

详细设计说明书

1 引言

1.1 编写目的：

教务管理系统是完成学生管理、班级管理、课程管理和教师管理几个管理模块，由教务工作人员系统给管理员、教师、以及学生多用户角色，各用户依据自己角色的不同而操作不同的功能模块，教务工作人员分别通过学生管理、班级管理、课程管理、教师管理和用户管理来进行这些模块的管理，包括这些模块的添加、修改、删除以及对课程的选择、学生信息的查询，教务工作人员同时通过教师管理模块了解学生学习的全部过程。本说明书确定系统的详细功能模块和数据结构，为后一阶段开发系统提供最详细的依据材料。

1.2 项目背景：

- (1) 开发软件的名称：天盛教务系统
- (2) 项目的任务开发者：单联天、盛家谱
- (3) 用户及实现软件的单位：中国石油大学（北京）

1.3 定义：

JDBC：Java 数据库连接，是 Java 语言中用来规范客户端程序如何来访问数据库的应用程序接口，提供了诸如查询和更新数据库中数据的方法。

SQL：结构化查询语言，是一种数据库查询和程序设计语言，用于存取数据以及查询、更新和管理关系数据库系统。

1.4 参考资料：

- [1]张俊,宋雪勤,曹慧敏,等.高校教务管理系统选课算法研究综述[J].攀枝花学院学报,2017(2):49-52.
- [2]白伟丽.学生选课系统的设计与实现[J]电脑与电信, 2018(10):38-40.
- [3]李帅,黄克敏,杨义,等.高校学生选课系统的研究与设计[J].电脑知识与技术,2017(9):99-100.

2 总体设计

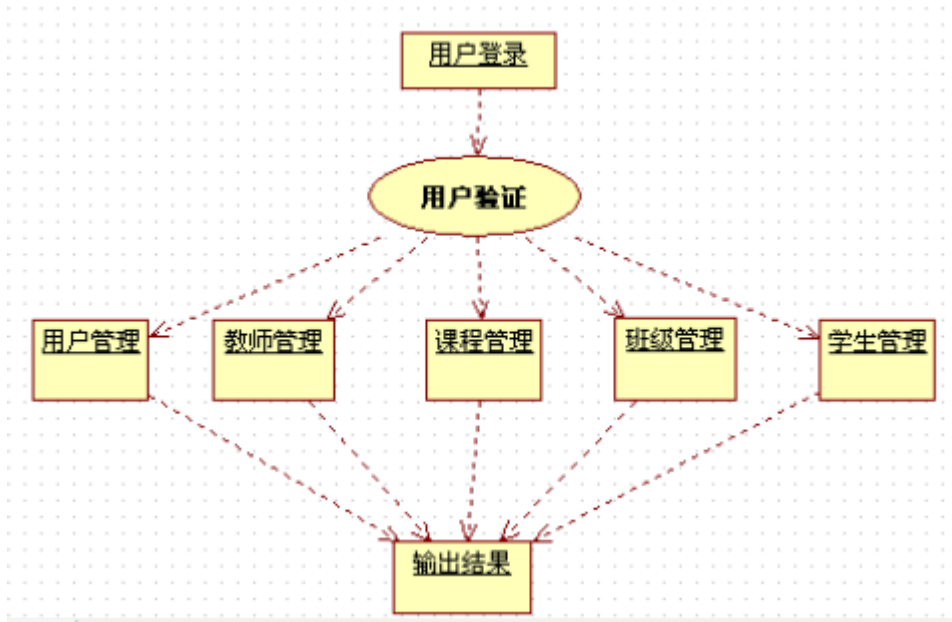
2.1 需求概述

高效的教学管理系统是当前信息化学校管理系统的需求。面对日益庞大的教学信息，如果是单纯的人工来管理或者是小型的不完善的系统来管理会很浪费时间、精力等，所以就需要有教学管理系统来提高管理工作的效率。通过建立这样的系统，可以做到信息的规范管理、快速查询和科学统计，从而减少管理方面的工作量。

本系统的设计面向高校的教学管理内容，经过充分地系统调研，发现本系统应包括院系信息、教研室信息、班级信息、教师信息、学生信息、课程信息、学生专业信息等需管理的信息。进一步还发现信息实体间需管理的信息有：专业分学期课程计划、学生选课及各类考试成绩登记、补考重修管理、研究生管理、班级管理、教师教学质量评估等内容。

