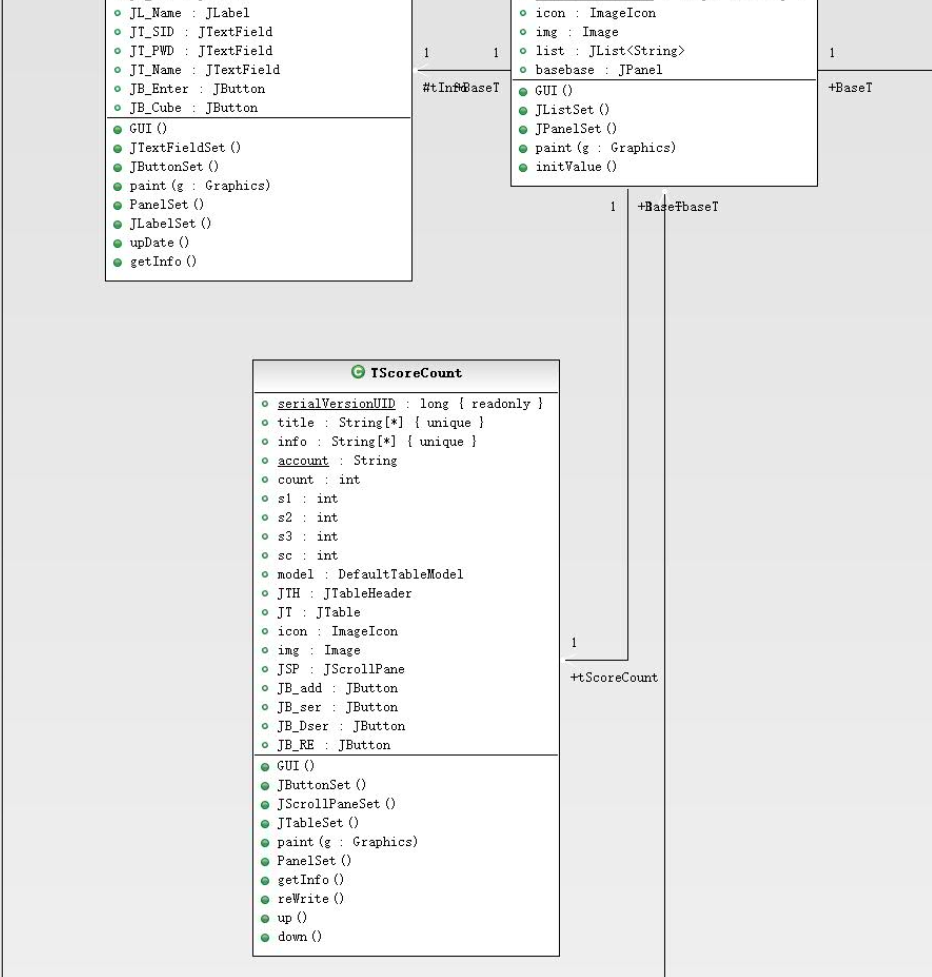
图1.2 教师登入功能和界面2

图1.3 教师登入功能和界面3

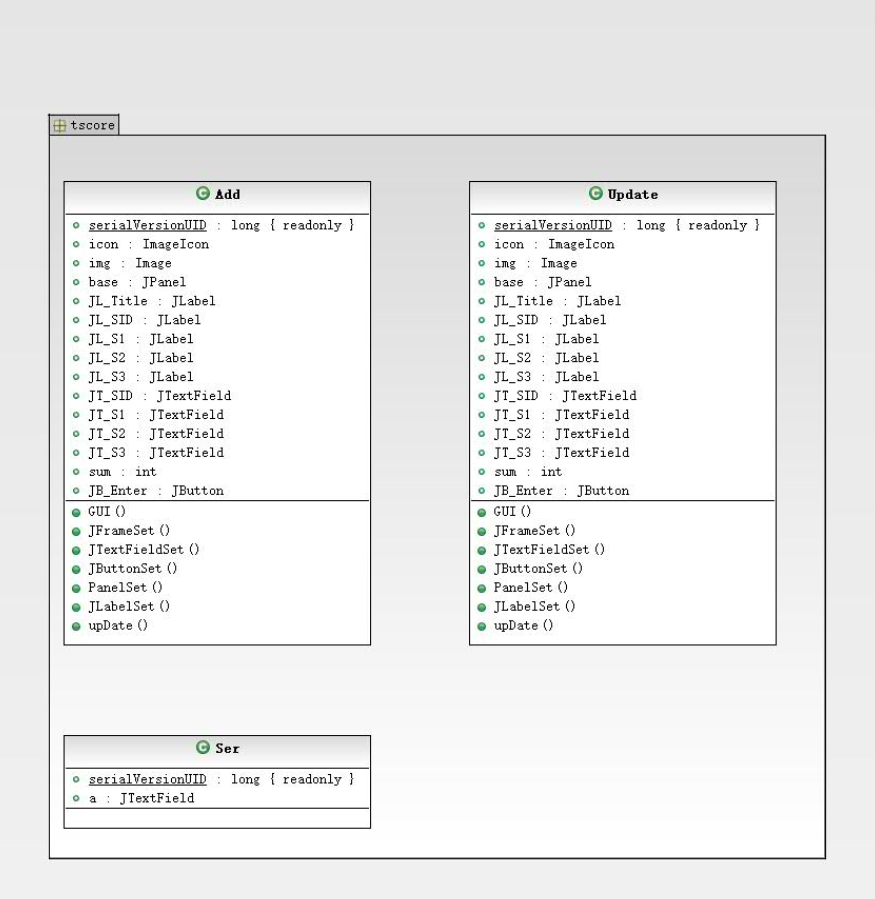
