五、详细设计说明书

[1．引言 1](#_Toc506973482)

[1.1编写目的 1](#_Toc506973483)

[1.2项目背景 1](#_Toc506973484)

[1.3定义 2](#_Toc506973485)

[1.4参考资料 2](#_Toc506973486)

[2．总体设计 2](#_Toc506973487)

[2.1需求概述 2](#_Toc506973488)

[2.2软件结构 2](#_Toc506973489)

[3．程序描述 2](#_Toc506973490)

[3.1功能 2](#_Toc506973491)

[3.2性能 2](#_Toc506973492)

[3.3输入项目 2](#_Toc506973493)

[3.4输出项目 2](#_Toc506973494)

[3.5算法 2](#_Toc506973495)

[3.6程序逻辑 2](#_Toc506973496)

[3.7接口 2](#_Toc506973497)

[3.8存储分配 2](#_Toc506973498)

[3.9限制条件 2](#_Toc506973499)

[3.10测试要点 2](#_Toc506973500)

# 1．引言

## 1.1编写目的

信息管理成为学校教学管理中十分重要又相当复杂的管理工作之一， 过去传

统的靠人力手工登记， 已经随着当前学生人数的增多， 越来越不符合教育和管理

的要求。而计算机的运算速度快， 处理能力强等特点， 恰好可以解决这一疑难问

题。因此为了保证学校的信息流畅， 工作高效， 设计一个学生成绩管理系统已经

是相当有必要了。 这不但能为广大教工人员减轻了成绩管理的工作， 而且对于推

动教学的发展也起到非常重要的作用。

我们开发其系统主要是为了帮助广大教师提高工作效率，节约资源，提高学

籍信息的精确度，实现学生成绩信息管理工作流程的系统化、规范化和自动化。

## 1.2项目背景

随着科学技术的不断提高 , 计算机科学日渐成熟 , 其强大的功能已为人们深

刻认识 , 它已进入人类社会的各个领域并发挥着越来越重要的作用。作为计算机

应用的一部分 , 使用计算机对学生信息进行管理 , 具有手工管理所无法比拟的优

点。例如，检索迅速、查找方便、可靠性高、存储量大、保密性好、寿命长、成

本低等。这些优点能够极大地提高学生档案管理的效率 , 也是企业的科学化、正

规化管理 , 与世界接轨的重要条件。因此，开发学生成绩管理系统很有必要。

系统名称： 学生成绩管理系统

用 户： 学校教务管理人员及各个学生教师

主管部门: 学校教务处

## 1.3定义

本软件属于小型数据库管理信息系统。学生管理主要是指计算机系学生的日常生活和学习的量化式管理，即将学生日常行为和主要学习的过程通过数字进行量化考核，以方便对于学生综合能力的实际测评，从而促进学生全面发展，在学生中树立公平合理的竞争氛围。本软件设计多种管理模式，适合不同角色如学校教务处，学院，老师，学生使用，控制不同用户的不同权限，具有良好的普适性。

【列出本档中用到的专门术语的定义和缩写词的原文。】

|  |  |  |
| --- | --- | --- |
| 专门术语 | 缩写词 | 原文 |
| 数据库管理系统 | DBMS | Database manege system |
| 只读内存镜像 | ROM | ReadOnly Memory image |
| 随机存取存储器 | RAM | random access memory |

