

分类号：TP315

U D C：D10621-408-(2018) 2978-0

密 级：公 开

编 号：2014121120

# 成 都 信 息 工 程 大 学

## 学 位 论 文

基于 ASP.NET 的企业员工绩效考核系统的设计与实现

论文作者姓名：	曾辉
申请学位专业：	网络工程
申请学位类别：	工学学士
指导教师姓名(职称)：	赵军（讲师）
论文提交日期：	2018 年 06 月 12 日

# 基于 ASP.NET 的企业员工绩效考核系统的设计与实现

## 摘 要

随着现代信息技术的不断发展，企业对人力资源信息化管理的需求越来越强烈，在经济快速增长的背景下，信息系统在企业发展过程中也起着日益重要的作用。因此结合先进的绩效管理理念，采用高效的开发技术，遵循软件工程的思想，完成绩效考核系统的构建，以实现员工绩效的信息化管理，有效挖掘员工自身的生产效能，从而增强企业的核心竞争力，就成为企业愈发关注的地方。

本系统以实际运用为开发背景，采用当前网络开发较为成熟的主流技术，将 ASP.NET 和 SQLServer 数据库技术相结合，设计并实现了一个基于 ASP.NET 平台 B/S 架构的企业员工绩效考核系统；实现了企业角色基础数据管理，角色分配管理，考核项目、标准管理等功能。完善了企业对于员工绩效考核的体系，提高了企业对于员工信息管理，员工的绩效考核的真实性、可靠性和效率。

本系统所需功能模块均已开发完成，并经过大量测试之后，系统稳定，数据准确可靠，能够为企业提供更经济、更高效、更可靠的绩效考核解决方案，已帮助企业挖掘员工生产效能，从而增强企业竞争力。

**关键词：**企业；绩效；考核；ASP.NET；SQLServer；B/S 架构

# **Design and Implementation of Enterprise Performance Evaluation based on ASP.NET**

## **Abstract**

With the continuous development of modern information technology, the demand for information management of human resources is becoming more and more intense. Under the background of rapid economic growth, the information system plays an increasingly important role in the process of enterprise development. Therefore, combining the advanced performance management concept, adopting the development technology of the University, following the idea of software engineering, completing the construction of the performance appraisal system, in order to realize the information management of employee performance, effectively excavate the productivity of the employee, and enhance the core competitive power of the enterprise, it becomes the place where the enterprise becomes more and more concerned.

This system takes the practical application as the development background, adopts the mature mainstream technology of current network development, combines ASP.NET and SQLServer database technology, designs and implements an enterprise employee performance appraisal system based on the B/S architecture of ASP.NET platform, and realizes the basic data management of the enterprise role, the role distribution management and the examination. The functions of nuclear project, standard management and so on. The company perfected the performance appraisal system for employees, improved the information management of employees, the authenticity, reliability and efficiency of employees' performance appraisal.

The functional modules needed in this system have been developed, and after a large number of tests, the system is stable, the data is accurate and reliable. It can provide more economical, more efficient and more reliable performance evaluation solutions for the enterprises. It has helped the enterprises to excavate the efficiency of the employees and thus enhance the competitiveness of the enterprises.

**Key words:** Enterprise; Performance; Evaluation; ASP.NET; SQLServer; B/S Architecture

# 目 录

论文总页数：40 页

1 引言.....	1
1.1 课题背景.....	1
1.2 国内外研究现状.....	1
1.3 本课题研究的意义.....	1
1.4 本课题的研究方法.....	2
2 系统实现相关技术.....	2
2.1 ASP. NET.....	2
2.2 SQLSERVER 数据库.....	2
2.3 B/S 结构.....	2
2.3 C#语言.....	3
3 系统需求分析.....	3
3.1 系统可行性分析.....	3
3.1.1 技术可行性.....	3
3.1.2 经济可行性.....	3
3.1.3 社会可行性.....	3
3.2 系统功能需求分析.....	4
3.2.1 系统管理模块.....	4
3.2.2 绩效考核管理模块.....	5
3.2.3 人员信息管理模块.....	6
3.2.4 角色分析.....	6
4 系统总体设计.....	6
4.1 系统流程设计.....	6
4.2 数据库设计.....	8
4.2.1 实体模型.....	8
4.2.2 数据实体关系.....	10
4.2.3 数据库表结构设计.....	11
4.3 系统页面设计.....	15
4.3.1 系统登录界面.....	15
4.3.2 人员信息管理模块页面.....	15

4.3.3 绩效考核管理模块.....	16
4.2.4 系统管理模块.....	18
5 系统实现.....	20
5.1 数据访问类库架构.....	20
5.2 系统登录.....	21
5.3 人员信息管理.....	22
5.3.1 我的信息管理.....	22
5.3.2 人员信息管理.....	23
5.4 绩效考核管理.....	24
5.4.1 我的考核项目.....	24
5.4.2 我的绩效得分.....	24
5.4.3 考核项目管理.....	25
5.4.4 人员绩效管理.....	26
5.5 系统管理.....	27
5.5.1 部门管理.....	27
5.5.2 职位管理.....	27
5.5.3 权限管理.....	28
5.5.4 职位分配.....	29
5.5.5 密码重置.....	30
6 系统测试.....	30
6.1 测试计划.....	30
6.2 测试用例.....	31
6.2.1 系统登陆.....	31
6.2.2 我的信息管理.....	31
6.2.3 人员信息管理.....	31
6.2.4 我的绩效考核项目.....	32
6.2.5 我的绩效得分.....	32
6.2.6 考核项目管理.....	33
6.2.7 人员绩效管理.....	33
6.2.8 部门管理.....	34
6.2.9 职位管理.....	34
6.2.10 权限管理.....	35
6.2.11 职位分配.....	35

6.2.12 密码重置.....	36
6.3 测试结果.....	36
结 语.....	37
参考文献.....	38
致 谢.....	39
声 明.....	40

# 1 引言

## 1.1 课题背景

随着现代信息技术的不断发展，企业对人力资源信息化管理的需求越来越强烈，在经济快速增长的背景下，信息系统在企业的发展过程中也起着日益重要的作用。

中国人民大学人力资源开发与管理研究中心副主任刘昕，在“2006 中国杰出人力资源管理者年会”上称，根据多年咨询实践，他发现绩效考核是中国国有企业进行有效管理员工的主要途径之一，当前多数企业在绩效考核实施过程中都有不同的误区并存在一些问题，如：绩效考核与发展战略相脱节，绩效考核时间僵化，考核指标脱离岗位职责，考核未形成有效的反馈机制等。他提出的这些问题，严重制约了企业的发展，导致企业的绩效改革没有形成改革的基础和依据。<sup>[1]</sup>

因此结合先进的绩效管理理念，采用高校的开发技术，遵循软件工程的思路，完成绩效考核系统的构建，以实现员工绩效的信息化管理，有效挖掘员工自身的生产效能，从而增强企业的核心竞争力，就成为企业愈发关注的地方。

## 1.2 国内外研究现状

我国绩效考核的发展起步较晚，十一届三中全会后，国家的工作重心转移到了经济建设方面，绩效考核在企业管理中发挥重要作用。但是，考核的标准和收入的分配没有建立科学的匹配关系，在实践和理论上，企业管理工作都相对落后于发达国家。毕竟中国大多数企业缺乏丰富的经验和系统的理论知识，对绩效考核的研究仍然处于起步阶段，而且不少的研究成果也只是局限于国外经验的介绍和国内企业绩效考核的初步研究，其任重而道远。

国外企业绩效考核有着悠久的历史，美国军方于 1813 年开始采用绩效考核，到 1842 年美国联邦政府开始对政府公务员进行绩效考核。关于绩效内涵的研究，学术界普遍认同并赞赏的主要是 Campbell 等人提出的观点：绩效是员工控制自身与组织目标相关的行为。对考核指标进行明确地研究是自 20 世纪 30 年代开始的。C. e. schneier, r. in. beatly, c. s. baird(1987)等人在论述“如何建构成功的绩效考核系统”时强调指出：最重要的是把握工作的本质，绩效评价的目标是减少评价时的主观性所产生的偏差。为了更加明确地澄清绩效考核中的不确定性和模糊性，加拿大学者从可测量的角度提出了六条选择考核指标的标准，分别是：确定性原则、可能性原则、客观性原则、非污染性原则、排他性原则和可验证性原则。<sup>[1]</sup>

## 1.3 本课题研究的意义

随着现代信息技术的不断发展，企业对人力资源信息化管理的需求越来越

强烈，在经济快速增长的背景下，信息系统在企业的发展过程中也起着日益重要的作用。

本系统研究和开发的目标就是帮助企业以更经济，更效率的方式进行员工的绩效考核，从而提高效率，节约成本，有效挖掘员工自身的生产效能，从而增强企业的核心竞争力

#### 1.4 本课题的研究方法

查阅图书馆和互联网上的企业绩效考核资料和文献，结合实际企业的考核流程，参考互联网上已有的绩效考核系统，理解系统需求并设计一个结合 SQL 数据库、基于 ASP.NET 平台 B/S 架构的系统，然后设计相关功能模块为：角色基础数据管理，角色分配管理，考核项目、标准管理。在实现整个系统后，进行大量测试，收集并分析测试数据，整理编写为论文。

