**《软件系统详细设计说明书》**

团队名称： 你又怎么了我错了行了吧

指导教师： 代祖华

完成时间： 2019年6月4日

**[学生信息关系系统]**

**目录**

[一.引言 3](#_Toc10555240)

[1.1编写目的 3](#_Toc10555241)

[1.2 背景 3](#_Toc10555242)

[1.3 参考资料 3](#_Toc10555243)

[二.程序系统结构 3](#_Toc10555244)

[三．程序设计说明 4](#_Toc10555245)

[3.1 程序描述 4](#_Toc10555246)

[3.2 程序功能 4](#_Toc10555247)

[3.2.1 功能分析 5](#_Toc10555248)

[3.3 运行环境 5](#_Toc10555249)

[3.4 输入项 6](#_Toc10555250)

[3.4.1学生板块 6](#_Toc10555251)

[3.4.2教师板块 7](#_Toc10555252)

[3.4.3管理员板块 7](#_Toc10555253)

[3.5 输出项 7](#_Toc10555254)

[3.5.1全部功能 8](#_Toc10555255)

[3.6 流程逻辑 8](#_Toc10555256)

[3.6.1功能数据流图 8](#_Toc10555257)

[四．主要功能设计说明 9](#_Toc10555259)

[五．测试要点 12](#_Toc10555260)

# 一.引言

## 1.1编写目的

本详细设计说明书，是在概要设计说明书的基础上进一步明确系统结构，详细的介绍系统的类与精化类的属性和操作，为进行后面的编码和测试做准备。

预期读者为：开发人员与系统使用者及用户代表

## 1.2 背景

A.待开发的软件系统的名称：学生信息管理系统；

B.本项目的任务提出者：代祖华老师；

C.开发者：团队开发（你又怎么了我错了行了吧）；

D.用户：高校的学生和教师（先立足于西北师范大学）；

E. 随着计算机网络技术的发展，给信息时代的人们带来了很大的方便。如今在Internet上，普及了更多，功能更加完整的各类管理系统。而针对于高校内使用的学生管理系统涉及到更高的安全性和方便性，这些在网上的资料很为零散。故我们为了整理和统一，为此进一步完善本系统的开发。使得此过程更加方便，更加透明，更加高效，以节省更多的人力和不必要的工作。

F. 而目前的此类系统或多或少的存在以下的问题：

1、不清楚建立网站的目的或没有根据自己的目的详细策划网站的功能，只适应一时之需；结果页面的设计，包括系统内的链接和功能未能真正提供方便；

2、安全性和其余功能与现实某些高校需求不符，在并不完全适合的情况下，体现出的不适合。或是针对某一点做出的管理系统，参差不齐。

G.通过对高校内学生信息及各方面管理工作的初步了解，从系统结构的组织，功能的实现，技术的要求以及可行性等多方面进行考虑，认为本课题是一个适应现今学生信息管理工作需求的计算机信息管理系统，具有一定的实际开发价值和使用价值。

1.3 参考资料

a．《构建之法》

b. 软件需求说明（GB8567-88）

c．《软件工程导论》

d.《学生信息管理系统开发内容和要求》

e.《学生信息管理系统需求规格说明书》

# 二.程序系统结构

本系统由用户操作模块、学生基本信息模块、宿舍模块、成绩模块、选课模块、学院模块、班级模块六个部分组成；

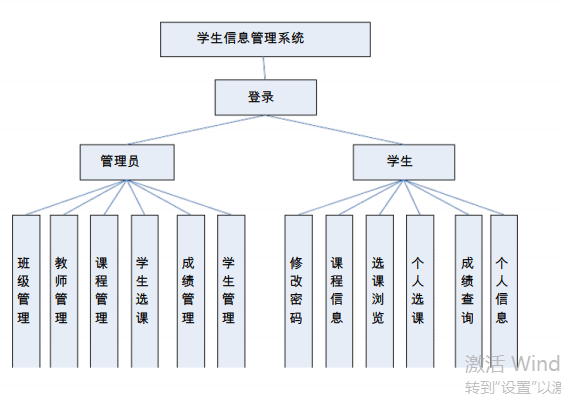


图2-1系统结构图

# 三．程序设计说明

## 3.1 程序描述

## 随着互联网的发展，利用 INTERNET 技术来实现“无纸办公”这个概念已经深入人 心，校园网作为学校信息化建设的一个平台在完成资源共享、互联网访问、教务管理、电子 备课等方面发挥了重要作用。服务教学、提高教学水平和效果是校园网建设的核心目标和核 心价值，如何利用现有的资源，来更好地服务于学校的教学，服务于学校的管理工作，是摆 在我们面前的一个共同的命题。就目前而言，学校的学籍管理方式还停留在手工操作基础 上，每个学期和学年，教师需要花费大量的时间来登录学生信息，而管理人员也需要花费较 多的时间来校对管理。同时又形成信息册在教师和教导处之间传递的延时性，遇到学生转入转出，又因为各个学校使用的学籍卡的不同，而造成信息的缺失或者遗漏。本系统就是为了 减轻教师繁复的誊写工作、加强学校学生信息管理的方便性。

## 3.2 程序功能

### 主要功能用来对学生的成绩进行收集和修改。具体包括学生成绩录入、修改、删除、查询、打印等功能，由于一些资料可以 Word 文档或 Excel 表格输入，故需要增加导入、导出数据功能。成绩录入需要分班级、分年级以及考试时间。成绩的录入、修改、删除只能由该课程的任课教师操作，其他人只有浏览和查询的权利。查询条件设置应该齐全，可以按班级查询，年级查询，也可以姓名查询，学号查询等等。