## 1.4参考资料

《项目开发计划》 数据库实践第五组

《需求规格说明书》 数据库实践第五组

《可行性研究报告》 数据库实践第五组

《软件工程导论》 . 陈明 . 机械工业出版社

《管理信息系统》 . 薛华成 . 清华大学出版社

《数据库原理开发及应用》 . 周屹. 清华大学出版社

# 2．总体设计

## 2.1需求概述

不同角色的用户登陆后可进行不同操作：

1.学生 查询学生个人信息，选课信息及选课成绩。

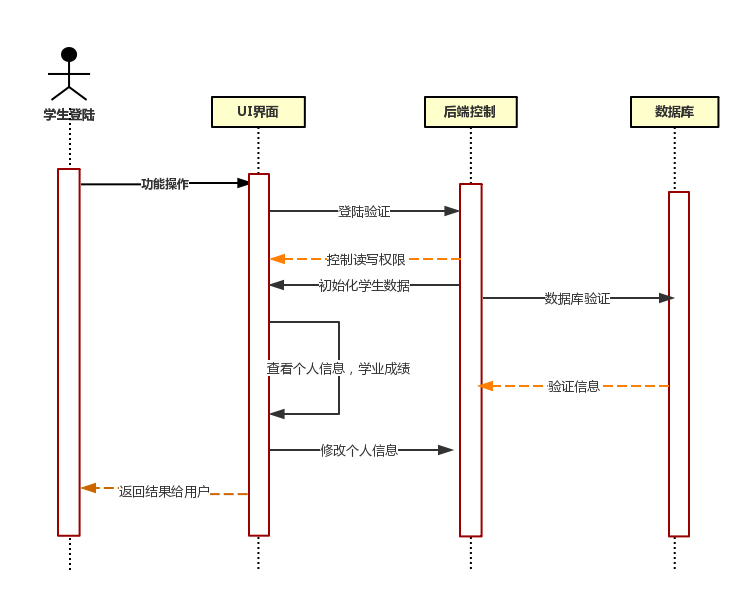
2.教师 查询教师个人信息，讲授课程信息及修改课程成绩。

3.学院教务处 查询学生，课程及教师信息，增加学生，增加老师，增加学院课程。

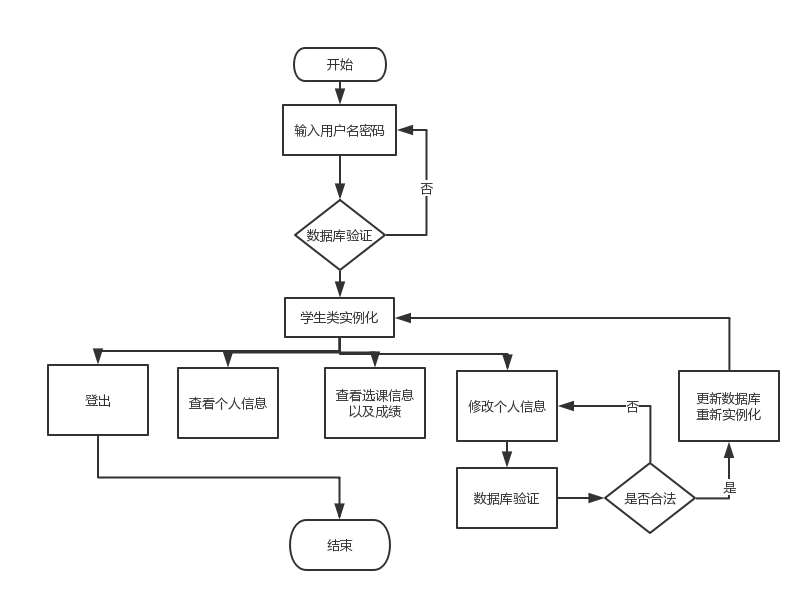
4.学校教务处 查询学院及学院下各个学生课程及教师。

### 2.1.1学生登陆系统

学生可以查看自己选课情况以及成绩情况，登陆系统之后会自动判断学生的身份信息，设定读写权限，页面会展示学生的照片，学号，姓名，入选年份，选课内容，当学生点击选课内容后会列出详细的选课详情，以及课程成绩，还有授课教师的其他信息，学生角色不具有修改数据库的权限。



学生登陆系统时序图

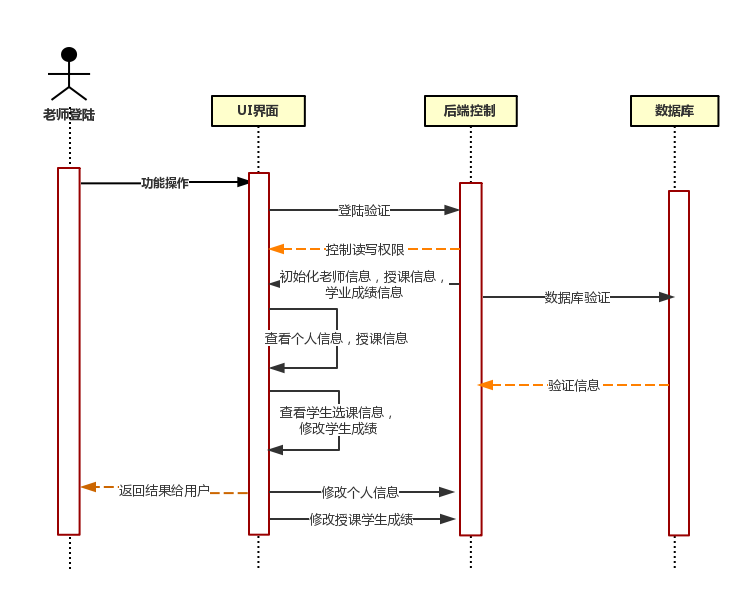


学校登陆系统流程图

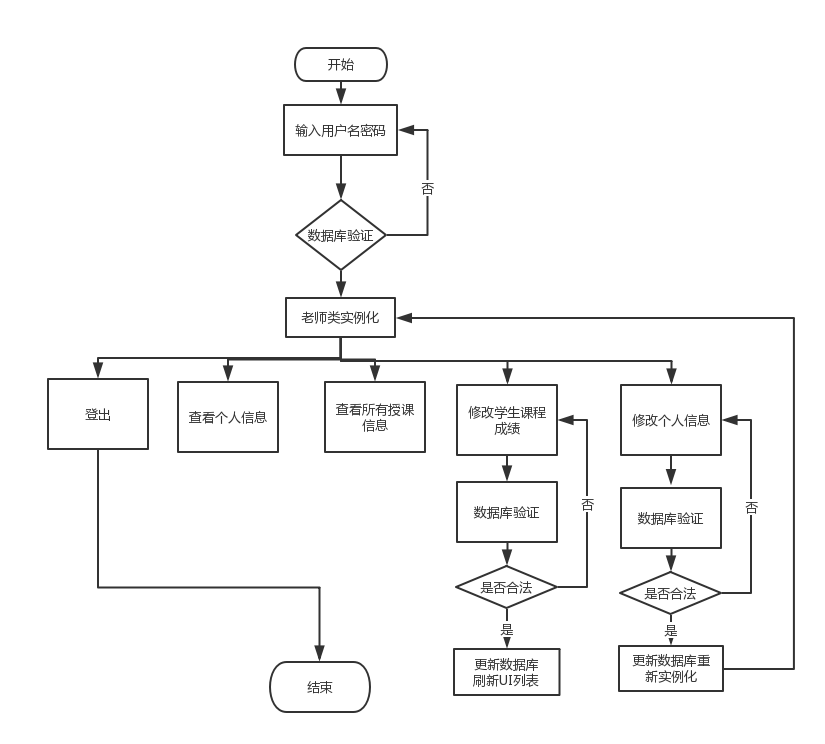
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 函数名 | 类 | 传入参数 | 返回值 | 备注 |
| Init() | Student | String INT | void | 初始化学生对象 |
| showCrouse() | Student | Course | Boolean | 从数据库读取该学生选课信息及成绩 |
| showInfo | Student | Cno | Boolean | 展示学生个人信息 |

### 2.2.2 老师登陆系统

老师可以查看自己所教授的课程，以及参与课程的学生成绩。UI界面会展示老师的个人信息，包括教工号码，教师职位，教师学历，工作简历，授课内容，教师角色可以查看选择此课程的学生信息，教师角色可以修改学生成绩。



