**目 录**

[1绪论 1](#_Toc41484992)

[1.1研究的背景及意义 1](#_Toc41484993)

[1.1.1 选题的背景 1](#_Toc41484994)

[1.1.2 国内外研究现状 1](#_Toc41484995)

[1.1.3 研究的意义 2](#_Toc41484996)

[1.2 系统目标 3](#_Toc41484997)

[2需求分析 4](#_Toc41484998)

[2.1 业务需求 4](#_Toc41484999)

[2.1.1主要业务流程 4](#_Toc41485000)

[2.2 功能需求 5](#_Toc41485001)

[2.2.1 角色分析 6](#_Toc41485002)

[2.2.2 业务功能 6](#_Toc41485003)

[2.3 非功能需求 14](#_Toc41485004)

[2.3.1环境需求 15](#_Toc41485005)

[2.3.2 性能需求 15](#_Toc41485006)

[2.3.3 安全需求 16](#_Toc41485007)

[3 总体设计 17](#_Toc41485008)

[3.1系统设计的原则 17](#_Toc41485009)

[3.2 系统体系结构设计 17](#_Toc41485010)

[3.3 系统功能模块设计 18](#_Toc41485011)

[4 数据库设计 19](#_Toc41485012)

[4.1概念结构设计 19](#_Toc41485013)

[4.1.1设计思路 19](#_Toc41485014)

[4.1.2 E-R图 19](#_Toc41485015)

[4.2 逻辑结构设计 23](#_Toc41485016)

[4.2.1 设计思路 23](#_Toc41485017)

[4.2.2 逻辑模型 24](#_Toc41485018)

[4.3 物理结构设计 24](#_Toc41485019)

[4.3.1 存取方式 24](#_Toc41485020)

[4.3.2 存储结构 24](#_Toc41485021)

[5 界面设计 29](#_Toc41485022)

[5.1 界面关系图或工作流图 29](#_Toc41485023)

[5.2 界面设计成果 29](#_Toc41485024)

[5.2.1 主界面 29](#_Toc41485025)

[5.2.2 子界面 31](#_Toc41485026)

[6 详细设计 34](#_Toc41485027)

[6.1系统主要功能模块介绍 34](#_Toc41485028)

[6.2会员单位管理 34](#_Toc41485029)

[6.2.1 功能结构设计 34](#_Toc41485030)

[6.2.2 类图设计 35](#_Toc41485031)

[6.2.3 顺序图设计 37](#_Toc41485032)

[6.2.4 核心处理流程设计 37](#_Toc41485033)

[6.3信息管理 38](#_Toc41485034)

[6.3.1 功能结构设计 38](#_Toc41485035)

[6.3.2 类图设计 39](#_Toc41485036)

[6.3.3 顺序图设计 41](#_Toc41485037)

[6.3.4 核心处理流程设计 41](#_Toc41485038)

[6.4专家库管理 42](#_Toc41485039)

[6.4.1 功能结构设计 42](#_Toc41485040)

[6.4.2 类图设计 43](#_Toc41485041)

[6.4.3 顺序图设计 45](#_Toc41485042)

[6.4.4 核心处理流程设计 45](#_Toc41485043)

[6.5收入与报送管理 46](#_Toc41485044)

[6.5.1 功能结构设计 46](#_Toc41485045)

[6.5.2 类图设计 47](#_Toc41485046)

[6.5.3 顺序图设计 49](#_Toc41485047)

[6.5.4 核心处理流程设计 49](#_Toc41485048)

[6.6技术转让管理 50](#_Toc41485049)

[6.6.1 功能结构设计 50](#_Toc41485050)

[6.6.2 类图设计 51](#_Toc41485051)

[6.6.3 顺序图设计 52](#_Toc41485052)

[6.6.4 核心处理流程设计 53](#_Toc41485053)

[6.7项目合作管理 54](#_Toc41485054)

[6.7.1 功能结构设计 54](#_Toc41485055)

[6.7.2 类图设计 54](#_Toc41485056)

[6.7.3 顺序图设计 56](#_Toc41485057)

[6.7.4 核心处理流程设计 56](#_Toc41485058)

[6.8招聘管理 57](#_Toc41485059)

[6.8.1 功能结构设计 57](#_Toc41485060)

[6.8.2 类图设计 58](#_Toc41485061)

[6.8.3 顺序图设计 60](#_Toc41485062)

[6.8.4 核心处理流程设计 60](#_Toc41485063)

[6.9简历管理 61](#_Toc41485064)

[6.9.1 功能结构设计 61](#_Toc41485065)

[6.9.2 类图设计 62](#_Toc41485066)

[6.9.3 顺序图设计 63](#_Toc41485067)

[6.9.4 核心处理流程设计 64](#_Toc41485068)

[7 编码 65](#_Toc41485069)

[7.1 代码实现与核心算法 65](#_Toc41485070)

[7.1.1协会管理员登录页面 65](#_Toc41485071)

[7.1.2新闻通知查询业务 66](#_Toc41485072)

[7.1.3技术转让查询业务 66](#_Toc41485073)

