



湖南工程职业技术学院

# 毕业设计

## 产品设计说明书

题 目 无忧人力资源管理的设计与实现

毕业设计类型 ☒ 产品设计 ☐ 工艺设计 ☐ 方案设计

专 业 名 称 软件技术

所 属 班 级 2018173802

学 生 姓 名 蔡智威

学 生 学 号 201817380213

指 导 老 师（学校） 胡瑶

指 导 老 师（企业） 邓佳豪

湖南工程职业技术学院教务处 制

# 目录

插图索引.....	III
附表索引.....	IV
第一章 项目概况.....	1
1.1 设计背景.....	1
1.2 设计意义.....	1
1.3 设计思路.....	1
第二章 需求分析.....	3
2.1 需求描述.....	3
2.2 用例建模.....	3
2.3 用例描述.....	4
2.3.1 部门信息用例描述.....	4
2.3.2 员工信息用例描述.....	6
2.3.3 招聘信息用例描述.....	7
2.3.4 培训管理用例描述.....	8
2.3.5 奖罚信息用例描述.....	9
2.3.6 公告信息用例描述.....	10
第三章 系统设计.....	11
3.1 数据库设计.....	11
3.1.1 概念模型.....	13
3.1.2 逻辑模型.....	14
3.1.3 物理模型.....	15
3.1.4 数据库脚本.....	16
3.2 系统架构设计.....	23
3.3 模块设计.....	24
3.3.1 部门管理信息类图.....	24
3.3.2 部门管理信息操作流程.....	25
3.3.3 员工信息类图.....	26
3.3.4 员工信息操作流程.....	27
3.3.5 培训信息类图.....	28
3.3.6 培训信息操作流程.....	29
3.3.7 招聘信息类图.....	29
3.3.8 招聘信息操作流程.....	30
3.3.9 奖罚信息类图.....	31
3.3.10 奖罚信息操作流程.....	32
3.3.11 公告信息类图.....	34
3.3.12 公告信息操作流程.....	34
第四章 产品实现.....	36
4.1 部门管理模块实现.....	36
4.1.1 界面设计.....	36
4.1.2 功能实现.....	37
4.2 员工信息模块实现.....	38
4.2.1 界面设计.....	38

4.2.2 功能实现.....	39
4.3 招聘信息模块实现.....	41
4.3.1 界面设计.....	41
4.3.2 功能实现.....	42
4.4 培训信息模块实现.....	43
4.4.1 界面设计.....	43
4.4.2 功能实现.....	45
4.5 奖惩信息模块实现.....	45
4.5.1 界面设计.....	45
4.5.2 功能实现.....	47
4.6 公告信息模块实现.....	47
4.6.1 界面设计.....	47
4.6.2 功能实现.....	49
第五章 产品测试.....	50
5.1 测试方法与策略.....	50
5.2 部门信息模块测试.....	50
5.3 员工信息模块测试.....	53
5.4 招聘信息模块测试.....	57
5.5 培训信息模块测试.....	59
5.6 奖惩信息模块测试.....	62
5.7 公告信息模块测试.....	63
第六章 结论.....	65
6.1 毕业设计成果特点.....	65
6.2 设计成果的实用价值或应用前景.....	65
6.3 不足之处或遗留未予解决的问题.....	65

## 插图索引

图 2-1 无忧入力资源管理用例模型图.....	4
图 3-1 无忧入力资源管理概念模型图.....	14
图 3-2 无忧入力资源管理逻辑模型图.....	15
图 3-3 无忧入力资源管理物理模型图.....	16
图 3-4 无忧人力资源管理系统架构设计图.....	24
图 3-5 部门信息管理类图.....	25
图 3-6 部门信息时序图.....	26
图 3-7 员工信息类图.....	27
图 3-8 员工信息时序图.....	27
图 3-9 培训信息类图.....	28
图 3-10 培训信息时序图.....	29
图 3-11 招聘信息类图.....	30
图 3-12 招聘信息时序图.....	31
图 3-13 奖罚信息类图.....	32
图 3-14 招聘信息时序图.....	33
图 3-15 公告信息类图.....	34
图 3-16 公告信息时序图.....	35
图 4-1 部门信息列表界面图.....	36
图 4-2 部门信息添加界面图.....	37
图 4-3 部门信息修改界面图.....	37
图 4-4 部门管理信息主要代码图.....	38
图 4-5 员工信息列表界面图.....	39
图 4-6 员工信息添加界面图.....	39
图 4-7 员工工资添加界面图.....	39
图 4-8 员工信息模块主要代码图.....	40
图 4-9 员工信息模块主要代码图.....	40
图 4-10 招聘信息列表界面图.....	41
图 4-11 招聘信息添加界面图.....	41
图 4-12 录用应聘者添加界面图.....	42
图 4-13 员工信息模块主要代码图.....	42
图 4-14 员工信息模块主要代码图.....	43
图 4-15 培训信息列表界面图.....	44
图 4-16 培训信息添加界面图.....	44
图 4-17 培训信息修改界面图.....	44
图 4-18 培训信息模块主要代码图.....	45
图 4-19 奖罚信息列表界面图.....	46
图 4-20 奖罚信息添加界面图.....	46
图 4-21 考勤信息列表界面图.....	46
图 4-22 奖罚信息模块主要代码图.....	47
图 4-23 公告信息列表界面图.....	48
图 4-24 公告信息添加界面图.....	48
图 4-25 查看公告内容.....	48
图 4-26 删除公告内容.....	49
图 4-27 公告信息模块主要代码图.....	49

## 附表索引

表 2-1 部门信息用例描述表.....	4
表 2-2 员工信息用例描述表.....	6
表 2-3 招聘信息用例描述表.....	7
表 2-4 培训信息用例描述表.....	8
表 2-5 奖惩信息用例描述表.....	9
表 2-6 公告信息用例描述表.....	10
表 3-1 Admini 表（管理员信息表） .....	11
表 3-2 punishments 表（奖惩信息表） .....	11
表 3-3 Recruitment 表（招聘信息表） .....	11
表 3-4 Stafftable 表（员工信息表） .....	12
表 3-5 Wage 表（工资信息表） .....	12
表 3-6 Departmental 表（部门管理信息表） .....	12
表 3-7 notice 表（公司公告信息表） .....	12
表 3-8 Clockin 表（考勤信息表） .....	13
表 3-9 Trainingform 表（企业培训信息表） .....	13
表 5-1 部门信息模块测试用例表.....	50
表 5-2 教练信息测试用例表.....	53
表 5-3 招聘信息模块测试用例表.....	57
表 5-4 培训信息模块测试用例表.....	60
表 5-5 奖惩信息模块测试用例表.....	62
表 5-6 公告信息模块测试用例表.....	63

# 第一章 项目概况

## 1.1 设计背景

随着互联网信息时代的高速发展,企业对人力资源管理也迎来了信息化发展。企业需要对公司部门、人员等信息进行一体化发展,提高公司的管理效率,所以我们决定做一个人力资源管理系统,来满足企业高效便捷的管理方式,可以满足企业对部门、员工、薪资、培训、工资、考勤等多方面的管理需求。

随着大数据、云计算、人工智能等技术逐步在向各产业领域下沉的大背景下,无忧人力资源管理系统实现研发、技术的与时俱进,持续发挥自身品牌、服务等方面的实力与优势,助推互联网企业管理系统的广泛应用。以企业管理为基础,无忧人力资源管理系统旨在逐步构建出完备、平顺、卓效的企业管理服务新生态体系,助力促进企业管理更加高效、便捷促进企业更加快速的发展。

## 1.2 设计意义

该伴随着我国经济不断的发展,物质生活水平不断的提高,人们开始对生活物质条件提出更高的要求。加上公司企业不断的增多,企业不断的发展壮大,企业管理也变得更加的需要,公司企业需要一个可以对人员、部门、薪资、考勤等一系列一体化的管理系统,而我设计的无忧人力资源管理系统就提供了一体化的管理系统,方便了企业对各个部门和人员进行管理,也方便的员工上班的打开方式,提供了便捷一体化管理方式,摆脱企业管理难的问题。

传统的管理方式大多以人工的方式进行直接管理,公司或企业根据部门的不同安排不同的管理人员,造成机构内部管理混乱,工作效率低下等。现今,在竞争日趋激烈的社会行业中,企业需要直接有效的方式对部门和人员进行管理,提高管理上的灵活性,实现企业稳步的发展与壮大。为适应企业信息化策略的要求,通过企业所参与的行业不同,来实现不同差异化的管理方式,提高管理效率。

## 1.3 设计思路

本系统目前的主要对象是企业。

本系统分为七大模块,分别是部门管理模块、员工管理模块、招聘信息管理模块、薪资管理模块、考勤管理模块、奖惩管理模块、公告管理模块。

部门管理模块:主要用来对部门进行相应的管理操作

员工管理模块：用来对员工进行相关的操作

招聘信息管理模块：用来对企业所需的员工进行招聘

薪资管理模块：管理员工的薪资信息

培训管理模块：发布公司员工的培训信息

奖惩管理模块：对公司员工的奖罚信息进行记录管理

公告管理模块：发布公司公告修改

设计开发流程如下：

1. 需求分析：包括需求调研，系统功能，性能分析，业务流程分析。
2. 概要设计：包括系统架构设计，功能模块设计，数据库设计。
3. 详细设计：包括方法设计、方法的封装、数据结构设计、属性使用设计。
4. 系统实现：包括编写代码，系统单元测试。
5. 系统测试：对项目功能进行系统测试。

## 第二章 需求分析

### 2.1 需求描述

系统中有多种角色，每种角色拥有不同的功能；管理员具有部门信息、员工信息、招聘信息、培训信息、奖罚信息、考勤信息、企业公告和员工工资管理全部功能。

管理员通过权限操作对系统的权限值进行增删改查，通过菜单权限分配实现各菜单的访问权限；企业对企业人员和部门信息进行增删改查，实现对企业人员和企业部门的增加和管理，企业对企业人员的工资和考勤进行增删改查，实现对企业员工的工资和企业员工考勤状况的管理，企业对企业招聘信息的增查，企业对招聘信息的管理，企业对企业管理培训进行增删改查，实现对企业培训的增加与管理，企业对企业公告进行增删改查，实现对企业公告的添加与管理。

### 2.2 用例建模

本系统中主要包括两类用户：员工、管理员，其中可以对这几类用户说明下。本系统中我实现的部门信息、员工信息、招聘信息、培训信息、奖罚信息、考勤信息、企业公告和员工工资管理为管理员。

根据前面的分析，可以得到无忧入力资源管理系统部门信息、员工信息、招聘信息、培训信息、奖罚信息、考勤信息、企业公告和员工工资管理的总体用例图如下图 2-1 所示：



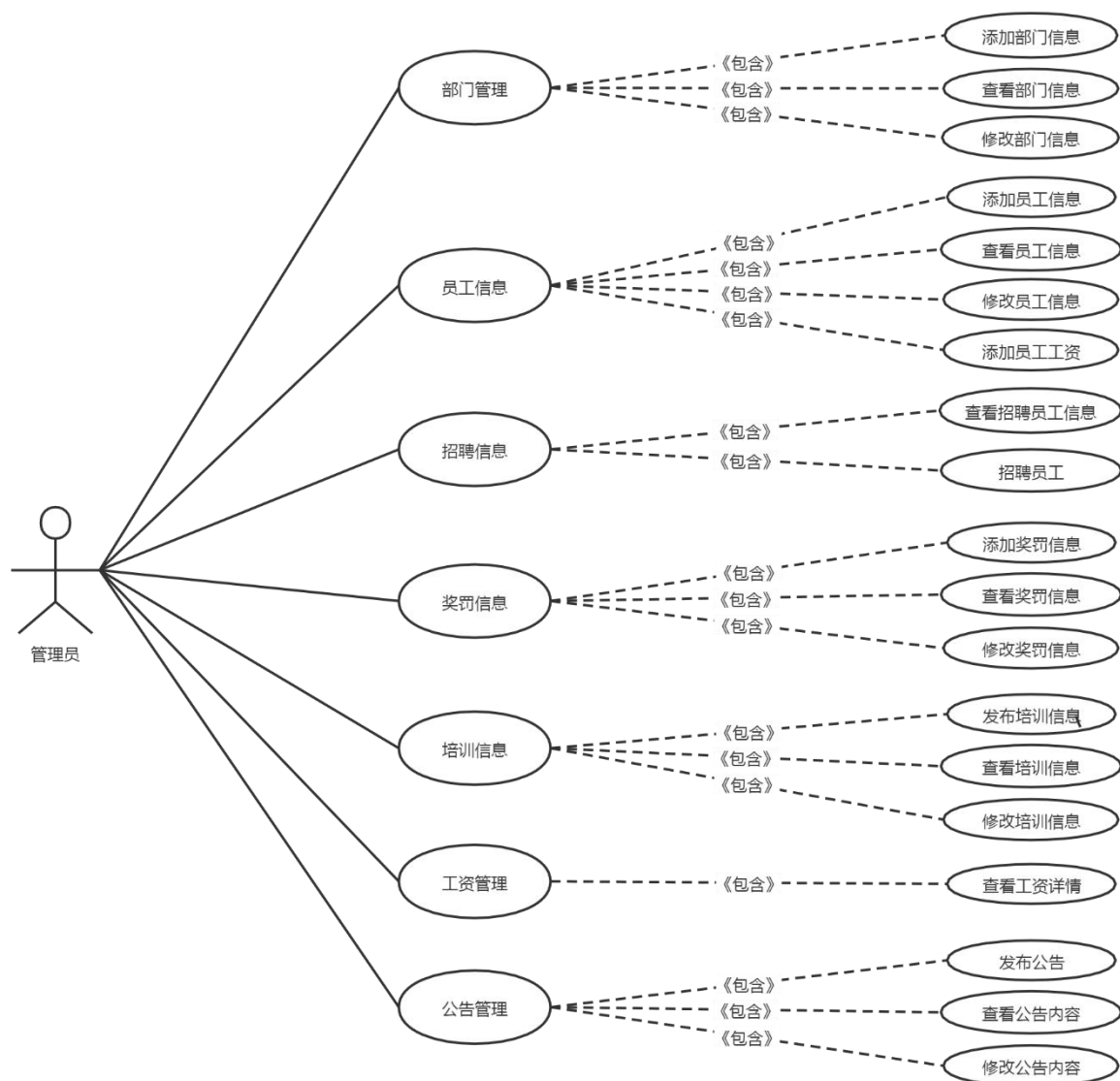


图 2-1 无忧入力资源管理用例模型图

## 2.3 用例描述

### 2.3.1 部门信息用例描述

部门信息：管理员登录后可以查看全部部门信息，可以新增或修改部门信息或是删除选中的部门信息。其具体的用例描述如下表 2-1 所示：

表 2-1 部门信息用例描述表

用例编号	CASE01
用例名称	部门信息

用例描述	系统管理员对部门信息进行 CRUD 操作
前置条件	系统管理员成功登录系统
后置条件	系统管理员在系统中新增了部门或修改了部门信息或查看部门的信息
活动步骤	<p>1-系统管理员登录系统</p> <p>2-系统显示部门列表</p> <p>3-系统管理员点击【新增部门】按钮，弹出新建部门界面，系统管理员输入部门信息，点击【添加】按钮，系统对输入的信息进行验证，将合法的信息保存，然后显示新增后的部门列表。</p> <p>4-系统管理员在部门列表中选择某个部门，点击【修改】，弹出部门修改界面，显示部门的当前信息，管理员更改部门信息，点击【保存】，系统对输入的信息进行验证，将合法的信息保存，然后显示修改后的部门列表。</p> <p>5-系统管理员在部门列表中选择某个部门，点击【查看】，弹出用户显示界面，显示部门的当前信息，点击【关闭】按钮关闭显示窗口。</p>
扩展点	无
数据格式	部门信息（部门 ID）

## 2.3.2 员工信息用例描述

员工信息：管理员登录后可以查看全部员工信息，可以新增或修改员工信息。其具体的用例描述如下表 2-2 所示：

表 2-2 员工信息用例描述表

用例编号	CASE02
用例名称	员工信息
用例描述	系统管理员对员工的信息进行 CRUD 操作
前置条件	系统管理员成功登录系统
后置条件	系统管理员在系统中新增了员工或修改了员工信息或查看全部员工的信息或添加员工的工资
活动步骤	<p>1-系统管理员登录系统</p> <p>2-系统显示员工列表</p> <p>3-系统管理员点击【新增员工】按钮，弹出新建员工界面，系统管理员输入员工信息，点击【添加】按钮，系统对输入的信息进行验证，将合法的信息保存，然后显示新增后的员工列表。</p> <p>4-系统管理员在员工列表中选择某个员工，点击【修改】，弹出员工修改界面，显示员工的当前信息，管理员更改员工信息，点击【保存】，系统对输入的信息进行验证，将合法的信息保存，然后显示修改后的员工列表。</p> <p>5-系统管理员在员工列表中选择员工，点击【查看】，弹出用户显示界面，显示员工的当前信息，点击【关闭】按钮关闭显示窗口。</p> <p>6-系统管理员在员工列表中选择某个员工，点击【添加工资】按钮，弹出用户界面，显示添加工资界面，管理员输入员工的工资详情点击【保存】按钮，将合法信息保存，然后显示新增后的工资界面。</p>
扩展点	无
数据格式	员工信息（员工 ID）

### 2.3.3 招聘信息用例描述

招聘信息：管理员登录后可以查看全部招聘信息，对所需要的的员工进行录取。其具体的用例描述如下表 2-3 所示：

表 2-3 招聘信息用例描述表

用例编号	CASE03
用例名称	招聘信息
用例描述	系统管理员对招聘的信息进行 CRUD 操作
前置条件	系统管理员成功登录系统
后置条件	系统管理员在系统中查看全部员工的信息或录取所需员工
活动步骤	1-系统管理员登录系统 2-系统显示招聘信息列表 3-系统管理员点击【新增招聘信息】按钮，弹出新建应聘者界面，系统管理员输入应聘者信息，点击【添加】按钮，系统对输入的信息进行验证，将合法的信息保存，然后显示新增后的应聘者列表。 4-系统管理员在招聘列表中选择应聘者，点击【查看】，弹出用户显示界面，显示应聘者的当前信息，点击【关闭】按钮关闭显示窗口。 5-系统管理员在员工列表中选择某个应聘者，点击【予以录取】按钮，弹出用户界面，显示录取员工界面，管理员点击【确定】按钮，将合法信息保存，然后显示新增后的员工界面。
扩展点	无
数据格式	招聘信息（应聘者 ID）

