五、详细设计说明书

五、详细设计说明书

1．引言 2

1.1 编写目的 2

1.2 项目背景 2

1.3 定义 2

1.4 参考资料 2

2．总体设计 . 3

2.1 需求概述 3

2.2 软件结构 3

3．程序描述 . 4

3.1 功能 4

3.2 性能 4

3.3 输入项目 5

3.4 输出项目 6

3.5 算法 6

3.6 程序逻辑 12

3.7 接口 17

3.8 存储分配 17

3.9 限制条件 18

3.10 测试要点 18

五、详细设计说明书

1．引言

1.1 编写目的

详细说明书的主要内容是关于我们团队所要设计的“教务管理系统”来编写的，随着学校的规模不断扩大，专业、班级、学生、教师等的数量急剧增加，有关学生课的各种信

息量也成倍增长， 而目前许多高校的学生成绩管理仍停留在人工管理阶段， 重复工作较多，工作量大，效率低。因此，迫切需要开发基于网络的学生成绩管理系统来提高管理工作的

效率。它主要介绍的是我们所要实现的功能与性能等，同时它对以后阶段的工作起指导作用，也是项目完成后系统验收的依据。

这份详细设计说明书主要面向系统分析员、程序员、测试员、实施员和最终用户。

1.2 项目背景

开发软件名称：教学管理系统

1.3 定义

Macromedia Dreamweaver 8

：网页设计工具

Microsoft SQL Server

：数据库工具

静态数据－－系统固化在内的描述系统实现功能的一部分数据

动态数据－－在软件运行过程中用户输入的后系统输出给用户的一部分数据， 也就是系

统要处理的数据。

数据字典－－数据字典中的名字都是一些属性与内容的抽象和概括， 它们的特点是数据

的“严密性”和“精确性”，没有半点含糊。

1.4 参考资料

数据库设计 沈洁 清华大学出版社 2007.3.1

《软件工程（第五版） 》张海藩 清华大学出版社

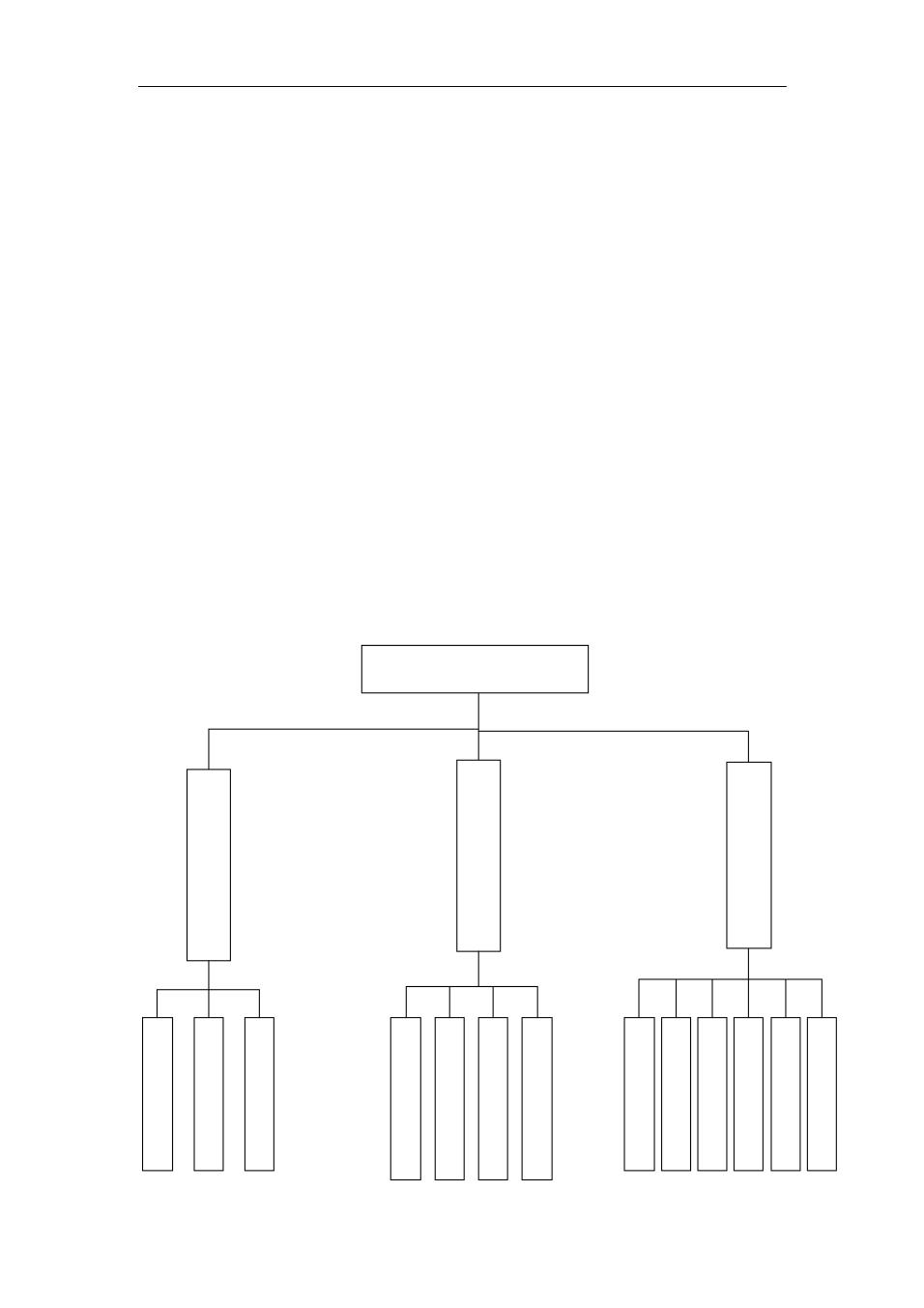
《软件工程模型与方法》 ，肖丁等，北京邮电大学出版社 。

* ASP.NET+Dreamweaver8案例精粹》 武新华等，西安电子科技大学出版社《信息系统应用与开发案例教程》 ，陈承欢，清华大学出版社。

Visual Basic 程序设计基础 林永兴 中国水利水电出版社 2011.1.1

JAVA 编程思想 埃克尔 机械工业出版社 2007.6.1

数据库系统实现 加西亚 莫里亚 机械工业出版社 2010.1.1

五、详细设计说明书

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。

本系统的设计模拟一般学校特别是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

系

统

管

理

的

基

本

信

息

教学信息管理系统

总 统

体 计

功 功

能 能

学 教 专

生 师 业

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用 | 对 |  |  |  |
| 数 | 课 |  |  |
| 户 |  |  |
| 据 | 程 |  |  |
| 登 | 库 | 查 |  |
| 表 | 的 |  |
| 录 | 询 |  |
| 的 | 选 |  |
| 功 | 基 |  |  |
| 本 | 修 |  |  |
| 能 | 操 |  |  |
|  |  |  |
|  | 作 |  |  |  |