2.2 软件结构：

如给出软件系统的结构图。



3 程序描述

3.1 登录模块

此页为登陆页面，用户通过输入用户名称和密码进入相应的页面。页面要求：用户名输入框，密码输入框，以及用户类型选择框。其中用户名为明文输入，密码框输入后显示*，用户类型框为单选框。当用户输入信息后作相应的验证并给出输入错误提示。

3.2 管理员管理模块

3.2.1 用户管理模块

功能：在用户ID处输入用户ID后点击检索将显示该用户ID的详细信息。

输入项目：鼠标键盘事件。

输出项目：浏览区和基本资料区。

测试要点：配置文件或数据库文件损坏，或没有足够内存可供使用时是否正常。

3.2.2 用户添加模块

功能：在现有用户资料的基础上添加。

输入项目：鼠标或键盘事件。

输出项目：更新后的数据库文件。

3.2.3 用户修改模块

功能：在现有用户资料的基础上修改。

输入项目：鼠标或键盘事件。

输出项目：更新后的数据库文件。

3.3 教师管理模块

功能：显示教师登录欢迎界面，进入正常工作状态。

输入项目：鼠标键盘事件。

输出项目：浏览区和基本资料区。

测试要点：配置文件或数据库文件损坏，或没有足够内存可供使用时是否正常。

3.3.1 教师添加模块

3.3.2 教师修改模块

3.3.3 教师删除模块

3.4 课程管理模块

功能：对用户操作进行反应，调出课程信息。

输入项目：鼠标键盘事件。

输出项目：浏览区和基本资料区。

测试要点：配置文件或数据库文件损坏，或没有足够内存可供使用时是否正常。

3.4.1课程添加模块

3.4.2课程修改模块

3.4.3课程删除模块

3.5学生管理模块

功能：显示学生的基本信息。

性能：要求在1~2秒内作出反应。

输入项目：鼠标键盘事件。

输出项目：浏览区和基本资料区。

3.2 算法：

3.2.1 用户登录验证方法：

`public int JudgeExist(Model.User.Users objUsers, out string strErr)`

`Model.User.Users objUsers` 为模型类，接收用户的账号和密码及权限信息，执行后返回值 0 和 1，当验证失败时返回 0，并用 `out string` 输出错误原因；当验证成功时返回 1，`out strErr` 为空。

3.2.2 修改用户密码的方法：

`public int ModifyUser(string UserName , Model.Users.Users objUsers, out string strErr)`

`string UserName` 是定义接收用户名的变量，根据用户名查找用户，`Model.Users.Users objUsers` 是模型类，得到用户的新密码，`out string strErr` 定义捕获错误的变量，当返回值为 1 则修改成功，返回值为 0，则出错，将错误赋给 `strErr`

3.2.3 查询的方法：

查询所有记录

`public int SearchUser(out DataTable objDataTable,out string strErr) out DataTable`

`objDataTable` 定义了一个表，`out string strErr` 定义捕获错误的变量，当返回值为 1 则查询成功，定义的表显示查询结果，返回值为 0，则出错，将错误赋给 `strErr`

查询某行记录

`public int SearchUser(string UserName , out DataTable objDataTable, out string`

`strErr) string UserName` 是定义接收用户名的变量，根据用户名查找用户的相关信息，`out DataTable objDataTable` 定义了一个表，`out string strErr` 定义捕获错误的变量，当返回值为 1 则查询成功，定义的表显示查询结果，返回值为 0，则出错，将错误赋给 `strErr`

3.2.4 学生信息管理的具体实现：

1.添加的方法：

`public int AddStu(Model.student.student objStu,out string strErr)`

`Model.student.student objStu` 是模型类，`out string strErr` 定义捕获错误的变量，`int` 为 1 则添加成功，同时添加用户成功，模型类存储学生的相关信息，为 0 则返回错误提示，将错误赋给 `strErr`

2.修改的方法：

`public int ModifyStu(int iStuld, string strOld, Model.student.student objStu, out string`

strErr) int iStuld, 是传入的学生 ID, string strOld 是传入的旧的学生编号, Model.student.student objStu 是模型类, out string strErr 定义捕获错误的变量 int 为 1 则修改成功, 同时修改用户成功, 模型类存储学生的相关信息, 为 0 则返回错误提示, 将错误赋给 strErr