教师登陆系统时序图

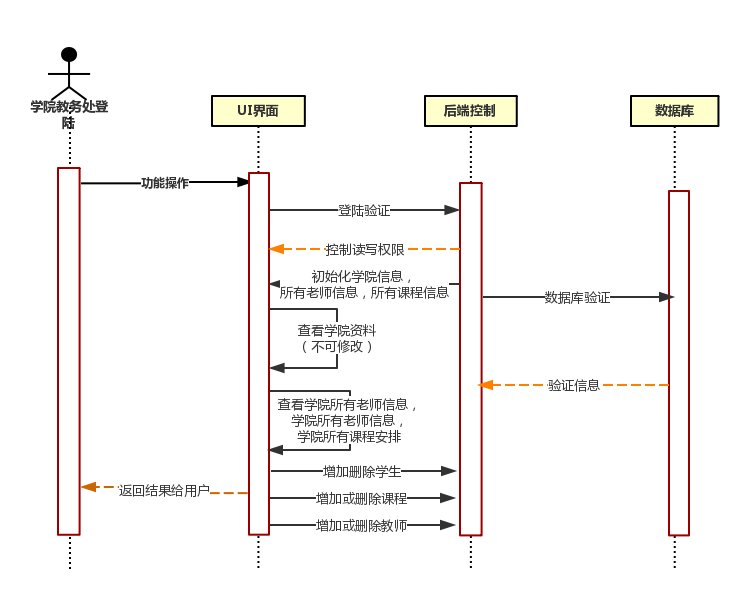


学校登陆系统流程图

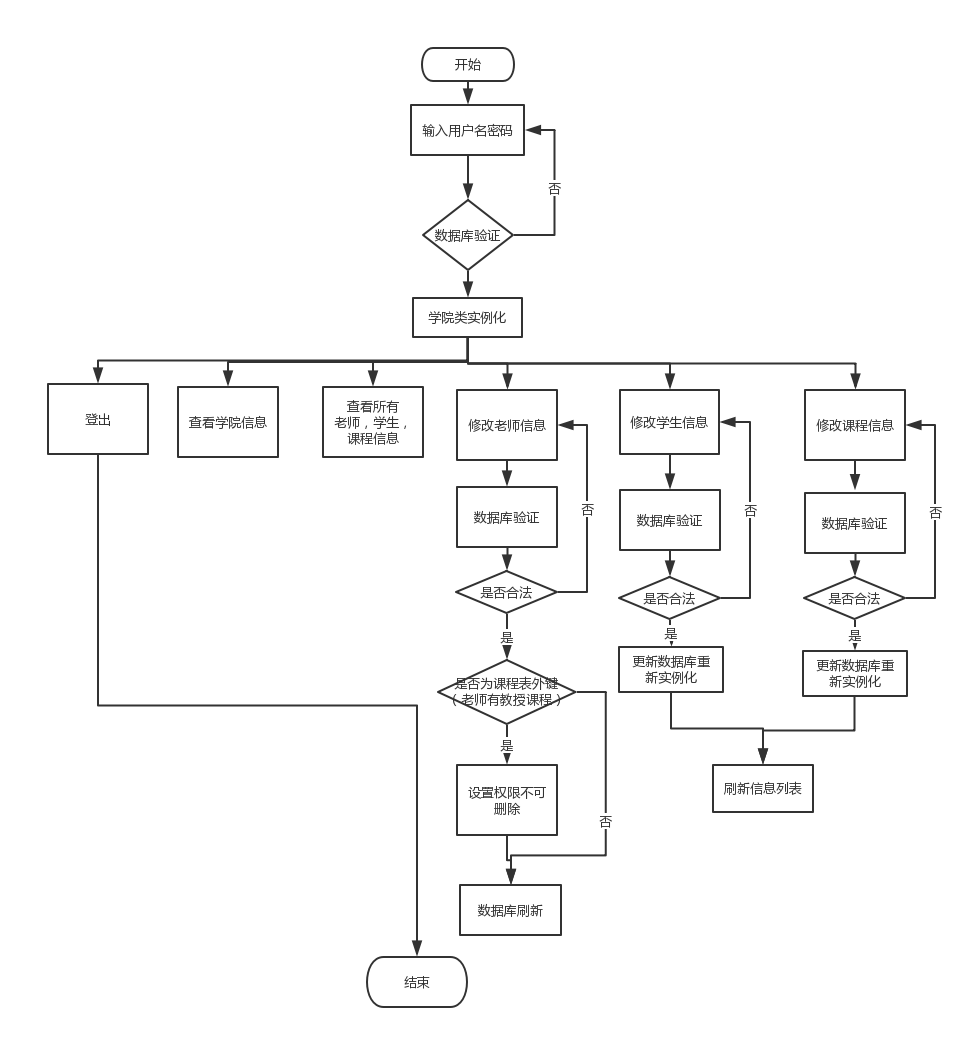
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 函数名 | 类 | 传入参数 | 返回值 | 备注 |
| Init() | Teacher | String INT | void | 初始化老师对象 |
| showCrouse() | Teacher | INT Tno | Boolean | 从数据库读取老师授课信息选课信息及成绩 |
| showInfo（） | Teacher | INT Tno | Boolean | 展示老师个人信息 |
| AddStu\_Cno | Teacher | Course | Boolean | 添加选课学生 |
| ChangeStu() | Teacher | Stu\_cno | Boolean | 修改学生成绩 |

### 2.2.3 学院教务处登陆系统

学院可以查询本学院学生，课程及教师信息，增加学生，增加老师，增加学院课程。学院界面会有三个列表，学院所有老师信息，学院所有学生信息，学院所有课程安排，学院可以修改或者增加学生实体，老师实体，还有课程实体，在删除时候会有一定权限限制。