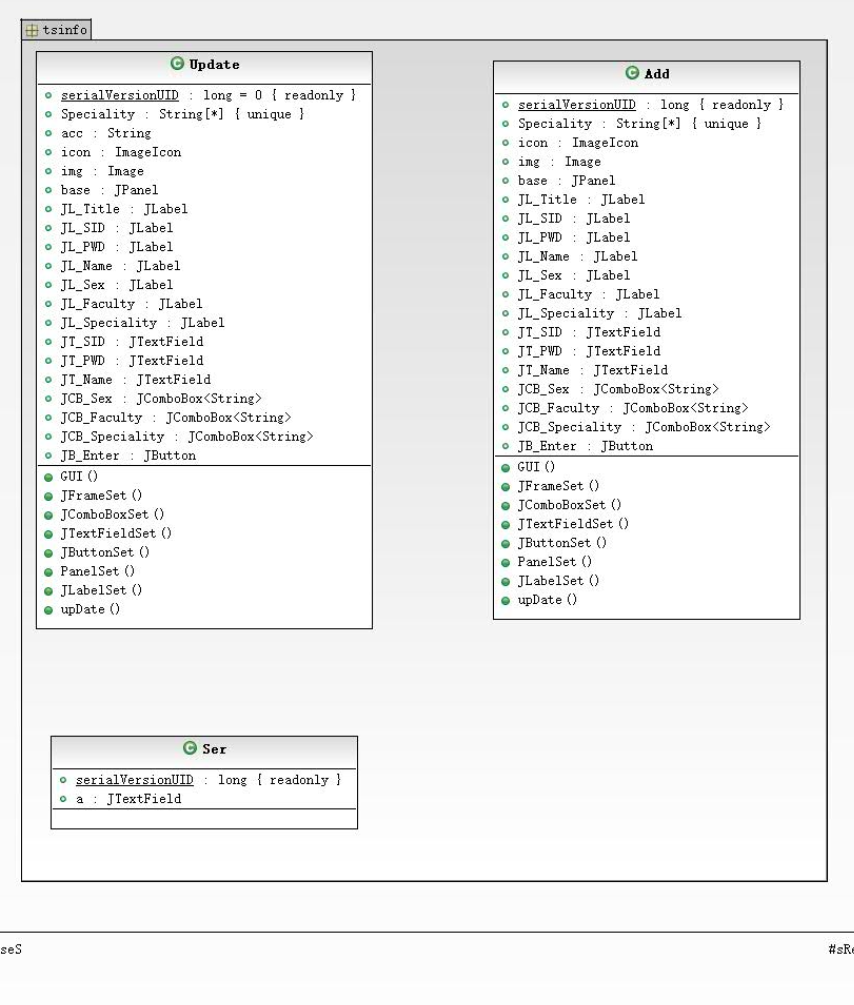
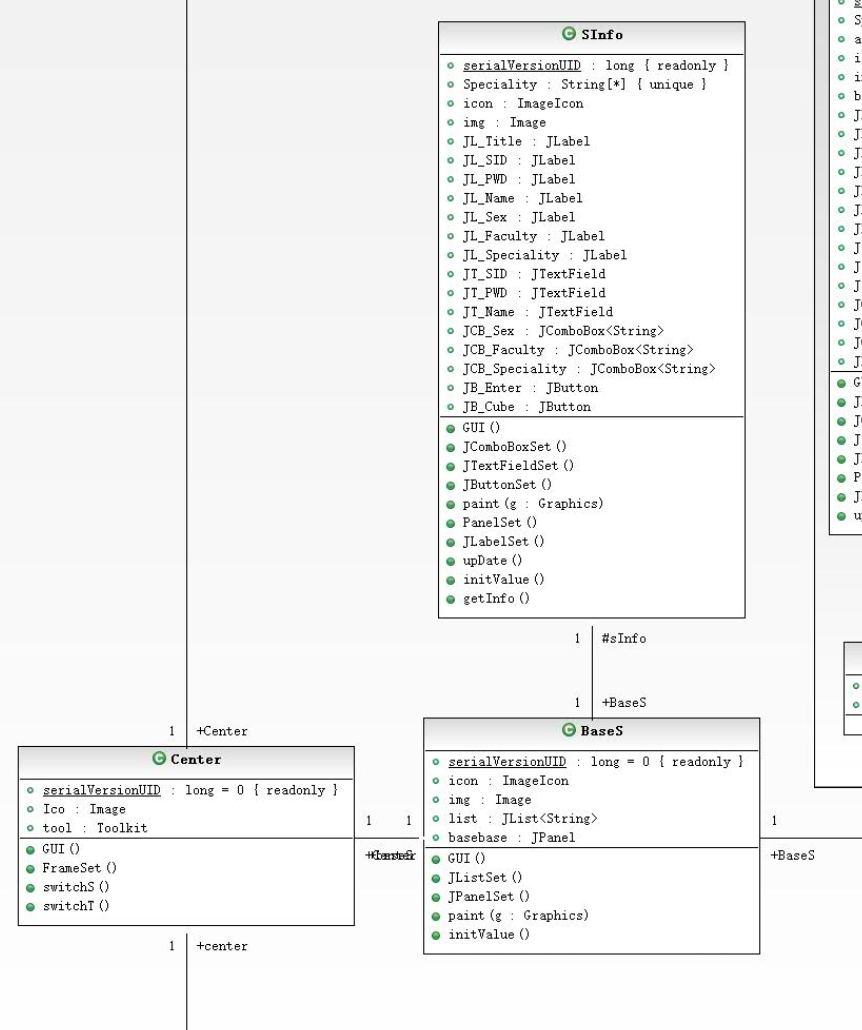
图1.4 教师登入功能和界面4 对学生分数操作的子功能