## 2 系统实现相关技术

### 2.1 ASP.NET

ASP.NET 是 Microsoft .net 的一部分，作为战略产品，不仅仅是 Active Server Page (ASP) 的下一个版本，它还提供了一个统一的 Web 开发模型，其中包括开发人员生成企业级 Web 应用程序所需的各种服务。ASP.NET 的语法在很大程度上与 ASP 兼容，同时它还提供一种新的编程模型和结构，可生成伸缩性和稳定性更好的应用程序，并提供更好的安全保护。它是一个已编译的、基于 .NET 的环境，可以与任何与 .NET 兼容的语言（包括 Visual Basic .Net、C# 和 Jscript.Net）创作应用程序。

### 2.2 SQLServer 数据库

SQL Server 是一个全面的、集成的、端到端的数据解决方案，它为企业中的用户提供了一个安全、可靠和高效的平台用于企业数据管理和商业智能应用。SQL Server 数据库为 IT 专家和信息工作者带来了强大的、熟悉的工具，同时减少了在从移动设备到企业数据系统的多平台上创建、部署、管理及使用企业数据和分析应用程序的复杂度。通过全面的功能集、和现有系统的集成性、以及对日常任务的自动化管理能力，SQL Server 数据库为不同规模的企业提供了一个完整的数据解决方案。

### 2.3 B/S 结构

B/S 结构（Browser/Server，浏览器/服务器模式），是 WEB 兴起后的一种网络结构模式，WEB 浏览器是客户端最主要的应用软件。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。客户机上只要安装一个浏览器（Browser 英 ['braʊzə] 美 ['braʊzə]），如 Netscape Navigator 或 Internet Explorer，服务器安装 SQL Server、Oracle、



MYSQL 等数据库。浏览器通过 Web Server 同数据库进行数据交互。

## 2.3 C#语言

C#是微软公司发布的一种面向对象的、运行于.NET Framework 之上的高级程序设计语言。并定于在微软职业开发者论坛(PDC)上登台亮相。C#是微软公司研究员 Anders Hejlsberg 的最新成果。C#看起来与 Java 有着惊人的相似；它包括了诸如单一继承、接口、与 Java 几乎同样的语法和编译成中间代码再运行的过程。但是 C#与 Java 有着明显的不同，它借鉴了 Delphi 的一个特点，与 COM（组件对象模型）是直接集成的，而且它是微软公司 .NET windows 网络框架的主角。

C#是一种安全的、稳定的、简单的、优雅的，由 C 和 C++衍生出来的面向对象的编程语言。它在继承 C 和 C++强大功能的同时去掉了一些它们的复杂特性（例如没有宏以及不允许多重继承）。C#综合了 VB 简单的可视化操作和 C++的高运行效率，以其强大的操作能力、优雅的语法风格、创新的语言特性和便捷的面向组件编程的支持成为.NET 开发的首选语言。

## 3 系统需求分析

### 3.1 系统可行性分析

该系统的目标是帮助企业用最小的时间、经济、人力成本进行绩效考核，通过查阅相关资料文献，根据需求调研分析报告，进行技术、经济、社会等多维度可行性分析，得出社会需求强烈、系统需求均可以较低的经济代价实现，因此开发本系统是可行的。

#### 3.1.1 技术可行性

从技术角度分析需求中的各个功能模块，可以用 C#语言开发基于 ASP.NET 结合 SQLServer 数据库的 B/S 架构系统，由于 C#语言、ASP.NET 平台框架、SQLServer 数据库相关技术均已比较成熟和完善，加之查阅绩效考核相关资料文献，集合自身专业知识和开发经验，因此开发本系统是技术可行的。

#### 3.1.2 经济可行性

由于开发本系统所用到的相关技术都较为成熟和完善，因此系统开发成本是较低的；由于系统被设计为 B/S 架构，因此只需搭建一台中高性能服务器，由此花费的硬件成本也是较低的；由于系统功能设计的功能节点和模块都较为灵活，适用性强，模块较为完善，能适应企业需求，因此后期维护成本是较低的。因此开发本系统是经济可行的。

#### 3.1.3 社会可行性

由于社会信息化发展迅速，企业对于人力资源信息化管理的需求愈发强烈，并且系统功能模块灵活，适用性强，模块较为完善，能适应企业需求，因此开

发本系统是社会可行的。

### 3.2 系统功能需求分析

开发该系统由人员信息管理，绩效考核管理，系统管理组成，用于人员绩效考核管理，本系统功能如图 1 所示：

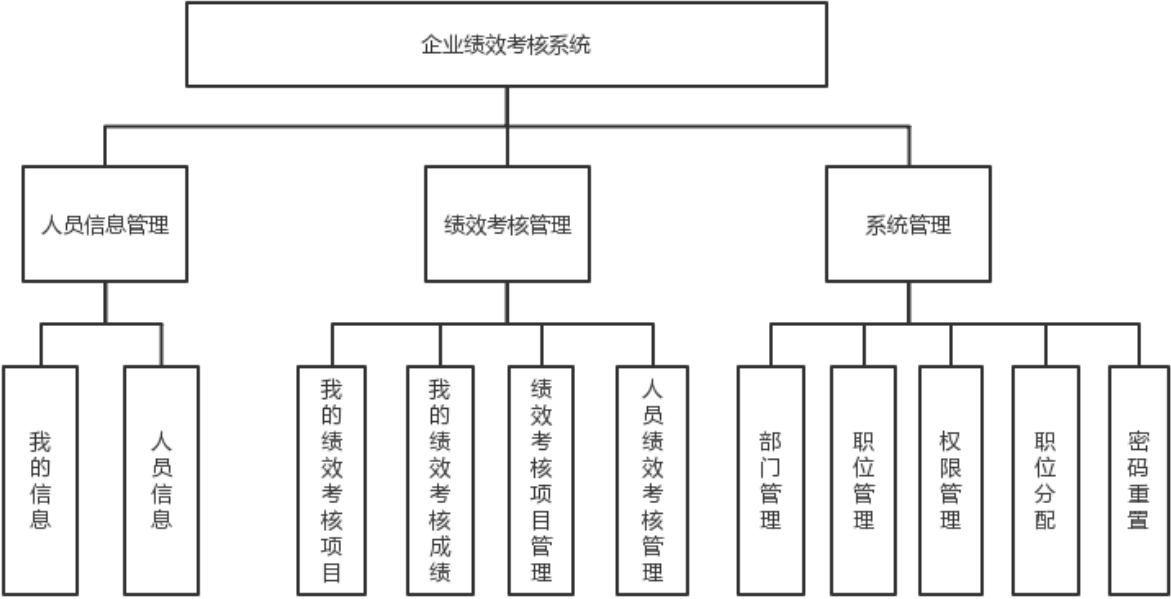


图 1 绩效考核功能图

#### 3.2.1 系统管理模块

系统管理模块主要提供系统管理员对系统的管理功能，包含：部门管理，职位管理，职位分配管理，权限管理，密码重置。其中：部门管理用于管理员根据自身企业部门组成维护部门信息；职位管理用于管理员根据企业部门职位信息维护职位；权限管理用于管理职位所对应的系统权限；职位分配是用于管理人员的职位信息，这四个模块是为了适应企业组成架构的差异和制定针对性的绩效考核项目和标准；密码重置是在员工忘记密码的情况下，为其重置密码。具体结构设计如图 2 所示：



图 2 系统管理模块

### 3.2.2 绩效考核管理模块

绩效考核项目及标准管理模块为所有人员提供查看绩效考核项目及标准和查看绩效考核得分功能，其中我的绩效考核项目是用于人员查看自己需要进行绩效考核的项目极其相信标准信息，其中包括：所有人的、我所在部门的、我所在职位的考核项目，我的绩效考核得分是用于人员根据考核年月查看自己以往绩效考核的得分记录；为部门管理员和系统管理员提供绩效考核项目及标准的定制以及为员工的绩效考核项目打分的功能，其中考核项目及标准管理是用于维护考核项目及考核标准信息以适应企业对于绩效考核的不同需要，人员绩效考核是用于为员工的绩效考核项目评分，形成绩效考核得分记录。如图 3 示：



图 3 绩效考核模块

3.2.3 人员信息管理模块

人员信息管理模块为所有人提供查看和修改自己个人信息功能；为部门管理员和系统管理员提供人员增删功能，其中我的信息管理是用于员工管理和维护自己的个人信息，人员信息管理是用于管理员维护员工信息。具体结构如图4所示：

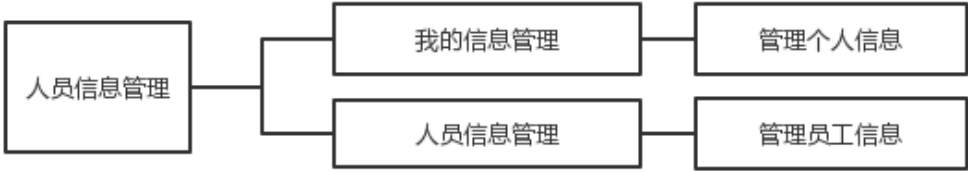


图4 人员信息管理模块

3.2.4 角色分析

由于不同企业有不同的部门架构和职位组成，因此，系统中设计了角色分配管理模块让企业管理人员根据企业实际情况进行动态调整。每种角色所对应的系统权限也可以由系统管理员动态分配。系统的角色大致分为：系统管理员、绩效考核人员和普通员工。

表1 绩效考核系统角色划分

角色	功能模块描述
系统管理员	个人信息管理，绩效考核项目、标准、评分管理，系统管理
绩效考核人员	个人信息管理，绩效评分管理
普通用户	个人信息管理，绩效评分查看