学院登陆系统时序图

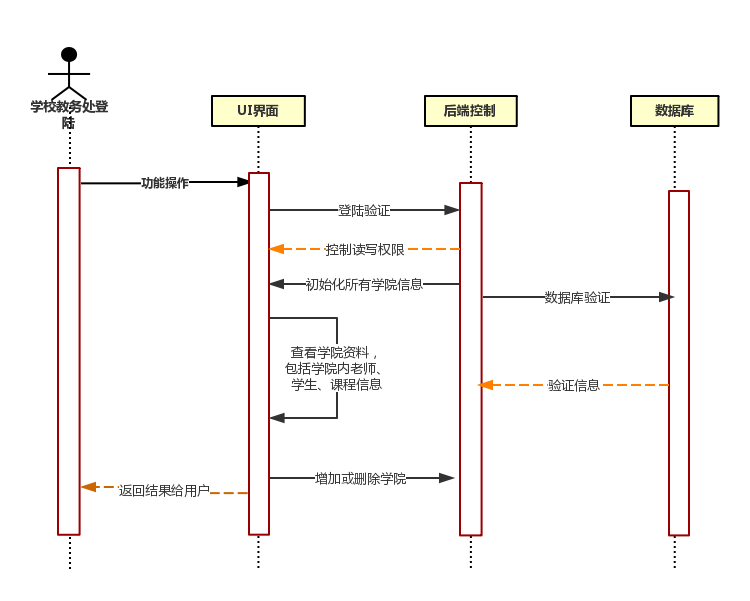


学校登陆系统流程图

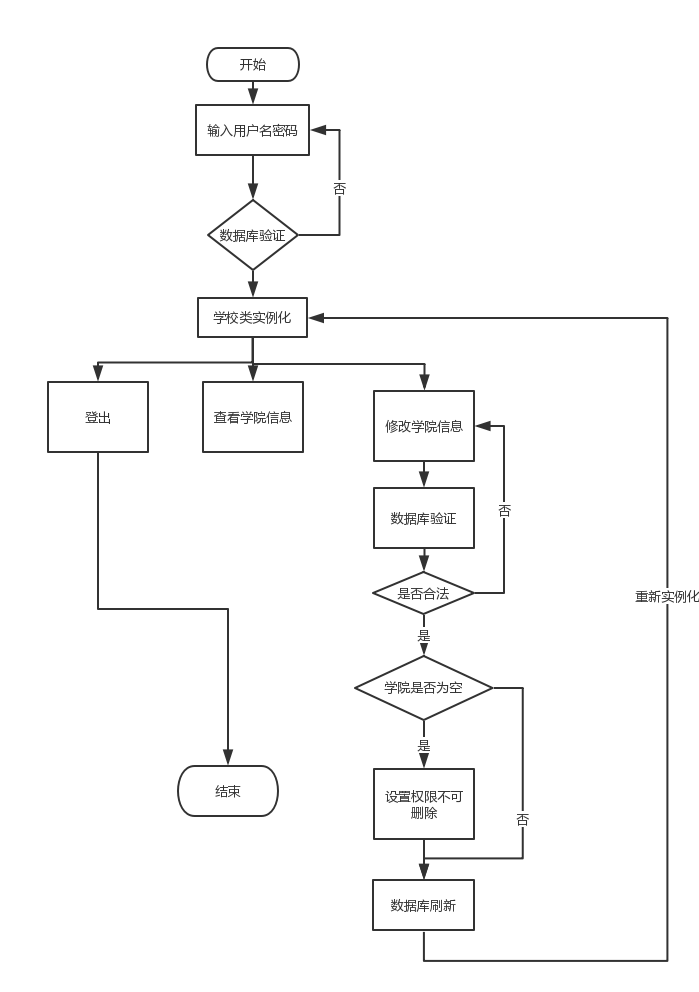
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 函数名 | 类 | 传入参数 | 返回值 | 备注 |
| Init() | Faculty | String INT | void | 初始化学院对象 |
| showCrouse() | Faculty | INT Fno | Boolean | 从数据库读取学院课程信息 |
| ShowStudent | Faculty | INT Fno | bollean | 读取展示学院所有学生 |
| ShowStudent |  | INT Fno | bollean | 读取展示学院所有老师 |
| showInfo（） | Faculty | INT Tno | Boolean | 展示学院信息 |
| AddCno | Faculty | Course | Boolean | 添加课程 |
| AddTeacher | Faculty | Teacher | Boolean | 添加老师 |
| AddStudent | Faculty | student | Boolean | 添加学生 |