图1.5 教师登入功能和界面5 对学生信息操作的子功能

图2.1 学生操作模块1

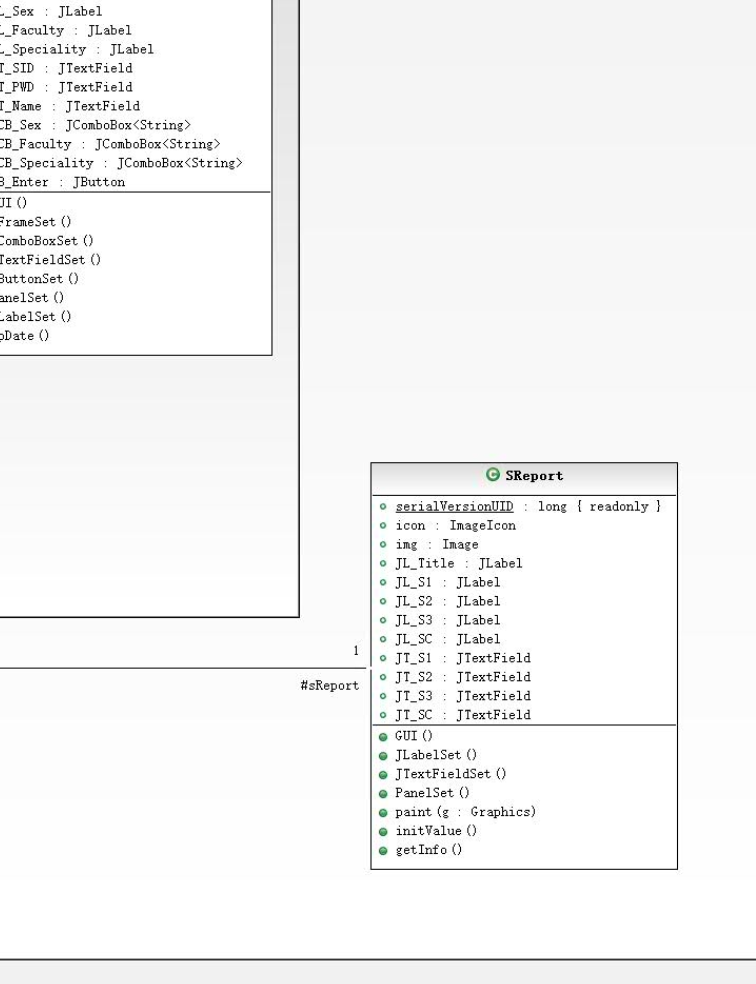


图2.2 学生操作模块2