学 老 课 专 学

院系

生 师 程 业 生 中班基 基 综 综 选 级或

本 本 合 合 课 学生

情 情 情 情 情 综合

情况

况 况 况 况 况

五、详细设计说明书

3．程序描述

3.1 功能

3． 1 . 1 功能分析

在教务管理系统中 , 主要有用户登陆、学生信息管理、课程信息管理、用户管理、教师信息管理、院系信息、教研室信息、班级信息、统计等功能，主要功能分析如下：

* 1）用户登陆

教务管理系统网站采用用户名及密码验证模式，进入教务管理系统网站前，用户必须选择

身份在登陆页面输入用户名及密码，只有验证通过的用户方可进入教务管理系统网站操作

主页面，才能对其中的内容进行查询或者操作 , 包括对学生、教师等信息的修改。

（2）学生信息管理

包括 3 个功能模块：学生个人信息的添加、学生信息的查询以及学生信息的修改于删除。

（3）课程信息管理

包括 3 个功能模块：课程基本添加与删除处理、成绩查询处理以及学生选课信息的查询处理。

（4）教师信息管理

包括 3 个功能模块：教师个人信息的添加、教师信息的查询以及教师信息的修改与删除。

（5）用户管理

包括 2 个功能模块：用户信息查询以及用户的删除与处理。

* 6）查询院系、教研室、班级信息：实现对于要查询的院系、教研室、班级信息的显示。
* 7）统计功能：统计学生、教师、课程、专业学生、院系中班级或学生综合情况。

3.2 性能

3． 2. 1 性能要求分析

（ 1）系统易操作性

所开发的系统应做到操作简单， 尽量使系统操作不受用户对电脑知识水平的限制， 如果在

以后要添加需要一些电脑知识水平限制的操作则在编写用户操作手册时， 将操作详细的添加

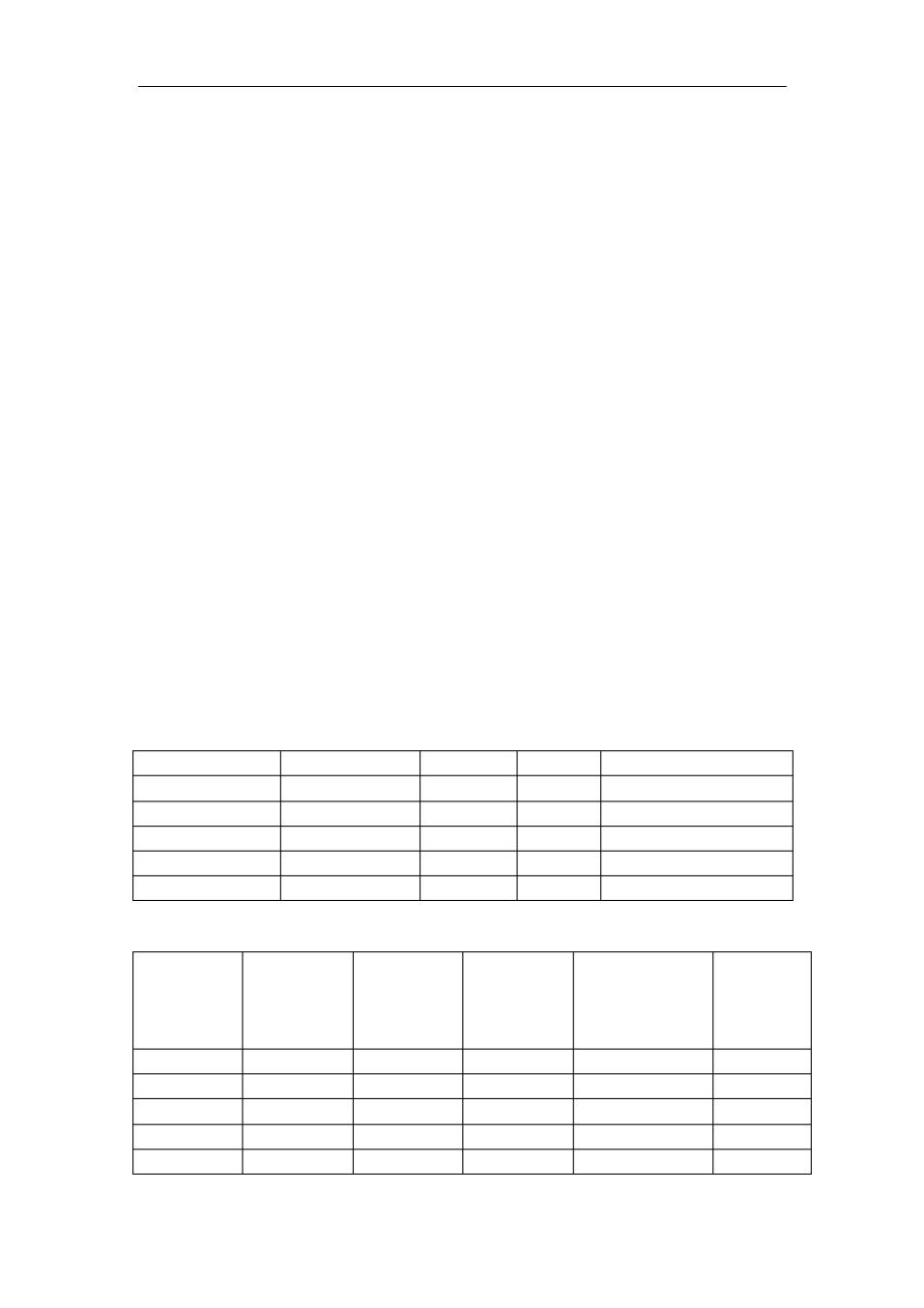
到其中。

（ 2）系统具有可维护性

因为系统涉及了教师、 课程、学生等信息， 比较重要， 数据库中的数据需定期修改、 添加、

删除等操作。 而系统可利用的空间及性能也随之下降， 为了使系统更好地运转， 我们可以对系统数据及一些简单的功能进行独立的维护及调整，或者添加一些功能。

（3）其他的性能要求：

五、详细设计说明书

输入要求：数据完整，详实 .