[7.1.4 简历管理 67](#_Toc41485074)

[7.2 代码优化分析 68](#_Toc41485075)

[8 测试 69](#_Toc41485076)

[8.1 测试方案设计 69](#_Toc41485077)

[8.1.1 测试策略 69](#_Toc41485078)

[8.1.2 测试进度安排 69](#_Toc41485079)

[8.1.3 测试资源 70](#_Toc41485080)

[8.1.4 关键测试点 70](#_Toc41485081)

[8.2 测试用例构建 70](#_Toc41485082)

[8.2.1 测试用例编写约定 70](#_Toc41485083)

[8.2.2 测试用例设计 71](#_Toc41485084)

[8.2.3 关键测试用例 72](#_Toc41485085)

[8.2.4 测试用例维护 73](#_Toc41485086)

[9 总结与展望 74](#_Toc41485087)

[9.1 设计工作总结 74](#_Toc41485088)

[9.2 未来工作展望 74](#_Toc41485089)

[谢 辞 75](#_Toc41485090)

[参考文献 76](#_Toc41485091)

[附录A 外文翻译—原文部分 78](#_Toc41485092)

[附录B 外文翻译—译文部分 83](#_Toc41485093)

[附录C 软件使用说明书 87](#_Toc41485094)

[附录D 主要源代码 90](#_Toc41485095)

# 1绪论

## 1.1研究的背景及意义

### 选题的背景

大数据及信息技术产业是南昌高新区服务外包产业基础上衍生的新兴产业，大数据产业的发展将成为引领南昌高新区经济增长的制高点。近年来，在省市的大力支持下，把握机遇稳步提升大数据及信息技术产业的实力，培育领头企业，扩大引入优势项目、助力企业创新发展及政策扶持等方面取得了长足进步，形成了较强的竞争优势。

到2020年末，南昌高新区将依托现有的产业研发优势，以大数据、VR/AR、工业4.0、智慧城市等前沿板块为抓手，以云计算、大数据、物联网、软件研发、“互联网+”等新一代信息技术产业为发力点，将高新区打造成国内领先的大数据及信息技术产业基地。

随着经济全球化浪潮和区域经济一体化趋势的不断推进，区域发展所带来的竞争越发激烈，产业格局变化日新月异。在当下错综复杂的发展形势，企业间唯有提升协作、相助发展、协同创新，才能在快速的发展浪潮中稳住位置、不断迈向未来。为此，在高新区管理会的大力倡议下，南昌高新区管委会成立了南昌高新区大数据及信息技术产业协会。

大数据及信息技术产业协会的成立将促进企业提升协作、互助发展、协同创新，进一步加强产业的竞争优势，为高新区大数据及信息技术产业发展提供澎湃的发展动力、开辟广阔的发展空间。为更好地利用信息化手段积极展开产业协会工作，协会决定分步建设《南昌高新区大数据及信息技术产业协会网站》项目。

本项目的设立与实施加强了协会会员之间的合作，使得协会资源得到了充分的利用，推动了协会的发展。促进了企业提升协作、相助发展、协同创新，进一步加强产业的竞争优势。降低协会的业务成本，提升协会工作效率，加强了协会信息化办公的能力。

### 国内外研究现状

大数据是随着信息技术高速发展来带的衍生物，更是信息化进程进入到新阶段的标志。大数据技术的发展与应用极大的推动了数字经济的构建与成长。信息化技术已经历过两次快速发展的浪潮，但信息化技术的成熟还远未到来。由于上世纪80年代个人计算机广泛普及，以单机应用为主要标志开启了数字化（信息化1.0）时代。上世纪90年代的互联网开始大规模投入商业领域，推动了以联网应用为主要特征的网络化（信息化2.0）。当前，以数据深度挖掘和融合应用为主要特征的信息化世界，正将我们引入一个新的智能化阶段（信息化3.0）。

在信息化的澎湃的发展浪潮中，数字化、网络化以及智能化是三条并行不悖的时代主线。数字化技术为信息化奠定了基础，依靠数字化技术我们能够轻易实现数据资源的获取和积累；网络化帮助我们构建稳定的平台，借助网格化技术促进数据资源的流通和汇集；智能化展现高效的能力，通过多渠道数据的融合分析将数据转换成信息呈现的类人智能，将极大的帮助人类更好地认知事物和处理问题。

信息技术从协助经济发展的辅助工具向成为推动经济发展的核心引擎的转变，是信息化发展进入到新阶段的另一个重要标志。随着信息技术在经济发展中的地位越来越重，一种新的经济范式应运而生——“数字经济”。“数字经济”是指以数字化知识和信息为核心生产要素、以现代信息网络体系为重要媒介载体、以有效使用的信息通信技术来提升效率和以优化经济结构为重要推动力的一系列经济活动。