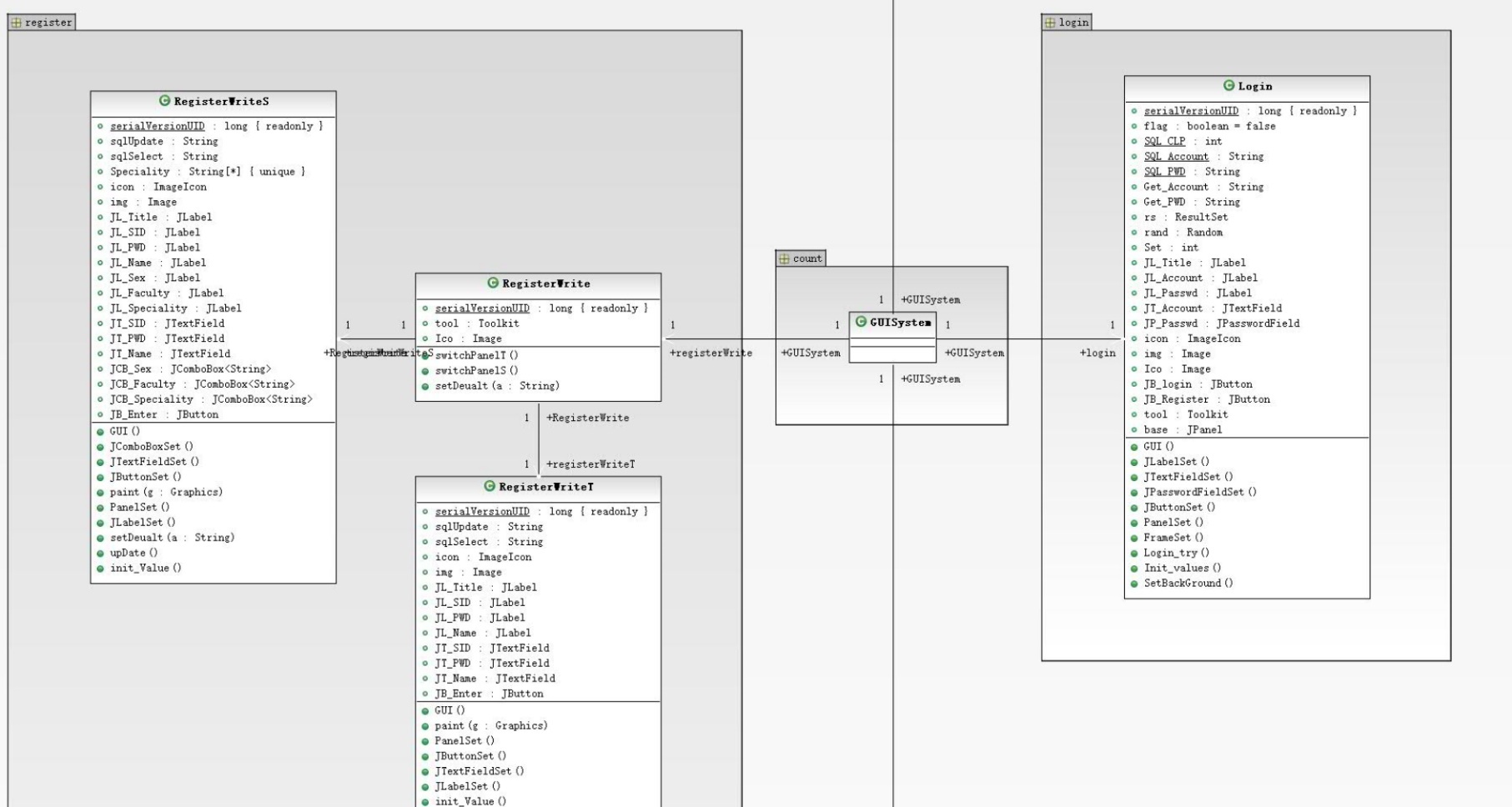


图3.1 中心缓存模块、登入模块、注册详细信息模块、注册详细信息跳转模块

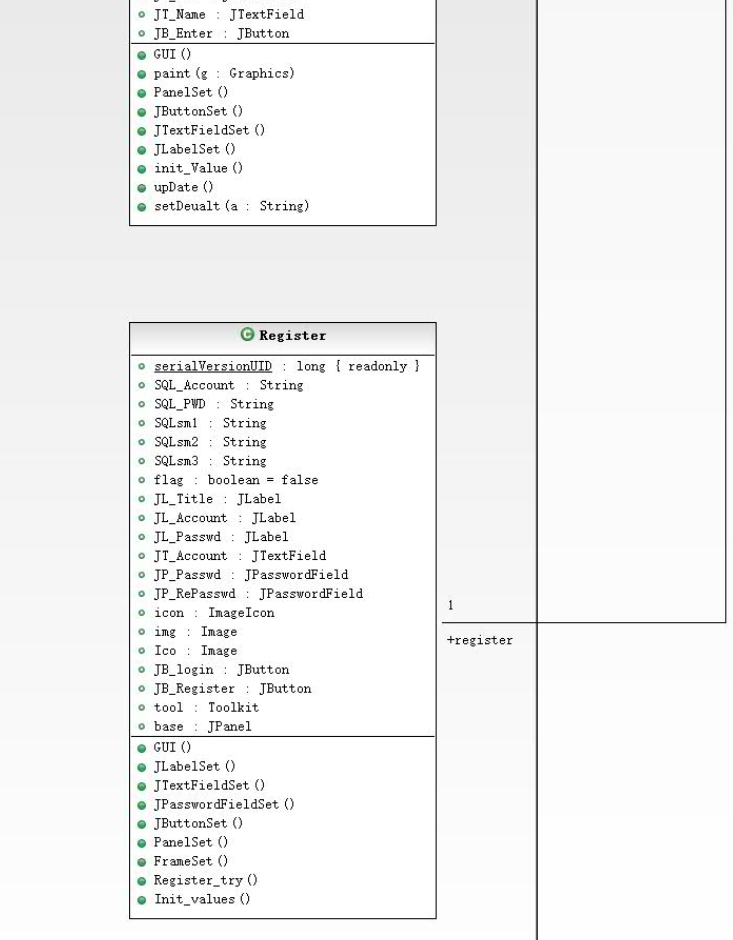