## 2.3.4 培训管理用例描述

员工培训信息：管理员登录后可以查看全部员工培训信息，对可以添加相应的培训信息和修改培训信息。其具体的用例描述如下表 2-4 所示：

表 2-4 培训信息用例描述表

用例编号	CASE04
用例名称	员工培训信息
用例描述	系统管理员对员工培训信息进行 CRUD 操作
前置条件	系统管理员成功登录系统
后置条件	系统管理员在系统中查看全部员工培训信息或添加新的员工培训信息，以及修改员工培训信息。
活动步骤	<p>1-系统管理员登录系统</p> <p>2-系统显示员工培训信息列表</p> <p>3-系统管理员点击【新增培训】按钮，弹出新建员工培训界面，系统管理员输入员工培训信息，点击【添加】按钮，系统对输入的信息进行验证，将合法的信息保存，然后显示新增后的员工培训列表。</p> <p>4-系统管理员在员工培训列表中选择培训信息，点击【查看】，弹出用户培训信息界面，显示员工培训的相应信息，点击【关闭】按钮关闭显示窗口。</p> <p>5-系统管理员在员工培训列表中选择某个培训信息，点击【修改】按钮，弹出培训信息修改界面，输入先要修改的内容，管理员点击【确定】按钮，将合法信息保存，然后显示修改后的员工培训列表界面。</p>
扩展点	无
数据格式	员工培训信息（员工 ID）

## 2.3.5 奖罚信息用例描述

员工奖罚信息：管理员登录后可以查看全部员工奖罚信息，对可以添加相应的奖罚信息和修改奖罚信息以及查看员工的考勤状况。其具体的用例描述如下表 2-5 所示：

表 2-5 奖罚信息用例描述表

用例编号	CASE05
用例名称	员工奖罚信息
用例描述	系统管理员对员工奖罚信息进行 CRUD 操作
前置条件	系统管理员成功登录系统
后置条件	系统管理员在系统中查看全部员工奖罚信息或添加新的员工奖罚信息，以及查看员工考勤状况。
活动步骤	1-系统管理员登录系统 2-系统显示员工培训信息列表 3-系统管理员点击【新增奖罚信息】按钮，弹出新建员工奖罚界面，系统管理员输入员工奖罚信息，点击【添加】按钮，系统对输入的信息进行验证，将合法的信息保存，然后显示新增后的员工培训列表。 4-系统管理员在员工奖罚信息列表中选择奖罚信息，点击【查看】，弹出用户奖罚信息界面，显示员工奖罚的相应信息，点击【关闭】按钮关闭显示窗口。 5-系统管理员点击【员工考勤】按钮，可以查看所有员工的考勤状况。
扩展点	无
数据格式	员工奖罚信息（员工奖罚 ID）

## 2.3.6 公告信息用例描述

公告信息：管理员登录后可以查看全部公告信息，对可以添加相应的公告信息和删除公告信息以及查看公告。其具体的用例描述如下表 2-6 所示：

表 2-6 公告信息用例描述表

用例编号	CASE06
用例名称	公告信息
用例描述	系统管理员对公告信息进行 CRUD 操作
前置条件	系统管理员成功登录系统
后置条件	系统管理员在系统中查看全部公告信息或添加新的公告信息，以及删除公告信息和查看详情。
活动步骤	1-系统管理员登录系统 2-系统显示员工培训信息列表 3-系统管理员点击【新增公告】按钮，弹出新建公告界面，系统管理员输入公告信息，点击【添加】按钮，系统对输入的信息进行验证，将合法的信息保存，然后显示新增后的公告列表。 4-系统管理员在公告信息列表中选择公告信息，点击【查看】，弹出公告界面，显示公告内容详情信息信息，点击【关闭】按钮关闭显示窗口。 5-系统管理员点击【删除】按钮，可以删除相应的公告信息。
扩展点	无
数据格式	公告信息（公告 ID）

# 第三章 系统设计

根据无忧人力资源管理系统的需求分析结果，本章主要设计了无忧人力资源管理系统数据库设计，以数据字典的形式一一罗列出表自身的组成结构，并使 PowerDesigner 工具设计了概念模型、逻辑模型和物理模型，客观清晰的展示了表与表之间的关系；本系统采用 MVC 架构设计模型、划分了系统的功能模块使用 UML 类图和 UML 时序图详细描述了各功能的设计思路。

## 3.1 数据库设计

无忧人力资源管理的数据库共设计了九张数据表，分别为奖罚信息表、招聘信息表、员工信息表、工资信息表、管理员信息表、部门管理信息表、公司公告信息表、考勤信息表、企业培训信息表。各表的具体结构如下表 3-1 到 3-6 所示：

表 3-1 Admini 表（管理员信息表）

字段名	中文名	类别	长度	允许为空	主键
Id	编号	int			Y
Name	用户名	nvarchar	10		
Password	密码	nvarchar	12		
UserNaem	账号	nvarchar	50	Y	

表 3-2 punishments 表（奖罚信息表）

字段名	中文名	类别	长度	允许为空	主键
EventsID	编号	int			Y
EmployeeID	员工编号	int	10		Y
Name	员工姓名	nvarchar	10	Y	
Reason	表彰内容	nvarchar	50	Y	
Datetime	发布时间	datetime	23	Y	
Amount	表彰奖励	nvarchar	50	Y	

表 3-3 Recruitment 表（招聘信息表）

字段名	中文名	类别	长度	允许为空	主键
RecruitmentID	招聘编号	int			Y
Name	姓名	nvarchar	10		
Phone	电话	nchar	11		
Age	年龄	nchar	10		
Sex	性别	nchar	2		
Nativeplace	籍贯	nvarchar	10	Y	
Education	学历	nvarchar	5	Y	
Position	应聘职位	nvarchar	10	Y	



Photos	照片	nchar	13	Y	
DepartmentID	应聘部门	int	10		
Introduction	自我简介	nvarchar	50		

表 3-4 Stafftable 表（员工信息表）

字段名	中文名	类别	长度	允许为空	主键
EmployeeID	员工编号	int			Y
Name	姓名	nvarchar	10		
Sex	性别	nchar	2		
Age	年龄	int	10		
Telephone	电话	nchar	13		
DepartmentID	所在部门	int	10		
EntryTime	入职日期	datetime	23		
Photos	照片	nvarchar	10	Y	
Position	职位	nvarchar	10		
Education	学历	nvarchar	5		
UserName	个人账号	nchar	18	Y	
Password	密码	varchar	20	Y	
Nativeplace	籍贯	nvarchar	10	Y	

表 3-5 Wage 表（工资信息表）

字段名	中文名	类别	长度	允许为空	主键
ID	工资编号	int	10		Y
EmployeeID	员工编号	int	10		
Name	员工姓名	nvarchar	10		
DepartmentID	部门编号	int	10		
Month	发放工资时间	datetime	23		
Basicwage	基本工资	decimal	18		
Overtime	考勤工资	decimal	18		
Gongz	按月奖励	decimal	18		
Insurance	加班工资	decimal	18		
Real	总工资	decimal	18		

表 3-6 Departmental 表（部门管理信息表）

字段名	中文名	类别	长度	允许为空	主键
DepartmentID	部门编号	int			Y
DepartmentName	部门名称	nvarchar	10		
DepartmentNun	部门简介	nvarchar	50		
Introduction	部门人数	nvarchar	50		
Basic	部门基本工资	numeric	18	Y	

表 3-7 notice 表（公司公告信息表）

字段名	中文名	类别	长度	允许为空	主键
id	公告编号	int	10		Y
name	公告名称	nvarchar	50		

Content	公告内容	nvarchar	MAX		
datetime	发布时间	datetime	23		
photo	配图	nvarchar	50	Y	

表 3-8 Clockin 表（考勤信息表）

字段名	中文名	类别	长度	允许为空	主键
Clockinid	考勤编号	int	10		Y
EmployeeID	员工编号	int	10		Y
Name	员工姓名	nvarchar	10		
Gotowork	上班打卡时间	datetime	23		
Gooffwork	下班打卡时间	datetime	23		

表 3-9 Trainingform 表（企业培训信息表）

字段名	中文名	类别	长度	允许为空	主键
TrainingID	培训编号	int	10		Y
Nametraining	培训名称	nvarchar	50		
Personnel	培训地点	nchar	10	Y	
Location	参与人员	nvarchar	50		
Reason	培训宗旨	nvarchar	50	Y	
Datetime	开始时间	datetime	23		
Datetimes	结束时间	datetime	23		

无忧入力资源管理的后台管理模块设计与实现主要使用到了上表的3-2, 3-3, 3-4, 3-5, 3-6, 3-7, 3-8, 3-9。其中3-4, 3-8表主要用于员工模块。

### 3.1.1 概念模型

根据前文的数据字典的描述，通过使用 PowerDesigner 工具进一步设计了数据库的概念模型结构，展现出各表的自身结构以及各表与表之间的关联。无忧入力资源管理的概念模型如下图 3-1 所示：

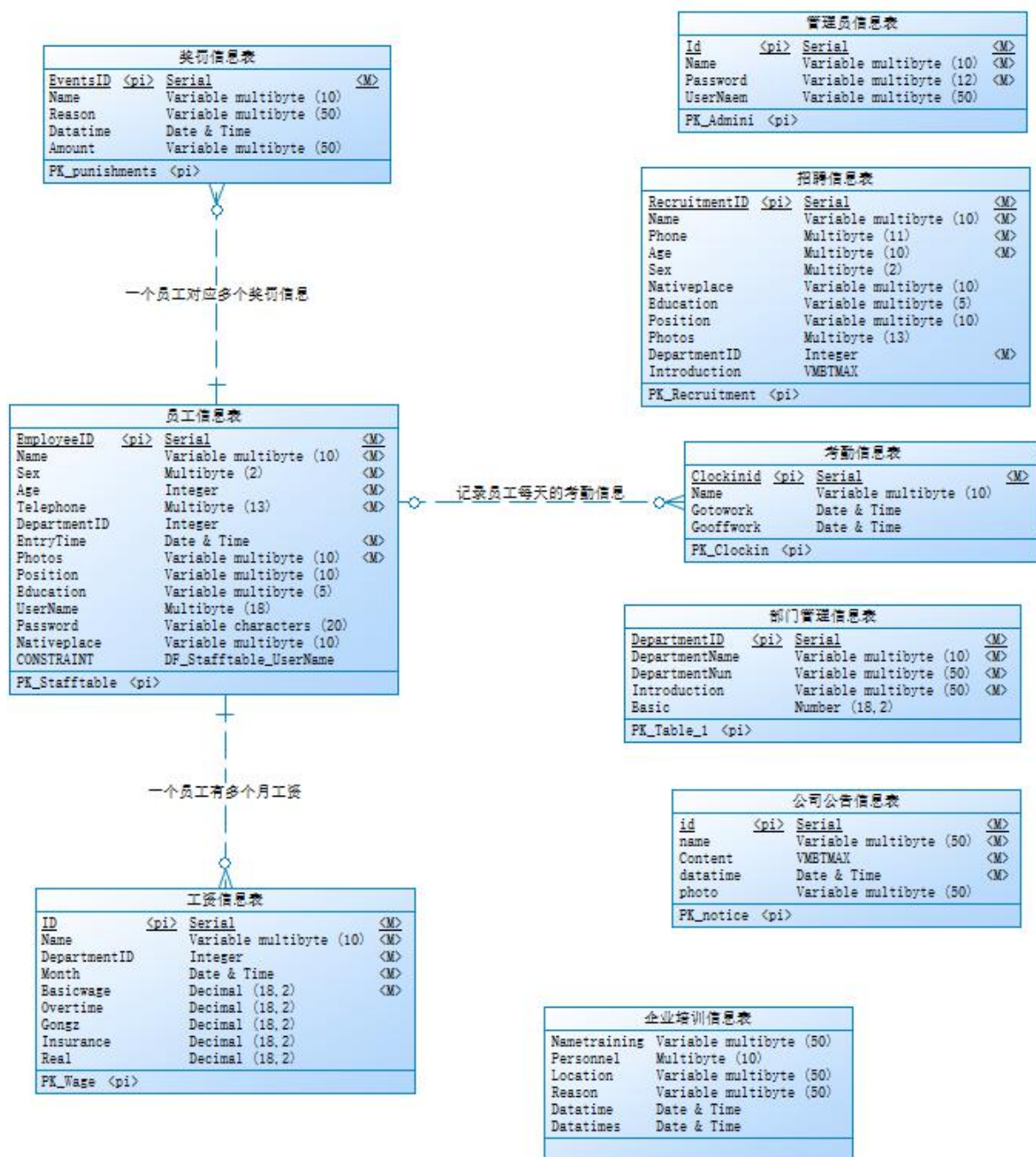


图 3-1 无忧入力资源管理概念模型图

### 3.1.2 逻辑模型

根据前文的数据字典的描述，通过使用 PowerDesigner 工具进一步设计了数据库的逻辑模型结构，展现出各表的自身结构以及各表与表之间的关联。

无忧入力资源管理的逻辑模型如下图 3-2 所示：

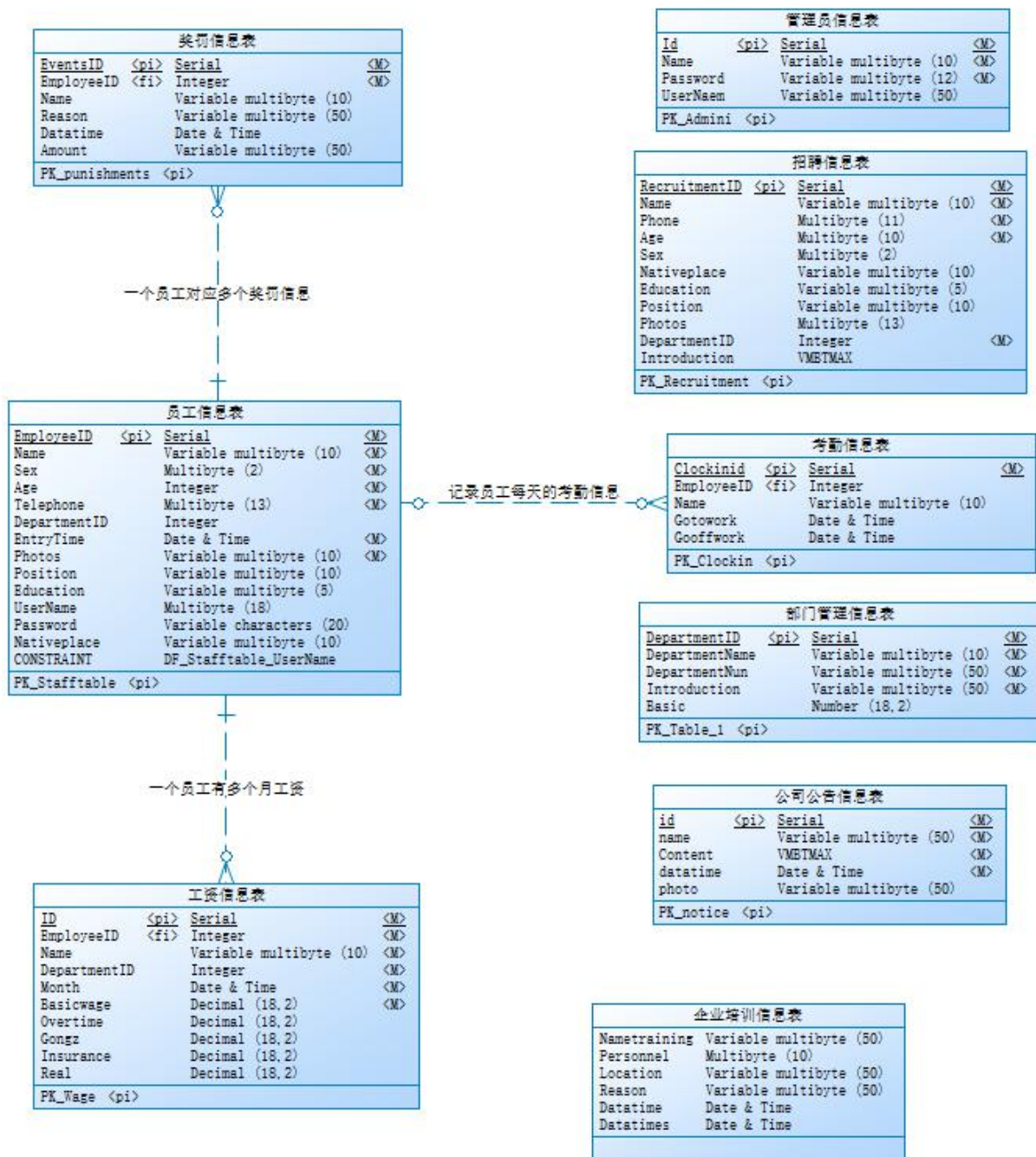


图 3-2 无忧入力资源管理逻辑模型图

### 3.1.3 物理模型

利用PowerDesigner 工具,在上节的逻辑模型基础上考虑各种具体的技术实现因素,进行数据库体系结构设计,真正实现数据在数据库中的存放。无忧入力资源管理的物理模型如下图 3-3 所示:

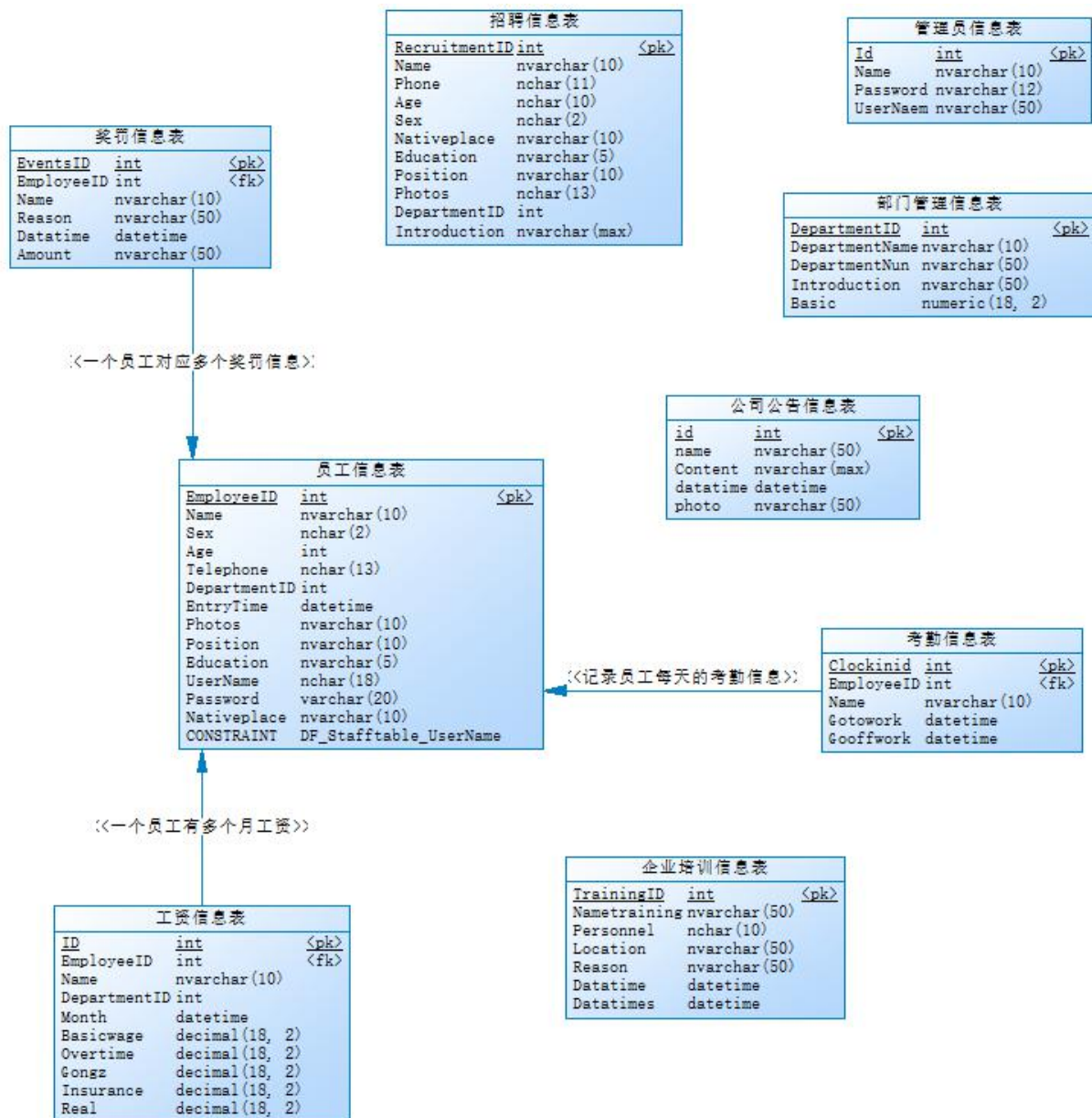


图 3-3 无忧入力资源管理物理模型图

### 3.1.4 数据库脚本

```
USE [master]
GO
/***** Object:  Database [Worry-freemanagement]    Script Date: 2020/9/9 10:04:22
*****/
CREATE DATABASE [Worry-freemanagement]
    CONTAINMENT = NONE
    ON PRIMARY
( NAME = N'Worry-freemanagement', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\Worry-freemanagement.mdf' , SIZE = 5120KB ,
MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )
```

```

LOG ON
( NAME = N'Worry-freemanagement_log', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\Worry-freemanagement_log.ldf' , SIZE = 2048KB ,
MAXSIZE = 2048GB , FILEGROWTH = 10%)
GO
ALTER DATABASE [Worry-freemanagement] SET COMPATIBILITY_LEVEL = 110
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [Worry-freemanagement].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [Worry-freemanagement] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [Worry-freemanagement] SET ANSI_NULLS OFF
GO
ALTER DATABASE [Worry-freemanagement] SET ANSI_PADDING OFF
GO
ALTER DATABASE [Worry-freemanagement] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [Worry-freemanagement] SET ARITHABORT OFF
GO
ALTER DATABASE [Worry-freemanagement] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [Worry-freemanagement] SET AUTO_CREATE_STATISTICS ON
GO
ALTER DATABASE [Worry-freemanagement] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [Worry-freemanagement] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [Worry-freemanagement] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [Worry-freemanagement] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [Worry-freemanagement] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [Worry-freemanagement] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [Worry-freemanagement] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [Worry-freemanagement] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [Worry-freemanagement] SET DISABLE_BROKER
GO
ALTER DATABASE [Worry-freemanagement] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO

```



```

ALTER DATABASE [Worry-freemanagement] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [Worry-freemanagement] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [Worry-freemanagement] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [Worry-freemanagement] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [Worry-freemanagement] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [Worry-freemanagement] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [Worry-freemanagement] SET RECOVERY FULL
GO
ALTER DATABASE [Worry-freemanagement] SET MULTI_USER
GO
ALTER DATABASE [Worry-freemanagement] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [Worry-freemanagement] SET DB_CHAINING OFF
GO
ALTER DATABASE [Worry-freemanagement] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [Worry-freemanagement] SET TARGET_RECOVERY_TIME = 0 SECONDS
GO
USE [Worry-freemanagement]
GO
/***** Object: Table [dbo].[Admini] Script Date: 2020/9/9 10:04:23 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Admini](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Name] [nvarchar](10) NOT NULL,
    [Password] [nvarchar](12) NOT NULL,
    [UserNaem] [nvarchar](50) NULL,
    CONSTRAINT [PK_Admini] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Clockin] Script Date: 2020/9/9 10:04:23 *****/
SET ANSI_NULLS ON

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Clockin](
    [Clockinid] [int] IDENTITY(1,1) NOT NULL,
    [EmployeeID] [int] NULL,
    [Name] [nvarchar](10) NULL,
    [Gotowork] [datetime] NULL,
    [Gooffwork] [datetime] NULL,
    CONSTRAINT [PK_Clockin] PRIMARY KEY CLUSTERED
(
    [Clockinid] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object:  Table [dbo].[Departmental]    Script Date: 2020/9/9 10:04:23 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Departmental](
    [DepartmentID] [int] IDENTITY(1,1) NOT NULL,
    [DepartmentName] [nvarchar](10) NOT NULL,
    [DepartmentNun] [nvarchar](50) NOT NULL,
    [Introduction] [nvarchar](50) NOT NULL,
    [Basic] [numeric](18, 2) NULL,
    CONSTRAINT [PK_Table_1] PRIMARY KEY CLUSTERED
(
    [DepartmentID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object:  Table [dbo].[notice]    Script Date: 2020/9/9 10:04:23 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[notice](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [name] [nvarchar](50) NOT NULL,
    [Content] [nvarchar](max) NOT NULL,
    [datetime] [datetime] NOT NULL,

```



```

        [photo] [nvarchar](50) NULL,
    CONSTRAINT [PK_notice] PRIMARY KEY CLUSTERED
    (
        [id] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO

/***** Object:  Table [dbo].[punishments]      Script Date: 2020/9/9 10:04:23 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[punishments](
    [EventsID] [int] IDENTITY(1,1) NOT NULL,
    [EmployeeID] [int] NOT NULL,
    [Name] [nvarchar](10) NULL,
    [Reason] [nvarchar](50) NULL,
    [Datatime] [datetime] NULL,
    [Amount] [nvarchar](50) NULL,
    CONSTRAINT [PK_punishments] PRIMARY KEY CLUSTERED
    (
        [EventsID] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO

/***** Object:  Table [dbo].[Recruitment]      Script Date: 2020/9/9 10:04:23 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Recruitment](
    [RecruitmentID] [int] IDENTITY(1,1) NOT NULL,
    [Name] [nvarchar](10) NOT NULL,
    [Phone] [nchar](11) NOT NULL,
    [Age] [nchar](10) NOT NULL,
    [Sex] [nchar](2) NULL,
    [Nativeplace] [nvarchar](10) NULL,
    [Education] [nvarchar](5) NULL,
    [Position] [nvarchar](10) NULL,
    [Photos] [nchar](13) NULL,
    [DepartmentID] [int] NOT NULL,
    [Introduction] [nvarchar](max) NULL,

```

```

CONSTRAINT [PK_Recruitment] PRIMARY KEY CLUSTERED
(
    [RecruitmentID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO

/***** Object: Table [dbo].[Stafftable]    Script Date: 2020/9/9 10:04:23 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Stafftable](
    [EmployeeID] [int] IDENTITY(1,1) NOT NULL,
    [Name] [nvarchar](10) NOT NULL,
    [Sex] [nchar](2) NOT NULL,
    [Age] [int] NOT NULL,
    [Telephone] [nchar](13) NOT NULL,
    [DepartmentID] [int] NULL,
    [EntryTime] [datetime] NOT NULL,
    [Photos] [nvarchar](10) NOT NULL,
    [Position] [nvarchar](10) NULL,
    [Education] [nvarchar](5) NULL,
    [UserName] [nchar](18) NULL,
    [Password] [varchar](20) NULL,
    [Nativeplace] [nvarchar](10) NULL,
    CONSTRAINT [PK_Stafftable] PRIMARY KEY CLUSTERED
    (
        [EmployeeID] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO

/***** Object: Table [dbo].[Trainingform]    Script Date: 2020/9/9 10:04:23 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Trainingform](
    [TrainingID] [int] IDENTITY(1,1) NOT NULL,

```

```

[Nametraining] [nvarchar](50) NULL,
[Personnel] [nchar](10) NULL,
[Location] [nvarchar](50) NULL,
[Reason] [nvarchar](50) NULL,
[Datatime] [datetime] NULL,
[Datatimes] [datetime] NULL,
CONSTRAINT [PK_Trainingform] PRIMARY KEY CLUSTERED
(
    [TrainingID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

/***** Object: Table [dbo].[Wage]    Script Date: 2020/9/9 10:04:23 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Wage](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [EmployeeID] [int] NOT NULL,
    [Name] [nvarchar](10) NOT NULL,
    [DepartmentID] [int] NOT NULL,
    [Month] [datetime] NOT NULL,
    [Basicwage] [decimal](18, 2) NOT NULL,
    [Overtime] [decimal](18, 2) NULL,
    [Gongz] [decimal](18, 2) NULL,
    [Insurance] [decimal](18, 2) NULL,
    [Real] [decimal](18, 2) NULL,
    CONSTRAINT [PK_Wage] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
ALTER TABLE [dbo].[notice] ADD CONSTRAINT [DF_notice_datatime] DEFAULT (getdate()) FOR
[datatime]
GO
ALTER TABLE [dbo].[punishments] ADD CONSTRAINT [DF_punishments_Datatime] DEFAULT
(getdate()) FOR [Datatime]
GO
ALTER TABLE [dbo].[Stafftable] ADD CONSTRAINT [DF_Stafftable_EntryTime] DEFAULT
(getdate()) FOR [EntryTime]

```

```

GO
ALTER TABLE [dbo].[Stafftable] ADD CONSTRAINT [DF_Stafftable_UserName] DEFAULT
(N'aa') FOR [UserName]
GO
ALTER TABLE [dbo].[Stafftable] ADD CONSTRAINT [DF_Stafftable_Password] DEFAULT
((1233456)) FOR [Password]
GO
ALTER TABLE [dbo].[Trainingform] ADD CONSTRAINT [DF_Trainingform_Datetime] DEFAULT
(getdate()) FOR [Datetime]
GO
ALTER TABLE [dbo].[Clockin] WITH CHECK ADD CONSTRAINT [FK_Clockin_Stafftable] FOREIGN
KEY([EmployeeID])
REFERENCES [dbo].[Stafftable] ([EmployeeID])
GO
ALTER TABLE [dbo].[Clockin] CHECK CONSTRAINT [FK_Clockin_Stafftable]
GO
ALTER TABLE [dbo].[punishments] WITH CHECK ADD CONSTRAINT [FK_punishments_Stafftable]
FOREIGN KEY([EmployeeID])
REFERENCES [dbo].[Stafftable] ([EmployeeID])
GO
ALTER TABLE [dbo].[punishments] CHECK CONSTRAINT [FK_punishments_Stafftable]
GO
ALTER TABLE [dbo].[Trainingform] WITH CHECK ADD CONSTRAINT
[FK_Trainingform_Trainingform] FOREIGN KEY([TrainingID])
REFERENCES [dbo].[Trainingform] ([TrainingID])
GO
ALTER TABLE [dbo].[Trainingform] CHECK CONSTRAINT [FK_Trainingform_Trainingform]
GO
ALTER TABLE [dbo].[Wage] WITH CHECK ADD CONSTRAINT [FK_Wage_Stafftable] FOREIGN
KEY([EmployeeID])
REFERENCES [dbo].[Stafftable] ([EmployeeID])
GO
ALTER TABLE [dbo].[Wage] CHECK CONSTRAINT [FK_Wage_Stafftable]
GO
USE [master]
GO
ALTER DATABASE [Worry-freemanagement] SET READ_WRITE
GO

```

## 3.2 系统架构设计

MVC 模型无忧人力资源管理系统使用 MVC 设计模式进行设计，有利于把模型数据与视图层实现有效分离，设置较为灵活的用户页面，提高用户体验效果。

MVC 设计模式包括业务模型、用户界面、控制器三层,MVC 是一种使用 MVC(Model View Controller 模型-视图-控制器)设计创建 Web 应用程序的模式:Model(模型)表示应用程序核心(比如数据库记录列表)。View(视图)显示数据(数据库记录)。Controller(控制器)处理输入(写入数据库记录)。MVC 模式同时提供了对 HTML、CSS 和 JavaScript 的完全控制。Model(模型)是应用程序中用于处理应用程序数据逻辑的部分。通常模型对象负责在数据库中存取数据。View(视图)是应用程序中处理数据显示的部分。通常视图是依据模型数据创建的。Controller(控制器)是应用程序中处理用户交互的部分。通常控制器负责从视图读取数据,控制用户输入,并向模型发送数据。MVC 分层有助于管理复杂的应用程序,因为您可以在一个时间内专门关注一个方面。例如,您可以在不依赖业务逻辑的情况下专注于视图设计。同时也让应用程序的测试更加容易。MVC 分层同时也简化了分组开发。不同的开发人员可同时开发视图、控制器逻辑和业务逻辑。MVC 模型结构如图 3-4 所示:

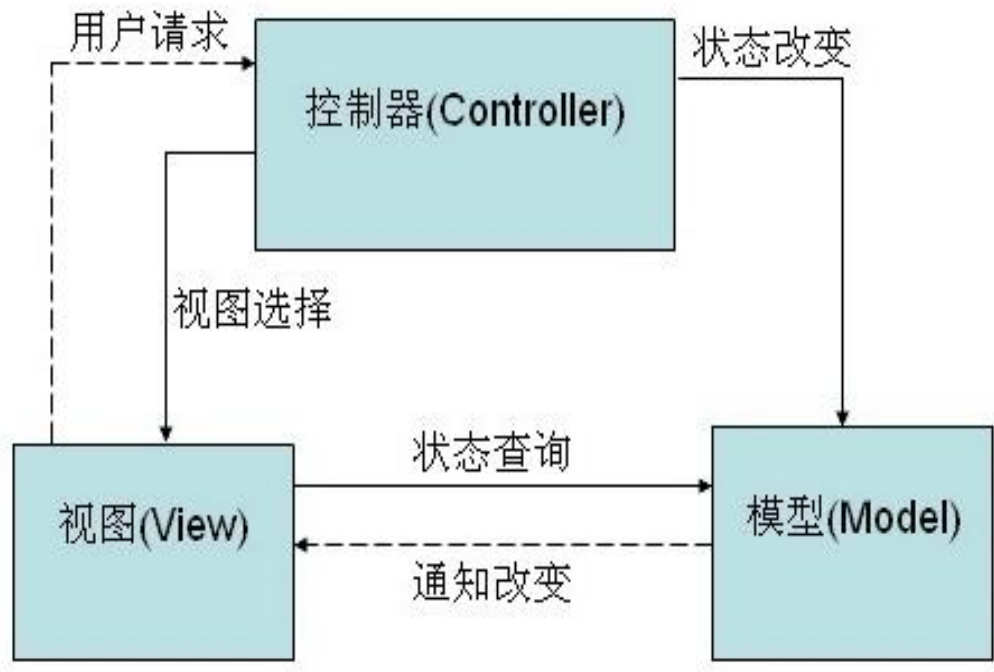


图 3-4 无忧人力资源管理系统架构设计图

### 3.3 模块设计

#### 3.3.1 部门管理信息类图

部门信息的类图如下图 3-5 所示。各类具体说明如下:

(1) LoginController 类: LoginController 类完成对部门信息的增删查改操作, 位于该项目目录 Worry\_freemanagement/Controllers/LoginController 处。方法包括

查看全体部门信息(Department())、查看部门信息(Delsit())、部门修改方法(Edit())、添加部门信息(Add())。

(2) Departmental 类: Departmental 类部门基本信息的类，位于该项目目录 Models/Departmental 处。属性包括部门名称 (DepartmentName)、部门编号 (DepartmentID)

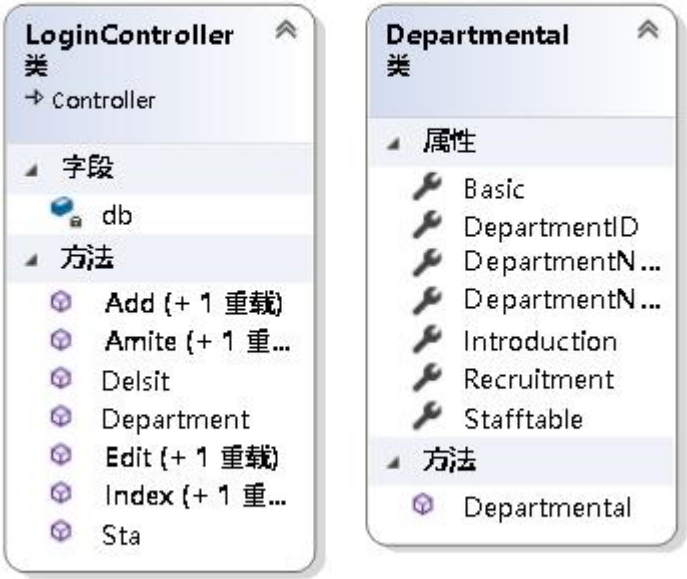


图 3-5 部门信息管理类图

### 3.3.2 部门管理信息操作流程

部门管理信息模块主要完成对部门基本信息的增删查改功能, 其操作流程如图 3-6 部门信息时序图所示:

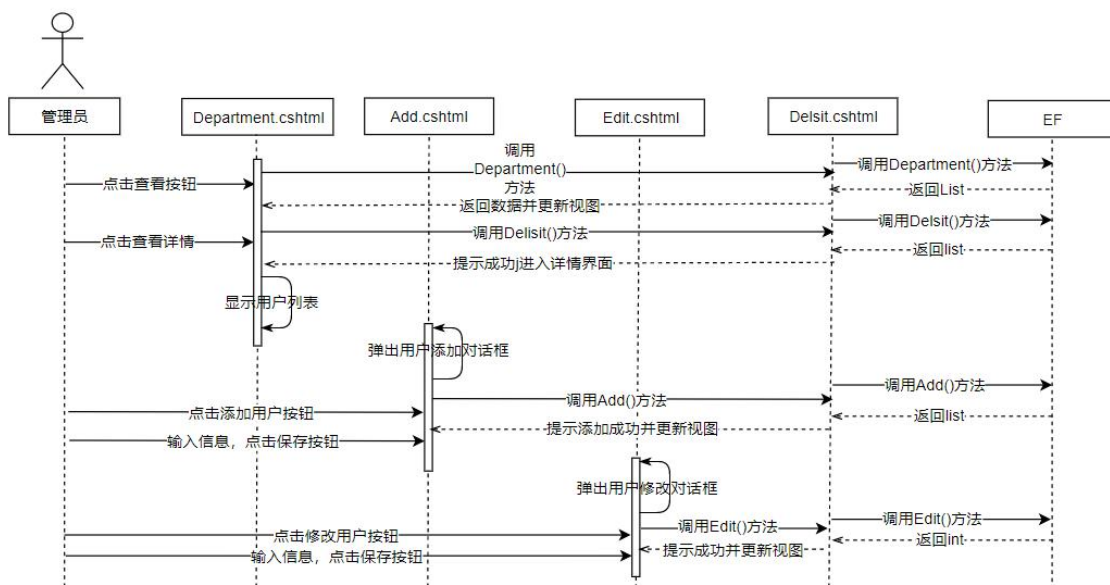


图 3-6 部门信息时序图

### 3.3.3 员工信息类图

员工信息的类图如下图 3-7 所示。各类具体说明如下：

(1) EmployeeController 类：EmployeeController 类完成对员工信息的增删查改操作，位于该项目目录 Worry\_freemanagement/Controllers/EmployeeController 处。方法包括员工查询方法（Index()）、查看员工详情方法（Delsit()）、员工添加方法（Add()）、员工工资方法（Updatecoach()）。