输出要求：简捷，快速，实时。

说明对于该软件的时间特性要求，

a． 响应时间：少于

3 秒

b． 更新处理时间：少于

2 秒

c． 数据的转换和传送时间少于

5 秒

说明对该软件的灵活性的要求，

即当需求发生某些变化时，

该软件对这些变化的适应能

力，操作方式上的变化：更人性化

1. 2. 2 逻辑模型的建立

经过以上调查分析， 明确以所开发系统的功能需求和性能要求， 发现了存在的问题， 弄清了系统的功能， 为系统逻辑模型的建立提供了依据。 系统分析的主要成果是建立系统的逻

辑模型，本系统的逻辑模型主要以系统的数据流图和数据词典作为描述功具。

3.3 输入项目

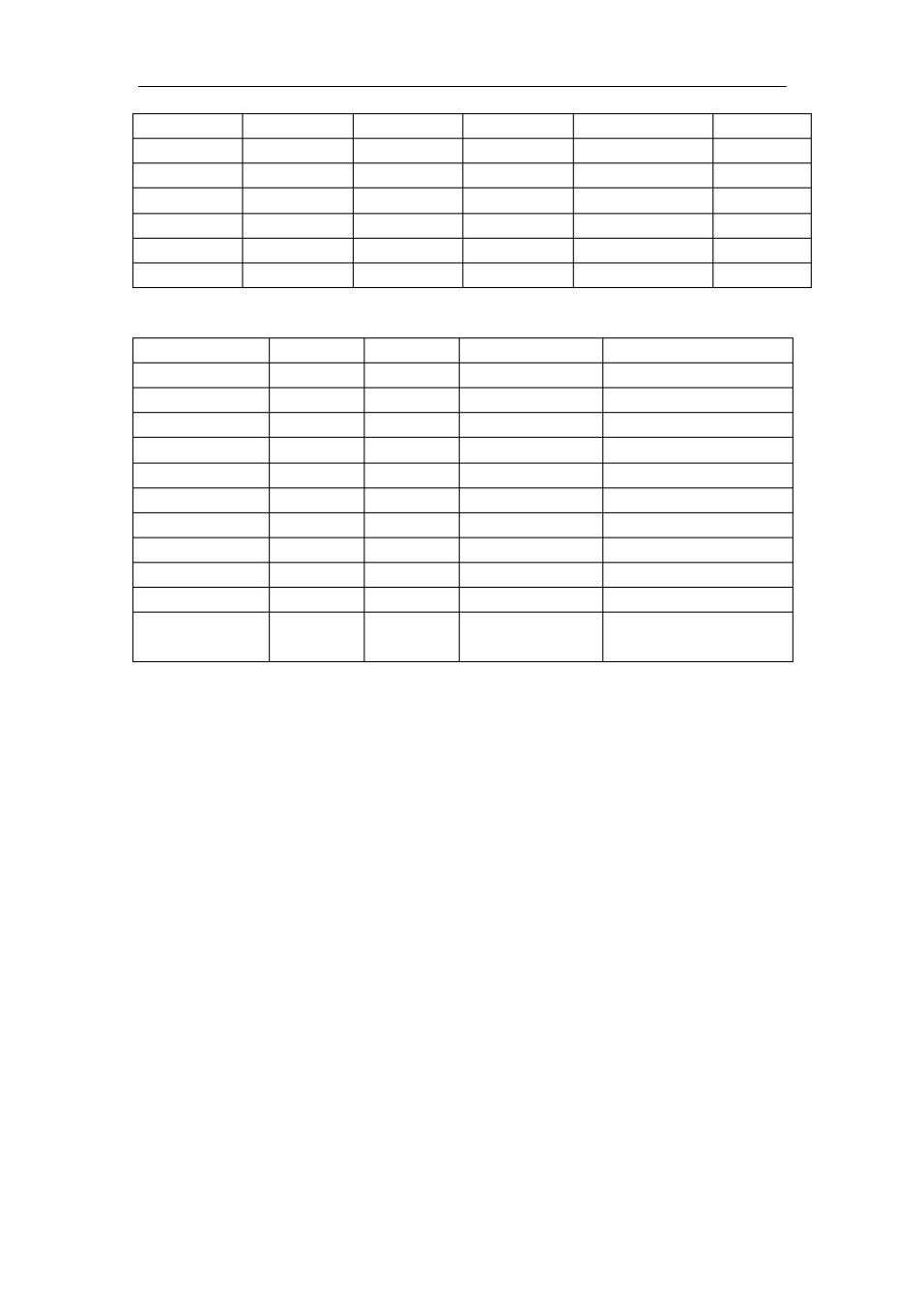
首先打开主界面， 在界面中选择要进行的操作， 若进行其他限制性操作则需要输入用户登录的口令，如：学号，教师职工号，和管理员的编号等。

具体如下例：

表名 <User>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 中文名称 | 类型 | 控制域 | 说明 |
| UserId | 用户 id | int | Public |  |
| UserName | 用户名 | string | Public |  |
| PassWord | 密码 | string | Public |  |
| Rights | 权限 | int | Public |  |
| Type | 删除标识 | int | Public | 为 0 时表示该生被删除 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 <Student> |  |  |  |  |  |  |
|  |  |  |  | 关系 ( 主键或外 |  |  |
| 字段名 | 中文名 | 数据类型 | 是否可为空 | 键 ) 注：如果是外 | 约束和默 |  |
| 键，则写上外键表 | 认值 |  |
|  |  |  |  |  |
|  |  |  |  | 名 |  |  |
| StuNo | 学号 | Char | 否 | 主键 | 自增 |  |
| StuName | 姓名 | Char | 否 |  |  |  |
| StuBirth | 出生日期 | Int | 否 |  |  |  |
| StuSex | 性别 | Char | 否 |  |  |  |
| MinZu | 民族 | Char | 否 |  |  |  |

五、详细设计说明书

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| StuClassId | 班级号 | Char | 是 | 外键 Class |
| EntTime | 入学时间 | Datatime | 否 |  |
| StuTel | 联系方式 | Char | 否 |  |
| StuAddr | 现住址 | Char | 否 |  |
| StuHuji | 户籍所在地 | Char | 否 |  |
| Remark | 备注 | Text | 是 |  |
| Type | 删除标志 | Int | 否 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表名 <Teacher> |  |  |  |  |  |
| 名称 | 中文名称 | 类型 | 控制域 |  | 说明 |
| TeaId | 教师 Id | int | Public |  |  |
| TeaNo | 教师编号 | string | Public |  |  |
| TeaName | 教师姓名 | string | public |  |  |
| Sex | 性别 | string | Public |  |  |
| BirthDate | 出生日期 | DateTime | Public |  |  |
| Degree | 学历 | string | Public |  |  |
| Rank | 职称 | String | Public |  |  |
| TeaTel | 电话号码 | string | Public |  |  |
| TeaAddress | 住址 | string | Public |  |  |
| Remark | 备注 | string | Public |  |  |
| Type | 删除标识 | int | Public | 当标识为 | 0 时表示该教 |
|  |  |  |  | 师已被删除 | |

3.4 输出项目

进入主界面， 普通用户主要可以进行的操作是查询学院、 系、及班级的大体情况以及其简介， 学生在普通用户基础上可以查询自己的学籍信息、 课程表、 选课情况和成绩等； 同样教师在学生可以操作的内容基础上， 可以查询学生的一些信息和自己的信息、 还可以对成绩

进行登记、修改等；管理员主要是对学生表、教师职工表、选课表、课程表及一些数据库的内容做修改、更新、删除等。

3.5 算法

3.5.1 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

stErr 为空。

3.5.2 修改用户密码的方法：

五、详细设计说明书

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

3.5.3 .