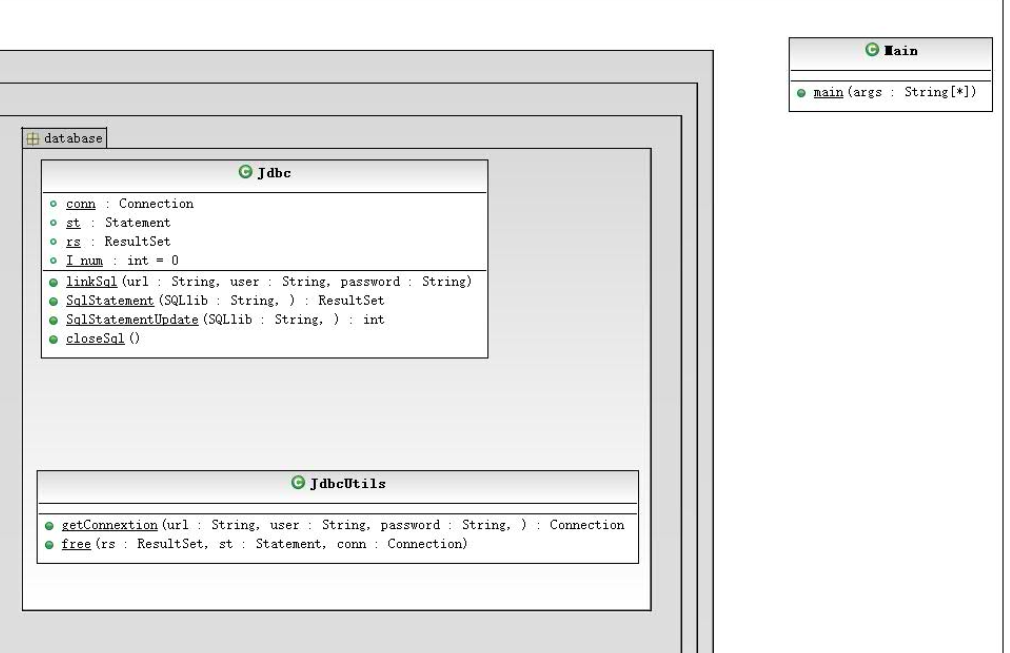
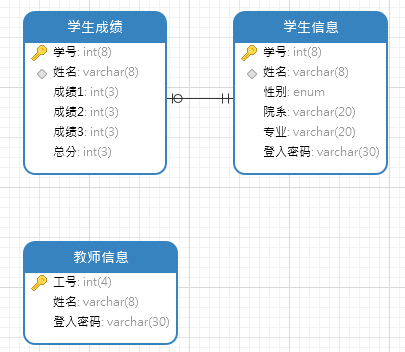
图3.2 注册模块