4 系统总体设计

4.1 系统流程设计

由于每一个企业的部门职位组成不尽相同，所以对应的绩效考核的项目及标准也不相同，因此，绩效考核必须灵活以适应各个企业的不同需求，基于此设计相关功能模块为：人员信息管理，绩效考核项目及标准管理，系统管理。

人员信息管理：由管理员输入新员工的用户名和初始密码，员工用其登陆系统，完善个人资料。

角色分配管理：系统把企业总体分为各个部门及其属下的各个职位，管理人员可添加相关部门和职位信息，把员工分配到相关职位，每个职位都对应了

不同的权限，在登陆系统后，会动态加载所对应的权限节点，员工可进行相关操作。

考核项目及标准：管理人员添加考核项目相关信息：考核名称，绩效分数，类型，说明等，然后将考核项目绑定到相关的部门或职位，部门考核人员可根据员工的考核项目达标情况进行绩效评分。

绩效评分：管理人员根据员工实际工作情况进行月度考核指标或年度考核指标的打分，即可得到员工考核的总分数。

绩效考核流程如图 5 所示：

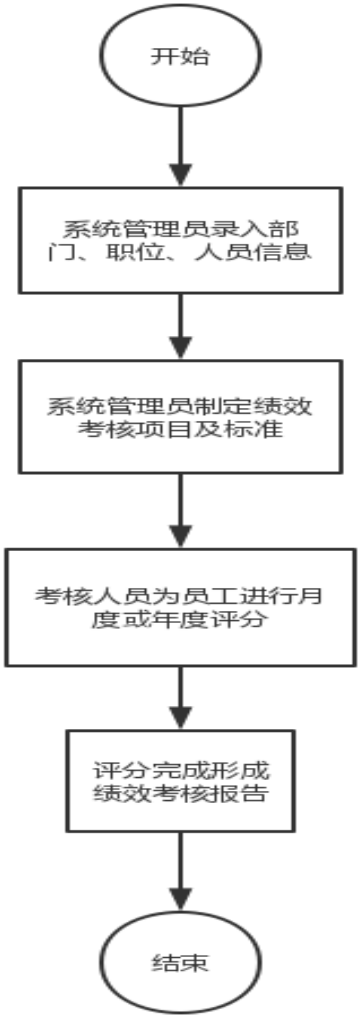


图 5 绩效考核流程图

## 4.2 数据库设计

### 4.2.1 实体模型

根据系统功能需求，设计了以下实体模型：员工实体、部门实体、职位实体、绩效考核项目实体、系统功能节点实体。

员工实体如图 6 所示：

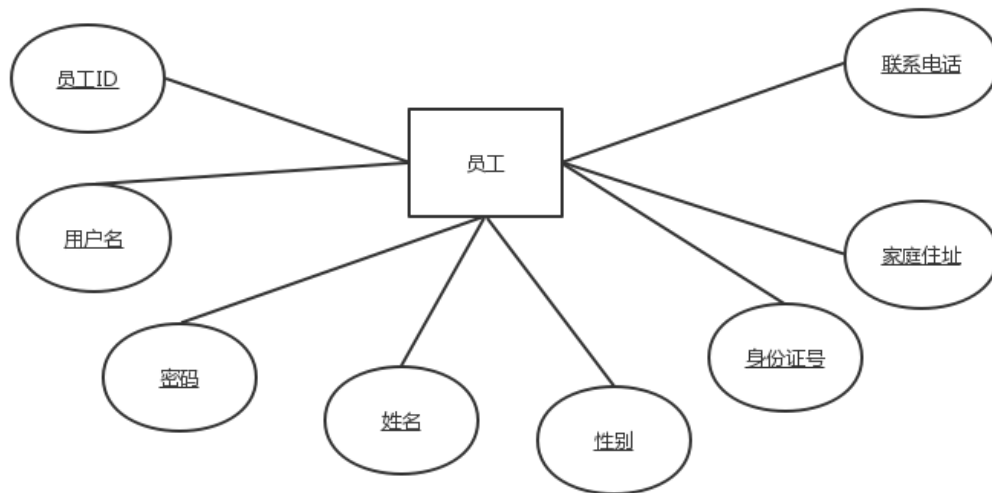


图 6 员工实体图

部门实体如图 7 所示：

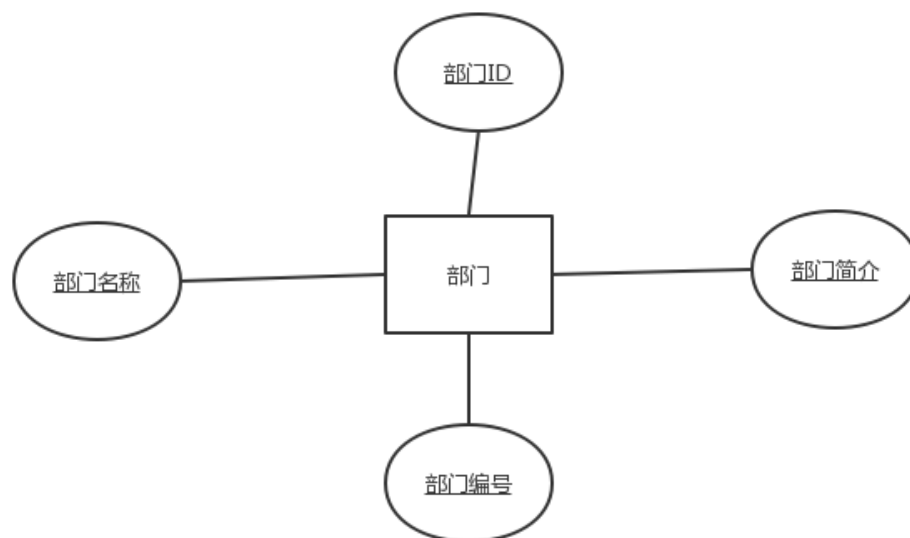


图 7 部门实体图

职位实体如图 8 所示：

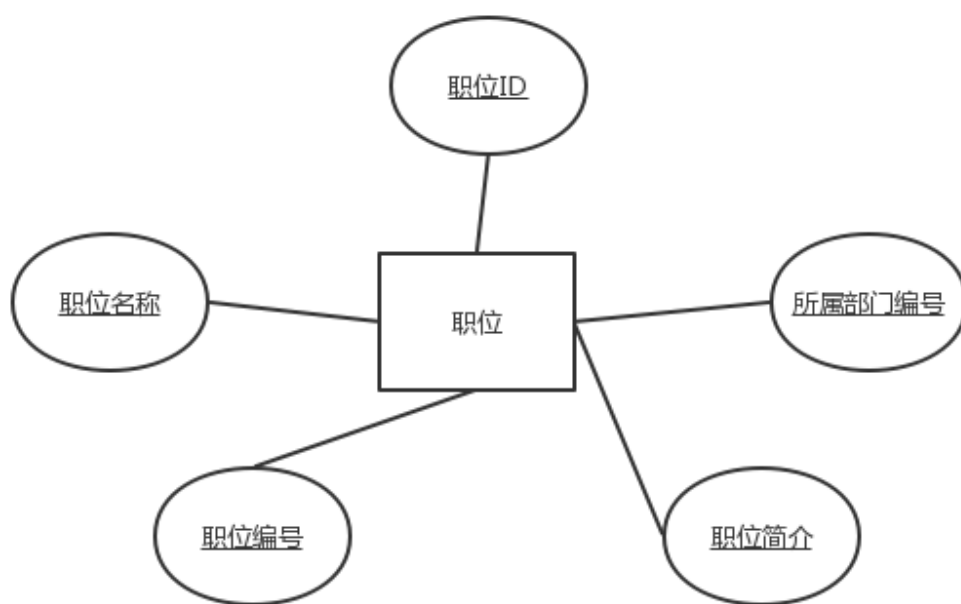


图 8 职位实体图

绩效考核项目实体如图 9 所示：

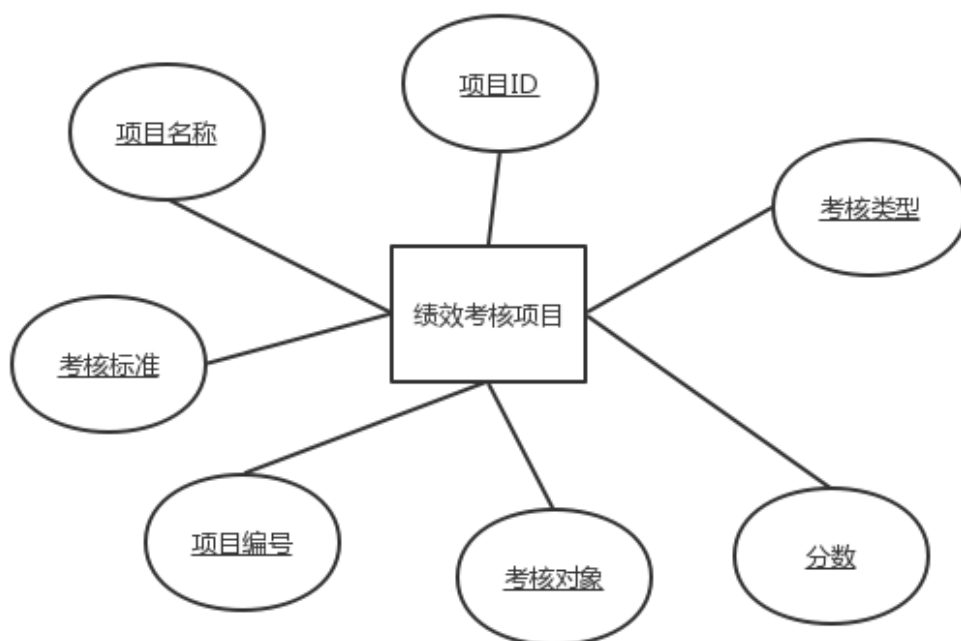


图 9 绩效考核项目实体图

系统功能节点实体如图 10 所示：

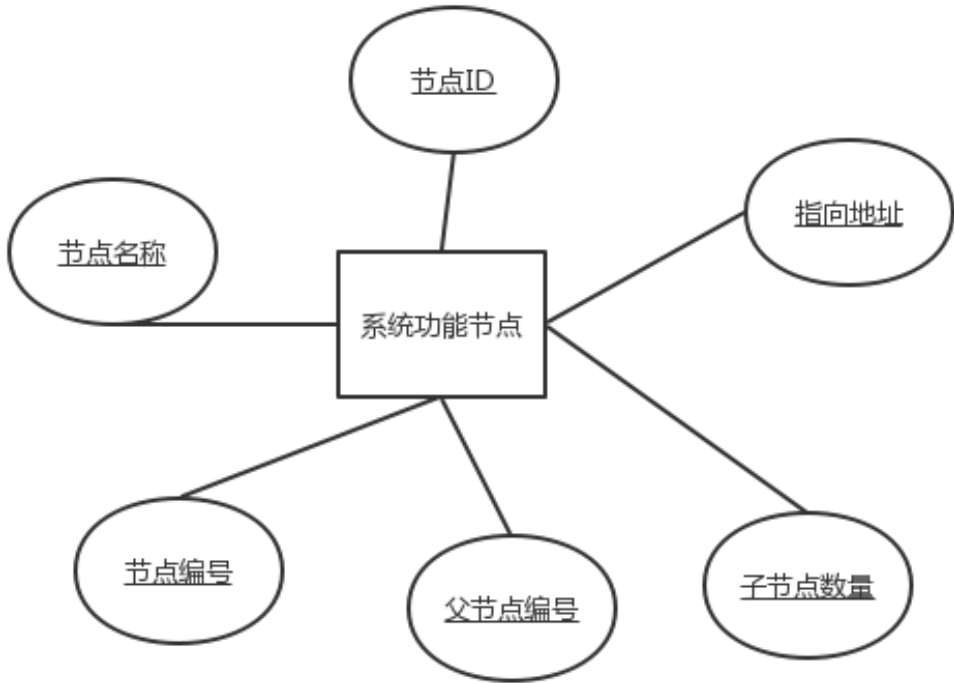


图 10 系统功能节点实体图

4.2.2 数据实体关系

在创建数据库之前，除了设计表实体模型，还要明确各个表之间的关联。数据库实体之间的关系可以用一个 ER 图来表示，如图 11 所示：



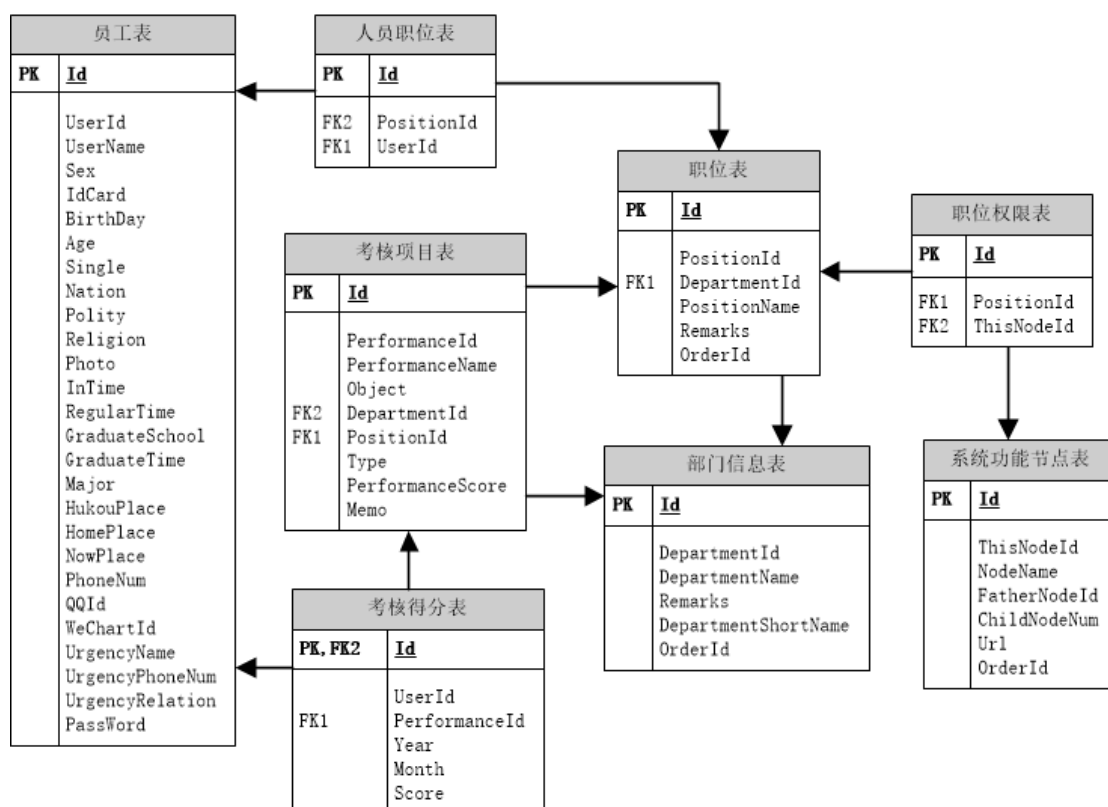


图 11 数据实体关系图

#### 4.2.3 数据库表结构设计

员工表主要包括了员工的登录账号和密码，姓名，性别，身份证号等其他基本信息字段。具体表结构如表 2 所示：

表 2 员工表

列名	数据类型	能否为空	说明
Id	int	否	用户 Id
UserId	Varchar(45)	否	用户账号
Password	Varchar(45)	否	用户密码
UserName	Varchar(45)	是	用户姓名
Sex	int	否	性别（0：女，1：男）
IdCard	Varchar(45)	是	身份证号
BirthDay	datetime	是	出生日期
Age	int	是	年龄
Single	Char(1)	是	是否已婚（1：是，0：否）
Nation	Varchar(45)	是	民族
Polity	Varchar(45)	是	政治面貌
Religion	Varchar(45)	是	宗教信仰