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

3.5.4 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

2 修改的方法：

strErr

public int

ModifyStu(int iStuId, string

strOld, Model.student.student

objStu,

out

string strErr)

int iStuId, 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

1. 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)

五、详细设计说明书

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 iStuId, string strStuNo, out string strErr) | | | | | |  |  |  |  |
| 删除学生和账号， int | | iStuId 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 。 |  |  |  |  |  |
| 3.5.5 教师管理功能的实现： | | | |  |  |  |  |  |  |
| 1 添加的方法： | |  |  |  |  |  |  |  |  |
| public | intAddTea(Model.teacher.teacher | | | | | objTea, | out | string | strErr) |
| Model.teacher.teacher | | objTea | | 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，模型类存储教师的相关信息，为 | | | | | | |  | 0 则返回错误提示， | |
| 将错误赋给 strErr | |  |  |  |  |  |  |  |  |

1. 修改的方法：

public int ModifyTea(int iTeaId, string strOld, Model.teacher.teacher objTea, out

string

strErr)

int iTeaId 是传入的教师 ID， string strOld 是传入的旧的教师编号，

Model.teacher.teacher objTea 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，模型类存储教师的相关信息，为 0 则返回错误

提示，将错误赋给 strErr 。

1. 查询的方法：

public int SearchTea (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

。

public int SearchTea(string objDataTable, out string strErr)

strTeaNo,

string

strTeaName,

out

DataTable

根据教师编号、教师姓名进行查询， string strTeaNo 定义接收教师编号的变量 ,string

strTeaName 定义接收教师姓名的变量 , 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，根据传入的教师编号和教师姓名查

询成功，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 DelTea(int iTeaId, string strTeaNo, out string strErr)

删除教师和账号， int iTeaId 定义接收教师 ID 的变量 , string strTeaNo 是定义接收教师

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 。

5 教师职务管理

a) 添加的方法

public int AddTeacherC\_C(Model.Teacher\_Course.Teacher\_Course objTeacherCourse,

五、详细设计说明书

Model.Teacher\_Class.Teacher\_Class objTClass, out string strErr)

Model.Teacher\_Course.Teacher\_Course

objTeacherCourse,

Model.Teacher\_Class.Teacher\_Class objTClass Model.teacher.teacher objTea

是模型

类， out string strErr 定义捕获错误的变量， int 为 1 则添加成功，模型类存储教师可

和教师课程班级以及课程班级的相关信息，为 0 则返回错误提示，将错误赋给 strErr

b) 查询的方法

public int SearchTCC(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 。 public int SearchTCC1(string TeaNo,string TeaName ,string

CourseName,string ClassName,out DataTable objDataTable, out string strErr)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 根据教师编号、教师姓名，课程名称，班级名称进行查询， | | | | string TeaNo | 定义接收教师编 |
| 号的变量 ,string | TeaName定义接收教师姓名的变量 , string | | | ClassName 定义了接收班级名 | |
| 称的变量 , string CourseName | | 定义了接收课程名称的变量 | | ,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，根据传入的教 |
| 师编号、教师姓名、课程姓名、班级姓名查询成功，将查询结果给表，返回值为 | | | | | 0，则将错 |
| 误赋给 strErr 。 |  |  |  |  |  |

1. 删除的方法

public int DelTCC(string TeaNo, string ClassName, string CourseName, out string

strErr)

删除教师职务信息， string TeaNo 是定义接收教师编号的变量 , string ClassName 定义

了接收班级名称的变量 , string CourseName 定义了接收课程名称的变量 , 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

d) 修改的方法

public int ModifyTCC(string TeaNo,int OldClassid,int OldCourseid,

Model.Teacher\_Class.Teacher\_Class objTCl, Model.Teacher\_Course.Teacher\_Course

objTCo, out string strErr)

修改教师所教班级，

的 班 级 ID 的 变

string TeaNo 定义接收教师编号的变量 ,int OldClassid

量 ,int OldCourseid 定 义 接 收 旧 的 课 程

ID

定义接收旧的 变 量

,

Model.Teacher\_Course.Teacher\_Course

objTCo,

Model.Teacher\_Class.Teacher\_Class

objTCl, Model.teacher.teacher objTea 是模型类， out string strErr 定义捕获错误的

变量，返回值 为 1 则添加成功， 模型类存储教师和教师课程班级以及课程班级的相关信息，

为 0 则返回错误提示，将错误赋给 strErr 。

public

int

ModifyTCC1(string

TeaNo,int

OldClassid,int

OldCourseid,

Model.Teacher\_Class.Teacher\_Class objTCl, Model.Teacher\_Course.Teacher\_Course objTCo, out string strErr)

修改教师所教课程， string TeaNo 定义接收教师编号的变量 ,int OldClassid 定义接收旧的

班 级 ID 的 变 量 ,int OldCourseid 定 义 接 收 旧 的 课 程 ID 的 变 量

Model.Teacher\_Course.Teacher\_Course objTCo, Model.Teacher\_Class.Teacher\_Class

,

objTCl, Model.teacher.teacher objTea

是模型类，

out string strErr

定义捕获错误的

变量，返回值 为 1 则添加成功，模型类存储教师可和教师课程班级以及课程班级的相关信息，为 0 则返回错误提示，将错误赋给 strErr 。

3.5.6 班级管理功能实现：

五、详细设计说明书

1. 添加的方法：

public int AddClass( Model.Class.Class objClass,out string strErr)

ModelClass objClass

是课程模型类。返回值定义为

0 和

1，当添加失败时，返回

0，

并用 out string strErr

输出出错原因；当添加成功时，返回

1， out string strErr

的输出为空。

2 修改的方法：

public

int ModifyClass(string

ClassNo, Model.Class.Class

objClass,

out string

strErr)

返回值定义为

0 和

1，当修改失败时，返回

0，并用

out string

strErr

输出出错原因；

当修改时，返回

1，out string strErr

的输出为空。

string iClassNo

定义接受班级编号

的变量 , Model.Class.Class objClass

为模型类

1. 查询的方法：

查询单行（按班级名和班级编号） ：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| public | int | SearchClass(string | ClassNo | ， stringClassName, out DataTable | | |
| objDataTable, out string strErr) | | |  |  |  |  |
| string iClassNo | | 定义接受班级编号的变量 , string strClassName | | | 定义接受班级名称的 | |
| 变量， out DataTable objDataTable | | | 是输出内存中的表的内容，用返回值定义为 | | | 0 和 1 |
| 判断查询成功或失败， 查询失败时用 out string | | | | strErr 输出出错原因， 成功时 out string | | |
| strErr | 输出为空。 | |  |  |  |  |
| 4 删除班级 | |  |  |  |  |  |
| public int DelClass(string ClassNo, out string strErr) | | | |  |  |  |
| 删除班级信息以及班级中的学生，班级课程表，教师班级表中相应记录 | | | | | string iClassNo | |
| 定义接受班级编号的变量 , out string | | | strErr | 输出出错原因，成功时 | out string | strErr |
| 输出为空。 | |  |  |  |  |  |
| 3.5.7 | 成绩管理功能实现： | |  |  |  |  |