(2) Stafftable 类：Stafftable 类员工基本信息的类，位于该项目目录 Models/Stafftable 处。属性包括员工名称（Name）、员工编号（EmployeeID）、所在部门（DepartmentName）。

(3) Departmental 类：Departmental 类部门基本信息的类，位于该项目目录 Models/Departmental 处。属性包括部门名称（DepartmentName）、部门编号（DepartmentID）。

(3) wage 类：wage 类部门基本信息的类，位于该项目目录 Models/wage 处。属性包括员工名称（Name）、工资编号（DepartmentID）。

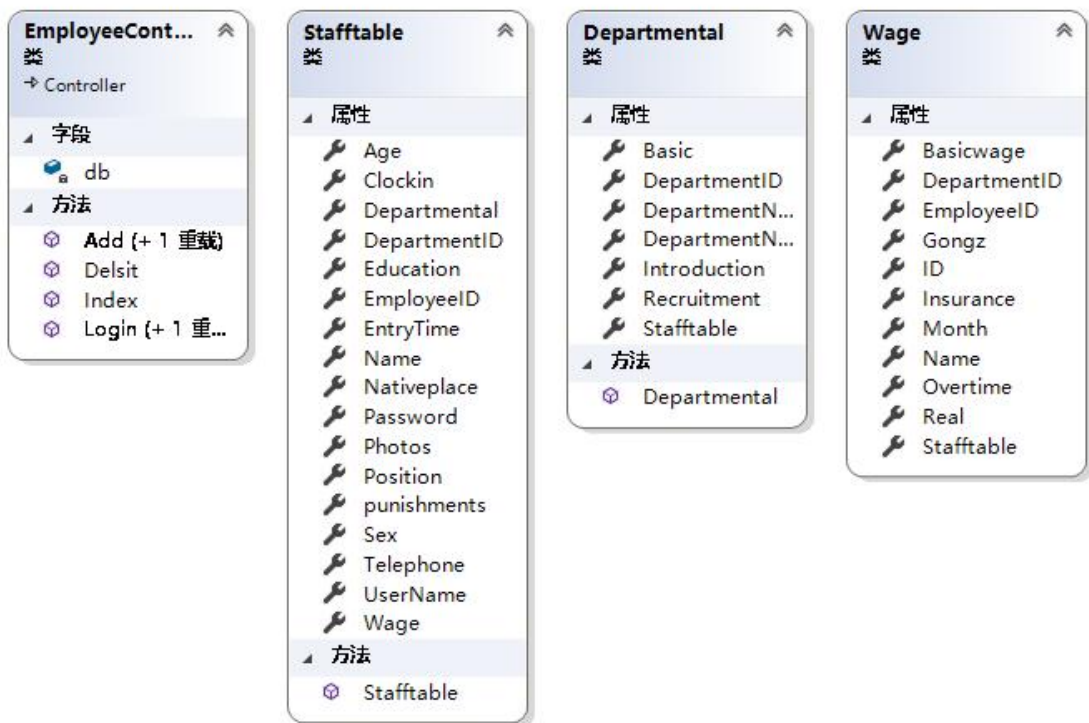


图 3-7 员工信息类图

### 3.3.4 员工信息操作流程

员工信息模块主要完成对员工基本信息的增删查改功能,其操作流程如图 3-8 员工信息时序图所示:

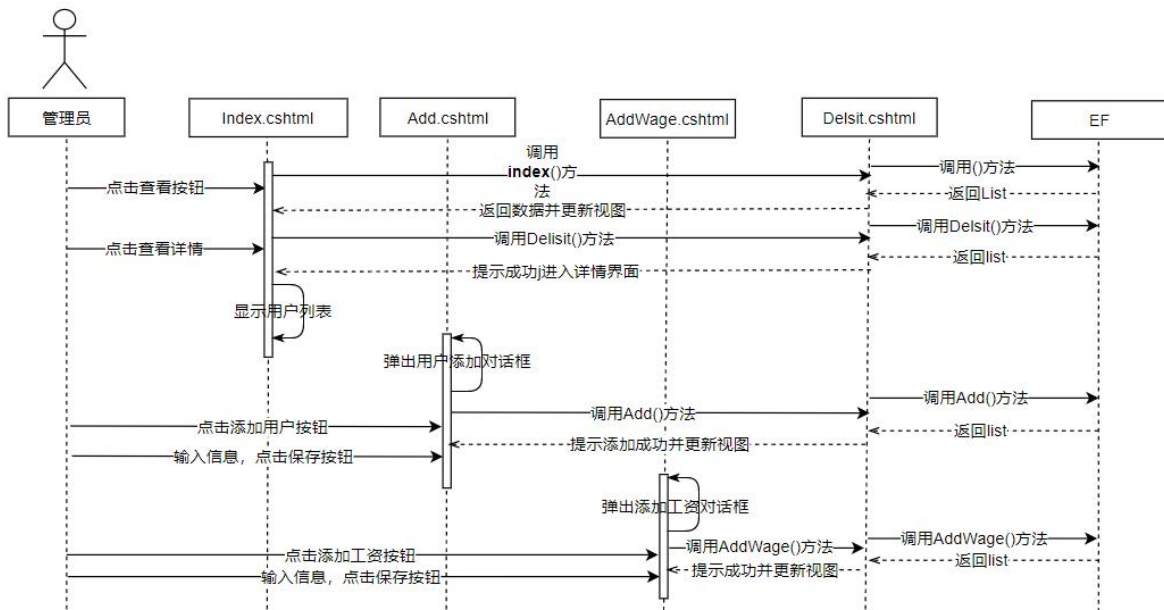


图 3-8 员工信息时序图



### 3.3.5 培训信息类图

员工信息的类图如下图 3-9 所示。各类具体说明如下：

(1) TrainingController 类：TrainingController 类完成对培训信息的增删查改操作，位于该项目目录 Worry\_freemanagement/Controllers/TrainingController 处。方法包括培训查询方法（Index()）、查看培训详情方法（Delsit()）、培训添加方法（Add()）、培训信息修改方法（alter()）。

(2) Stafftable 类：Stafftable 类员工基本信息的类，位于该项目目录 Models/Stafftable 处。属性包括员工名称（Name）、员工编号（EmployeeID）、所在部门（DepartmentName）。

(3) Trainingform 类：Trainingform 类部门基本信息的类，位于该项目目录 Models/Trainingform 处。属性包括培训名称（Nametraining）、培训编号（TrainingID）。

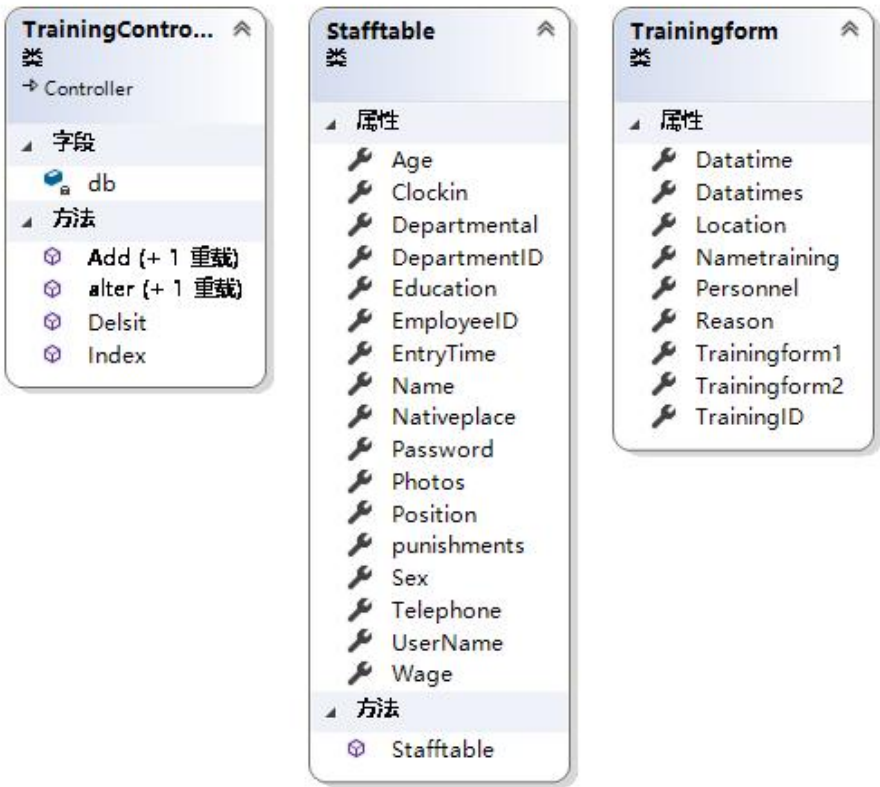


图 3-9 培训信息类图

### 3.3.6 培训信息操作流程

培训信息模块主要完成对员工基本信息的增删查改功能,其操作流程如图 3-10 培训信息时序图所示:

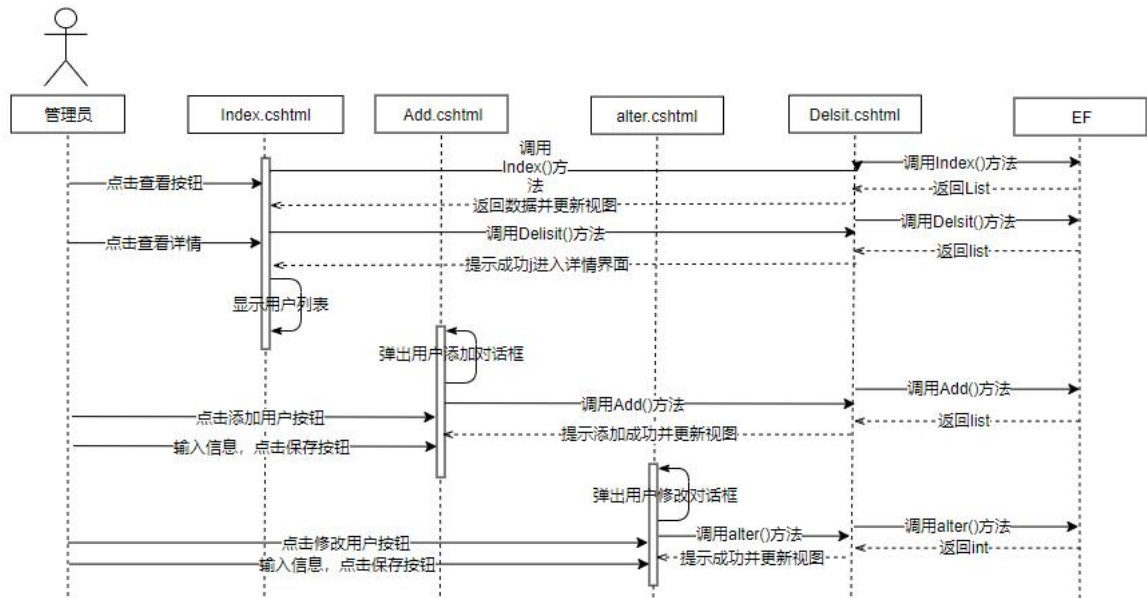


图 3-10 培训信息时序图

### 3.3.7 招聘信息类图

招聘信息的类图如下图 3-11 所示。各类具体说明如下:

(1) RecruitmentController 类:RecruitmentController 类完成对招聘信息的增删查改操作, 位于该项目目录

Worry\_freemanagement/Controllers/RecruitmentController 处。方法包括招聘查询方法 (Index())、查看招聘详情方法 (Delsit())、招聘添加方法 (Add())、录用员工方法 (Hire ())。

(2) Stafftable 类: Stafftable 类员工基本信息的类, 位于该项目目录 Models/Stafftable 处。属性包括员工名称 (Name)、员工编号 (EmployeeID)、所在部门 (DepartmentName)。

(3) Recruitment 类: Recruitment 类部门基本信息的类, 位于该项目目录 Models/Recruitment 处。属性包括招聘名称 (Name)、招聘编号 (RecruitmentID)。

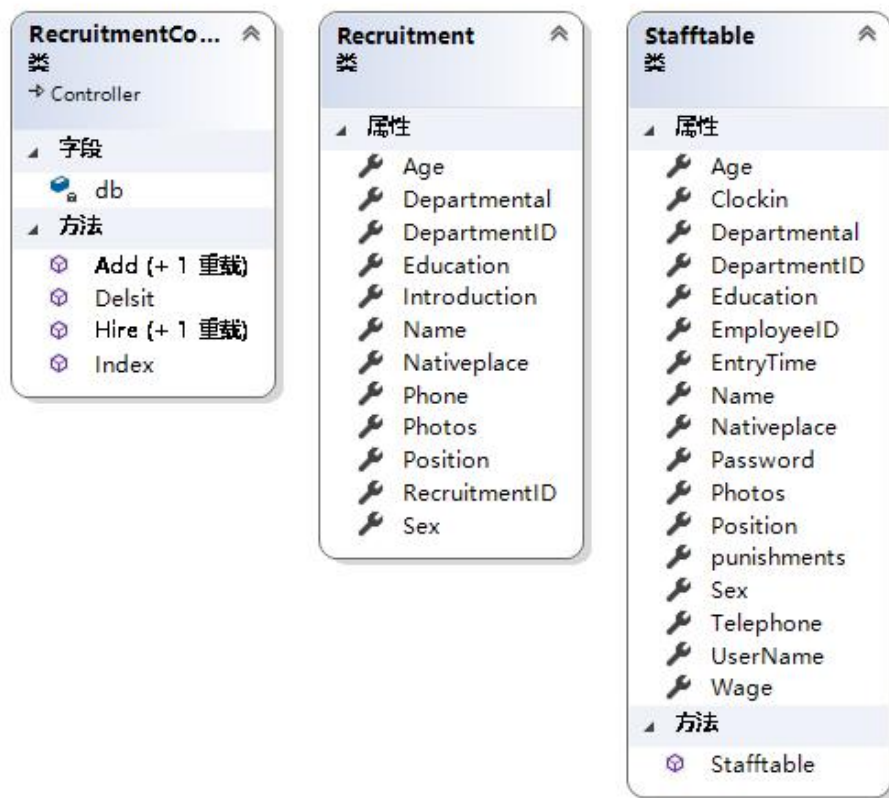


图 3-11 招聘信息类图

### 3.3.8 招聘信息操作流程

招聘信息模块主要完成对员工基本信息的增删查改功能,其操作流程如图 3-12 招聘信息时序图所示:

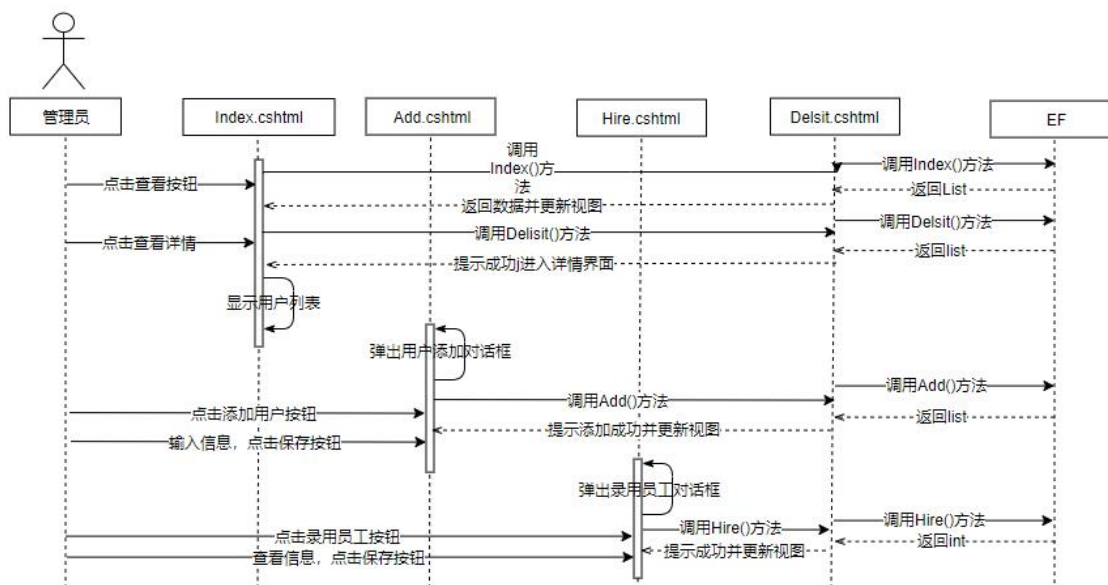


图 3-12 招聘信息时序图

### 3.3.9 奖罚信息类图

奖罚信息的类图如下图 3-13 所示。各类具体说明如下：

(1) punishmentsController 类: punishmentsController 类完成对招聘信息的增删查改操作，位于该项目目录

Worry\_freemanagement/Controllers/punishmentsController 处。方法包括招聘查询方法 (Index())、查看奖罚详情方法 (Delsit())、奖罚信息添加方法 (Add())、员工考勤查看 (Clioic())。

(2) Stafftable 类: Stafftable 类员工基本信息的类，位于该项目目录 Models/Stafftable 处。属性包括员工名称 (Name)、员工编号 (EmployeeID)、所在部门 (DepartmentName)。

(3) punishments 类: punishments 类部门基本信息的类，位于该项目目录 Models/punishments 处。属性包括姓名 (Name)、奖罚编号 (EventsID)。

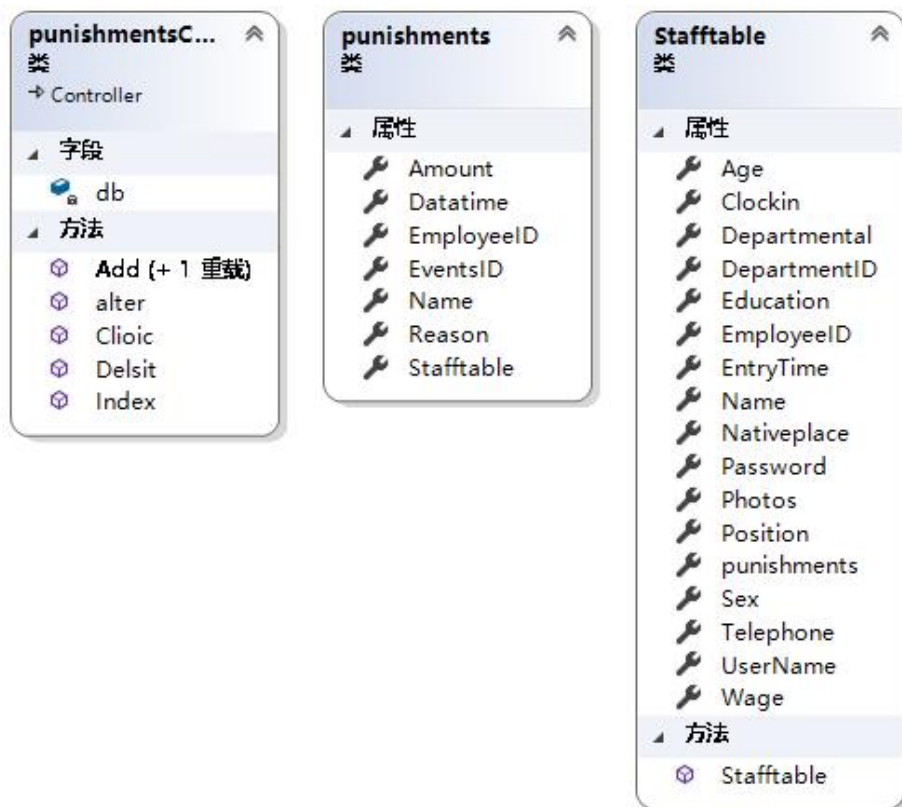


图 3-13 奖罚信息类图

### 3.3.10 奖罚信息操作流程

奖罚信息模块主要完成对员工奖罚信息的增删查改功能, 其操作流程如图 3-14 奖罚信息时序图所示:

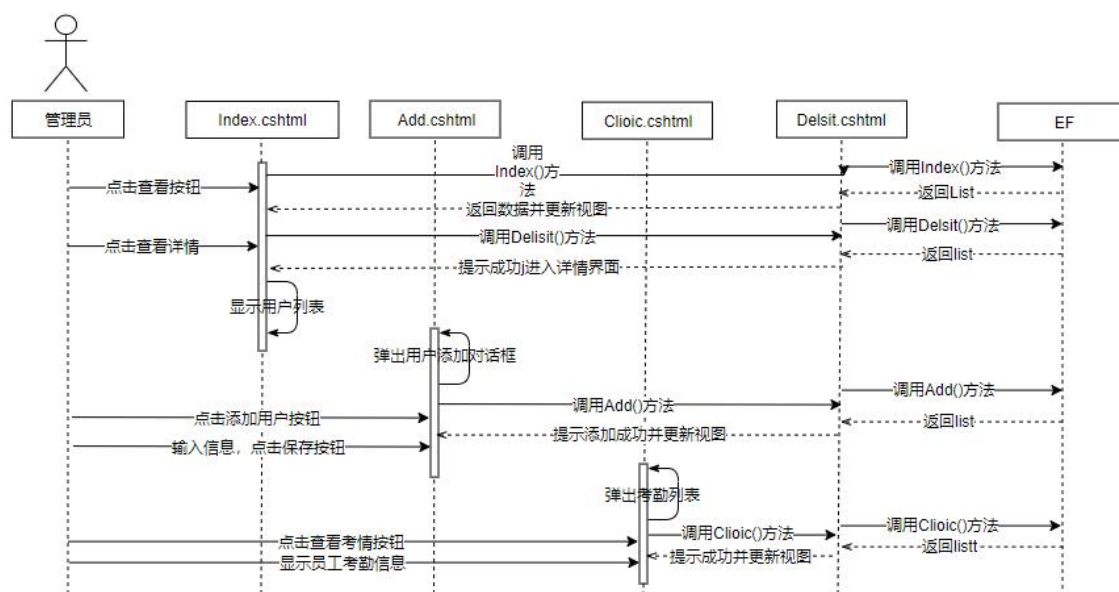


图 3-14 招聘信息时序图

### 3.3.11 公告信息类图

公告信息的类图如下图 3-15 所示。各类具体说明如下：

(1) noticeController 类:noticeController 类完成对公告信息的增删查改操作，位于该项目目录 Worry\_freemanagement/Controllers/noticeController 处。方法包括招聘查询方法（select()）、查看奖罚详情方法（Content()）、奖罚信息添加方法（Index()）、删除公司信息(Delete())。

(2) notice 类: notice 类员工基本信息的类，位于该项目目录 Models/notice 处。属性包括公告名称（name）、公告编号(id)、公告类容(Content)。

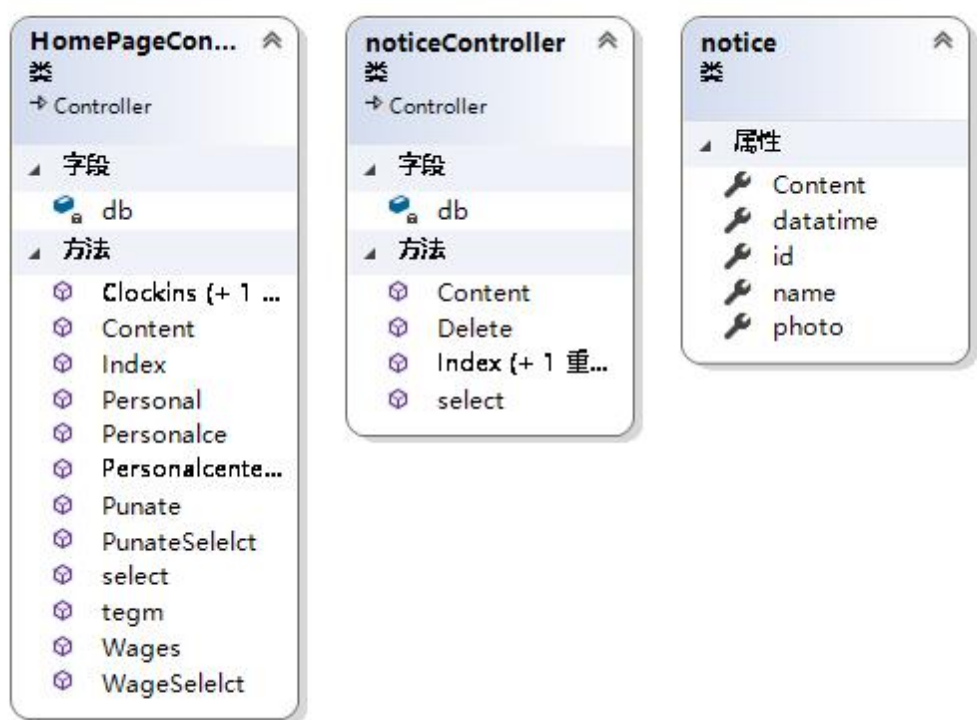


图 3-15 公告信息类图

### 3.3.12 公告信息操作流程

公告信息模块主要完成对公司公告信息的增删查改功能，其操作流程如图 3-16 奖罚信息时序图所示：

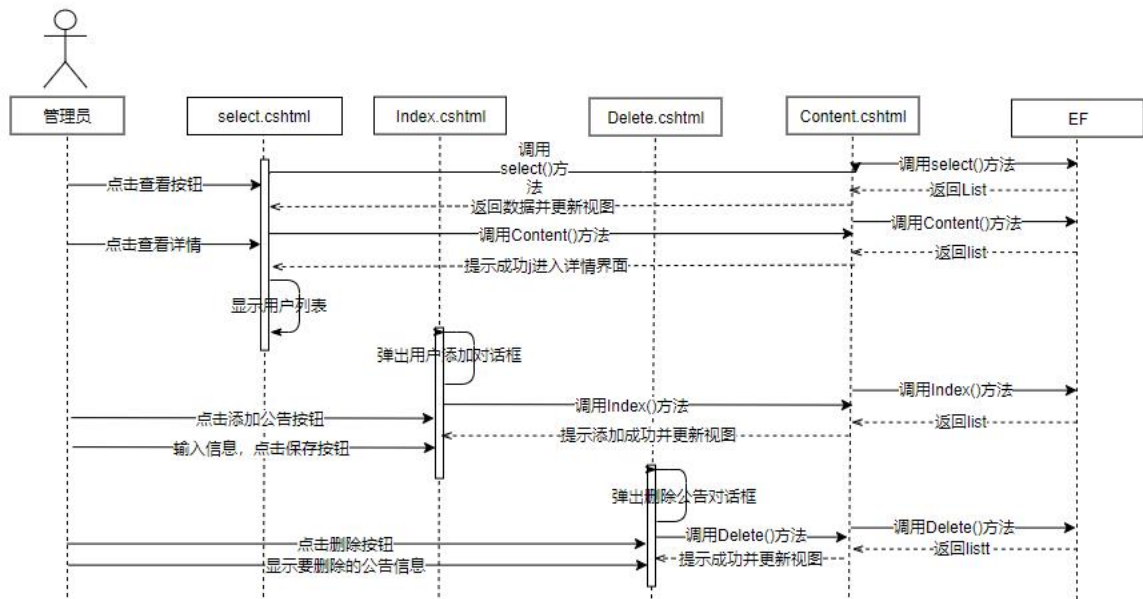


图 3-16 公告信息时序图



# 第四章 产品实现

## 4.1 部门管理模块实现

部门管理模块包括部门信息展示、新增部门、修改部门信息、查看部门详情、搜索框（部门名称）模糊查询功能。

### 4.1.1 界面设计

部门信息列表界面如图 4-1 所示, 点击修改按钮跳转到部门信息修改页面，部门信息修改界面如图 4-3 所示，点击查看部门按钮查看对应的部门信息，部门信息添加界面如图 4-2 所示，添加成功后弹出返回发哦相应的界面：

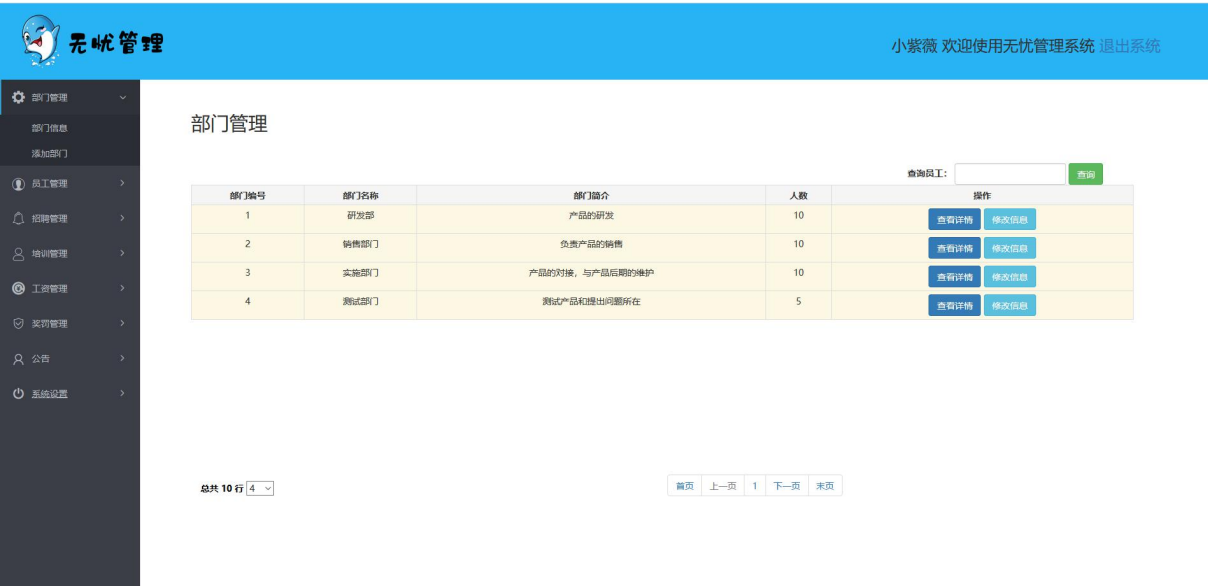


图 4-1 部门信息列表界面图

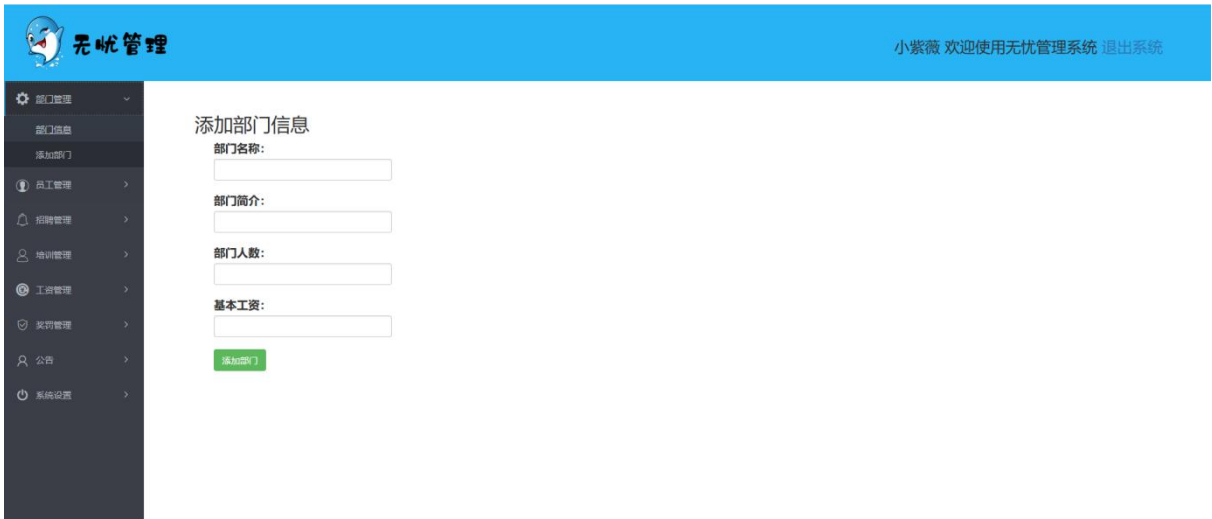


图 4-2 部门信息添加界面图

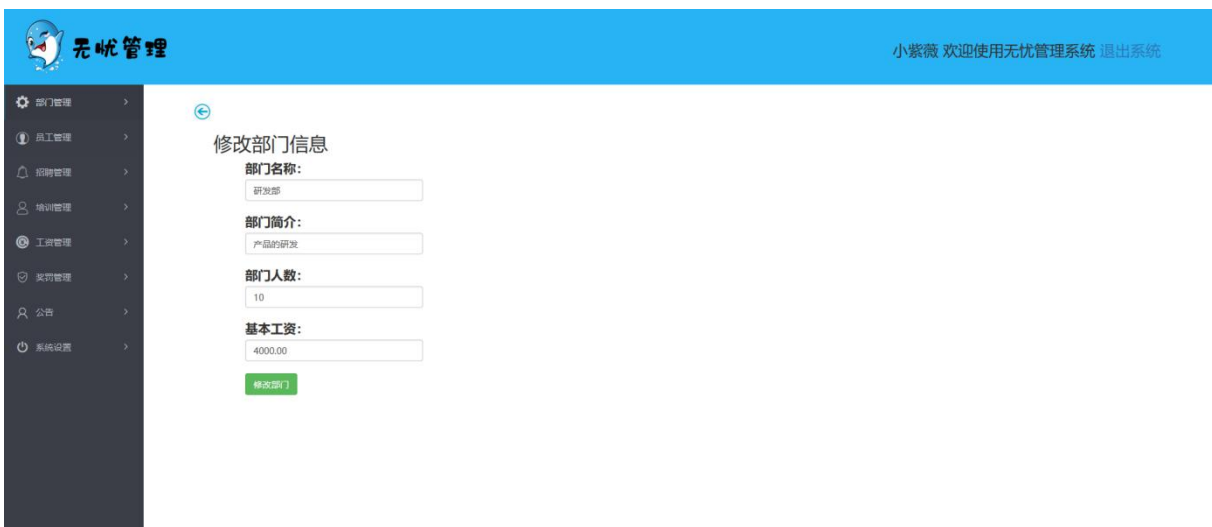


图 4-3 部门信息修改界面图

## 4.1.2 功能实现

部门信息模块的主要代码

```

/// <summary>
/// 查看所有部门
/// </summary>
/// <param name="Name"></param>
/// <param name="pageIndex"></param>
/// <param name="pageCount"></param>
/// <returns></returns>
0 个引用
public ActionResult Department(string Name = "", int pageIndex = 1, int pageCount = 4)
{
    //List<Departmental> list = db.Departmental.ToList();
    //return View(list);
    int totalCount = db.Departmental.OrderBy(p => p.DepartmentID).Where(p => p.DepartmentName.Contains(Name) || Name == "").Count();
    //总页数
    double totalPage = Math.Ceiling((double)totalCount / pageCount);
    //获得用户集合，分页查询Skip() 跳过指定数量的集合 Take() 从过滤后返回的集合中再从第一行取出指定的行数
    List<Departmental> tad = db.Departmental.OrderBy(p => p.DepartmentID).Where(p => p.DepartmentName.Contains(Name) || Name == "")
        .ToList().Skip((pageIndex - 1) * pageCount).Take(pageCount).ToList();
    ViewBag.pageIndex = pageIndex;
    ViewBag.pageCount = pageCount;
    ViewBag.totalCount = totalCount;
    ViewBag.totalPage = totalPage;
    ViewBag.Name = Name;
    return View(tad);
}
/// <summary>
/// 查看部门信息
/// </summary>
/// <param name="id"></param>
/// <returns></returns>
0 个引用
public ActionResult Delsit(int id)
{
    Departmental dep = db.Departmental.Find(id);
    int sum = db.Stafftable.Where(p=>p.DepartmentID==id).Count();
    ViewBag.sum = sum;
    return View(dep);
}

```

图 4-4 部门管理信息主要代码图

## 4.2 员工信息模块实现

员工信息模块包括员工信息展示、员工信息的添加、查看员工信息、添加员工工资，搜索框（员工姓名）模糊查询功能。

### 4.2.1 界面设计

员工信息列表界面如图 4-5 所示，点击添加按钮跳转到员工信息添加页面，员工信息添加界面如图 4-6 所示，点击查看详情按钮查看对应的员工信息，员工工资添加界面如图 4-7 所示，添加成功后进入工资管理页面：

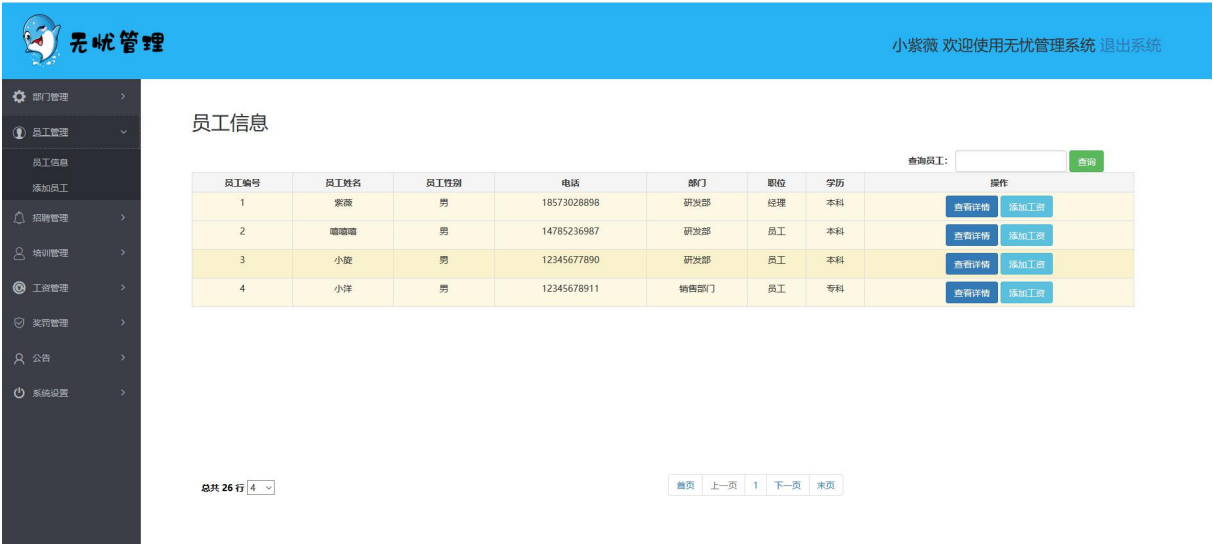


图 4-5 员工信息列表界面图



图 4-6 员工信息添加界面图



图 4-7 员工工资添加界面图

## 4.2.2 功能实现

员工信息模块的主要代码

```

/// <summary>
/// 添加员工工资
/// </summary>
/// <returns></returns>
0 个引用
public ActionResult AddWage(int id)
{
    Stafftable ste = db.Stafftable.Find(id);
    return View(ste);
}

[HttpPost]
0 个引用
public ActionResult AddWage(Wage wa, string Basicwage, string Gongz, string Overtime, string Insurance)
{
    Wage wag1 = db.Wage.OrderByDescending(p => p.ID).FirstOrDefault(p => p.EmployeeID == wa.EmployeeID);
    if (wag1 == null)
    {
        wa.Real = decimal.Parse(Basicwage) + decimal.Parse(Gongz) + decimal.Parse(Overtime) + decimal.Parse(Insurance);
        db.Wage.Add(wa);
        db.SaveChanges(); return RedirectToAction("Index", "Wages");
    }
    else
    {
        string colTime = string.Format("{0:Y}", wa.Month);
        string nawTime = string.Format("{0:Y}", wag1.Month);
        if (colTime == nawTime)
        {
            return Content("<script>alert(' 本月工资已成功发放! ');history.go(-1)</script>");
        }
        else
        {
            wa.Real = decimal.Parse(Basicwage) + decimal.Parse(Gongz) + decimal.Parse(Overtime) + decimal.Parse(Insurance);
            db.Wage.Add(wa);
            db.SaveChanges(); return RedirectToAction("Index", "Wages");
        }
    }
}
}