Photo	Varchar(45)	是	照片
InTime	datetime	是	入职时间
RegularTime	datetime	是	转正时间
GraduateSchool	Varchar(45)	是	毕业院校
GraduateTime	datetime	是	毕业时间
Major	Varchar(45)	是	所学专业
HukouPlace	Varchar(45)	是	户口所在地
HomePlace	Varchar(45)	是	家庭住址
NowPlace	Varchar(45)	是	现居住地址
PhoneNum	Varchar(45)	是	电话号码
QQId	Varchar(45)	是	QQ 号
WeChartId	Varchar(45)	是	微信号
UrgencyName	Varchar(45)	是	紧急联系人姓名
UrgencyPhoneNum	Varchar(45)	是	紧急联系人电话
UrgencyRelation	Varchar(45)	是	与紧急联系人关系

员工职位表包含了员工账号和职位编号，存储员工的职位信息。具体表结构如表 3 所示：

表 3 员工职位表表

列名	数据类型	能否为空	说明
Id	int	否	Id
PositionId	Varchar(45)	否	职位编号
UserId	Varchar(45)	否	员工账号

职位表包含了职位编号，职位名称，所属部门，职位简介等信息字段。具体表结构如表 4 所示：

表 4 职位表

列名	数据类型	能否为空	说明
OrderId	int	否	Id
DepartmentId	Varchar(45)	否	所属部门编号
PositionName	Varchar(45)	否	职位名称
Remarks	Varchar(45)	否	职位简介

PositionId	Varchar(45)	否	职位编号
------------	-------------	---	------

职位权限表包含了职位编号与其对应的系统功能节点编号，用于动态设定和加载不同职位所对应的节点权限。具体表结构如表 5 所示：

表 5 职位权限表

列名	数据类型	能否为空	说明
Id	int	否	Id
PositionId	Varchar(45)	否	职位编号
ThisNodeId	Varchar(100)	否	功能节点编号

系统功能节点表包含节点编号，节点名称，子节点数量，父节点编号，指向地址，用于存储系统功能节点信息。具体表结构如表 6 所示：

表 6 系统功能节点表

列名	数据类型	能否为空	说明
ThisNodeId	Varchar(45)	否	该节点编号
NodeName	Varchar(45)	否	节点名称
FatherNodeId	Varchar(45)	否	父节点编号
ChildNodeNum	Varchar(45)	否	子节点数量
Url	Varchar(00)	否	指向地址
OrderId	int)	否	Id

考核得分表包含员工账号，考核项目编号，考核时间年月，考核得分，用于记录用户不同时间的绩效考核得分。具体表结构如表 7 所示：

表 7 考核得分表表

列名	数据类型	能否为空	说明
Id	int	否	Id
UserId	Varchar(45)	否	员工账号
PerformanceId	Varchar(45)	否	考核项目编号
Year	int	否	考核年份
Month	int	否	考核月份
Score	DECIMAL(9,1)	是	考核得分