图4 启动模块和数据传输模块

### 2.5.2、数据库设计



## 2.6、项目详细设计

### 2.6.1、Main类

功能：连接数据库并且开启一个窗体程序。

### 2.6.2、JdbcUtils类

功能： 1.注册并加载JDBC驱动

2.建立连接方法(getConnextion)

读取传入的url、user、password建立连接并返回连接

3.关闭连接方法(free)

关闭传入的ResultSet、Statement、Connection

### 2.6.3、Jdbc类

Jdbc类是对JdbcUtils的再封装类，添加了一些常用功能。

功能： 1.建立并存储连接(linkSql)

根据传入的url、user、password建立并存储连接。

使用了JdbcUtils的getConnextion方法

2.使用查询语句并返回结果集(SqlStatement)

3.使用更新语句并返回受影响行数(SqlStatementUpdate)

4.关闭数据库(closeSql)

使用了JdbcUtils的free方法

5\*.实现数据库的表的增删查改同步(Update)

该方法在2.0版的Jdbc类中才有，本设计的是1.0

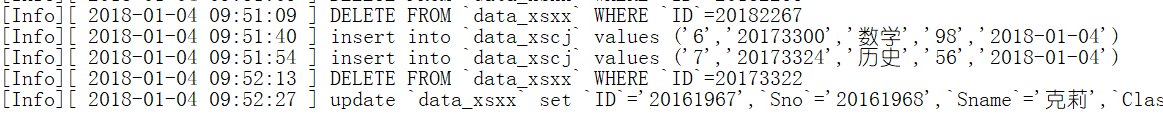
 三个不同的子方法分别实现增同步、删同步和改同步

图2.1 同步的截图

### 2.6.4、GUISystem类

这个类在调用Main类的main方法时，预先缓存了大部分的窗体，在运行程序过程中可以明显感受到程序非常流畅。这也为后面的一些类间的同步和通信建立起了基础。

功能： ● 预先缓存Login(登陆)、Register(注册)、RegisterWrite(注册信

息)、Center(中心功能基础界面)四个重要的类。

● RegisterWrite类对RegisterWriteS类和RegisterWriteT类进行了缓存。

● Center类对BaseS、BaseT类进行了缓存。

● BaseS类对SInfo类和SReport类进行了缓存。

● BaseT类对TInfo类、TScore类、TScoreCount类和TSInfo类进行了缓存。

### 2.6.5、Login类

这是程序启动时最先使用的类。通过此类打开窗口。

功能：

1. 自动更换背景

每次启动时，它会随机产生一个数值，并加载相应的图片。

有的图片会和组件产生干扰，所以在相应的加载方法中，又对组件背景和字体颜色的调整。见图2.3。

1. 提示功能

账户输入框中未聚焦时显示“请输入账户”，聚焦后消失。

密码输入框未聚焦时显示“请输入密码”，聚焦后消失。

1. 限制功能

账户框中只能输入数字，而且无法超过8个。(超过8个的字符会被消除)

1. 精确的登入判断功能(Login\_try)

自动判断账户密码是否正确、若正确会转到主窗体并调用对应的功能进行加载。

1. 注册界面的跳转
2. 精确的预先错误判断和警告

精确地判断密码错误、账号错误和账号是否存在，并告知用户。

1. 预先处理错误

本设计所有类全部进行了预先错误判断，经100多次测试后，已经确定不会再有异常能够出现了。见图2.2

细节： 每次窗体激活的时候，默认聚焦在登陆按钮上面。

关闭窗体前，会默认调用Jdbc的closeSql方法。

任何不同的动作，都会引起组件样式的变更。(聚焦、移动到组件上等)

