**基于C#.NET学生学籍管理系统**

**的设计与实现**

**设**

**计**

**指**

**导**

**2015年11月**

# 第1章 系统分析

## 1.1 需求分析

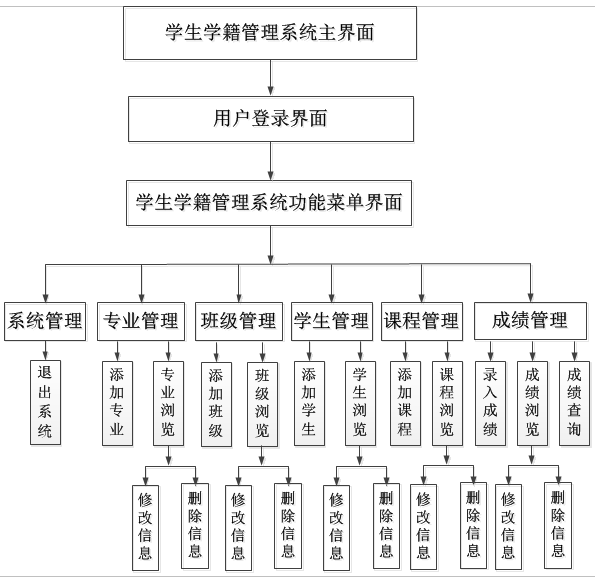
随着社会的不断发展，社会各行各业对人才的需求不断增加，培养人才的高等学校为满足社会需求，学校学生数量急剧增加，在此开发学生学籍管理系统有助于提高学生信息管理的工作效率。

学生学籍管理系统采用前台应用程序结合后台SQL Server2008数据库的C/S架构的开发模式，运用当前较流行的C#语言为开发语言，程序的代码及结构都得到了优化，提高了程序的运行效率。

学生学籍管理系统在Visual Studio.NET2010环境中结合SQL Server2008数据库进行开发，该环境提供了大量可供选择的数据控件、开发人员可以很方便地建立与数据库的连接，并在此基础上，利用各种常用的组件对数库进行操作。

## 1.2系统功能模块分析

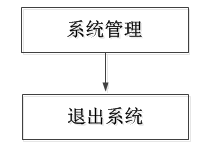
学生学籍系统要求实现系统管理、专业管理、班级管理、学生管理、课程管理及成绩管理几大功能。系统总体结构图如图1.1所示：



**图1.1 系统功能模块结构图**

### 1.2.1 系统管理

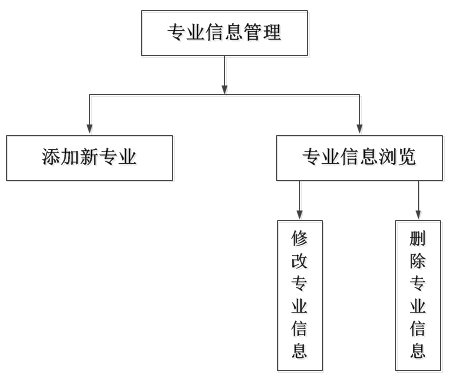
系统管理模块实现用户系统退出功能，系统管理在学生学籍管理系统窗体（窗体名：StuScoSystemMain）中以菜单形式设计，模块结构图如图2.2所示



**图2.2系统管理功能模块图**

### 1.2.2 专业管理

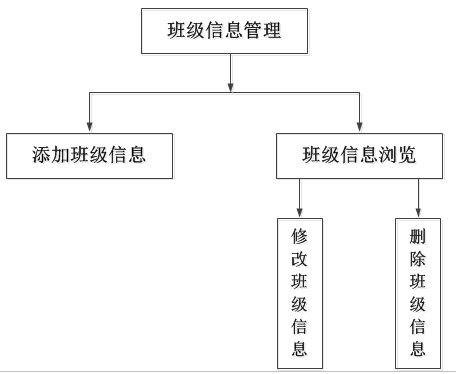
专业管理模块实现添加专业信息和专业信息浏览功能，其中专业信息浏览还需实现修改专业信息和删除专业信息子功能，专业管理在学生学籍管理系统窗体（窗体名：StuScoSystemMain）中以菜单形式设计，模块结构图如图2.3所示