考核项目表包含考核项目编号，考核项目名称，考核对象，部门编号，职位编号，考核类型，考核项目总分，考核标准。具体表结构如表 8 所示：

表 8 考核项目表

列名	数据类型	能否为空	说明
PerformanceId	Varchar(45)	否	考核项目编号
PerformanceName	Varchar(45)	否	考核项目名称
Object	Char(1)	否	考核对象（所有、部门、职位）
DepartmentId	Varchar(45)	是	部门编号
PositionId	Varchar(45)	是	职位编号
Type	Char(1)	否	考核类型（Y 年度考核 M 月度考核）
PerformanceScore	DECIMAL(9, 1)	否	考核项目总分
Memo	text	是	考核标准

部门表包含部门编号，部门名称，部门简介等字段。具体表结构如表 9 所示：

表 9 部门表

列名	数据类型	能否为空	说明
OrderId	Int	否	Id
DepartmentId	Varchar(45)	否	部门编号
DepartmentName	Char(1)	否	部门名称
Remarks	Varchar(45)	是	部门简介
DepartmentShortName	Varchar(45)	是	部门简称

## 4.3 系统页面设计

### 4.3.1 系统登录界面



图 12 系统登录

### 4.3.2 人员信息管理模块页面

人员信息管理模块包括我的信息管理和人员信息管理，具体界面图如下：



图 13 我的信息管理



图 17 人员信息管理

### 4.3.3 绩效考核管理模块

绩效考核管理模块包括我的考核项目，我的绩效得分，考核项目管理，人员绩效管理功能，具体界面图如下：



图 17 我的绩效项目



图 18 我的绩效得分



图 19 考核项目管理



图 20 人员绩效管理

### 4.2.4 系统管理模块

系统管理模块包括部门管理，职位管理，权限管理，职位分配，密码重置功能，具体界面图如下：



图 21 部门管理



图 22 职位管理





图 23 权限管理



图 24 职位分配



图 25 密码重置

## 5 系统实现

### 5.1 数据访问类库架构

为了更加方便地对数据库进行查询、新增、修改、删除操作，运用三层架构地思想、构建一个数据访问类库，来执行操作语句，并返回相应结果，核心代码：

获取数据库链接字符串和执行查询语句方法代码：

```
static private string ConnString =  
System.Web.Configuration.WebConfigurationManager.ConnectionStrings["ConnString"].ToString(); //数据库链接字符串
```

```
#region 执行SQL方法
```

```
///输入SQL 返回DataSet
```

```
///SQLString 执行的SQL语句
```

```
static public DataSet GetResult(string SQLString) //输入SQL 返回DataSet
```

```
{
```

```
    DataSet ds = new DataSet();
```

```
    using (SqlConnection conn = new SqlConnection(ConnString))
```

```
    {
```

```
        SqlCommand cmd = new SqlCommand();
```

```
        cmd.Connection = conn;
```

```
        try
```

```
        {
```

```
            cmd.CommandText = SQLString;
```

```
            conn.Open();
```

```
            SqlDataAdapter ad = new SqlDataAdapter(cmd);
```

```
            conn.Close();
```

```
            ad.Fill(ds);
```

```
        }
```

```
        catch
```

```
        {
```

```
            throw;
```

```
        }
```

```
        finally
```

```
        {
```

```
            conn.Close();
```

```
        }
```

```
    }
```

执行新增、修改、删除操作方法：

```
/// <summary>
```

```
/// 无返回的SQL执行方法
```

```
/// </summary>
```

```
/// <param name="SQLString">执行的SQL语句</param>
```

```
public static void GetNoResult(string SQLString) //无返回
```

```

{
    SqlCommand cmd = new SqlCommand();
    cmd.CommandTimeout = 600;
    cmd.CommandType = CommandType.Text;
    using (SqlConnection conn = new SqlConnection(ConnString))
    {
        conn.Open();
        cmd.Connection = conn;
        cmd.CommandText = SQLString;
        try
        {
            cmd.ExecuteNonQuery();
            conn.Close();
        }
        catch
        {
            conn.Close();
            throw;
        }
    }
}

```

## 5.2 系统登录

系统登录的功能是验证使用者，并且在验证成功后跳转到系统内部页面。员工需要输入员工账号、密码、验证码，验证准确无误后，方可跳转到系统内部页面，并且将用户相关信息用 Cookie 存储到客户端，在以后的 Form 表单提交时会持续验证；若有误，会提示员工重新输入。核心代码如下：

```

public static string UserLogin(string UserId, string PassWord, Page p) //验证账号密码
{
    string Rst = "OK";
    Parameters param = new Parameters();
    param.Add("@UserId", UserId);
    param.Add("@Password", PassWord);
    string StrSQL = @"select
a.Id,a.UserId,a.UserName,b.PositionId,c.DepartmentId,pwdcompare(@Password,a.Password)Login
n
FROM dbo.UserInfo a
LEFT JOIN PeoplePosition b ON b.UserId = a.UserId
LEFT JOIN dbo.PositionInfo c ON
b.PositionId=c.PositionId
WHERE a.UserId=@UserId ORDER BY a.Id DESC";
    DataTable dt= DataAccess.GetResult(StrSQL,param).Tables[0];
    if (dt.Rows.Count > 0 && (dt.Rows[0]["Login"].ToString() == "1")) //验证用户名
与密码

```

```

        {
            DataRow dr= dt.Rows[0];
            User myinfo = new User();
            myinfo.Id = dr["Id"].ToString();
            myinfo.UserId = UserId;
            myinfo.UserName = dr["UserName"].ToString();
            myinfo.DepartmentId = dr["DepartmentId"].ToString();
            myinfo.PositionId = dr["PositionId"].ToString();

            Guid GUID = new Guid();
            string StrGUID = GUID.ToString();
            FormsAuthenticationTicket authTicket = new FormsAuthenticationTicket(1,
            UserId + StrGUID, DateTime.Now, DateTime.Now.AddMinutes(1440), false,
            myinfo.GetRoleString());

            //加密,并保存角色信息
            string encryptedTicket = FormsAuthentication.Encrypt(authTicket);
            FormsAuthentication.SetAuthCookie(UserId + StrGUID, false);//设置cookie
名字
            HttpCookie authCookie = new
            HttpCookie(FormsAuthentication.FormsCookieName, encryptedTicket);
            p.Response.Cookies.Add(authCookie);
        }
        else
        {
            Rst = "Error";
        }
        return Rst;
    }
}

```

## 5.3 人员信息管理

### 5.3.1 我的信息管理

该功能模块针对系统内所有员工进行修改和查询个人信息，在员工首次登陆到系统后看，需要员工自行修改和完善个人信息并提交保存，核心代码如下：

```

public void Save(string Id)//保存我的信息
{
    Parameters param = new Parameters();
    param.Add("@UserId", UserId.Value);
    param.Add("@UserName", UserName.Value);
    if (PhotoUpload.HasFile)//上传照片
    {
        param.Add("@Photo",DataAccess.UploadFile(PhotoUpload, "Photo",
ViewState["Photo"].ToString()));
    }
}

```

```

    }
    else
    {
        param.Add("@Photo", ViewState["Photo"].ToString());
    }
    param.Add("@Id", Id);
    string SQL = @"update UserInfo set
UserId=@UserId,UserName=@UserName,IdCard=@IdCard,BirthDay=@BirthDay,Single=@Single,
Nation=@Nation,Sex=@Sex,Polity=@Polity,

InTime=@InTime,RegularTime=@RegularTime,GraduateSchool=@GraduateSchool,Major=@Major,
GraduateTime=@GraduateTime,PhoneNum=@PhoneNum,QQId=@QQId,

WeChartId=@WeChartId,UrgencyName=@UrgencyName,UrgencyPhoneNum=@UrgencyPhoneNum,
HukouPlace=@HukouPlace,HomePlace=@HomePlace,NowPlace=@NowPlace,Photo=@Photo

where Id=@Id";
    DataAccess.GetNoResult(SQL, param);
}

```

### 5.3.2 人员信息管理

该功能模块是针对管理员对员工基本信息进行增加、删除、修改和查询操作。核心代码如下：

```

protected void Add_ServerClick(object sender, EventArgs e)//新增员工
{
    try
    {
        Parameters param = new Parameters();
        param.Add("@UserId", NewPersonId.Value);
        param.Add("@PassWord", NewPersonPassword.Value);
        if (DataAccess.GetResult("select 1 from UserInfo where
UserId=@UserId",param).Tables[0].Rows.Count==0)
        {
            DataAccess.GetNoResult("INSERT UserInfo (UserId,PassWord)
VALUES (@UserId,pwdencrypt(@PassWord))",param);
            Response.Write("<script>alert('添加成功');</script>");
        }
        else
        {
            Response.Write("<script>alert('人员编号存在，请重新填写');</script>");
        }
    }
}

```

```

        catch
        {
            Response.Write("<script>alert('添加失败');</script>");
        }
        finally
        {
            NewPersonId.Value = "";
            NewPersonPassword.Value = "";
            InitBind(1, true);
        }
    }
}