```

图 4-8 员工信息模块主要代码图

```

/// <summary>
/// 添加员工
/// </summary>
/// <returns></returns>
0 个引用
public ActionResult Add()
{
    var de = db.Departmental.ToList();
    ViewBag.de = de;
    return View();
}

[HttpPost]
0 个引用
public ActionResult Add(Stafftable stafftable, HttpPostedFileBase file)
{
    if (ModelState.IsValid)
    {
        if (file != null)
        {
            string fileName = Path.GetFileName(file.FileName);
            //2. 判断文件是否是图片
            //string hz = Path.GetExtension(fileName);
            if (fileName.EndsWith(".jpg") || fileName.EndsWith(".png") || fileName.EndsWith(".jpeg") || fileName.EndsWith(".gif"))
            {
                //3. 保存图片到项目文件夹当中
                file.SaveAs(Server.MapPath("~/image/" + fileName));
                //4. 将图片文件名，绑定到该用户的photo字段中
                stafftable.Photos = fileName;
            }
        }
        //通过注册时间填写的账号去数据库中查找账号是否存在
        Stafftable st = db.Stafftable.Where(p => p.Name == stafftable.Name).FirstOrDefault();
        if (st == null)
        {
            stafftable.EntryTime = System.DateTime.Now;
            db.Stafftable.Add(stafftable);
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        else
        {
            //ModelState.AddModelError("", "该账号已存在");
            var de = db.Departmental.ToList();
            ViewBag.de = de;
            return Content("<script>alert(' 该员工已存在');window.open(' " + Url.Content("~/Employee/Index") + "', '_self')</scri");
        }
    }
    else
    {
        return View();
    }
}
}

```

图 4-9 员工信息模块主要代码图

## 4.3 招聘信息模块实现

招聘信息模块包括招聘信息展示、招聘信息添加、查看招聘信息、录取应聘者，搜索框（应聘者姓名）模糊查询功能。

### 4.3.1 界面设计

招聘信息列表界面如图 4-10 所示, 点击添加按钮跳转到招聘信息添加页面, 招聘信息添加界面如图 4-11 所示, 点击查看详情按钮查看对应的招聘信息, 招聘录取界面如图 4-12 所示, 点击录取后进入员工管理页面:

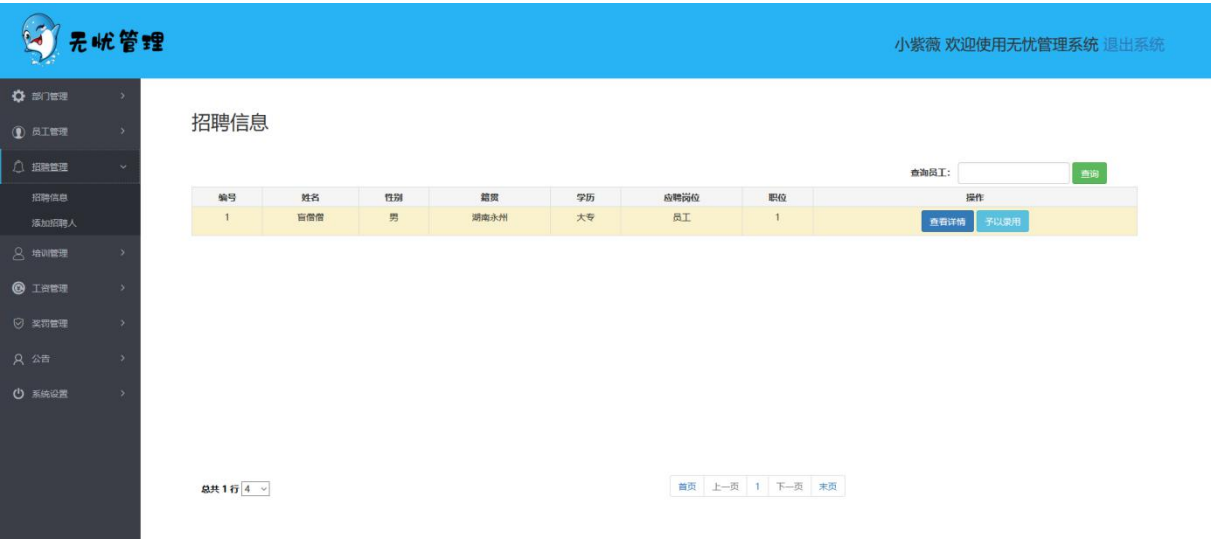


图 4-10 招聘信息列表界面图



图 4-11 招聘信息添加界面图

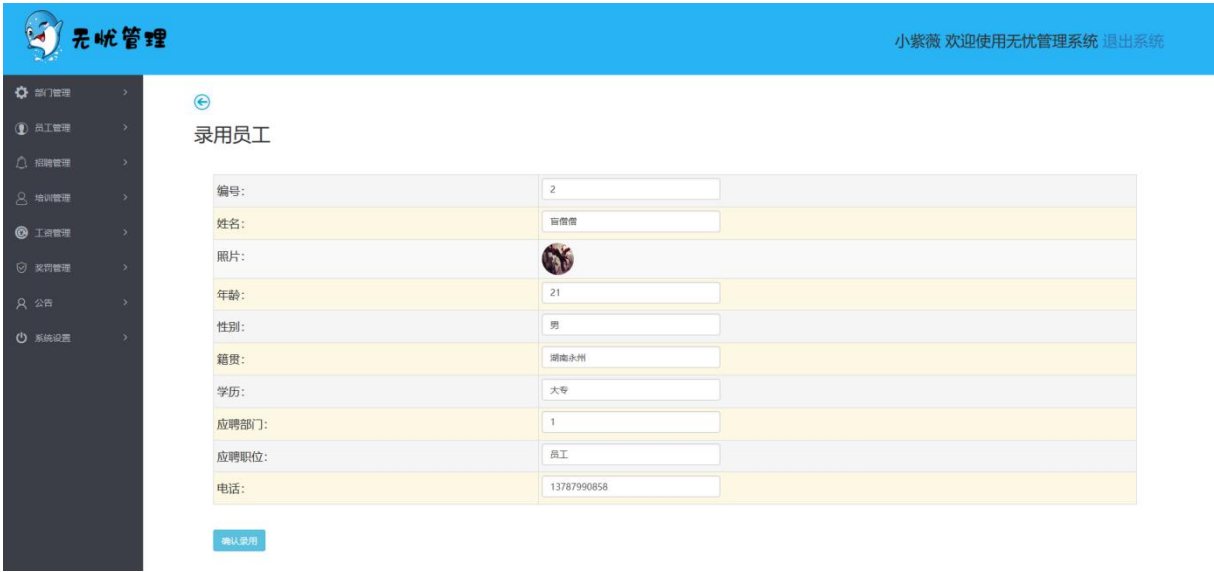


图 4-12 录用应聘者添加界面图

### 4.3.2 功能实现

员工信息模块的主要代码

```
/// <summary>
/// 添加新的应聘者
/// </summary>
/// <param name="id"></param>
/// <returns></returns>
0 个引用
public ActionResult Add()
{
    var de = db.Departmental.ToList();
    ViewBag.de = de;
    return View();
}
[HttpPost]
0 个引用
public ActionResult Add(Recruitment rec, HttpPostedFileBase file)
{
    if (file != null)
    {
        string fileName = Path.GetFileName(file.FileName);
        //2. 判断文件是否是图片
        //string hz = Path.GetExtension(fileName);
        if (fileName.EndsWith(".jpg") || fileName.EndsWith(".png") || fileName.EndsWith(".jpeg") || fileName.EndsWith(".gif"))
        {
            //3. 保存图片到项目文件夹当中
            file.SaveAs(Server.MapPath("~/image/" + fileName));
            //4. 将图片文件名, 绑定到该用户的photo字段中
            rec.Photos = fileName;
        }
    }
    if (ModelState.IsValid)
    {
        db.Recruitment.Add(rec);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    else
    {
        return View();
    }
}
```

图 4-13 员工信息模块主要代码图

```

/// <summary>
/// 录用员工
/// </summary>
/// <param name="id"></param>
/// <returns></returns>
0 个引用
public ActionResult Hire(int id)
{
    Recruitment er = db.Recruitment.Find(id);

    return View(er);
}
[HttpPost]
0 个引用
public ActionResult Hire(Stafftable sta, HttpPostedFileBase file, int id)
{
    if (file != null)
    {
        string fileName = Path.GetFileName(file.FileName);
        //2. 判断文件是否是图片
        //string hz = Path.GetExtension(fileName);
        if (fileName.EndsWith(".jpg") || fileName.EndsWith(".png") || fileName.EndsWith(".jpeg") || fileName.EndsWith(".gif"))
        {
            //3. 保存图片到项目文件夹当中
            file.SaveAs(Server.MapPath("~/image" + fileName));
            //4. 将图片文件名, 绑定到该用户的photo字段中
            sta.Photos = fileName;
        }
    }
    sta.EntryTime = System.DateTime.Now;
    sta.Password = "123456";
    sta.UserName = "aa";
    db.Configuration.ValidateOnSaveEnabled = false;
    db.Stafftable.Add(sta);
    db.SaveChanges();
    db.Configuration.ValidateOnSaveEnabled = true;
    db.Recruitment.Remove(db.Recruitment.Find(id));
    db.SaveChanges();
    return RedirectToAction("Index", "Employee");
}

```

图 4-14 员工信息模块主要代码图

## 4.4 培训信息模块实现

培训信息模块包括培训信息展示、培训信息添加、查看培训信息、修改培训信息。

### 4.4.1 界面设计

培训信息列表界面如图 4-15 所示, 点击添加按钮跳转到培训信息添加页面, 培训信息添加界面如图 4-16 所示, 点击查看详情按钮查看对应的培训信息, 培训信息修改界面如图 4-17 所示, 点击修改按钮进入培训信息修改界面:



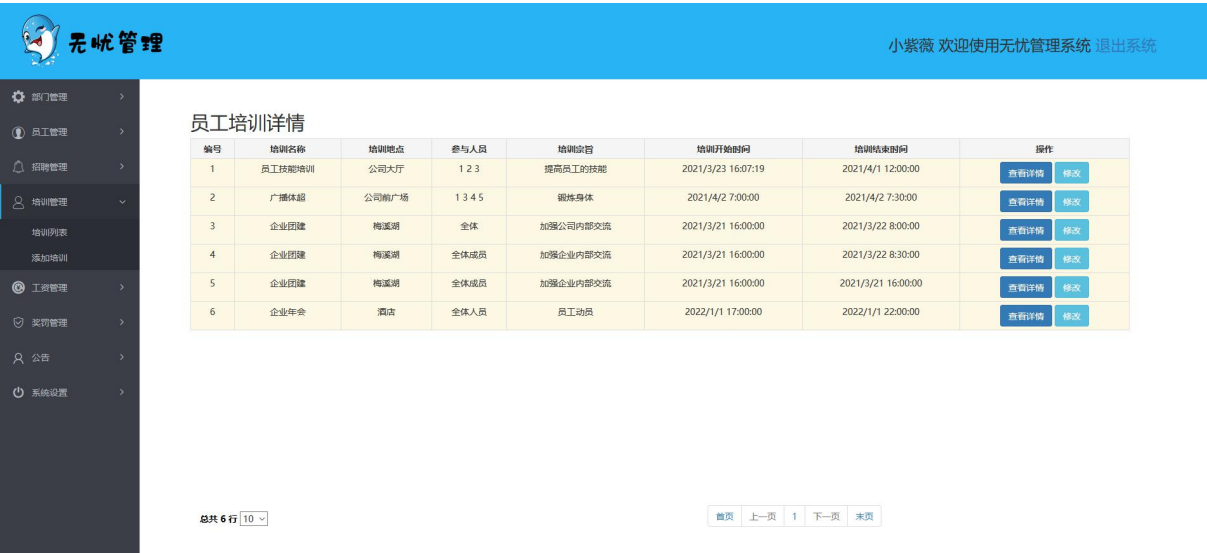


图 4-15 培训信息列表界面图

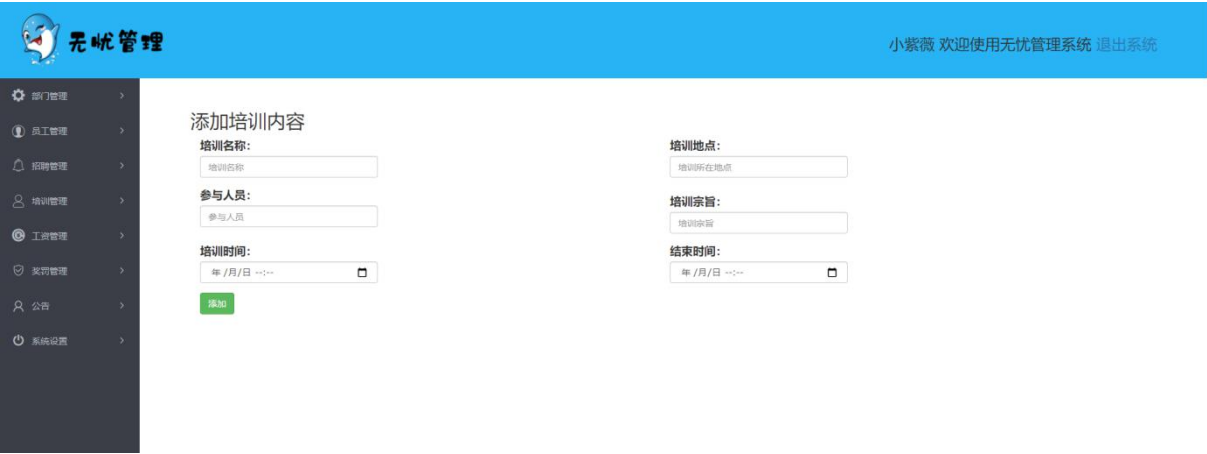


图 4-16 培训信息添加界面图

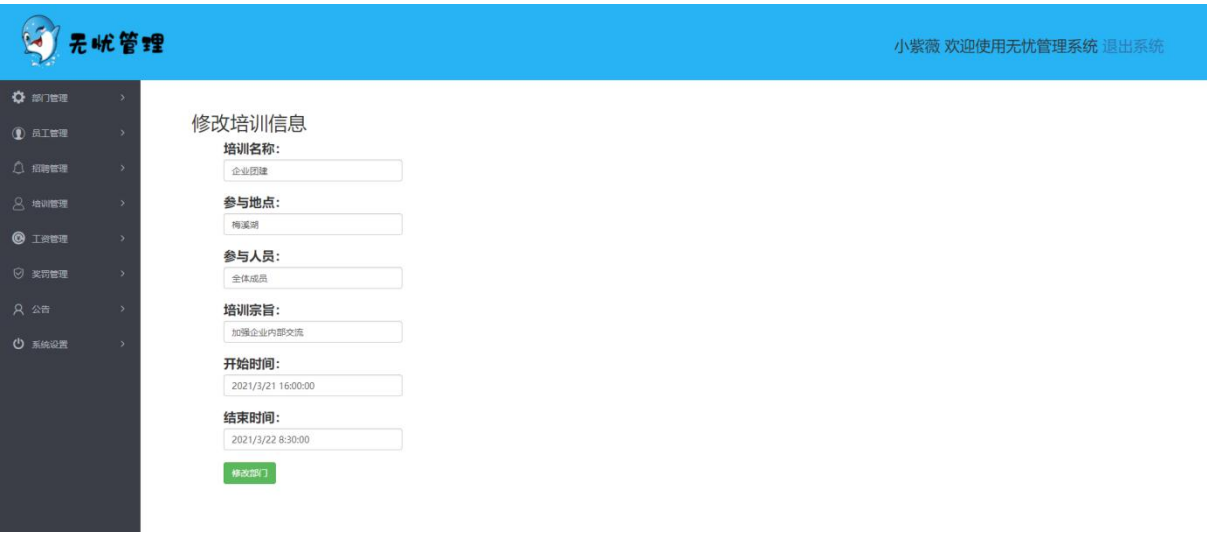


图 4-17 培训信息修改界面图

## 4.4.2 功能实现

培训信息模块增删查改的主要代码

```
Entities1 db = new Entities1();
// GET: Training
/// <summary>
/// 员工培训列表
/// </summary>
/// <returns></returns>
0 个引用
public ActionResult Index(string Name = "", int pageIndex = 1, int pageCount = 4)
{
    //List<Trainingform> list = db.Trainingform.ToList();
    //return View(list);
    int totalCount = db.Trainingform.OrderBy(p => p.TrainingID).Count();
    //总页数
    double totalPage = Math.Ceiling((double)totalCount / pageCount);
    //获得用户集合，分页查询Skip()跳过指定数量的集合 Take() 从过滤后返回的集合中再从第一行取出指定的行数
    List<Trainingform> tra = db.Trainingform.OrderBy(p => p.TrainingID).ToList().Skip((pageIndex - 1) * pageCount).Take(pageCount).ToList();
    ViewBag.pageIndex = pageIndex;
    ViewBag.pageCount = pageCount;
    ViewBag.totalCount = totalCount;
    ViewBag.totalPage = totalPage;
    return View(tra);
}
/// <summary>
/// 查看培训详情
/// </summary>
/// <returns></returns>
0 个引用
public ActionResult Delsit(int id)
{
    Trainingform er = db.Trainingform.Find(id);
    return View(er);
}
```