**图2.3 专业管理功能模块图**

### 1.2.3 班级管理

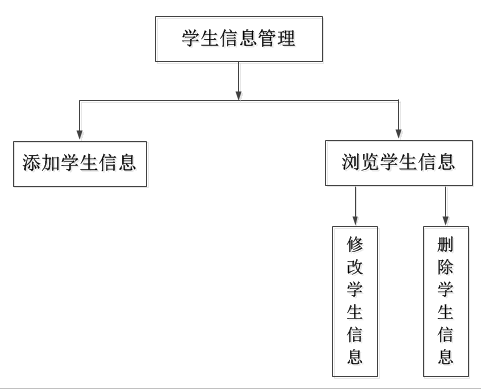
班级管理模块实现添加班级信息和班级信息浏览功能，其中班级信息浏览功能还需实现修改班级信息和删除班级信息子功能，班级管理在学生学籍管理系统窗体（窗体名：StuScoSystemMain）中以菜单形式设计，模块结构图如图2.4所示



**图2.4班级管理功能模块图**

### 1.2.4 学生管理

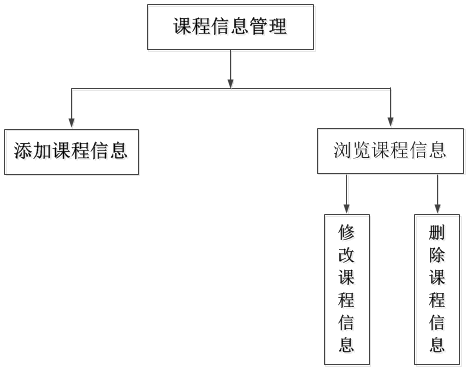
学生管理模块实现添加学生信息和学生信息浏览功能，其中学生信息浏览功能还需要实现修改学生信息和删除学生信息子功能，学生管理在学生学籍管理系统窗体（窗体名：StuScoSystemMain）中以菜单形式设计，模块结构图如图2.4所示



**图2.4 学生管理功能模块图**

### 1.2.5课程管理

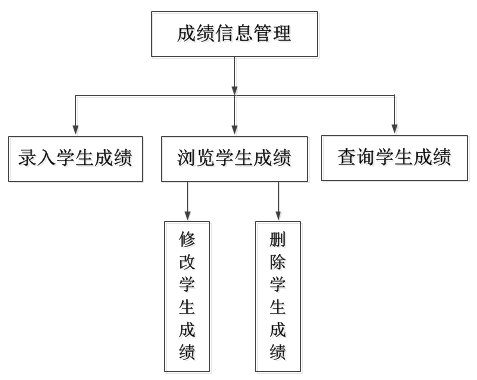
课程管理模块实现添加课程信息和浏览课程信息功能，其中浏览课程信息功能还需实现修改课程信息和删除课程信息子功能，课程管理在学生学籍管理系统窗体（窗体名：StuScoSystemMain）中以菜单形式设计，模块结构图如图2.5所示



**图2.5 课程管理功能模块图**

### 1.2.6 成绩管理

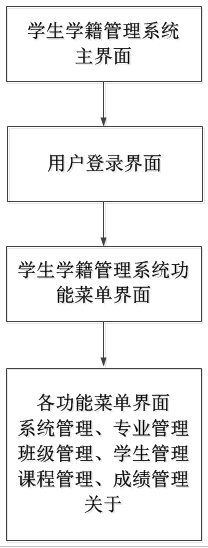
成绩管理模块实现录入成绩信息、浏览学生成绩和学生成绩查询功能，其中浏览学生信息功能还需实现修改成绩信息和删除成绩信息子功能，成绩管理在学生学籍管理系统窗体（窗体名：StuScoSystemMain）中以菜单形式设计，模块结构图如图2.6所示