### 2.2.4 学校教务处登系统

学校可以查询各个学院的信息，并且进行任何数据库内容的修改，可以视为数据管理最高权限者。



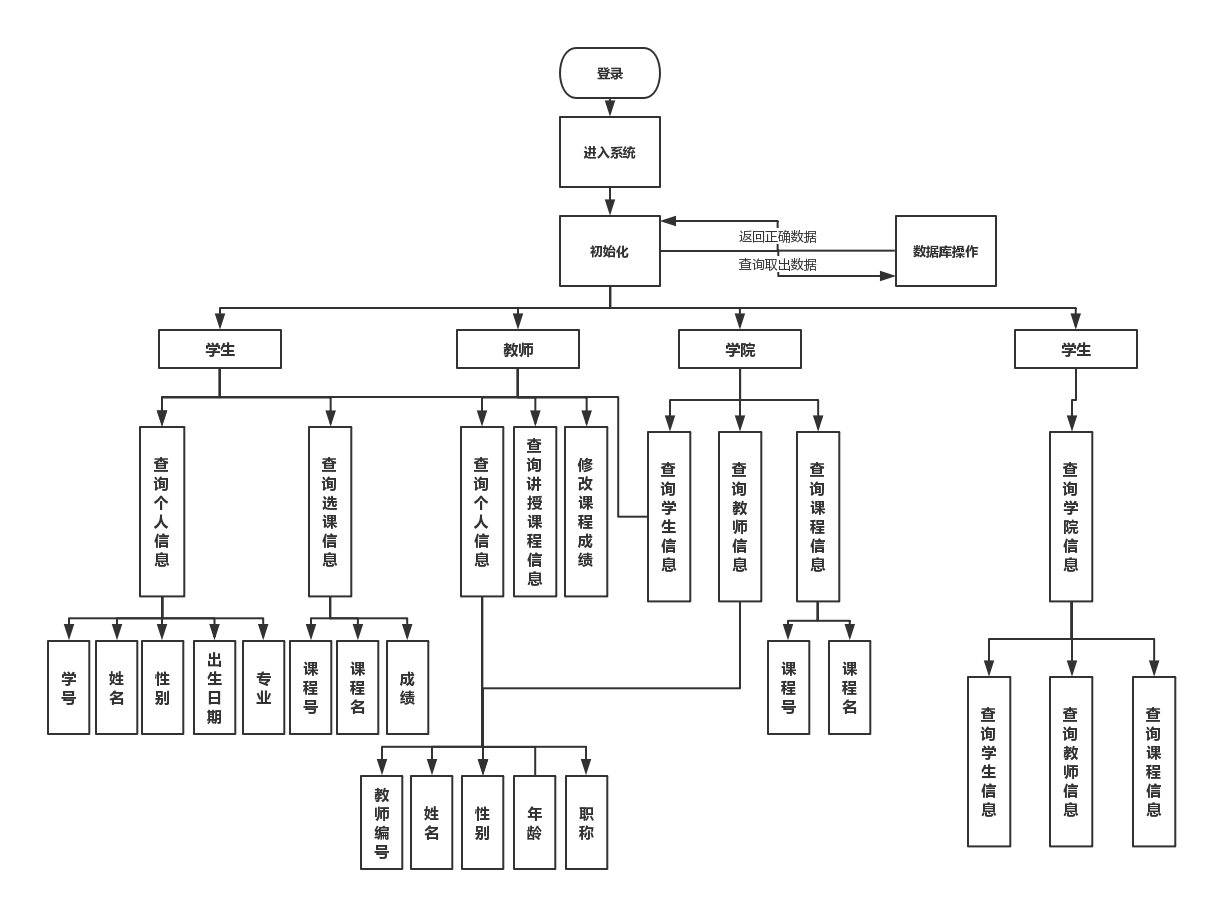
学校登陆系统时序图



学校登陆系统流程图

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 函数名 | 类 | 传入参数 | 返回值 | 备注 |
| Init() | School | String INT | void | 初始化学校对象 |
| showCrouse() | School | INT Fno | Boolean | 从数据库读取学院课程信息 |
| ShowFaculty | School | void | Boolean | 展示所有学院信息 |
| ShowStudent | School | INT Fno | bollean | 读取展示学院所有学生 |
| ShowStudent | School | INT Fno | bollean | 读取展示学院所有老师 |
| showInfo（） | School | INT Tno | Boolean | 展示学院信息 |
| AddCno | School | Course | Boolean | 添加课程 |
| AddTeacher | School | Teacher | Boolean | 添加老师 |
| AddStudent | School | student | Boolean | 添加学生 |
| AddFaculty | School | Faculty | Bollean | 添加学院 |
|  |  |  |  |  |

## 2.2软件结构

 软件为三层框架，MVC，及MODEL ,WIEW,CONTROL，view和control主要由JAVAfX控制，model主要由mysql数据库实现。

软件整体架构图

# 3．程序描述

## 3.1功能

### 3.1.1整体设计模块

整体模块划分图

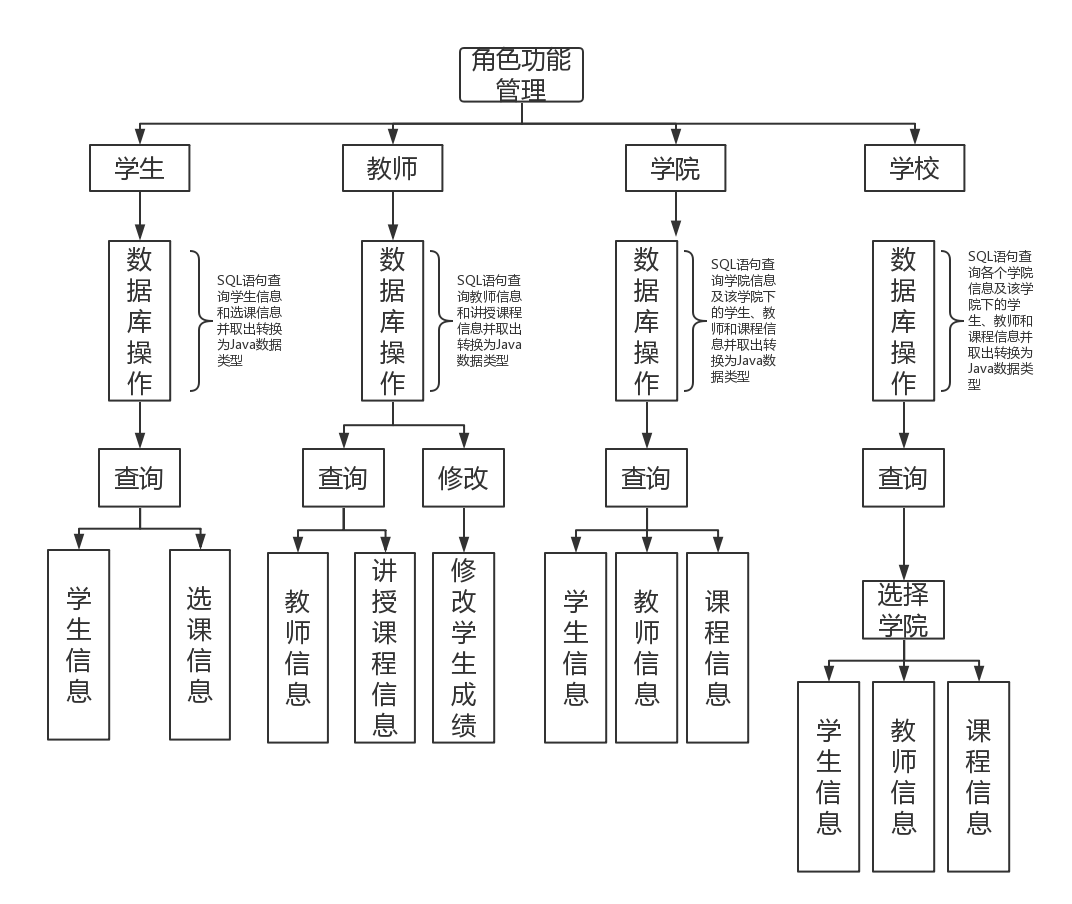
主要涉及三大功能模块，数据库，用户界面，角色权限控制，数据库使用JDBC连接mysql，用户界面使用JAVAfx相关控件，使用数据监听模式，将数据与列表list控件双向绑定，修改数据后触发事件，更新数据库操作。角色权限有两部分，登陆时根据用户的用户名密码判断角色身份，实现不同的初始化，角色进行修改数据库内容操作时候经行第二部份权限控制。