只要按下回车(任意聚焦状态，除了注册聚焦)，就会调用Login\_try方法。

在调用Login\_try成功后，跳转到其他页面时会调用init\_values()方法来刷新这个界面，保证回到这个界面后不留下痕迹。

本类在Login\_try成功后，使用SQL\_Account和其他类进行“通信”。

登入会判断是否为设定的默认值或者是空的情况，遇到后会弹出警告。

 除了这些以外，还有很多相关的细节，几乎所有的类都有这样的细节。

图2.2 什么都没输并不会触发异常(因进行了预先错误处理)

图2.3 登陆界面

### 2.6.6、Register类

提供注册功能的类

功能:

1. 自动判断是否为学生或是老师进行注册
2. 提示功能
3. 限制功能
4. 登入界面或者注册详细信息界面的跳转(Register\_try)
5. 精确的预先错误判断和警告

是否输入账户

是否输入密码

2次密码是否一致

账户是否存在

账户和密码是否合法

1. 预先处理错误

对5种能够触发异常的情况进行了处理

图2.4 注册界面

### 2.6.7、RegisterWrite类

本类是注册消息信息的骨架，负责判断注册的账户信息后，加载对应的功能。

功能：

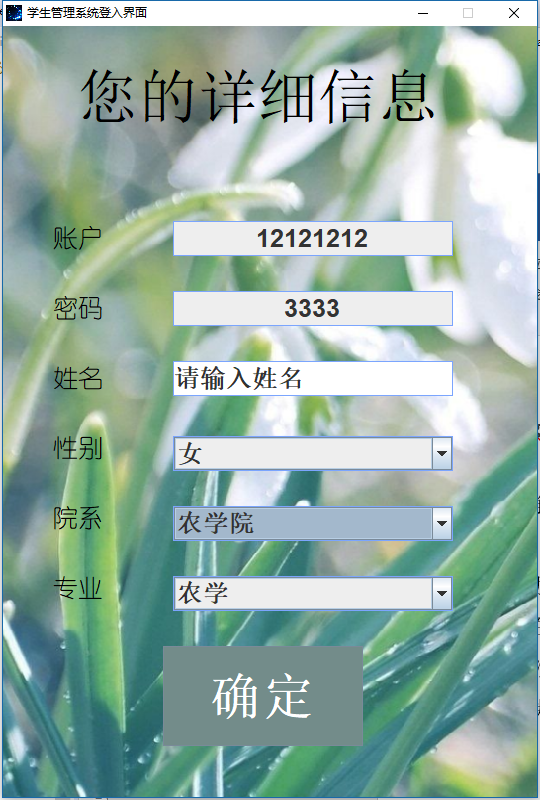
1. 预先判断加载哪个功能，并预先填入已有的账户信息(setDeualt)
2. 加载学生的注册详情填写功能(switchPanelT)
3. 加载教师注册详情填写功能(switchPanelS)
4. 错误预先判断，以避免异常的产生(所有类都有，后面不赘述了)

图2.5 教师的注册详情填写

图2.6 学生的注册详情填写

### 2.6.8、RegisterWriteT类

预先将Register中注册的信息填写到本类窗体并锁定(无法更改)

功能：完成姓名的填写

### 2.6.9、RegisterWriteS类

预先将Register中注册的信息填写到本类窗体并锁定(无法更改)

功能： 完成信息的填写

院系和专业是相关联的，当院系出现更改是，专业组合框会进行重新加载并重绘界面。

### 2.6.10、Center类

主要窗体功能的骨架，会根据账号的类别来加载相应的功能

功能：

1. 加载学生的功能面板(switchS)

加载前会调用学生面板的初始化函数(initValue)进行数据预加载。

1. 加载老师的功能面板(switchT)

加载前会调用老师面板的初始化函数(initValue)进行数据预加载。



图2.5 教师的功能面板

图2.6 学生的功能面板

### 2.6.11、BaseS类

对SInfo(学生个人信息)、SReport(学生成绩单)类进行预加载(包含数据加载)

图2.8 SReport

### 2.6.12、SInfo类

提供了学生对个人信息的修改，修改密码时会使用PwdCh类，当然，修改信息界面的信息都是预加载的(如果有的话)，院系和专业还是关联的。见图2.7。

### 2.6.13、SReport类

提供了学生对个人信息的查看，仅此功能。(当然也是预加载的)