1 添加的方法：

public int AddScore(Model.Score objScore,out string strErr)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model.Score objScore | 是课程模型类。返回值定义为 | 0 和 1，当添加失败时，返回 | | 0， |
| 并用 out string strErr | 输出出错原因；当添加成功时，返回 | | 1， out string strErr |  |
| 的输出为空。 |  |  |  |  |
| 2 修改的方法： |  |  |  |  |

public int ModifyScore(string CourseNo,Model.string StuNo,float Score,out string strErr)

返回值定义为 0 和 1，当添加失败时，返回 0，并用 out string strErr 输出出错原因；

当修改成功时，返回 1， out string strErr 的输出为空 ,string CourseNo 定义了接收课程

编号的变量 ,string StuNo 定义接收学生编号的变量 ,float Scoure 分数变量。

1. 查询的方法：

public int SearchScore(Model.Score objScore,out DataTable objDataTable ,out string

五、详细设计说明书

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| strErr) |  |  |  |  |  |  |  |
| 查询所有成绩 Model.Score objScore | | | 是课程模型类。返回值定义为 | | | 0 和 1，当添加失 | |
| 败时，返回 0，并用 out string strErr | | | 输出出错原因；当查询成功时，返回 | | | 1，out string | |
| strErr | 的输出为空。 |  |  |  |  |  |  |
| public | int SearchScoreStudent(sting | | StuNo, string TeaNo,string CourseName, string | | | | |
| ClassName, Model.Score objScore,out DataTable objDataTable ,out string strErr) | | | | | |  |  |
| 按学生、班级、课程、教师查询成绩 | | | sting | StuNo 定义了接收学生编号的变量 | | | , string |
| TeaNo接收教师编号 ,string CourseName, | | | 接收课程名称 string ClassName | | | 接收班级名称 , | |
| Model.Score objScore | | 是课程模型类。返回值 | | int 定义为 | 0 和 1，当添加失败时，返回 0， | | |
| 并用 out string strErr | | 输出出错原因；当查询成功时，返回 | | | 1， out string strErr | | 的输 |
| 出为空。 | |  |  |  |  |  |  |

1. 删除的方法：

public int DelScore(int id,out string strErr)

int id 定义编号， out string strErr 输出出错原因，成功时 out string strErr 输出

为空。

3.5.8 课程信息管理功能：

1. 添加的方法：

public int AddCourse(Model.Course.Course objCourse, out string strErr)

Model.Course

objCourse

是课程模型类。 返回值定义为

0 和

1，当添加失败时， 返回

0，

并用

out string strErr

输出出错原因；当添加成功时，返回

1， out string strErr

的输

出为空。

2. 修改的方法：

public int ModifyCourse(int iCourseId, Model.Course.Course objCourse, out

string strErr)

返回值定义为

0 和

1，当修改时，返回

0，并用

out string strErr

输出出错原因；当

修改时，返回

1， out string strErr

的输出为空。

1. 查询的方法：

public int SearchCourse(out DataTable objDataTable, out string strErr)

查询所有课程：

public int SearchCourse(

sting

CourseNo,string

CourseName,out

DataTable

objDataTable, out string strErr)

根据课程编号，课程名查询单行课程

收课程名 , out DataTable objDataTable

判断查询成功或失败，查询失败时用

strErr 输出为空。

sting CourseNo 接收课程编号 ,string CourseName 接

是输出内存中的表的内容，用 int 定义为 0 和 1

out string strErr 输出出错原因，成功时 out string

4 删除课程

public int DelCourse(string CourseNo, out string strErr)

string CourseNo 接收课程编号，返回值为 0 和 1，判断查询成功或失败，查询失败时为

用 out string strErr 输出出错原因，成功时为 1， out string strErr 输出为空。

0

3.5.9 统计功能实现：

五、详细设计说明书

1．计算总评分

public int sum(int , int )

Int 是一个学生的其中一门课程的成绩，另一个是下一门的成绩，依次的相加就是一个学

生的总评分，返回值为 int 类型的数据。若此时学生无成绩，则返回值为 0.

1. 计算全班平均总评分

public int average(int a[])

一个班级的学生总成绩按数组的形式存放，是整形的数组形式，将数据先求和在求平均值，

若有成绩返回值为计算结果，无成绩返回 0.

3．统计各等级学生人数

public b[] stat(int a[] ，count b ，char sex,count1 c ,char Sage,count2 d)

统计班级人数， 此时 count 作为计数的变量， 依次对数组进行遍历， 若对应的是我们限制要求的数据，则 count++ ，遍历结束返回 return （ count ）。

当我们要按性别统计人数时，我们的限制条件就是 sex ，返回 return （ count1 ），若按年龄

统计则限制条件是 Sage，返回 return （ count2 ）。

3.5.10 用户修改密码功能

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

2. 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)