### 3.1.2 单体模块设计

#### 登录模块：

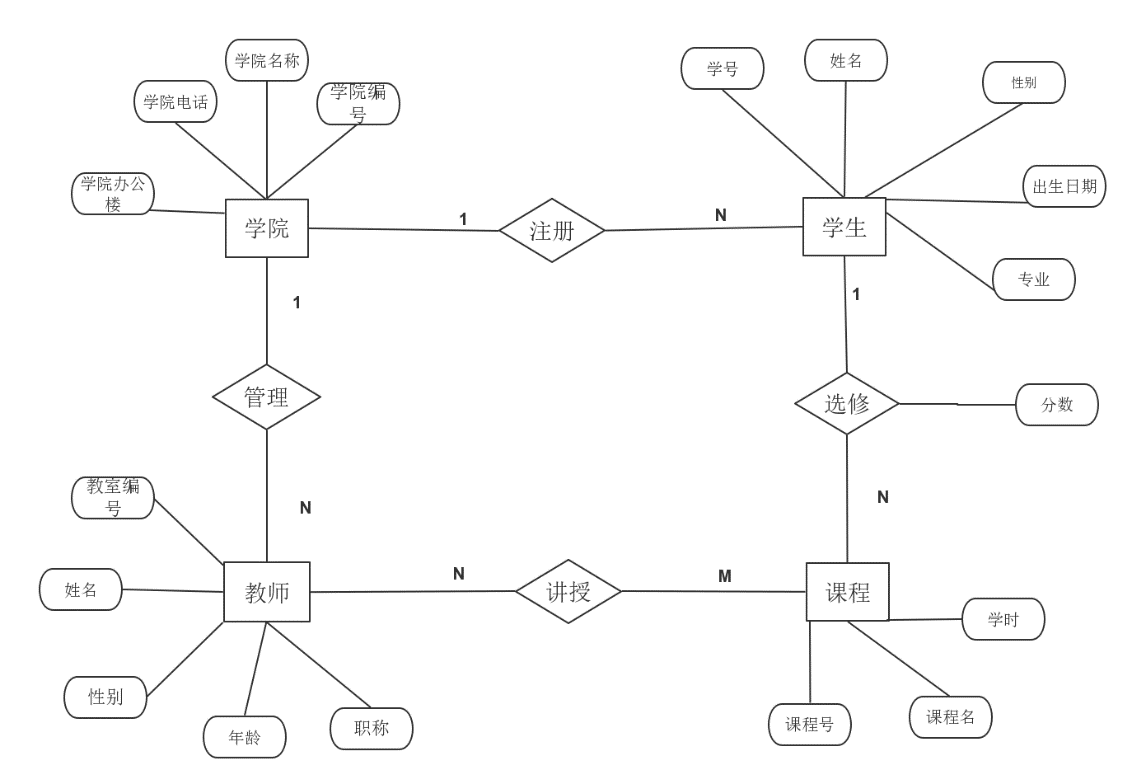
登录模块图

#### 角色功能管理模块：

角色功能管理模块图

#### 数据库模块

数据库模块图

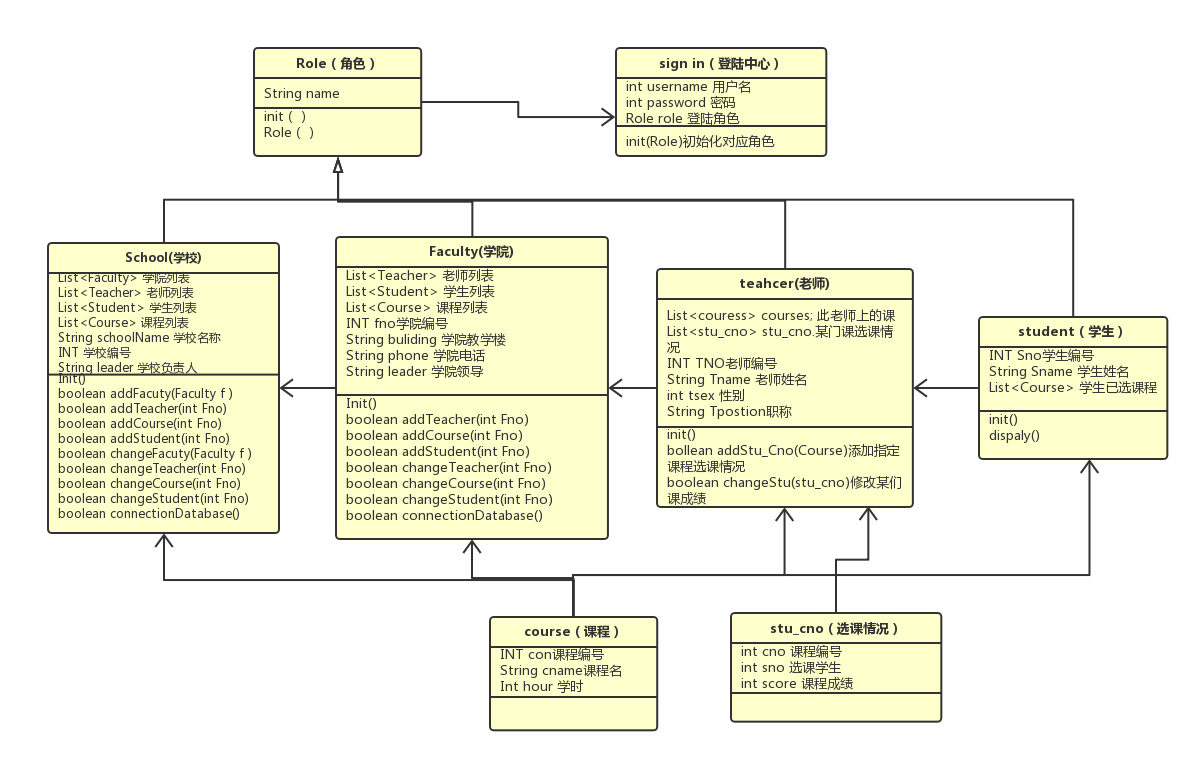


全局ER图

#### 用户界面模块

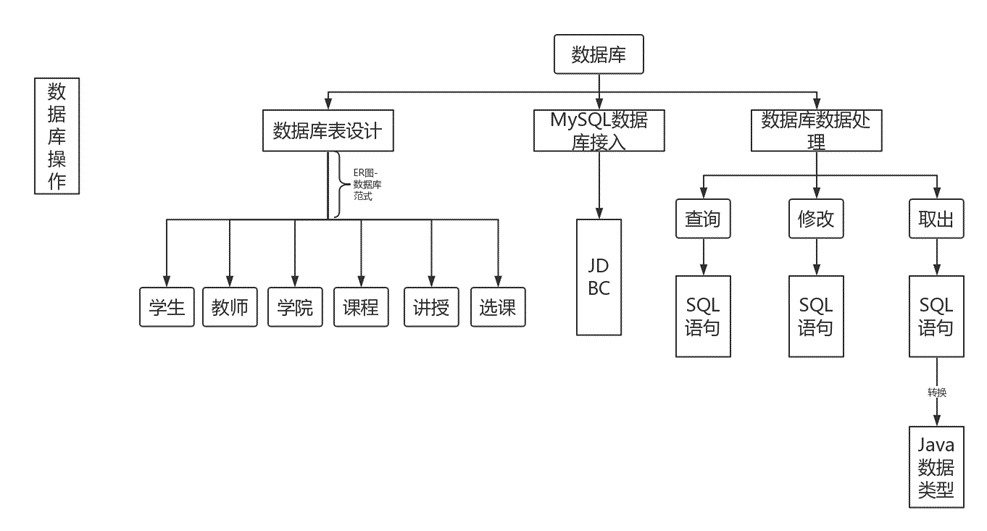
用户界面模块图

### 3.1.3 类设计，函数设计

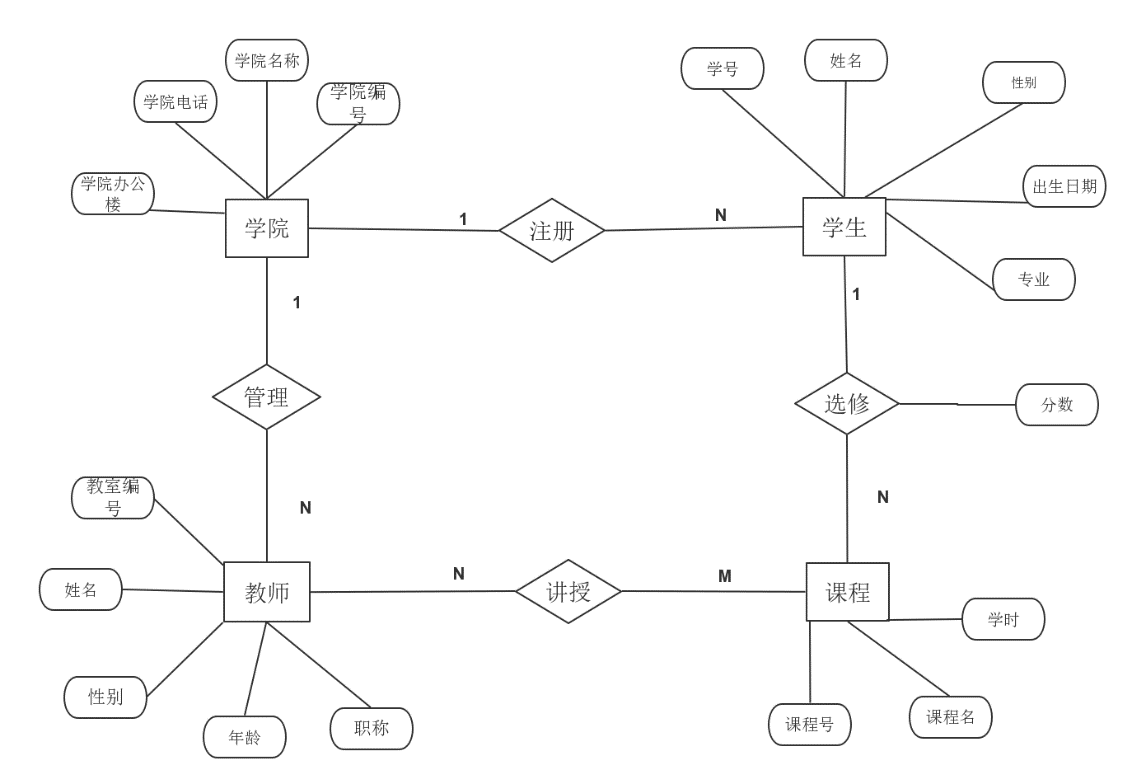
本系统设计学生，教师，学院，学校，四个不同角色，根据角色权限和功能需求设计类。

系统全局类图

### 3.1.4 数据库设计方案

1. 整体模块设计

2.全局ER图



3.表设计

数据库设计有6张表，红色表示主键，蓝色表示外键，根据ER图设计完成，具有良好的可拓展性。

学院：faclty(Fno,Fname,Fphone,Fbuilding)

教师：teacher(Tno,Tname,Tsex,Tage,Tpostion,Fno)

学生：student(Sno,Sname,Ssex,Sbirth,Fno)

讲授：tea\_cou(Tno,Cno)

课程：course(Cno,Cname,Chour)

选修：stu\_cou(Sno,Cno,Score)

faculty学院信息表的结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 字段说明 | 类型 | 位数 | 属性 | 备注 |
| 1 | FNO | 学院编号 | INT | 11 | 不可为空 | 主键 |
| 2 | Fname | 学院名称 | VARCHAR | 45 | 不可为空 |  |
| 3 | Fphone | 学院电话 | VARCHAR | 45 | 不可为空 |  |
| 4 | Fbuildin | 学院地点 | VARCHAR | 4 | 不可为空 |  |
| 5 | Fleader | 学院领导 | VARCHAR | 45 | 不可为空 |  |

teacher教师信息表的结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 字段说明 | 类型 | 位数 | 属性 | 备注 |
| 1 | TNO | 教师编号 | INT | 11 | 不可为空 | 主键 |
| 2 | Tname | 教师姓名 | VARCHAR | 45 | 不可为空 |  |
| 3 | Tsex | 性别 | VARCHAR | 45 | 不可为空 |  |
| 4 | Tage | 年龄 | VARCHAR | 4 | 不可为空 |  |
| 5 | Tpostion | 教师职称 | VARCHAR | 45 | 不可为空 |  |
| 6. | Fno | 所属学院 | INT | 11 | 不可为空 | 外键 |