### 2.6.14、PwdCh类

本类提供了密码修改的功能，并且会把修改完的信息回写到SInfo或者Tinfo类中。(当然也是自动判断)。本类是独立一个线程运行的，自动区分是与TInfo还是SInfo类进行通信。

### 2.6.15、BaseT类

对TInfo、TScore、TScoureCount和TSinfo类进行预加载(包含数据加载)

图2.14 TInfo窗体

### 2.6.16、TInfo类

可以修改教师的个人信息

### 2.6.17、TSInfo类

对学生信息的增删查改方法

功能：

1. 字串查询

查询时会调用Ser类方法，输入在Ser中的数据会写入缓存中供本类调用。

自动搜索学号、姓名、性别、院系和专业有关的字段，并列出。

1. 添加信息

会调用 Add 类进行操作。

1. 修改信息

会预先将相应的学生信息写入Update类中，方便操作。

1. 取消查询

其实就是类似“复位”的操作，这个操作会重新加载表单。

1. 删除操作

删除一个学生的信息以后，如果该学生有成绩记录，相应的成绩记录也会被删除，并重新加载TScore类和TScoreCount类中的列表信息。

### 2.6.18、TScore类

对学生成绩信息进行增删查改的类

功能：

1. 字串查询

查询时会调用Ser类方法，输入在Ser中的数据会写入缓存中供本类调用。自动搜索学号和姓名有关的字段，并列出。

1. 添加

添加学生的成绩信息，需要至少输入学生的学号，若已有成绩记录则会报错。调用 Add 类方法。

1. 取消查找

重新加载列表。

1. 修改

调用Update类方法，预先加载相应数值在其界面中(如果有的话)。

1. 删除

删除成绩信息时，会自动重载TScoreCount类中的数据，以保证信息的一致。

### 2.6.19、TScoreCount类

这个类实现了高分排序、低分排序和学号排序的功能，在排序过程中会自动计算各成绩平均值，并注入到缓存中，点击平均值按钮会显示各科目以及总分的平均值。

* + 1. Add、Ser、和Update类

这三个类有2份，一份为TSInfo类服务，另一份为TScount类服务。

这三个类都是独立的，有自己的线程，还会为与其服务的类进行通信。

## 2.7、项目测试

测试方法：人工测试，下面是各个方法测试情况：

Login类：

最小化后恢复对界面无影响

不输入直接登入会触发设置的限制之一（账户未输入），并弹出警告

试图往账户框随便输入，触发事件，任何输入的字符只要不是数字会自动消除，在第八位时，不管怎么按，都无法输入（被消除了）

试图不输入密码，触发限制并弹出警告（请输入密码）

试图进行SQL注入，无法在账户框输入除数字外的其他字符（别想了）

Register类：

最小化后恢复对界面无影响

试图不输入任何数据就注册，触发限制（未输入有效信息）

试图不输入密码注册，触发限制（未输入有效密码）

试图只输入一位密码进行注册，触发限制（输入密码不得小于4位）

试图注册已存在的用户，触发限制（该账号已存在）

试图输入非4位工号或非8位学号，触发限制（账号必须是四位或八位）

RegisterWriteT类：

直接确认，无任何问题，字符框中的提示信息会自动被过滤，数据库相应的属性填写的是“没有任何信息”（不占用空间，即NULL，不是字符NULL）

RegisterWriteS类

直接确认，无任何问题，字符框中的提示信息会自动被过滤，数据库相应的属性填写的是“没有任何信息”（不占用空间，即NULL，不是字符NULL）

SInfo类：

最小化后恢复，对界面产生重大影响（界面消失），经测试后，发现最小化后恢复将界面变成了1x1的像素点，现已在paint方法中进行了修复处理，该问题已解决。后面的测试表明SReport、TInfo、TSInfo、TScore和TScoreCount类中都有这样的重大BUG，现在都已经修复。

SInfo、SReport、TInfo、TSInfo、TScore和TScoreCount类都通过了上面列举的测试。

测试结果：已无任何BUG

测试结论：该程序实现了某教师提出的所有需求，且无任何BUG

## 2.8、项目总结

该程序活用了各种监听事件，使得程序的安全性，用户体验提升了一大截。

使用paint方法的调用的规则解决意想不到的重大BUG。（每次paint时，会预先进行界面大小的强制调整）

进行本课程设计的时候，对大多数的swing组件进行了使用，熟悉了一些组件的使用方法，比如JTable。

## 2.9、参考文献

Oracle JDK 9 Documentation

——https://docs.oracle.com/javase/9/