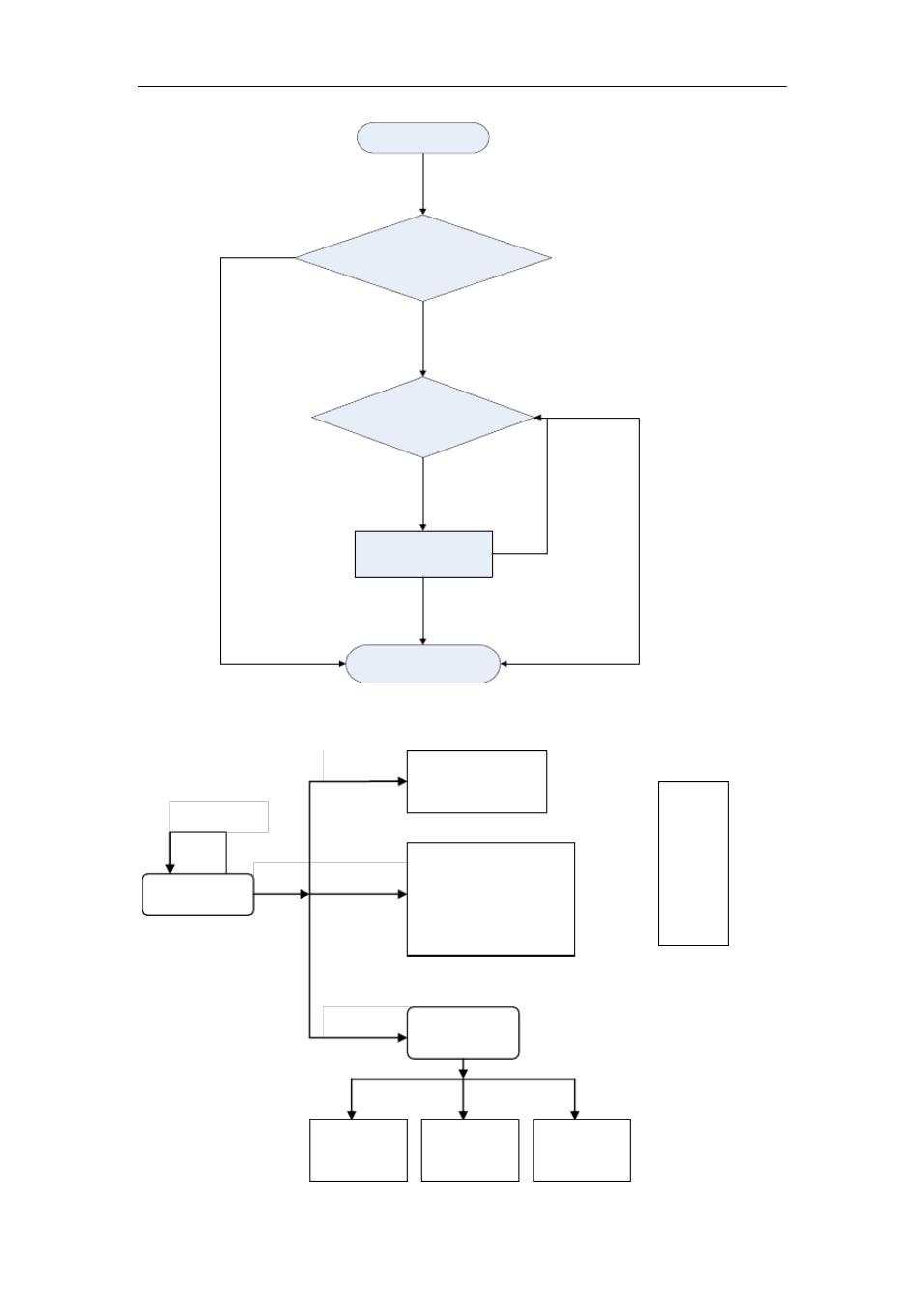
string UserName 是定义接收用户名的变量，根据用户名查找用户的相关信息， out

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

3.6 程序逻辑

登录功能的实现流程图：

五、详细设计说明书

No

began

If(StuNo=this.StuNo

||TeaNo=this.TeaNo||

UserId=this.UserId)

And Password

Yes

search 、 update 、 delete 或

exit();

yes Update

No

Update datebase

End or Putting in is

Wrong!