```

## 5.4 绩效考核管理

### 5.4.1 我的考核项目

我的绩效考核项目功能旨在让考核员工查看自己的考核项目以及考核标准，更好地达到考核项目要求，核心代码如下：

```

protected void InitBind(int PageNo, Boolean IfAll)//加载属于我的考核项目
{
    string SQLC = @"a.Id,a.PerformanceId,a.PerformanceName,a.PerformanceScore,
                    case a.Object when 'A' then '全体人员'
                        when 'D' then (select DepartmentName from
DepartmentInfo t where t.DepartmentId=a.DepartmentId)
                        when 'P' then '('+ d.DepartmentName +')'+
c.PositionName
                    end Object,
                    case a.Type when 'M' then '月度考核' when 'Y' then '年度考核' end
Type";
    string SQLT = @" from Performance a
                    LEFT JOIN dbo.PositionInfo c ON c.PositionId = a.PositionId
                    LEFT JOIN dbo.DepartmentInfo d ON d.DepartmentId =
c.DepartmentId ";
    string SQLW = @" where a.Object='A' or (a.Object='D' and a.DepartmentId='" +
SetAndGetUser.DepartmentId + "') or (a.Object='P' and a.PositionId ='" +
SetAndGetUser.PositionId + "')";
    String SQLO = "Order by a.Id ASC";
    PageLoad.PageLoad.DataPage(Paged1, PagedInfo1, GridView1, SQLC, SQLT, SQLW,
SQLO, PageNo, IfAll);
}

```

### 5.4.2 我的绩效得分

该模块是用于员工查看其当前或历史的月度或年度考核的得分，以及详细评分信息，核心代码如下：

```

public void InitBind(string UserId,string Time,int PageNo,Boolean IfAll)//加载考核项目和分数

```

信息

```
{
    ViewState["UserId"] = UserId;
    ViewState["Time"] = Time;
    Year.Value = Time;
    string SQLC = @"a.Id,a.PerformanceId,a.PerformanceName,a.PerformanceScore,
                    case a.Object when 'A' then '全体人员'
                                when 'D' then (select DepartmentName from
DepartmentInfo t where t.DepartmentId=a.DepartmentId)
                                when 'P' then '('+ d.DepartmentName +')'+
c.PositionName
                    end Object,
                    '年度考核' Type,
                    isnull((select t.Score from PerformanceScore t where t.UserId='" +
UserId + "' and t.Year='" + Time + "' and t.PerformanceId=a.PerformanceId),0)Score";
    string SQLT = @" from Performance a
                    LEFT JOIN dbo.PositionInfo c ON c.PositionId = a.PositionId
                    LEFT JOIN dbo.DepartmentInfo d ON d.DepartmentId =
c.DepartmentId ";
    string SQLW = @" where a.type='Y' and (a.Object='A' or (a.Object='D' and
a.DepartmentId=(SELECT y.DepartmentId FROM PeoplePosition x LEFT JOIN
dbo.PositionInfo y ON y.PositionId = x.PositionId WHERE x.UserId='" + UserId + "')) or
(a.Object='P' and a.PositionId =(SELECT x.PositionId FROM PeoplePosition x WHERE
x.UserId='" + UserId + "')))";
    String SQLO = " Order by a.Id ASC";
    PageLoad.PageLoad.DataPage(Paged1, PagedInfo1, GridView1, SQLC, SQLT, SQLW,
SQLO, PageNo, IfAll);//gridview显示详细分数信息
    TotalScore.Value = DataAccess.GetResult("select ISNULL(SUM(t.Score),0) TotalScore
from PerformanceScore t left join Performance k on t.PerformanceId=k.PerformanceId where
t.UserId='" + UserId + "' and t.Year='" + Time + "' and
k.Type='Y'").Tables[0].Rows[0]["TotalScore"].ToString();
} //计算总得分
```

### 5.4.3 考核项目管理

由于各个企业的绩效考核项目和标准并不相同，因此，设计该功能用于考核管理人员根据企业自身情况制定和维护考核项目及其标准，增强灵活性和适用性核心代码如下：

```
protected void Save_ServerClick(object sender,EventArgs e)//保存绩效考核项目信息
{
    string Id = "";
    string SQL = "";
    if (Request["Status"].ToString().ToUpper() == "Add".ToUpper())
    {
```

```

        string PerformanceId = Performance.Get();
        Parameters param = new Parameters();
        param.Add("@PerformanceId", PerformanceId);
        SQL = @"Insert into Performance (PerformanceId) values (@PerformanceId)
SELECT SCOPE_IDENTITY() AS ID";
        Id = DataAccess.GetResult(SQL, param).Tables[0].Rows[0][0].ToString();
        Performance.Save(Id);
        Response.Write("<script>alert('保存成功');</script>");
    }

```

#### 5.4.4 人员绩效管理

该模块是用于考核管理人员为员工各个绩效考核项目评分，形成详细的考核记录以供查询。

考核管理人员需要在列表页选择月度还是年度绩效考核，并选择考核的年月，若是查询历史记录，列表页会显示历史考核记录分数，点击任意员工编号即可查看其详细评分记录，若是新增，则选择考核类型和新增考核记录的年月，点击任意员工编号即可对该年月进行绩效评分，保存即可形成考核记录，并自动计算分数，需要注意的是，改变查询条件后，必须点击查询按钮。核心代码如下：

```

public void SaveScore(string UserId)//保存人员绩效分数
{
    if(string.IsNullOrEmpty(Month.Value))
    {
        return;
    }
    string[] YM = Month.Value.Split('-');
    int Year = Int32.Parse(YM[0].ToString());
    int Mon = Int32.Parse(YM[1].ToString());
    DataAccess.GetNoResult("delete PerformanceScore where UserId='" + UserId + @"'
and Year='" + Year + "' and Month='" + Mon);//先删除已有的考核分数
    string StrSQL = "";
    foreach (GridViewRow gv in GridView1.Rows)
    {
        string Score = ((TextBox)gv.FindControl("GetScore")).Text.Trim();
        StrSQL += @" insert into  PerformanceScore
(UserId,PerformanceId,Year,Month,Score) values ('" + UserId + "','" + gv.Cells[1].Text.ToString()
+ "','" + Year + "','" + Mon + "','" + Score + "') ";
    }
    DataAccess.GetNoResult(StrSQL);//全部作为新增，保存考核分数信息
}

```



## 5.5 系统管理

### 5.5.1 部门管理

该模块是用于企业依据自身情况构建部门组织，使考核项目种类多样化，考核更加准确，增强系统灵活性，适用性。核心代码如下：

```
protected void Save_ServerClick(object sender, EventArgs e)//保存部门信息
{
    string StrSQL;
    Parameters param = new Parameters();
    param.Add("@DepartmentId", DepartmentId.Value);
    param.Add("@DepartmentName", DepartmentName.Value);
    param.Add("@DepartmentShortName", DepartmentShortName.Value);
    param.Add("@OrderId", OrderId.Value.Trim());
    param.Add("@Remarks", Remarks.Value.Trim());
    if (Request["Status"].ToString() == "Add")
    {
        StrSQL = "Insert into DepartmentInfo ( DepartmentId, DepartmentName,
DepartmentShortName,OrderId,Remarks) values ( @DepartmentId, @DepartmentName,
@DepartmentShortName, @OrderId ,@Remarks)";
        DataAccess.GetNoResult(StrSQL, param);
        Response.Write("<script>alert('操作成功!');window.location='DepartmentInfo_Edit.aspx?Status=" + Request["Status"].ToString() +
"";</script>");
    }
    else
    {
        StrSQL = "Update DepartmentInfo set
DepartmentId=@DepartmentId,DepartmentName=@DepartmentName,DepartmentShortName=@
DepartmentShortName,OrderId=@OrderId,Remarks=@Remarks where Id=@ID";
        param.Add("@ID", Request["ID"].ToString());
        DataAccess.GetNoResult(StrSQL, param);
        Response.Write("<script>alert('操作成功!');</script>");
    }
}
```

### 5.5.2 职位管理

职位管理模块用于管理人员根据企业自身情况动态构建部门中职位，使考核项目种类多样化，考核更加准确，增强系统灵活性，适用性。需要选择对应的部门。核心代码如下：

```
protected void Save_ServerClick(object sender, EventArgs e)//保存职位信息
{
    string StrSQL;
```

```

Parameters param = new Parameters();
param.Add("@PositionId", PositionId.Value);
param.Add("@PositionName", PositionName.Value);
param.Add("@DepartmentId", DepartmentId.Value);
param.Add("@OrderId", OrderId.Value.Trim());
param.Add("@Remarks", Remarks.Value.Trim());
if (Request["Status"].ToString() == "Add")
{
    StrSQL = "Insert into PositionInfo ( PositionId, PositionName,
DepartmentId,OrderId,Remarks) values ( @PositionId, @PositionName, @DepartmentId,
@OrderId ,@Remarks)";
    DataAccess.GetNoResult(StrSQL, param);
    Response.Write("<script>alert('操作成功!');window.location='PositionInfo_Edit.aspx?Status=" + Request["Status"].ToString() +
"";</script>");
}
else
{
    StrSQL = "Update PositionInfo set
PositionId=@PositionId,PositionName=@PositionName,DepartmentId=@DepartmentId,OrderId
=@OrderId,Remarks=@Remarks where Id=@ID";
    param.Add("@ID", Request["ID"].ToString());
    DataAccess.GetNoResult(StrSQL, param);
    Response.Write("<script>alert('操作成功!');</script>");
}
}
}