**图2.6 成绩管理功能模块图**

## 1.3 系统业务流程设计

首先设计系统运行的主窗体，设计名为StuMain。当单击[进入系统]按钮后，系统关闭主窗体SruMain，显示学生学籍管理系统的[登录]界面，设计的窗体名是Loginfrm，用户输入用户名、密码，通过验证正确后，系统弹出登录成功消息框，单击[确定]，系统关闭登录窗体Loginfrm，显示学生学籍管理系统的主界面，设计的窗体名是StuScoSystemMain。在StuScoSystemMain窗体上分别设计[系统管理]、[专业管理]、[班级管理]、[学生管理]、[课程管理]、[成绩管理]和[关于]功能菜单。实现业务流程图如图1.7所示。



**图1.7业务功能流程图**

# 第2章数据库设计

学生学籍管理系统中需要采集大量信息，包括学生信息、班级信息、课程信息等，如果不合理有效组织数据表的结构，以及合理设置每张表所包含的字段，则在后期进行数据的整理及汇总时，会增加开发人员的编程难度，造成效率降低。根据对系统分析，数据信息可归纳以下内容：

一个专业可包含有多个班级；

一个班级对应一个入学年份及学制；

一个班级可以包含多名学生；

每一位学生拥有唯一的学号；

一个班级对应连续的两个学期；

一个班级在一个学期可以开设多门课程；

每一门课程需要记录每一名学生的成绩；

根据上述数据间的对应关系及依赖关系，可以设计数据库中应包含的数据表如下：

专业信息：描述所开设的专业名称；

班级名称：描述班级名称、所属专业、入校时间、学制及辅导员信息；

学生信息：描述学生学号、姓名、性别、专业、班级等基础信息；

课程信息：描述课程名称、所开班级信息；

成绩信息：描述学生学号、课程、学期、成绩信息；

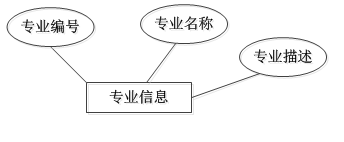
用户信息：描述用户名称、密码信息

## 2.1数据库概念结构设计

根据数据库需求分析，基本可确定并概括出程序中所包含的实体及实体间的关系，为后续的数据库逻辑结构设计提供指导。下面通过E-R实体关系图对实体进行描述。

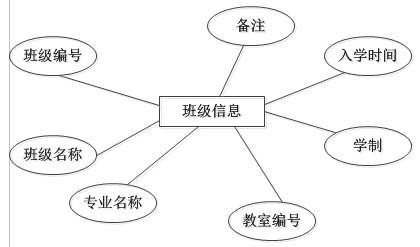
### 2.1.1 专业信息实体

专业信息实体用E-R图描述为：专业编号、专业名称、专业描述。E-R图如图2.1所示。



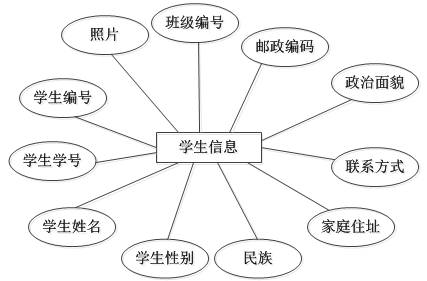
### 2.1.2 班级信息实体

班级信息实体用E-R图描述为：班级名称、入学年份、专业信息、学制、辅导员、教室编号。E-R图如图2.2所示。



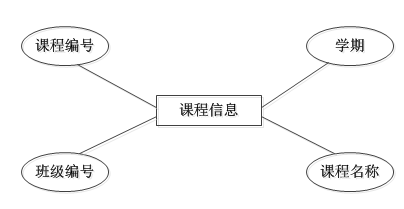