处理流程：

老师

失败

成功 管理员

用户登录

添加、修改、删

除学生成绩

计算总评成绩 、全

班平均总评分、 成绩

不及格、及格、中、

良、优等分数段的学

生人数

处

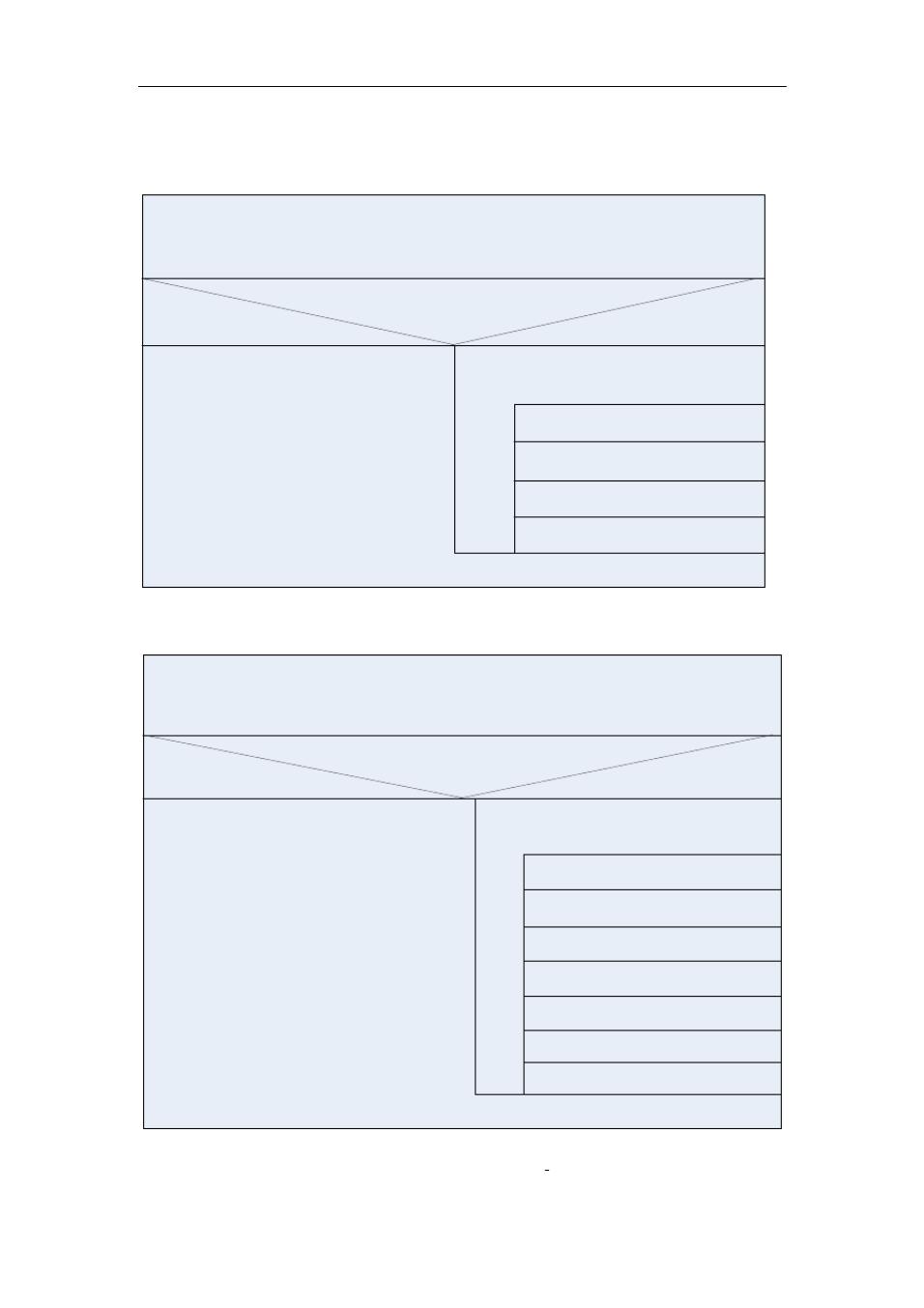
理

流

学生 查看成绩

全 班 单 科 全 班 各 科 个人成绩

成绩 汇总成绩

五、详细设计说明书

查询学生信息以及学生可用的功能：

学生输入的学号、密码

验证是否正确，符合数据库中的内容

NO Yes

继续

基本信息查询

退出，并提示学号或密码错误

课程表查询

成绩查询

选课和修改所选课程

教师信息及其可以进行的操作：

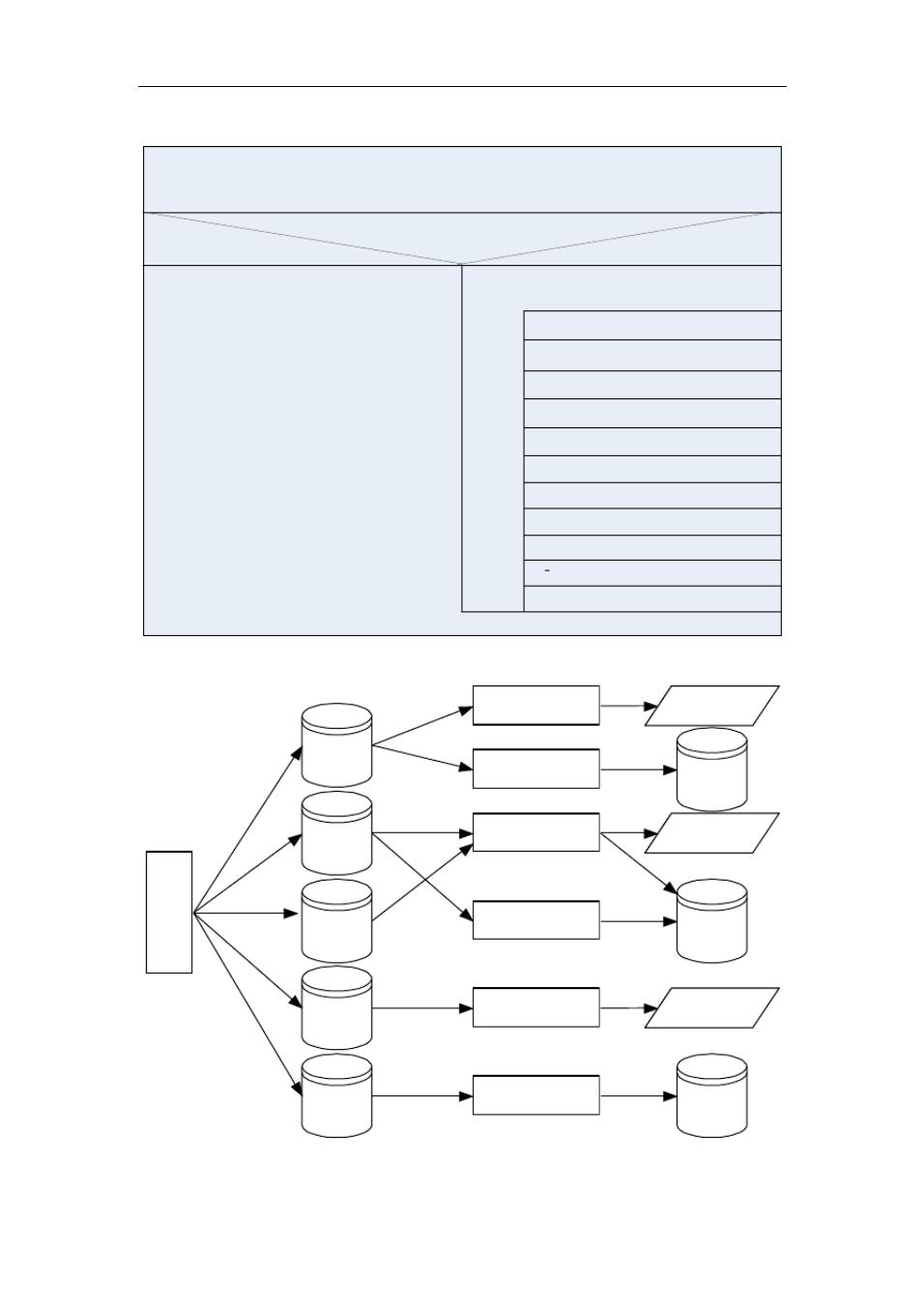
输入教师职工号、密码

验证是否正确，符合数据库中的内容

NO Yes

继续

|  |  |  |
| --- | --- | --- |
|  | 基本信息查询 |  |
| 退出并提示职工号或密码错误 | 课程表查询 |  |
|  |  |
|  | 成绩查询 |  |
|  | 成绩的登陆和修改 |  |
|  | 学生基本信息查询 |  |
|  | 打印学生成绩单 |  |
|  | 查看教师福利情况 |  |

五、详细设计说明书

管理员的登录及其可以进行的操作：

管理员登陆 ID 号及密码

验证是否正确，符合数据库中的内容

NO Yes

继续

|  |  |  |
| --- | --- | --- |
|  | 基本信息查询 |  |
| 退出并提醒 | 课程表查询 |  |
| ID 号或密码错误 |  |