```

### 5.5.3 权限管理

权限管理模块用于管理人员根据企业自身情况动态管理不同职位的系统功能节点权限，员工只能使用有权限的功能模块，增强系统灵活性，适用性。核心代码如下：

```

protected void Save_ServerClick(object sender,EventArgs e)
{
    DataAccess.GetNoResult("DELETE dbo.PositionPower WHERE PositionId =
"" + RoleGrid.SelectedDataKey[0].ToString() + "");//删除该职位下所有节点信息
    foreach (TreeNode roots in jd.Nodes)//遍历权限节点树
    {
        if (roots.Checked == true)//如果节点被选中，则新增保存
        {
            Parameters param = new Parameters();
            param.Add("@PositionId",
RoleGrid.SelectedDataKey[0].ToString());

```

```

        param.Add("@ThisNodeId", roots.Value);
        DataAccess.GetNoResult("INSERT dbo.PositionPower
(PositionId,ThisNodeId) VALUES (@PositionId,@ThisNodeId)", param);
        if (roots.ChildNodes.Count > 0)
        {
            SaveChildNodes(roots,
RoleGrid.SelectedDataKey[0].ToString());//保存子节点
        }
    }
    Response.Write("<script>alert('保存成功! ');</script>");
}
protected void SaveChildNodes(TreeNode root,string PositionId)//保存子节点
{
    foreach(TreeNode tn in root.ChildNodes)
    {
        if(tn.Checked==true)
        {
            Parameters param = new Parameters();
            param.Add("@PositionId", PositionId);
            param.Add("@ThisNodeId", tn.Value);
            DataAccess.GetNoResult("INSERT dbo.PositionPower
(PositionId,ThisNodeId) VALUES (@PositionId,@ThisNodeId)", param);
            if (tn.ChildNodes.Count > 0)
            {
                SaveChildNodes(tn, PositionId);//如果该节点存在子节点，则递归调用
            }
        }
    }
}

```

#### 5.5.4 职位分配

职位分配模块用于管理人员根据企业自身情况动态管理不同职位的员工，为员工分配职位，使员工使用有权限的功能模块，增强系统灵活性，适用性。核心代码如下：

```

protected void Save_ServerClick(object sender, EventArgs e)
{
    if (RoleGrid.SelectedRow == null || UserGrid.Rows.Count < 1) return;

    string sql = " Delete from PeoplePosition where PositionId=" +
RoleGrid.SelectedDataKey[0].ToString() + "";//删除该职位下所有人
    if (UserId.Value != "")
        sql += " and a.UserId like '%" + (UserId.Value) + "%";
    if (UserName.Value != "")

```

```

        sql += " and a.UserName like '%" + (UserName.Value) + "%'";
    DataAccess.GetNoResult(sql);
    for (int i = 0; i < UserGrid.Rows.Count; i++)
    {
        HtmlInputCheckBox myBox =
(HtmlInputCheckBox)UserGrid.Rows[i].Cells[1].FindControl("UserId");//读取用户ID
        if (myBox.Checked == true)//如果被勾选，则重新新增保存
        {
            Parameters param = new Parameters();
            param.Add("@PositionId", RoleGrid.SelectedDataKey[0].ToString());
            param.Add("@UserID", myBox.Value);
            DataAccess.GetNoResult("Insert Into PeoplePosition(PositionId,UserID)
values (@PositionId,@UserID)", param);
        }
    }
    Response.Write("<script>alert('操作成功');</script>");
}

```

### 5.5.5 密码重置

密码模块用于当员工忘记或丢失密码后，管理人员为其重置密码，增强系统灵活性，适用性。系统也为每个员工都配置了一个修改密码的功能。核心代码如下：

```

protected void EduPerson_Click(object sender, EventArgs e)
{
    if (CheckUser(UserId.Text.Trim()) == true)
    {
        string StrSQL = "update UserInfo set PassWord=pwdencrypt('' + Password.Value
+ '') where UserId=''' + UserId.Text.Trim() + ''';//直接重置密码
        DataAccess.GetResult(StrSQL);
        Response.Write("<script>alert('用户" + EduPerson.Text.Trim() + "的密码重置成
功，请牢记！')</script>");
        UserId.Text = "";
        Password.Value = "";
    }
    else
    {
        Response.Write("<script>alert('用户名错误,请核对后重试!')</script>");
    }
}

```

## 6 系统测试

### 6.1 测试计划

本次测试对象为企业员工绩效考核系统；测试范围为整个系统；测试方法

为黑盒测试；测试硬件环境为英特尔 i7-410HQ 处理器，8G 内存，1TB 机械硬盘；测试软件环境环境为 Windows 10 企业版操作系统搭载 IIS10，数据库使用 SQLServer2012；测试工具为谷歌浏览器；测试人员为本人。

## 6.2 测试用例

### 6.2.1 系统登陆

用户在登录时，需要输入账号，密码，以及选择用户类型。用户类型包括系统管理员、资产管理员和普通老师，登录测试用例如表 10 所示：

表 10 登录界面测试用例

用例编号	用例摘要	操作步骤	预期结果	实际结果
T-01-01	登录系统	1) 输入用户名、 密码、验证码 2) 点击登陆按钮	1)跳转到系统功能页 面	1)跳转到系统功能页 面
T-01-02	登陆系统	1) 输入错误用户 名、密码、验证码 2) 点击登陆按钮	1) 提示“账号或密码 错误”	1) 提示“账号或密码 错误”

### 6.2.2 我的信息管理

我的信息管理包括查询、更改我的个人基本信息，具体测试用例 11 所示：

表 11 我的信息管理测试用例

用例编号	用例摘要	操作步骤	预期结果	实际结果
T-02-01	查看我的信 息	1) 点击我的 信息管理节 点	1) 显示我的信息	1) 显示我的信息
T-02-02	管理我的信 息	1) 填写全部信息 2) 点击保存按钮	1)提示“操作成功”， 更新信息	1)提示“操作成功”， 更新信息

### 6.2.3 人员信息管理

人员信息管理包括查询、添加、修改和删除员工基本信息，具体测试用例如表 12 所示：

表 12 人员信息管理测试用例

用例编号	用例摘要	操作步骤	预期结果	实际结果
T-03-01	查看员工信 息	1) 点击人员信息 管理节点	1)显示人员信息列表	1)显示人员信息列表

T-03-02	新增员工	1) 点击新增按钮 2) 填写员工账号、密码 3) 点击保存按钮	1) 提示“操作成功”，列表显示新员工	1) 提示“操作成功”，列表显示新员工
T-03-03	维护员工信息	1) 点击查询列表中的人员账号 2) 填写全部信息 3) 点击保存按钮	1) 提示“操作成功”，更新信息	1) 提示“操作成功”，更新信息
T-03-04	删除员工	1) 列表的删除按钮 2) 确认删除	1) 提示“操作成功”，更新列表	1) 提示“操作成功”，更新列表

### 6.2.4 我的绩效考核项目

我的绩效考核项目功能旨在让考核员工查看自己的考核项目以及考核标准，更好地达到考核项目要求，具体测试用例如表 13 所示：

表 13 我的考核项目测试用例

用例编号	用例摘要	操作步骤	预期结果	实际结果
T-04-01	查看我的考核项目	1) 点击我的考核项目节点	1) 显示我的考核项目列表，包含：所有人、所在部门、所任职位的考核项目	1) 显示我的考核项目列表，包含：所有人、所在部门、所任职位的考核项目
T-04-02	查看考核项目详细信息	1) 点击考核项目列表中的考核项目编号	1) 跳转到详细页面，并显示对应项目详细信息	1) 跳转到详细页面，并显示对应项目详细信息

### 6.2.5 我的绩效得分

该模块是用于员工查看其当前或历史的月度或年度考核的得分，以及详细评分信息，具体测试用例如表 14 所示：

表 14 我的绩效得分测试用例

用例编号	用例摘要	操作步骤	预期结果	实际结果
T-05-01	查看我的绩效得分	1) 点击我的绩效得分节点	1) 显示当前年月的绩效考核得分信息	1) 显示当前年月的绩效考核得分信息
T-05-02	条件查询	1) 选择考核类型、考核年月	1) 显示条件查询的考核得分信息	1) 显示条件查询的考核得分信息

		2) 点击查询按钮		
T-05-03	查看考核得分详细信息	1) 点击考核得分列表中的考核记录	1) 跳转到详细页面, 并显示对应得分详细信息	1) 跳转到详细页面, 并显示对应得分详细信息

### 6.2.6 考核项目管理

该功能用于考核管理人员根据企业自身情况制定和维护考核项目及其标准, 增强灵活性和适用性, 具体测试用例如表 15 所示:

表 15 考核项目管理测试用例

用例编号	用例摘要	操作步骤	预期结果	实际结果
T-06-01	查看考核信息	1) 点击考核项目管理节点	1) 显示考核项目列表	1) 显示考核项目列表
T-06-02	新增考核项目	1) 点击新增按钮 2) 填写员工账号、密码 3) 点击保存按钮	1) 提示“操作成功”, 列表显示新考核项目	1) 提示“操作成功”, 列表显示新考核项目
T-06-03	维护考核项目信息	1) 点击考核项目编号 1) 填写全部信息 2) 点击保存按钮	1) 提示“操作成功”, 显示新增考核项目信息	1) 提示“操作成功”, 显示新增考核项目信息
T-06-04	删除考核项目	1) 点击列表的删除按钮 2) 确认删除	1) 提示“操作成功”, 更新列表	1) 提示“操作成功”, 更新列表