student学生信息表的结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 字段说明 | 类型 | 位数 | 属性 | 备注 |
| 1 | SNO | 学生编号 | INT | 11 | 不可为空 | 主键 |
| 2 | Sname | 学生姓名 | VARCHAR | 45 | 不可为空 |  |
| 3 | Ssex | 性别 | VARCHAR | 45 | 不可为空 |  |
| 4 | Sage | 年龄 | VARCHAR | 4 | 不可为空 |  |
| 5 | Sbirth | 入学时间 | VARCHAR | 45 | 不可为空 |  |
| 6. | Fno | 所属学院 | INT | 11 | 不可为空 | 外键 |

course课程信息表的结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 字段说明 | 类型 | 位数 | 属性 | 备注 |
| 1 | CNO | 课程编号 | INT | 11 | 不可为空 | 主键 |
| 2 | Cname | 课程名字 | INT | 45 | 不可为空 |  |
| 3 | Chour | 课程学时 | INT | 45 | 不可为空 |  |
| 4 | Cstart | 开课时间 | Date | 11 |  |  |
| 5 | Fno | 所属学院 | INT | 11 | 不可为空 |  |

Sno\_cno学生选课信息表的结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 字段说明 | 类型 | 位数 | 属性 | 备注 |
| 1 | SNO | 学生编号 | INT | 11 | 不可为空 | 主键，外键 |
| 2 | Cno | 课程编号 | INT | 45 | 不可为空 | 主键，外键 |
| 3 | Score | 课程成绩 | INT | 45 | 不可为空 |  |

tno\_cno教师讲授表的结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 字段说明 | 类型 | 位数 | 属性 | 备注 |
| 1 | TNO | 教师编号 | INT | 11 | 不可为空 | 主键，外键 |
| 2 | CNO | 课程编号 | INT | 11 | 不可为空 | 主键，外键 |
|  |  |  |  |  |  |  |

### 3.1.4 UI设计方案



用户登陆界面



UI操作界面

## 3.2性能

# 1、高吞吐量 满足高并发下的大数据量交互需求，满足数据备份或ETL过程的大数据量迁移。具体需求信息获取参照以上数据库应用需求。

# 2、负载均衡 满足高并发下数据库的负载均衡能力，需求分析需要收集数据库的部署架构、负载均衡策略等数据信息。

# 3、读写分离 获取需求的要点是明确哪些是写节点，哪些是读节点，并且切换的策略什么，数据同步的策略是什么。

# 4、分区分片（分库分表） 获取需求的要点是把握数据的垂直切换和水平分库概念。明确需要对哪些数据块进行切分，分别分散到哪几台数据库主机上；需要对哪些大表进行数据水平切分，并且分布到哪些DB或table中。通过需求分析，做出数据切分的合理性判断，以及做出系统可测性的判断。

# 5、高并发 根据以上的数据库应用需求，进一步制定数据库的高并发需求，估算出单台数据库的API接口压力和需要满足的并发能力。

# 6、高可用性 高可用性可能也综合涉及到数据的多项能力，主要应用的是集群技术，HA容错及互备技术，体现的是无故障运行。获取需求的要点是明确高可用性技术架构，了解HA采用的工作方式，以及掌握故障切换方法和数据一致性验证需求。

## 3.6程序逻辑

程序流程图

## 3.7接口

## 3.8存储分配

## 3.9限制条件

### 3.9.1运行环境

硬件环境：1、PC机，CPU 1G Hz以上，内存 1G字节，硬盘空间10G 处理器：100M Hz以上 ROM+RAM：32M以上

软件环境：Windows 2000 以上版本的32位*系统*,含:Windows 2000、Windows XP、Windows 2003、Vista、Win7、Windows 2008

### 3.9.2数据库技术

（1）数据库数据显示技术

通过什么方式和形式把数据库中的学生数据显示到界面上供管理者查阅，目前的基本技术都是基于Sql语言的混合编程实现遍历数据库，对数据的提取，然后通过GUI编程在界面上显示。

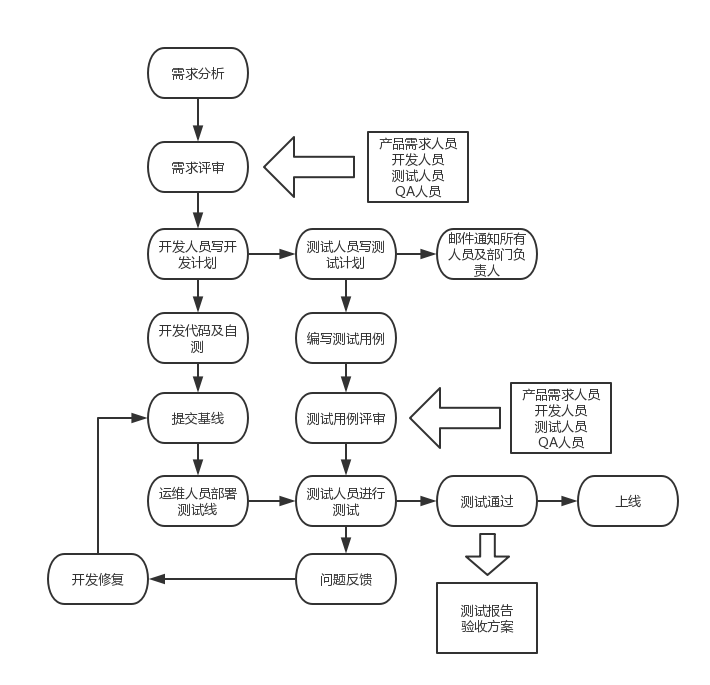
目前的界面编程和Sql语言的混合编程已经比较成熟，比如用于界面编程的有C、Java、Qt、C++等等，并且他们基本都是开源的，不会存在侵权和成本的问题。

（2）数据库数据修改技术

对数据库的操作基本的还是离不开Sql语言的混合编程，主要是通过对按钮事件的捕捉，通过对应的Sql代码实现对数据库相关操作

## 3.10测试要点

### 3.10.1测试计划



### 3.10.2质量保证计划