### 3.2.1 功能分析

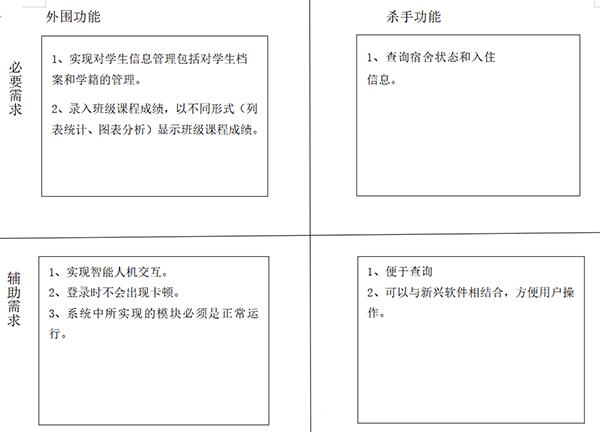


图3-1系统功能图

## 3.3 运行环境

1.设备：

A．server 要求内存在256以上，CPU2.0GB以上

B．Clinet内存在128以上，CPU奔腾III以上，最大支持20台分机连接到主机上

C．Tomcat服务器版本在7.0以上

2.支持软件：

## A.操作系统：windows xp 以上操作

## B.系统数据库管理系统：mysql

## C.编程软件：myeclipse，Navicat

## 3.4 输入项

1.用户输入端：

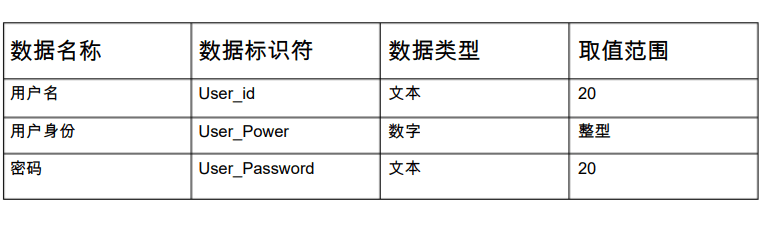


图2-2用户输入项表示图

### 3.4.1学生模块

学生登录：

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| 角色 | Char(3) |  |
| 用户名 | Char（6） | 主键 |
| 密码 | Char（6） |  |

学生表：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否允许为空 | 说明 |
| 学号 | Char（6） | Not null | 主键 |
| 姓名 | Char（8） | Not null |  |
| 专业名 | Char(10) | null |  |
| 性别 | Bit | null | 男 1，女 0 |
| 出生日期 | SmallDatetime | null |  |
| 总学分 | tinyint | null |  |
| 照片 | image | null |  |
| 备注 | text | null |  |

### 3.4.2教师模块

课程表：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否允许为空 | 说明 |
| 课程号 | Char（3） | Not null | 主键 |
| 课程名 | Char（16） | Not null |  |
| 开课学期 | tinyint | Not null | 只能为 1-8 |
| 学时 | tinyint | Not null |  |
| 学分 | tinyint | null |  |

学生选课表：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否允许为空 | 说明 |
| Char（6） | Not null | Not null | 主键 |
| 课程号 | Char（3） | Not null | 主键 |
| 成绩 | tinyint | null |  |
| 学分 | tinyint | null |  |

### 3.4.3管理员模块

管理员登录：

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| 角色 | Char(3) |  |
| 用户名 | Char(6) | 主键 |
| 密码+ | Char(6) | Not null |

## 3.5 输出项

点击功能按钮会实现相应的动作响应，实现对于各个模块表的增删改查功能

## 3.6 流程逻辑

数据流图说明：本系统由教师管理学生的基本信息及选课信息，故展示教师模块为例:

### 3.6.1教师功能数据流图

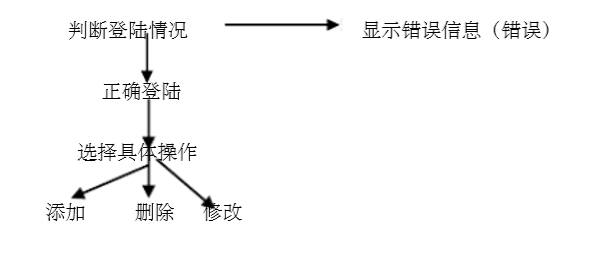


图3-3教师功能数据流图

# 四．主要功能设计说明

由于系统需求分析时更改了定位，之前所做的页面设计太过死板和简单，系统还在开发中，没有成型的界面效果，故此处做文字说明。

学生：可登录、选课

教师：可登陆后对于学生进行全面的管理，附加全部模块的查询

管理员：对所有人进行增删改查的管理

# 五.测试要点

测试工作分为四个阶段：单元测试、组装测试、确认测试、系统测试

单元测试：采用白盒法和黑盒法相结合的方法，对于逻辑结构复杂的模块采用白盒法，对于以输入、输出为主的模块采用黑盒法测试，以提高测试的效率。

组装测试：自底向上的增量测试。

确认测试：由用户参与按需求规格说明书验收。

系统测试：采用人工测试方法。

1. 系统环境模块测试是为了检测系统环境模块，数据响应是否正确，数据能否正确，并进行仔细核对。
2. 基本信息测试是基本信息模块是本系统的一个重要模块，本模块能否正确运行关系到系统设计成败的关键，所以有必要对这个模块进行专门测试。以弥补设计过程中的不足，以便于及早发现和修改问题。

尚未解决的问题：

1. 数据库的设计，起初定为五个表，可是在分析过程中发现至少需要分解为八个表，才能达到最优范式。
2. 编程过程中接口的混乱定义
3. 模块划分问题