### 6.2.7 人员绩效管理

该模块是用于考核管理人员为员工各个绩效考核项目评分, 形成详细的考核记录以供查询, 具体测试用例如表 16 所示:

表 16 人员绩效管理测试用例

用例编号	用例摘要	操作步骤	预期结果	实际结果
T-07-01	查看人员得分信息	1) 点击人员绩效管理节点	1) 显示所有员工绩效考核记录	1) 显示所有员工绩效考核记录
T-07-02	新增考核得分信息	1) 选择考核类型和考核年月等条件, 点击“查询”按钮	1) 提示“操作成功”, 并更新员工考核记录	1) 提示“操作成功”, 并更新员工考核记录

		钮，显示条件查询结果，点击任意人员编号 2) 填写每项考核项目的得分 3) 点击保存按钮		
T-07-03	维护考核项目信息	1) 条件查询后选择人员编号 1) 填写考核得分得分信息 2) 点击保存按钮	1) 提示“操作成功”，并更新员工考核记录	1) 提示“操作成功”，并更新员工考核记录

### 6.2.8 部门管理

该模块是用于企业依据自身情况构建部门组织，使考核项目种类多样化，考核更加准确，增强系统灵活性，适用性，具体测试用例如表 17 所示：

表 17 部门管理测试用例

用例编号	用例摘要	操作步骤	预期结果	实际结果
T-08-01	查看部门信息	1) 点击部门管理管理节点	1) 显示部门信息列表	1) 显示部门信息列表
T-08-02	新增部门	1) 点击新增按钮 2) 填写全部部门信息 3) 点击保存按钮	1) 提示“操作成功”，列表显示新部门	1) 提示“操作成功”，列表显示新部门
T-08-03	维护部门信息	1) 点击查询列表中的部门编号 2) 填写全部信息 3) 点击保存按钮	1) 提示“操作成功”，更新信息	1) 提示“操作成功”，更新信息
T-08-04	删除部门	1) 点击列表的删除按钮 2) 确认删除	1) 提示“操作成功”，更新列表	1) 提示“操作成功”，更新列表

### 6.2.9 职位管理

职位管理模块用于管理人员根据企业自身情况动态构建部门中职位，使考核项目种类多样化，考核更加准确，增强系统灵活性，适用性，具体测试用例如表 18 所示：



表 18 职位管理测试用例

用例编号	用例摘要	操作步骤	预期结果	实际结果
T-09-01	查看职位信息	1) 点击职位管理 管理节点	1)显示职位信息列表	1)显示职位信息列表
T-09-02	新增职位	1) 点击新增按钮 2) 填写全部部门 信息 3) 点击保存按钮	1)提示“操作成功”， 列表显示新职位	1)提示“操作成功”， 列表显示新职位
T-09-03	维护职位信息	1) 点击查询列表 中的职位编号 2) 填写全部信息 3) 点击保存按钮	1)提示“操作成功”， 更新信息	1)提示“操作成功”， 更新信息
T-09-04	删除职位	1) 点击列表的删 除按钮 2) 确认删除	1)提示“操作成功”， 更新列表	1)提示“操作成功”， 更新列表

#### 6.2.10 权限管理

权限管理模块用于管理人员根据企业自身情况动态管理不同职位的系统功能节点权限，员工只能使用有权限的功能模块，增强系统灵活性，适用性，具体测试用例如表 19 所示：

表 19 权限管理测试用例

用例编号	用例摘要	操作步骤	预期结果	实际结果
T-10-01	查看权限	1) 点击职位列表 中任意职位	1) 显示勾选节点树	1) 显示勾选节点树
T-10-02	维护权限	1) 选择职位列表 任意职位 2) 勾选权限节点 树 3) 点击保存按钮	1)提示“操作成功”， 更新权限节点树	1)提示“操作成功”， 更新权限节点树

#### 6.2.11 职位分配

职位分配模块用于管理人员根据企业自身情况动态管理不同职位的员工，为员工分配职位，使员工使用有权限的功能模块，增强系统灵活性，适用性，具体测试用例如表 20 所示：

表 20 职位分配测试用例

用例编号	用例摘要	操作步骤	预期结果	实际结果
T-11-02	查看职位人员	1) 点击职位列表中任意职位	1)显示该职位所有人员	1)显示该职位所有人员
T-11-02	维护职位人员	1) 选择职位列表任意职位 2) 勾选人员表中该职位人员 3) 点击保存按钮	1)提示“操作成功”，显示该职位所有人员	1)提示“操作成功”，显示该职位所有人员

### 6.2.12 密码重置

密码模块用于当员工忘记或丢失密码后，管理人员为其重置密码，增强系统灵活性，适用性，具体测试用例如表 21 所示：

表 21 密码重置测试用例

用例编号	用例摘要	操作步骤	预期结果	实际结果
T-12-01	重置密码	1) 输入账号和新密码 2) 点击重置按钮	1)提示“操作成功”，员工能用新密码登陆系统	1)提示“操作成功”，员工能用新密码登陆系统

## 6.3 测试结果

表 22 测试结果

编号	测试项目	用例编号	执行结果
1	登录系统	T-01-01, T-01-02	Pass
2	我的信息管理	T-02-01, T-02-02	Pass
3	人员信息管理	T-03-01, T-03-02, T-03-03, T-03-04	Pass
4	我的绩效考核项目	T-04-01, T-04-02	Pass
5	我的绩效得分	T-05-01, T-05-02, T-05-03	Pass
6	考核项目管理	T-06-01, T-06-02, T-06-03, T-06-04	Pass
7	人员绩效管理	T-07-01, T-07-02, T-07-03	Pass
8	部门管理	T-08-01, T-08-02, T-08-03, T-08-04	Pass
9	职位管理	T-09-01, T-09-02, T-09-03, T-09-04	Pass
10	权限管理	T-10-01, T-10-02	Pass
11	职位分配	T-11-01, T-11-02	Pass
12	密码重置	T-12-01	pass

## 结 语

经过三个多月的研究、设计、代码编写、调试测试，基本实现了需求中所描述的功能模块，经过测试，系统能够正常稳定地运行，并且由于系统功能权限节点和绩效考核项目及标准是由管理员动态管理和分配，因此能满足不同企业的要求，更加灵活，适用性更强。在查找相关资料文献的过程中，让我充分了解了企业绩效考核的模式、项目、标准，这有利于以后在企业工作过程中的学习、生活和成长；在查询开发文档和开发功能模块的过程中，巩固了已经学到的专业知识，拓展了额外知识，锻炼了系统项目设计、开发的能力，让我获益匪浅。

但是，系统还是有许多不够完善的地方，比如：页面设计不够整洁美观、不具备数据的导入导出 Excel 功能、功能节点还不够丰富、没有做页面的权限验证等等，其中导入和导出 Excel 功能可以在 AccessDatabaseEngine 引擎之后在程序中调用接口即可实现，而丰富功能节点、页面权限验证、页面设计还需要额外的学习才能逐渐将之实现和完善。这些不足反映出我所学的知识还不够，技能还不熟，这将不断提醒和鞭策我学习和进步。

## 参考文献

- [1] 杨雪梅. 企业绩效考核现状研究[J]. 现代交际, 2011, (09)
- [2] 方睿等编著. 数据库原理及应用[M]. 北京: 机械工业出版社, 2010. 11
- [3] 王海春等著. ASP.NET 开发与应用实践[M]. 西安: 西安电子科技大学出版社, 2016. 1
- [4] 熊平. 浅析绩效考核[J]. 现代经济信息, 2010, (04)
- [5] 刁建忠, 张荣进等著. C#程序设计与应用[M]. 清华大学出版社, 2010. 07
- [6] 潘文林. 数据库应用系统开发案例与实践教程[M]. 清华大学出版社, 2005. 09
- [7] 张景峰. ASP 程序设计教程[M]. 北京: 水利水电出版社, 2007
- [8] 张海藩. 软件工程导论[M]. 清华大学出版社, 2013. 08

## 致 谢

本文是在赵军老师的热情关心和指导下完成的，他渊博的知识和严谨的治学态度使我受益匪浅，对顺利完成本课题起到了极大的作用。在此向他表示我最衷心的感谢！

在论文完成过程中，本人还得到了其他老师和同学的热心帮助，本人向她们表示深深的谢意！

最后向在百忙之中评审本文的各位专家、老师表示衷心的感谢！

作者简介：

姓 名：曾辉

性别：男

出生年月：1995 年 4 月 18 日

民族：汉

E-mail:1004705955 @qq. com

## 声 明

本论文的工作是 2017 年 12 月至 2018 年 6 月在成都信息工程大学网络空间安全学院完成的。文中除了特别加以标注地方外，不包含他人已经发表或撰写过的研究成果，也不包含为获得成都信息工程大学或其他教学机构的学位或证书而使用过的材料。

关于学位论文使用权和研究成果知识产权的说明：

本人完全了解成都信息工程大学有关保管使用学位论文的规定，其中包括：

- (1) 学校有权保管并向有关部门递交学位论文的原件与复印件。
- (2) 学校可以采用影印、缩印或其他复制方式保存学位论文。
- (3) 学校可以学术交流为目的复制、赠送和交换学位论文。
- (4) 学校可允许学位论文被查阅或借阅。
- (5) 学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

除非另有科研合同和其他法律文书的制约，本论文的科研成果属于成都信息工程大学。

特此声明！

作者签名：

2018 年 06 月 12 日