### 2.1.3 学生信息实体

学生信息实体用E-R图描述为：学号、姓名、性别、专业、入学年份、民族、家庭住址、政治面貌。E-R图如图2.3所示。



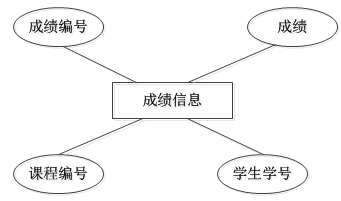
### 2.1.4 课程信息实体

课程信息实体E-R图描述为：课程名称、班编号、开设学期。E-R图如图2.4所示。



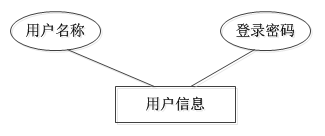
## 2.1.5 成绩信息实体

成绩信息实体E-R图描述为：课程名称、学生学号、开设学期、课程成绩。E-R图如图2.5所示。



### 2.1.6 用户信息实体

用户信息实体E-R图描述为：用户名称、登录密码。E-R图如图2.6所示。



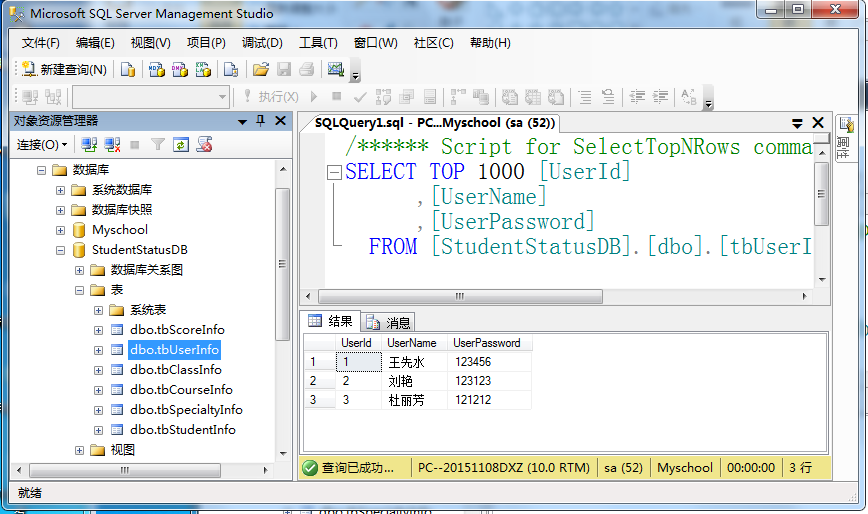
## 2.2数据库逻辑结构设计

数据库逻辑结构设计主要是为最终确定的数据库所需要的各种信息表及每张信息表所包含的字段服务的。学生学籍管理系统设计的数据库名为StudentStatusDB，在该数据库下分别设计用户信息表tbUserInfo、学生成绩信息表tbScoreInfo、班级信息表tbClassInfo、课程信息表tbCourseInfo、学生信息表tbStudentInfo、学生专业信息表tbSpecialtyInfo。学生学籍管理系统数据库结构设计如表2.1所示：

**表2.1 学生学籍系统数据库（StudentStatusDB）结构**

|  |  |
| --- | --- |
| **数据表名称** | **数据表信息描述** |
| tbUserInfo | 用户登录信息表 |
| tbClassInfo | 班级信息表 |
| tbScoreInfo | 学生成绩信息表 |
| tbSpecialtyInfo | 学生专业信息表 |
| tbStudentInfo | 学生信息表 |
| tbCourseInfo | 课程信息表 |

在数据库SQL Server2008中设计的数据及表如图2.7所示：



**图2.7 学生学籍数据库结构设计**

### 2.2.1用户登录信息表设计

用户登录信息表名称是tbUserInfo，在该表中设计的字段是：用户编号、用户名称、用户密码。设计结果如表2.2所示。

**表2.2 用户登录信息表tbUserInfo**

|  |  |  |
| --- | --- | --- |
| **字段名** | **类型** | **描述** |
| UserId | int | 用户编号 |
| UserName | nchar(50) | 用户名称（主键） |
| UserPassword | nchar(50) | 用户密码 |