图 4-18 培训信息模块主要代码图

## 4.5 奖罚信息模块实现

奖罚信息模块包括奖罚信息展示、奖罚信息添加、查看奖罚信息、查看员工考情状况信息。

### 4.5.1 界面设计

奖罚信息列表界面如图 4-19 所示, 点击添加按钮跳转到奖罚信息添加页面, 奖罚信息添加界面如图 4-20 所示, 点击查看详情按钮查看对应的奖罚信息, 员工考勤状况如图 4-21 所示:

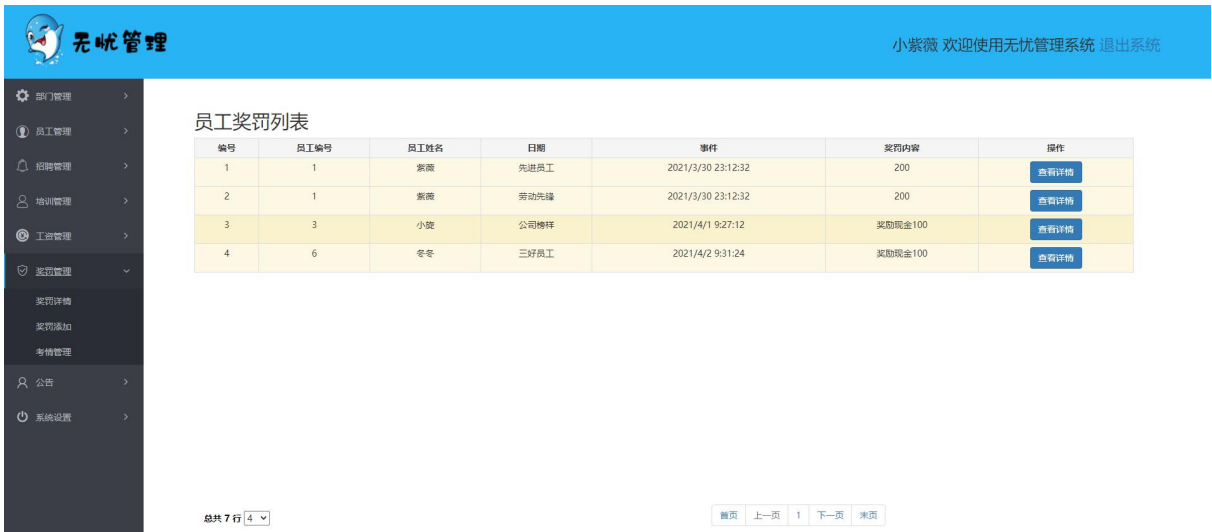


图 4-19 奖罚信息列表界面图

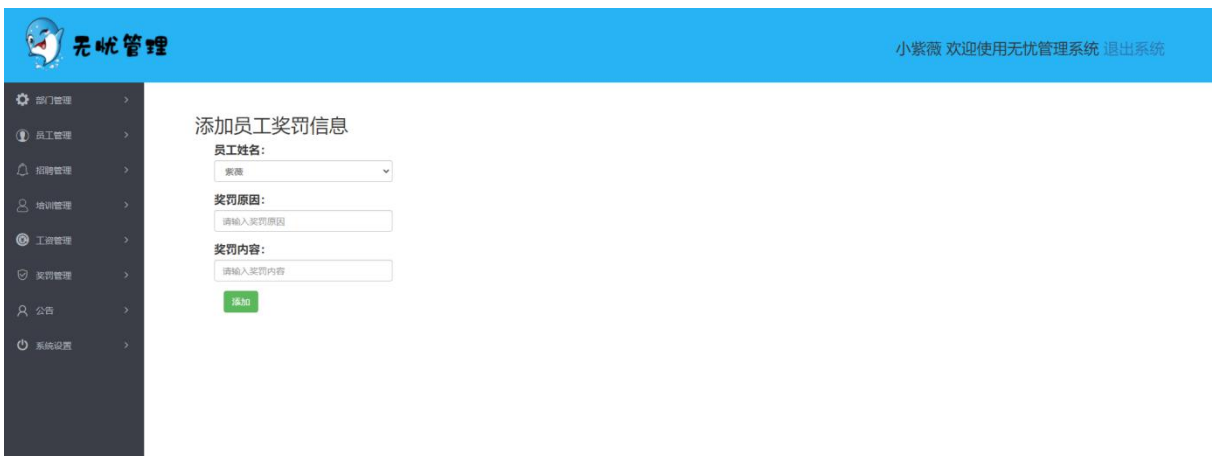


图 4-20 奖罚信息添加界面图

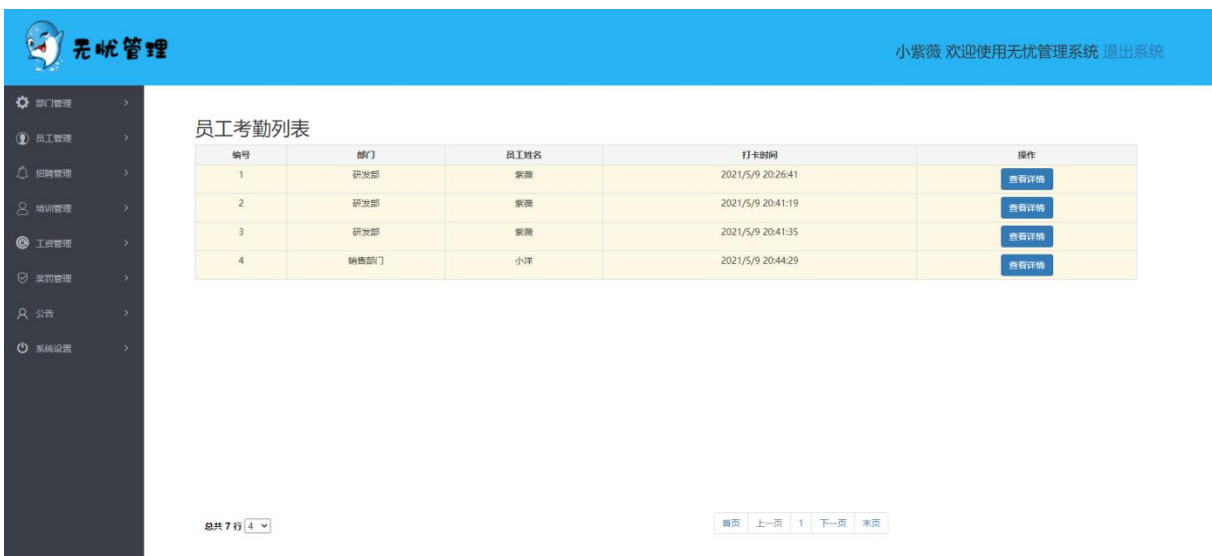


图 4-21 考勤信息列表界面图

## 4.5.2 功能实现

奖罚信息模块增删查改的主要代码

```
//List<punishments> list = db.punishments.ToList();
//return View(list);
int totalCount = db.punishments.OrderBy(p => p.EventsID).Count();
//总页数
double totalPages = Math.Ceiling((double)totalCount / pageCount);
//获得用户集合，分页查询Skip(<)跳过指定数量的集合 Take() 从过滤后返回的集合中再从第一行取出指定的行数
List<punishments> sta = db.punishments.OrderBy(p => p.EventsID).ToList().Skip((pageIndex - 1) * pageCount).Take(pageCount).ToList();
ViewBag.pageIndex = pageIndex;
ViewBag.pageCount = pageCount;
ViewBag.totalCount = totalCount;
ViewBag.totalPage = totalPages;
ViewBag.Name = Name;
return View(sta);
}
/// <summary>
/// 查看奖罚信息
/// </summary>
/// <returns></returns>
0 个引用
public ActionResult Delsit(int id)
{
    punishments pum = db.punishments.Find(id);
    return View(pum);
}
//查看员工考勤情况
0 个引用
public ActionResult Clcioic(string Name = "", int pageIndex = 1, int pageCount = 4)
{
    int totalCount = db.punishments.OrderBy(p => p.EventsID).Count();
    //总页数
    double totalPages = Math.Ceiling((double)totalCount / pageCount);
    //获得用户集合，分页查询Skip(<)跳过指定数量的集合 Take() 从过滤后返回的集合中再从第一行取出指定的行数
    List<Clockin> clo = db.Clockin.OrderBy(p => p.Clockinid).ToList().Skip((pageIndex - 1) * pageCount).Take(pageCount).ToList();
    ViewBag.pageIndex = pageIndex;
    ViewBag.pageCount = pageCount;
    ViewBag.totalCount = totalCount;
    ViewBag.totalPage = totalPages;
    ViewBag.Name = Name;
    return View(clo);
}
```

图 4-22 奖罚信息模块主要代码图

## 4.6 公告信息模块实现

公告信息模块包括奖罚信息展示、公告信息添加、查看公告信息、删除公告信息。

### 4.6.1 界面设计

公告信息列表界面如图 4-23 所示, 点击添加按钮跳转到公告信息添加页面, 公告信息添加界面如图 4-24 所示, 点击标题信息查看对应的公告信息, 如图 4-25 所示, 点击删除按钮, 可以删除对应的公告信息如图 4-26:



图 4-23 公告信息列表界面图

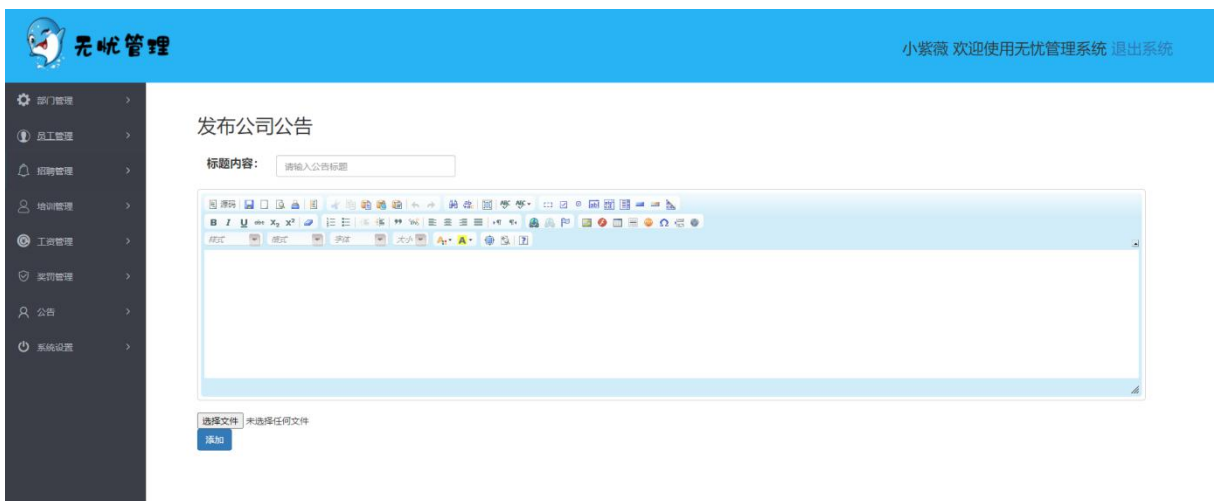


图 4-24 公告信息添加界面图



图 4-25 查看公告内容

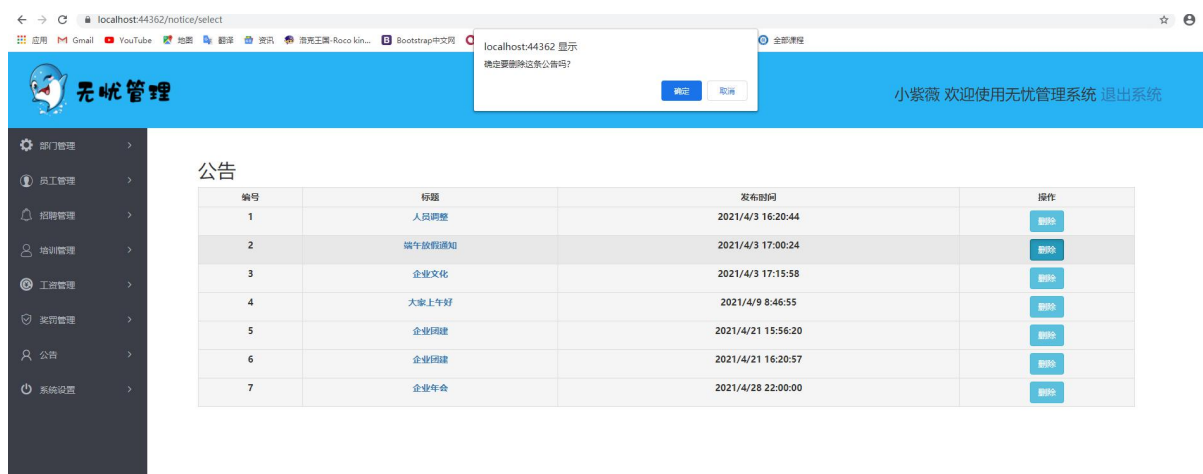


图 4-26 删除公告内容

## 4.6.2 功能实现

公告信息模块增删查改的主要代码

```
// GET: notice
/// <summary>
/// 添加公告
/// </summary>
/// <returns></returns>
0 个引用
public ActionResult Index()
{
    return View();
}
[ValidateInput(false)]
[HttpPost]
0 个引用
public ActionResult Index(notice not, HttpPostedFileBase file)
{
    if (ModelState.IsValid)
    {
        if (file != null)
        {
            string fileName = Path.GetFileName(file.FileName);
            //2. 判断文件是否是图片
            //string hz = Path.GetExtension(fileName);
            if (fileName.EndsWith(".jpg") || fileName.EndsWith(".png") || fileName.EndsWith(".jpeg") || fileName.EndsWith(".gif"))
            {
                //3. 保存图片到项目文件夹当中
                file.SaveAs(Server.MapPath("~/image/" + fileName));
                //4. 将图片文件名, 绑定到该用户的photo字段中
                not.photo = fileName;
            }
        }
        not.datetime = System.DateTime.Now;
        db.notice.Add(not);
        db.SaveChanges();
        return RedirectToAction("select", "notice");
    }
    else
    {
        return View();
    }
}
```

图 4-27 公告信息模块主要代码图

# 第五章 产品测试

## 5.1 测试方法与策略

本项目在所有功能编码完成且系统能完整运行的状态下进行系统测试，以确认各项需求的功能是否能正常运行，并给出测试结果。

本系统采用人工测试的方式进行黑盒测试对系统功能进行验证，使用到的测试用例设计方法包括：边界值分析、等价类划分、错误猜测法、因果图法等。

本项目的测试策略包括：

### 1. 功能测试

功能测试验证本系统能按照设计要求实现模块的各个功能，数据应完整、界面美观、操作方便。

### 2. 数据项测试

- 输入字母和数字等数据项是否能够正确显示，并添加到系统中；
- 鼠标点击、图形模式的数据项（如滑动条）是否正常工作；
- 是否能够识别非法数据；
- 提示消息是否可理解。

### 3. 业务测试

功能测试完成后进行业务测试，业务测试关注的要点是业务流程，及数据流从软件中的一个模块流到另一个模块的过程中的正确性。

## 5.2 部门信息模块测试

部门信息模块的测试主要验证管理员是否能正确的实现部门信息的增删改查功能，并对管理员输入的数据进行合法性验证，对输入的非法数据给出提示，并验证提示信息可理解性。该模块的测试用例及测试执行结果如下表 5-1 所示：

表 5-1 部门信息模块测试用例表

用例编号	测试项目	测试标题	重要级别	预置条件	输入	执行步骤	预期输出	测试结果
------	------	------	------	------	----	------	------	------

1	添加部门信息	【添加】按钮功能测试	高	管理员登录成功，打开新增部门页面	无	点击【添加部门】按钮	弹出“添加部门”页面	通过
2	添加部门信息	输入全部正确的新增数据，进行【取消】按钮功能测试	高	管理员登录成功，打开“添加部门信息”窗口	1、部门名称：研发部门 2、部门简介：企业研发部门 3、部门人员：5人 4、部门工资：4500	输入以上数据，点击【取消】按钮	不保存当前新增内容，关闭当前窗口，回到列表页	通过
3	添加部门信息	不输入部门名称（非空验证）数据，进行保存	高	管理员登录成功，打开“添加部门信息”窗口	1、部门名称：研发部门 2、部门简介：企业研发部门 3、部门人员：5人 4、部门工资：4500	输入以上数据，点击【保存】按钮	提示：请输入部门名称	通过
4	添加部门信息	不输入部门简介（非空验证）数据，进行保存	高	管理员登录成功，打开“添加部门信息”窗口	1、部门名称：研发部门 2、部门简介：企业研发部门 3、部门人员：5人 4、部门工资：4500	输入以上数据，点击【保存】按钮	提示：添加成功（部门简介可以为空）	通过
5	添加部门信息	不输入部门人数（非空验证）数据，进行保存	高	管理员登录成功，打开“添加部门信息”窗口	1、部门名称：研发部门 2、部门简介：企业研发部门 3、部门人员：5人 4、部门工资：4500	输入以上数据，点击【保存】按钮	提示：请输入部门人数	通过



6	添加部门信息	不输入部门工资（非空验证）数据，进行保存	高	管理员登录成功，打开“添加部门信息”窗口	1、部门名称：研发部门 2、部门简介：企业研发部门 3、部门人员：5人 4、部门工资：4500	输入以上数据，点击【保存】按钮	提示：请输入部门工资	通过
7	修改部门信息	【修改】按钮功能测试	高	管理员登录成功，打开部门信息页面	无	点击【修改】按钮	弹出“修改部门”页面，显示带入的部门信息	通过
8	修改部门信息	“修改部门信息”窗口文字显示及UI排版正确性验证	中	管理员登录成功，打开“修改部门信息”窗口	无	查看页面显示	页面文字显示及UI排版正确	通过
9	修改部门信息	输入全部正确的修改数据，进行【保存】按钮功能测试	高	管理员登录成功，打开“修改部门信息”窗口	1、部门名称：研发部门 2、部门简介：企业研发部门 3、部门人员：5人 4、部门工资：4500	输入以上数据，点击【保存】按钮	保存当前修改内容，关闭当前窗口，返回列表页	通过
10	修改部门信息	修改部门人数正确性验证	高	管理员登录成功，打开部门信息页面	无	查看部门信息列表页数据显示	列表页相应内容随之更新	通过
11	修改部门信息	输入全部正确的修改数据，进行【取消】按钮功能测试	高	管理员登录成功，打开“修改部门信息”窗口	1、部门名称：研发部门 2、部门简介：企业研发部门 3、部门人员：5人 4、部门工资：4500	输入以上数据，点击【取消】按钮	不保存当前编辑内容，关闭当前窗口，返回列表页，列表页相应内容前后不变	通过

12	修改部门信息	输入全部正确的修改数据，进行“【取消】”按钮功能测试	高	管理员登录成功，打开“修改部门信息”窗口	1、部门名称：研发部门 2、部门简介：企业研发部门 3、部门人员：5人 4、部门工资：4500	输入以上数据，点击“取消”按钮	不保存当前编辑内容，关闭当前窗口，返回列表页，列表页相应内容前后不变	通过
13	修改部门信息	修改部门工资记录正确性验证	高	管理员登录成功，打开部门工资页面	无	查看部门信息页数据显示	列表页相应内容随之更新	通过