3.查询的方法:

public int SearchStu(out DataTable objDataTable, out string strErr, string strErr) out DataTable objDataTable 定义了一个表, out string strErr 定义捕获错误的变量, 当返回值为 1 则查询成功, 定义的表显示查询结果, 返回值为 0, 则出错, 将错误赋给 strErr。 public int SearchStu(string strStuNo,string StuName,string ClassName, out DataTable objDataTable, out string strErr)

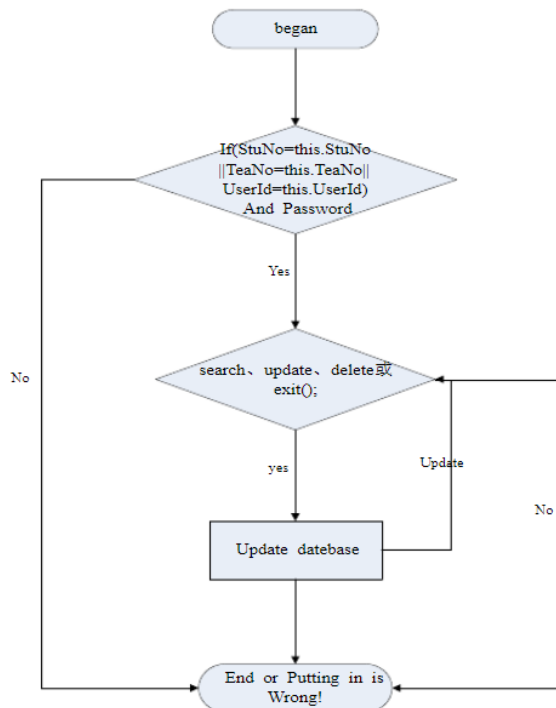
管理员根据学生编号、学生姓名、学生所在班级进行查询, string strStuNo 定义接收学生编号的变量,string StuName 定义接收学生姓名的变量,string ClassName 定义接收学生所在班级的变量, out DataTable objDataTable 定义了一个表, out string strErr 定义捕获错误的变量, 当返回值为 1 时, 根据传入的学生编号和学生姓名和所在班级查询学生成功, 将查询结果给表, 返回值为 0, 则将错误赋给 strErr。

4.删除的方法:

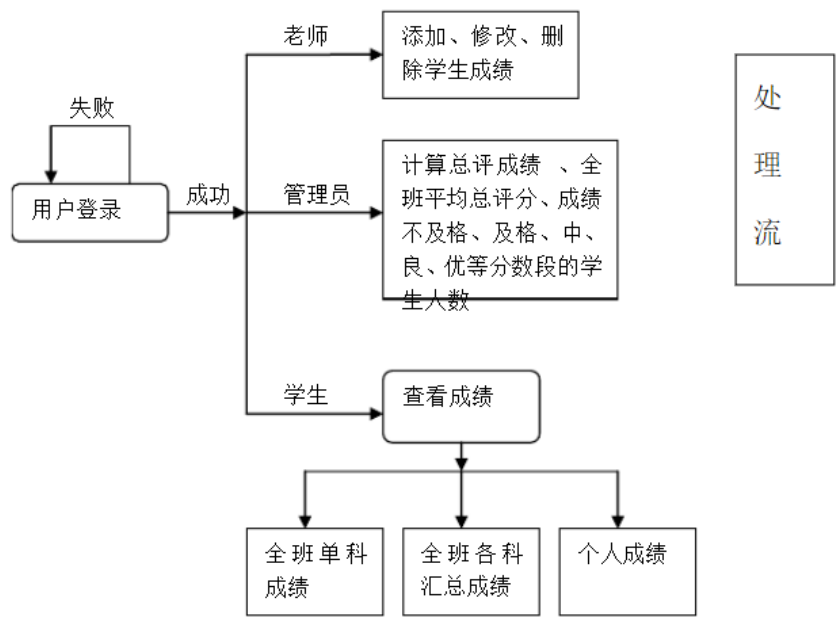
public int DelStu(int iStuld, string strStuNo, out string strErr) 删除学生和账号, int iStuld 定义接收学生 ID 的变量, string strStuNo 是定义接收学生编号的变量, out string strErr 定义捕获错误的变量, 当返回值为 1 时, 删除成功, 当返回值为 0 时, 出错, 则将错误赋给 strErr。

3.3 程序逻辑:

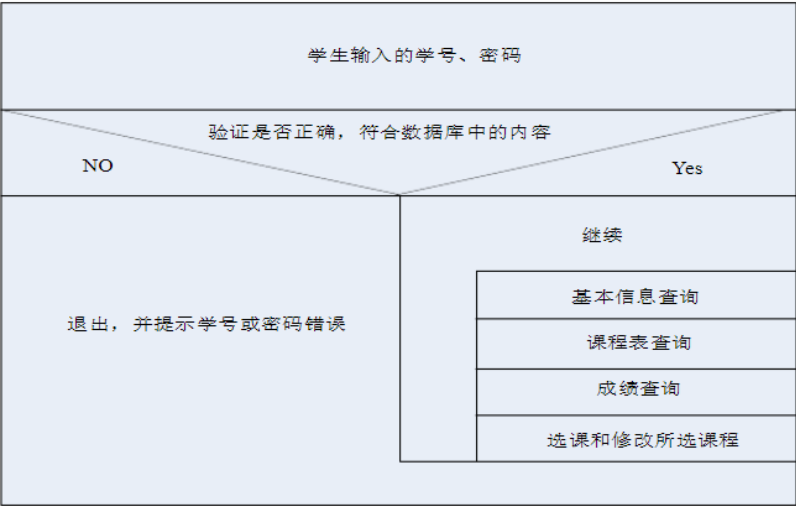
登录功能的实现流程图:



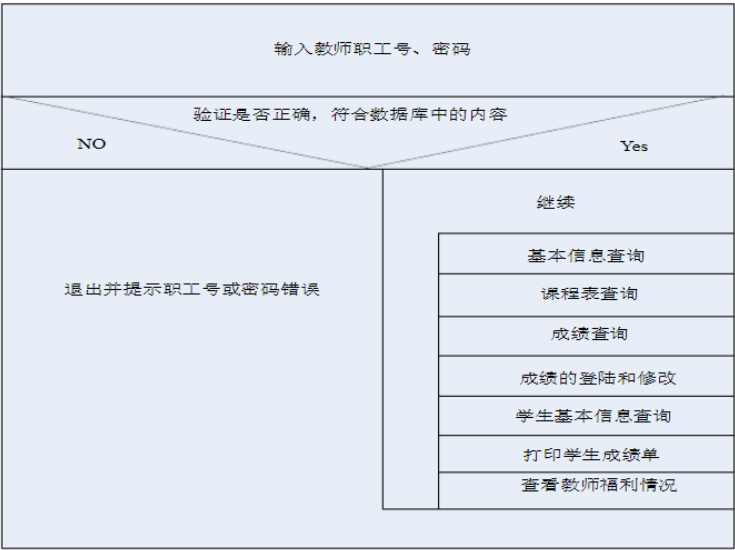
处理流程：



查询学生信息以及学生可用的功能：



教师信息及其可以进行的操作：



管理员的登录及其可以进行的操作：



3.4 接口

(1) 存储分配；

在编程时进行动态内存分配是非常必要的。它可以在程序运行的过程中帮助分配所需的内存，而不是在进程启动的时候就进行分配。然而，有效地管理这些内存同样也是非常重要的。在大型的、复杂的应用程序中，内存泄漏是常见的问题。当以前分配的一片内存不再需要使用或无法访问时，但是却并没有释放它，那么对于该进程来说，会因此导致总可用内存的减少，这时就出现了内存泄漏。尽管优秀的编程实践可以确保最少的泄漏，但是根据经验，当使用大量的函数对相同的内存块进行处理时，很可能会出现内存泄漏。尤其是在碰到错误路径的情况

下更是如此，所以存储分配很重要，在我们建立数据库时一定要分配好内存。

(2) 限制条件。

- 1.所建议系统的运行寿命的最小值
- 2.进行系统方案选择比较的时间
- 3.硬件、软件、运行环境和开发环境方面的条件和限制
- 4.可利用的信息和资源
- 5.系统投入使用的最晚时间。

3.5 测试要点：

- 1.测试方案:使用以界面为基础的测试

以界面为基础的测试仅仅依靠软件与其运行环境之间的界面来选择和产生测试数据，而不管软件的具体需求和具体实现细节。包括软件输入，输出数据的类型取值范围以及取值的概率分布等等。

- 2.测试项目:该测试计划主要包括对软件各模块进行测试。

- 1.系统登录窗口测试
- 2.系统管理模块测试
- 3.修改密码功能模块测试
- 4.学生、教师、及一些课程等的录入测试
- 5.学生、教师、及一些课程等删除测试
- 6.学生、教师、及一些课程等更新测试
- 7.学生、教师、及一些课程等增加测试