### 2.2.2 学生成绩信息表设计

学生成绩信息表名称是tbScoreInfo，在该表中设计的字段有：成绩编号（主键）、学期、学生学号、课程名称、课程成绩。设计结果如表2.3所示。

**表2.3 学生成绩信息表tbScoreInfo**

|  |  |  |
| --- | --- | --- |
| **字段名** | **类型** | **描述** |
| filedaScoreId | int | 成绩编号（主键） |
| filedSemester | nvarchar(50) | 学期 |
| filedStudentNumber | nvarchar(50) | 学生学号 |
| filedCourseName | nvarchar(50) | 课程名称 |
| filedScore | int | 课程成绩 |

### 2.2.3 班级信息表设计

### 2.2.4 学生信息表设计

### 2.2.5 学生专业信息表

### 2.2.6 课程信息表设计

## 2.3设置数据表间的依赖关系

在常规下，数据库中设计的数据表都不是独立存在的，而这些表间存在一定的依赖关系，称为关系。如学生信息表中的[班级]字段来源于班级信息表中的现有班级。如果数据库中的信息不能满足正常的依赖关系，则会破坏数据库的完整性、一致性。

根据学生学籍管理系统的E-R图分析哪些表间的字段存在依赖关系，分析结果如下：

班级信息表中的专业信息来源于专业信息表；

学生信息表中的班级信息来源于班级信息表；

课程信息表中的班级来源于班级信息表；

学生成绩表中的学号来源于学生信息表；

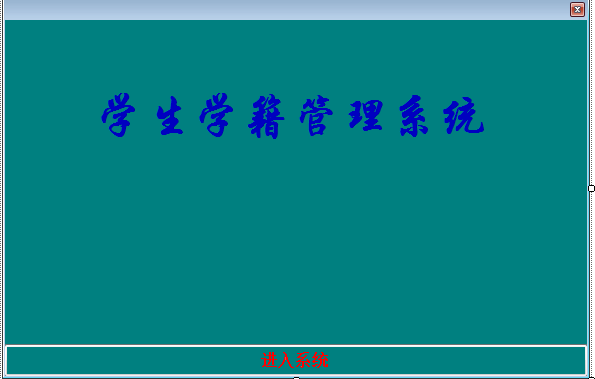
### 2.3.1 在SQL Server 2008中设置本系统表间的关系

# 第3章系统设计

## 3.1系统主界面设计

当系统运行后显示的主界面，在主界面中设计标签信息“学生学籍管理系统”，设计按钮信息“进入系统”。当显示主界面后，用户只要单击“进入系统”按钮，则关闭当前主界面，跳转到用户登录界面。

软件运行效果图如图3.1所示。



**图3.1 程序运行主界面图**

### 3.1.1设计实现过程

启动开发环境，在解决方案StuScoreSystem的StuScoMain项目中添加Windows窗体并命名为StuMain，设置窗体的name值为为软件运行的界面。在界面上设计两个控件分别是显示“学生学籍管理系统”的标签控件，加载“进入系统”的按钮控件。

在项目StuScoMain中分别添加“登录”窗体并命名为Loginfrm；“学生学籍管理系统”工作界面窗体并命名为StuScoSystemMain

在主界面中实现的代码设计如下：

namespace StuScoMain

{

public partial class StuScoMain : Form

{

public StuScoMain()

{

InitializeComponent();

}

private void btnLogin\_Click(object sender, EventArgs e)

{

Loginfrm loca = new Loginfrm();//当前窗体的单击事件中事例化要显示的窗体对象

loca.Show();//显示当前窗体的下个窗体

this.Hide();//隐藏当前窗体

}

}

}

## 3.2系统登录界面设计