成绩查询

成绩的登陆和修改

学生基本信息查询

打印学生成绩单

查看教师福利情况

对教师、学生信息更新

对教师、学生信息的添加

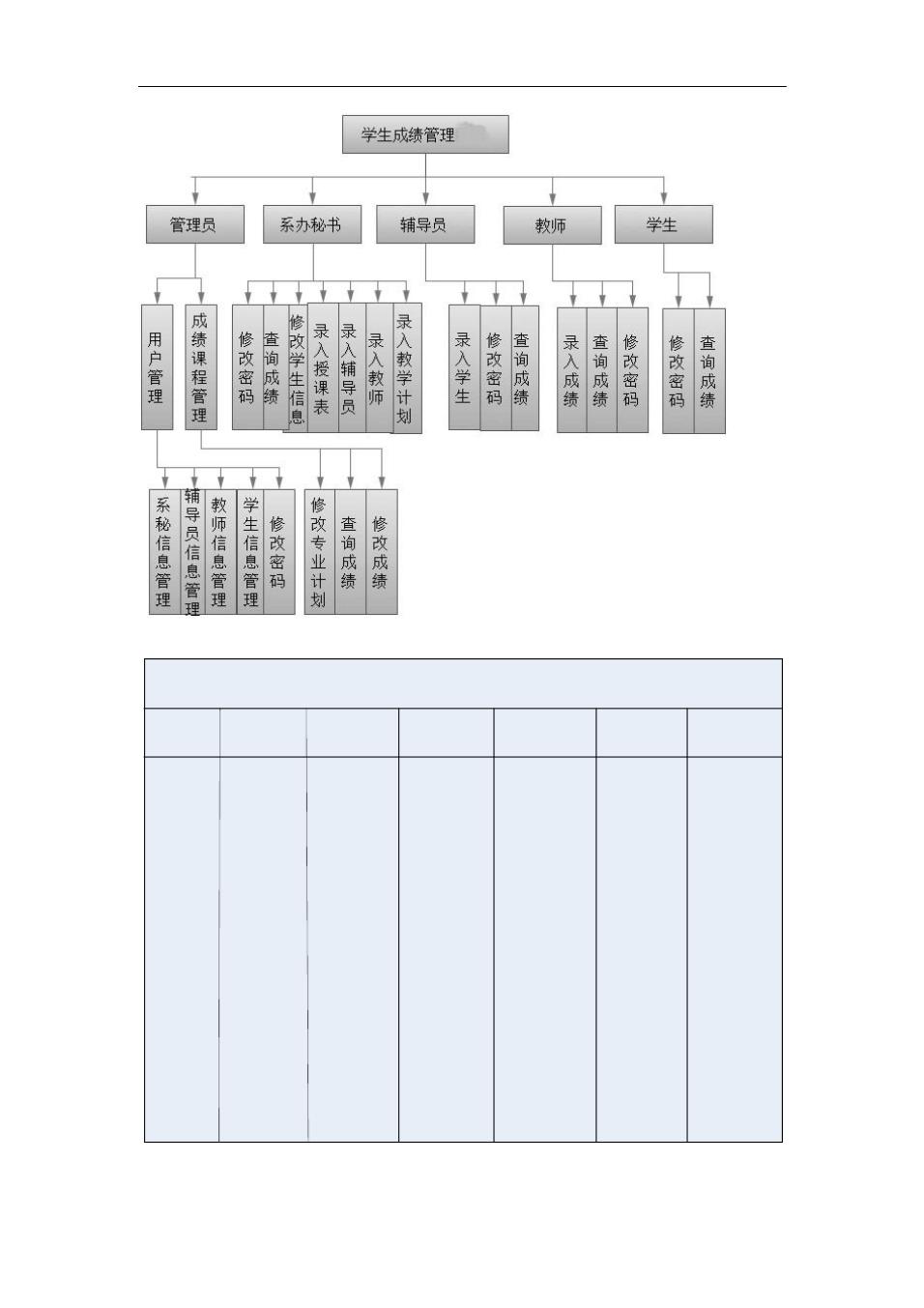
删除已经不再需要的数据

对数据库内容的维护

学生数据查询的流程图：

|  |  |  |  |
| --- | --- | --- | --- |
|  | 输入学生学号 | 学生基本信息 |  |
| 学生信息 |  |  |  |
| 表 | 录入学生信息 | 学生信息 |  |
|  |  |
|  |  | 表 |  |
| 课程表 | 输入学号 | 学生选课信息 |  |
|  |  |  |
| 数据 | 选择您要添加的课 |  |  |
| 查询 |  |  |
| 程 | 选课表 |  |
| 选课表 |  |
| 学生成绩 | 输入学生学号 | 学生成绩信息 |  |
| 表 |  |  |  |
| 用户信息 | 选择编辑或修改的 | 用户信息 |  |
| 用户信息 |  |
| 表 | 表 |  |
|  |  |

学生成绩管理：

五、详细设计说明书

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 统计功能的实现内容： | |  |  |  |  |  |  |  |  |
|  |  |  | 输入要统计的内容 |  |  |  |  |  |  |
| Case1 ： | Case2 ： | Case3 ： | Case4 ： | Case5 ： | | Case6 ： | | Case7 ： |  |
| 学生 | 教师 | 课程 | 专业 | 院系 | | 班级 | | 成绩 |  |
| 统 | 统 | 统 | 统 |  | 统 |  | 统 | 统 |  |
| 计 | 计 | 计 | 计 |  | 计 |  | 计 | 计 |  |
| 学 | 教 | 课 | 有 |  | 学 |  | 每 | 学 |  |
| 生 | 师 | 程 | 多 |  | 校 |  | 个 | 生 |  |
| 的 | 的 | 的 | 少 |  | 有 |  | 专 | 成 |  |
| 各 | 情 | 基 | 专 |  | 多 | 看 | 业 | 绩 |  |
| 种 | 况 | 本 | 业 |  | 少 | 有 | ， |  |
| 方 | 是 |  |
| 情 | ， | 情 | ， | 学 | 多 | 可 |  |
| 便 | 否 |  |
| 况 | 如 | 况 | 以 | 院 | 少 | 以 |  |
| 学 | 需 |  |
|  | ： | ， | 及 | ， | 个 | 方 |  |
|  | 所 | 如 | 专 | 校 | 并 | 要 | 班 | 便 |  |
|  | 属 | ： | 业 | 的 | 显 | 增 | 级 | 的 |  |
|  | 院 | 学 | 所 | 扩 | 示 | 加 | ， | 看 |  |
|  | 展 | 班 |  |
|  | 系 | 分 | 要 | 每 | 在 | 出 |  |
|  |  | 级 |  |
|  |  | ， | 学 |  | 个 | 招 | 教 |  |
|  |  |  |  |  |
|  |  | 学 | 习 |  | 院 |  | 生 | 学 |  |
|  |  | 时 | 的 |  | 的 |  | 时 | 的 |  |
|  |  | ， | 课 |  | 专 |  | 可 | 质 |  |
|  |  | 先 | 程 |  | 业 |  | 以 | 量 |  |
|  |  | 行 |  |  | ， |  | 查 |  |  |
|  |  | 课 |  |  |  |  |  |  |  |

五、详细设计说明书

3.7 接口

外部设计：

界面登录：

内部接口：

学生登录： void Stu\_Login()

教师登陆： void Tea\_Login （）

管理员登录： void User\_Login （）

查询： char search(char a[])

修改： string update （ String a[] ）

删除： int delete （int a ）

计算总评分： int sum(int , int )

计算全班平均总评分 int average(int a[])

统计各等级学生人数 b[] stat(int a[])

统计信息： string total （ string a[]

）

修改用户密码：

int ModifyUser(string UserName , Model.Users.Users objUsers, out

string strErr)

3.8 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，

五、详细设计说明书

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。

3.9 限制条件

|  |  |
| --- | --- |
| 由于我们组的开发知识有限，软硬件可能出现故障，且开发工作计划在 | 2011 年 11 月 |

1. 日完成。具体如下：

a． 所建议系统的运行寿命的最小值；

b． 进行系统方案选择比较的时间；

c． 硬件、软件、运行环境和开发环境方面的条件和限制；

d． 可利用的信息和资源；

e． 系统投入使用的最晚时间。

3.10 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

11．

13．

14．

15．

16．

17．

18．

19．

20．

21．

22．

23．

24．

25．

26．

27．