### 5.3 员工信息模块测试

员工关系信息模块的测试主要验证管理员是否能正确的实现员工信息的增删改查功能，并对管理员输入的数据进行合法性验证，对输入的非法数据给出提示，并验证提示信息的可理解性。该模块的测试用例及测试执行结果如下表 5-2 所示：

表 5-2 教练信息测试用例表

用例编号	测试项目	测试标题	重要级别	预置条件	输入	执行步骤	预期输出	测试结果
1	添加员工信息	【添加员工】按钮功能测试	高	管理员登录成功，打开新增员工页面	无	点击【添加】按钮	弹出“添加员工”页面	通过
2	添加员工信息	输入全部正确的新增数据，进行【取消】按钮功能测试	高	管理员登录成功，打开“添加员工信息”窗口	1、员工姓名：张三 2、电话：13789770858 3、学历：大专 4、年龄：21 5、性别：男 6、职位：员工 7、部门：研发部 8、照片：1.jpg 9、账号：zs 10、	输入以上数据，点击【取消】按钮	不保存当前新增内容，关闭当前窗口，返回列表页	通过

					密码： 123456 11、 籍贯：湖南 岳阳			
3	添加 员工 信息	不输入姓 名（非空 验证）数 据，进行 保存	高	管理员登录 成功，打开 “添加员工 信息”窗口	1、员工姓 名：张三 2、 电话： 1378977085 8 3、学历：大 专 4、年龄： 21 5、性别： 男 6、职位： 员工 7、部 门：研发部 8、照片：1、 jpg 9、账 号：zs 10、 密码： 123456 11、 籍贯：湖南 岳阳	输入以上 数据，点 击【保存】 按钮	提示：请 输入员工 姓名	通过
4	添加 员工 信息	不输入联 系电话 （非空验 证）数据， 进行保存	高	管理员登录 成功，打开 “添加员工 信息”窗口	1、员工姓 名：张三 2、 电话： 1378977085 8 3、学历：大 专 4、年龄： 21 5、性别： 男 6、职位： 员工 7、部 门：研发部 8、照片：1、 jpg 9、账 号：zs 10、 密码： 123456 11、 籍贯：湖南 岳阳	输入以上 数据，点 击【保存】 按钮	提示：请 输入联系 电话	通过
5	添加 员工 信息	不输入账 号（非空 验证）数 据，进行 保存	高	管理员登录 成功，打开 “添加员工 信息”窗口	1、员工姓 名：张三 2、 电话： 1378977085 8 3、学历：大	输入以上 数据，点 击【保存】 按钮	提示：请 输入员工 账号	通过

					专 4、年龄：21 5、性别：男 6、职位：员工 7、部门：研发部 8、照片：1、jpg 9、账号：zs 10、密码：123456 11、籍贯：湖南岳阳			
6	添加员工信息	不输入密码（非空验证）数据，进行保存	高	管理员登录成功，打开“添加员工信息”窗口	1、员工姓名：张三 2、电话：13789770858 3、学历：大专 4、年龄：21 5、性别：男 6、职位：员工 7、部门：研发部 8、照片：1、jpg 9、账号：zs 10、密码：123456 11、籍贯：湖南岳阳	输入以上数据，点击【保存】按钮	提示：请输入员工密码	通过
7	添加员工信息	不输入员工图片（非空验证）数据，进行保存	高	管理员登录成功，打开“添加员工信息”窗口	1、员工姓名：张三 2、电话：13789770858 3、学历：大专 4、年龄：21 5、性别：男 6、职位：员工 7、部门：研发部 8、照片：1、jpg 9、账号：zs 10、密码：123456 11、	输入以上数据，点击【保存】按钮	提示：请输入员工图片	通过

					籍贯：湖南岳阳			
8	添加员工工资	【添加工资】按钮功能测试	高	管理员登录成功，打开新增员工工资页面	无	点击【添加工资】按钮	弹出“添加员工工资”页面	通过
9	添加员工工资	输入全部正确的新增数据，进行【取消】按钮功能测试	高	管理员登录成功，打开“添加员工工资”窗口	1、员工姓名：张三 2、基本工资：3600.00 3、月份：2021.4.20 4、加班补贴：500.00 5、按月奖金：1500.00 6、考勤工资：500.00	输入以上数据，点击【取消】按钮	不保存当前新增内容，关闭当前窗口，回到列表页	通过
10	添加员工工资	不输入月份（非空验证）数据，进行保存	高	管理员登录成功，打开“添加员工工资”窗口	1、员工姓名：张三 2、基本工资：3600.00 3、月份：2021.4.20 4、加班补贴：500.00 5、按月奖金：1500.00 6、考勤工资：500.00	输入以上数据，点击【保存】按钮	提示：请输入月份	通过
11	修改员工信息	不输入加班补贴（非空验证）数据，进行保存	高	管理员登录成功，打开“添加员工工资”窗口	1、员工姓名：张三 2、基本工资：3600.00 3、月份：2021.4.20 4、加班补贴：500.00 5、按月奖金：1500.00 6、考勤工资：500.00	输入以上数据，点击【保存】按钮	提示：请输入加班补贴	通过
12	修改	不输入按	高	管理员登录	1、员工姓	输入以上	提示：请	通过

	员工信息	月奖金（非空验证）数据，进行保存		成功，打开“添加员工工资”窗口	名：张三 2、基本工资：3600.00 3、月份：2021.4.20 4、加班补贴：500.00 5、按月奖金：1500.00 6、考勤工资：500.00	数据，点击【保存】按钮	输入按月奖金	
13	修改员工信息	不输考勤工资（非空验证）数据，进行保存	高	管理员登录成功，打开“添加员工工资”窗口	1、员工姓名：张三 2、基本工资：3600.00 3、月份：2021.4.20 4、加班补贴：500.00 5、按月奖金：1500.00 6、考勤工资：500.00	输入以上数据，点击【保存】按钮	提示：请输入考勤工资	通过

## 5.4 招聘信息模块测试

招聘信息模块的测试主要验证管理员是否能正确的实现招聘信息的增删改查功能，并对管理员输入的数据进行合法性验证，对输入的非合法数据给出提示，并验证提示信息可理解性。该模块的测试用例及测试执行结果如下表 5-3 所示：

表 5-3 招聘信息模块测试用例表

用例编号	测试项目	测试标题	重要级别	预置条件	输入	执行步骤	预期输出	测试结果
1	添加招聘信息	【添加】按钮功能测试	高	管理员登录成功，打开新增招聘信息页面	无	点击【添加】按钮	弹出“添加招聘信息”页面	通过
2	添加招聘信息	输入全部正确的新增数据，	高	管理员登录成功，打开“添加招聘	1、员工姓名：李四 2、电话：	输入以上数据，点击【取消】	不保存当前新增内容，关闭	通过

		进行【取消】按钮功能测试		信息”窗口	13789770858 3、学历：大专 4、年龄：21 5、性别：男 6、职位：员工 7、部门：研发部 8、照片：2.jpg 9、籍贯：湖南岳阳 10、简介：能够胜任	按钮	当前窗口，返回列表页	
3	添加招聘信息	不输入姓名（非空验证）数据，进行保存	高	管理员登录成功，打开“添加招聘信息”窗口	1、员工姓名：李四 2、电话：13789770858 3、学历：大专 4、年龄：21 5、性别：男 6、职位：员工 7、部门：研发部 8、照片：2.jpg 9、籍贯：湖南岳阳 10、简介：能够胜任	输入以上数据，点击【保存】按钮	提示：请输入姓名	通过
4	添加招聘信息	不输入联系电话（非空验证）数据，进行保存	高	管理员登录成功，打开“添加招聘信息”窗口	1、员工姓名：李四 2、电话：13789770858 3、学历：大专 4、年龄：21 5、性别：男 6、职位：员工 7、部门：研发部 8、照片：2.jpg 9、籍贯：湖南岳阳 10、简介：能够胜任	输入以上数据，点击【保存】按钮	提示：请输入联系电话	通过

					介：能够胜任			
5	添加招聘信息	不输入员工图片（非空验证）数据，进行保存	高	管理员登录成功，打开“添加招聘信息”窗口	1、员工姓名：李四 2、电话：13789770858 3、学历：大专 4、年龄：21 5、性别：男 6、职位：员工 7、部门：研发部 8、照片：2.jpg 9、籍贯：湖南岳阳 10、简介：能够胜任	输入以上数据，点击【保存】按钮	提示：请添加图片	通过
6	添加招聘信息	【录用】按钮功能测试	高	管理员登录成功，打开录用员工页面	无	点击【录用】按钮	弹出“录用”页面	通过
7	添加招聘信息	对录用员工，进行【取消】按钮功能测试	高	管理员登录成功，打开“录用员工”窗口	2、员工姓名：李四 2、电话：13789770858 3、学历：大专 4、年龄：21 5、性别：男 6、职位：员工 7、部门：研发部 8、照片：2.jpg 9、籍贯：湖南岳阳 10、简介：能够胜任	输入以上数据，点击【取消】按钮	不添加当前新增内容，关闭当前窗口，返回列表页	通过

## 5.5 培训信息模块测试

培训信息模块的测试主要验证管理员是否能正确的实现培训信息的增删改查功能，



并对管理员输入的数据进行合法性验证，对输入的非法数据给出提示，并验证提示信息  
的可理解性。该模块的测试用例及测试执行结果如下表 5-4 所示：

表 5-4 培训信息模块测试用例表

用例编号	测试项目	测试标题	重要级别	预置条件	输入	执行步骤	预期输出	测试结果
1	添加培训信息	【添加】按钮功能测试	高	管理员登录成功，打开新增培训信息页面	无	点击【添加】按钮	弹出“添加培训信息”页面	通过
2	添加培训信息	输入全部正确的新增数据，进行【取消】按钮功能测试	高	管理员登录成功，打开“添加培训信息”窗口	1、培训名称：技能培训 2、培训地点：公司会议室 3、参与人员：开发部门全体人员 4、培训宗旨：提高员工技能 5、培训时间：18:00 6、结束时间：20:00	输入以上数据，点击【取消】按钮	不保存当前新增内容，关闭当前窗口，返回列表页	通过
3	添加培训信息	不输入培训名称（非空验证）数据，进行保存	高	管理员登录成功，打开“添加培训信息”窗口	1、培训名称：技能培训 2、培训地点：公司会议室 3、参与人员：开发部门全体人员 4、培训宗旨：提高员工技能 5、培训时间：18:00 6、结束时间：20:00	输入以上数据，点击【保存】按钮	提示：请输入培训名称	通过
4	添加培训信息	不输入培训地点（非空验证）数据，	高	管理员登录成功，打开“添加培训信息”窗口	1、培训名称：技能培训 2、培训地点：公司	输入以上数据，点击【保存】按钮	提示：请输入培训地点	通过

		进行保存			会议室 3、 参与人员： 开发部门全 体人员 4、 培训宗旨： 提高员工技 能 5、培训 时间：18； 00 6、结束 时间：20:00			
5	添加 培训 信息	不输入参 与员工 （非空验 证）数据， 进行保存	高	管理员登录 成功，打开 “添加培训 信息”窗口	1、培训名 称：技能培 训 2、培训 地点：公司 会议室 3、 参与人员： 开发部门全 体人员 4、 培训宗旨： 提高员工技 能 5、培训 时间：18； 00 6、结束 时间：20:00	输入以上 数据，点 击【保存】 按钮	提示：请 输入参与 员工	通过
6	添加 培训 信息	不输入培 训宗旨 （非空验 证）数据， 进行保存	高	管理员登录 成功，打开 “添加培训 信息”窗口	1、培训名 称：技能培 训 2、培训 地点：公司 会议室 3、 参与人员： 开发部门全 体人员 4、 培训宗旨： 提高员工技 能 5、培训 时间：18； 00 6、结束 时间：20:00	输入以上 数据，点 击【保存】 按钮	提示：请 输入培训 宗旨	通过
7	添加 培训 信息	不输入培 训时间 （非空验 证）数据， 进行保存	高	管理员登录 成功，打开 “添加培训 信息”窗口	3、员工姓 名：李四 2、电话： 1378977085 8 3、学历：大 专 4、年龄： 21 5、性别：	输入以上 数据，点 击【保存】 按钮	提示：请 输入培训 时间	通过

					男 6、职位： 员工 7、部 门：研发部 8、照片： 2.jpg 9、 籍贯：湖南 岳阳 10、简 介：能够胜 任			
--	--	--	--	--	---	--	--	--

## 5.6 奖罚信息模块测试

奖罚信息模块的测试主要验证管理员是否能正确的实现奖罚信息的增删改查功能，并对管理员输入的数据进行合法性验证，对输入的非法数据给出提示，并验证提示信息可理解性。该模块的测试用例及测试执行结果如下表 5-5 所示：

表 5-5 奖罚信息模块测试用例表

用例编号	测试项目	测试标题	重要级别	预置条件	输入	执行步骤	预期输出	测试结果
1	添加奖罚信息	【添加】按钮功能测试	高	管理员登录成功，打开新增奖罚信息页面	无	点击【添加】按钮	弹出“添加奖罚信息”页面	通过
2	添加奖罚信息	输入全部正确的新增数据，进行【取消】按钮功能测试	高	管理员登录成功，打开“添加奖罚信息”窗口	1、姓名：张三 2、奖罚原因：优秀员工 3、奖罚时间：2021.4.20 4、奖罚内容：200 元	输入以上数据，点击【取消】按钮	不保存当前新增内容，关闭当前窗口，回到列表页	通过
3	添加奖罚信息	不输入奖罚名称（非空验证）数据，进行保存	高	管理员登录成功，打开“添加奖罚信息”窗口	1、姓名：张三 2、奖罚原因：优秀员工 3、奖罚时间：2021.4.20 4、奖罚内容：200 元	输入以上数据，点击【保存】按钮	提示：请输入奖罚名称	通过

4	添加奖罚信息	不输入奖罚原因（非空验证）数据，进行保存	高	管理员登录成功，打开“添加奖罚信息”窗口	1、姓名：张三 2、奖罚原因：优秀员工 3、奖罚时间：2021.4.20 4、奖罚内容：200 元	输入以上数据，点击【保存】按钮	提示：请输入奖罚原因	通过
5	添加奖罚信息	不输入奖罚时间（非空验证）数据，进行保存	高	管理员登录成功，打开“添加奖罚信息”窗口	1、姓名：张三 2、奖罚原因：优秀员工 3、奖罚时间：2021.4.20 4、奖罚内容：200 元	输入以上数据，点击【保存】按钮	提示：请输入奖罚时间	通过
6	添加奖罚信息	不输入奖罚内容（非空验证）数据，进行保存	高	管理员登录成功，打开“添加奖罚信息”窗口	1、姓名：张三 2、奖罚原因：优秀员工 3、奖罚时间：2021.4.20 4、奖罚内容：200 元	输入以上数据，点击【保存】按钮	提示：请输入奖罚内容	通过

## 5.7 公告信息模块测试

公告信息模块的测试主要验证管理员是否能正确的实现公告信息的增删改查功能，并对管理员输入的数据进行合法性验证，对输入的非合法数据给出提示，并验证提示信息可理解性。该模块的测试用例及测试执行结果如下表 5-6 所示：

表 5-6 公告信息模块测试用例表

用例编号	测试项目	测试标题	重要级别	预置条件	输入	执行步骤	预期输出	测试结果
1	添加公告信息	【添加】按钮功能测试	高	管理员登录成功，打开新增公告信息页面	无	点击【添加】按钮	弹出“添加公告信息”页面	通过
2	添加	输入全部	高	管理员登录	1、公告标	输入以上	不保存当	通过

	公告信息	正确的新增数据，进行【取消】按钮功能测试		成功，打开“添加公告信息”窗口	题：放假通知 2、公告内容：临时放假一天 3、公告图片：3. jpg	数据，点击【取消】按钮	前新增内容，关闭当前窗口，回到列表页	
3	添加公告信息	不输入公告标题（非空验证）数据，进行保存	高	管理员登录成功，打开“添加公告信息”窗口	1、公告标题：放假通知 2、公告内容：临时放假一天 3、公告图片：3. jpg	输入以上数据，点击【保存】按钮	提示：请输入公告标题	通过
4	添加公告信息	不输入公告内容（非空验证）数据，进行保存	高	管理员登录成功，打开“添加公告信息”窗口	1、公告标题：放假通知 2、公告内容：临时放假一天 3、公告图片：3. jpg	输入以上数据，点击【保存】按钮	提示：请输入公告内容	通过
5	添加公告信息	不输入公告图片（非空验证）数据，进行保存	高	管理员登录成功，打开“添加公告信息”窗口	1、公告标题：放假通知 2、公告内容：临时放假一天 3、公告图片：3. jpg	输入以上数据，点击【保存】按钮	提示：请输入公告图片	通过
6	添加公告信息	【删除】按钮功能测试	高	管理员登录成功，删除公告信息页面	无	点击【删除】按钮	弹出“是否要删除公告信息”页面	通过
7	添加公告信息	【删除】按钮功能测试，点击取消	高	管理员登录成功，删除公告信息页面	无	点击【取消】按钮	不进行删除操作，关闭当前窗口，回到列表页	通过

## 第六章 结论

### 6.1 毕业设计成果特点

无忧人力资源管理系统管理员拥有部门信息、员工信息、招聘信息、培训信息、奖惩信息、工资信息、公告信息的增删查改操作。通过此管理系统的开发，将软件技术专业授课阶段学习的前沿理论知识 C#、MVC、面向对象、数据库等与实践联系，融会贯通，不但提高了理论水平，还提高了项目开发的能力。项目采用 ASP.NET MVC 进行搭建，Web 前端采用 Html+CSS、Bootstrap 等流行的技术，结合 Javascript 和 jQuery 实现动态特效。Web 后端采用 MVC+EF 框架实现业务逻辑的处理和数据库访问。

### 6.2 设计成果的实用价值或应用前景

随着互联网消费时代的到来,如何方便、简捷地进行企业管理，方便企业管理人员进行相关的操作。管理人员只需要进入后台管理系统，选择相应的操作就可以对相关的信息进行操作，所以我们决定做一个企业人力支援管理系统，无忧人力资源管理系统提供了便捷的管理操作，可以方便企业的管理人员进行公司的管理，从解决企业管理难，分工不明确的问题，一个功能完善、简单易用的信息管理系统不但能有效地减轻企业管理人员的工作负担，而且提高了企业人员的管理效率。

### 6.3 不足之处或遗留未予解决的问题

无忧人力管理系统-部门信息、员工信息等模块的设计与实现是基于 B/S 管理模式的结构平台和面向对象的设计思想，并利用一些与 MVC 相关的技术，对于无忧人力资源管理系统各部门、人员等信息模块要求的功能已全部完整的实现。尽管如此，本系统在面向企业使用过程中，仍遇到了某些潜在的问题，但大体上还是完成了无忧人力资源管理的功能需求。

本系统的不足之处：

1. 员工管理系统没有设置分部门显示的方式，需要手动去查询。
2. 培训信息无法批量添加人员。
3. 数据库的恢复和备份没有完全实现自动化。
4. 界面不够美观，需要调整样式及色彩搭配。

以上这些问题，将在今后的工作中进行完善。同时我们会在使用该系统的过程中，不断总结反馈的信息，希望该系统能在未来广泛推广使用，服务更多的人群。