“数字经济”是以新一代信息技术和产业为基础，继农业经济、工业经济之后的迎来的全新经济形态。从经济构成上来看，农业经济属于单层结构，以农业为主体结构并配合以其他行业，以人力、畜力和自然力为经济动力，使用手工工具，以家庭为单位达到自给自足，社会分工并不明显，行业间关系相对独立；工业经济是两层结构，即提供能源动力和行业制造装备的制造业，以及工业化发展后的各行各业，并形成分工合作的工业体系。数字经济则可分为三层结构：提供核心经济动力的是信息技术及其装备产业、深度信息化的各行各业以及跨行业数据融合应用的数据增值产业。当前，数字经济正渐渐成型，未来我们将进入信息技术引领经济发展的爆发期、黄金期！

我国作为人口大国和制造大国，数据产能不容小觑，大数据资源极为丰富。随着数字中国建设的推进，各行业的数据资源采集以及应用的能力不断加强，会带来更多的数据积累。然而，当下我国大数据与实体经济融合并不融洽，行业大数据应用的广度和深度略显匮乏，大数据生态系统形成和发展的问题亟待解决。

我国大数据与实体经济融合不够深入，主要问题表现在：与大数据相关的基础设施配置不到位，导致海量数据采集难度大；大数据与实体经济的融合缺乏有效引导与支撑，得不到大数据助力的实体经济数字化转型缓慢；大数据生态系统下缺乏自主可控的数据互联共享平台等。当前，工业互联网已经成为互联网发展的新领域，然而仍存在不少问题急需解决。我国当前要推动大数据技术与企业相互结合，着手推动相应的工业互联网平台的建设，快速抢占工业领域的大数据基础服务市场。

### 研究的意义

随着国家对政务信息化进程的稳步推进，各级政府已经不断积累了大量与大众生产生活休戚相关的数据，并成为大数据时代下最具价数据的保有者。如何盘活政务信息数据，更好地支撑政府完成决策并更好的提供便民服务，进而促进大数据事业蓬勃发展，是事关全局的关键。

国务院于2015年9月发布《促进大数据发展行动纲要》，行动纲要指出当前大数据发展的重要任务之一就是“加快政府数据开放共享，推动资源整合，提升治理能力”，纲要明确了时间节点，2017年要基本形成跨部门数据资源共享共用格局；2018年要建成以政府为主导的数据共享开放平台，打通政府部、企事业单位间的数据壁垒，并在部分领域开展应用试点服务；2020年达到政府数据集的普遍开放。随后，国务院和国务院办公厅又陆续印发了一系列相关文件，来推进政务信息资源的共享管理、政务信息系统的整合共享、互联网+政务服务试点、政务服务一网一门一次改革等，推进跨越层级、跨越地域、跨越系统、跨越部门、跨越业务的政务信息系统整合、互联、协同和数据共享，以政务大数据为原材料支撑“放管服”改革落地，推进建设数字政府和智慧政府。目前，我国政务领域的数据开放共享已取得了引人注目的进展和较为显著的效果。

本项目的设立与实施加强了协会会员之间的合作，使得协会资源得到了充分的利用，推动了协会的发展。促进了企业提升协作、互助发展、协同创新，进一步加强产业的竞争优势。降低协会的业务成本，提升协会工作效率，加强了协会信息化办公的能力。这项研究能更好地利用信息化手段积极展开产业协会工作，提供高效、快捷服务，推动各企业资源共享、协同创新。同时，提升协会管理水平和效率，切实为大数据及信息技术产业的发展发挥积极作用，进一步积聚资源，为大数据及信息技术产业协会发展提供更有力的支持和服务。

## 系统目标

为更好地利用信息化手段积极展开产业协会工作，协会决定分步建设《南昌高新区大数据及信息技术产业协会网站》项目，搭建一个集信息发布、资源共享、供需撮合、会员管理功能为一体的一站式服务平台，为企业的新闻、产品或技术、人才供需、项目合作等方面提供高效、快捷服务，推动各企业资源共享、协同创新。同时，提升协会管理水平和效率，切实为大数据及信息技术产业的发展带来积极的推动作用，进一步积聚资源，为大数据及信息技术产业协会发展提供更强劲有力的支持和服务。

根据以上分析，按照“统一规划、分步实现、应用主导、资源共享”的总体原则，本网站建设定位于服务大数据及信息技术产业协会，主要建设 “协会官网、协会管理员后台、会员单位后台”。

1. 协会官网：主要展示产业协会相关的新闻中心（新闻动态：图片新闻、视频新闻、混合新闻）、通知公告、产业政策等信息。
2. 协会管理员后台：官网支撑服务系统，主要提供网站用户权限管理、新闻管理、通知管理、公告管理、关于协会、会员单位管理、网站运行管理等功能。
3. 会员单位后台：搭建一个协会会员单位与非会员单位或用户相互交流分享的平台，提供人才需求、项目合作、技术转让等资源共享以及合作撮合管理。

# 2需求分析

## 2.1 业务需求

1. 游客用户能够进入网站前台，实现新闻展示、人才招聘信息查看、会员单位信息查看、简历投递等功能。
2. 协会管理员能够进入协会管理员后台，实现协会管理员对协会官网产生的所有业务数据的管理，如新闻通知、会员单位信息、专家库信息、招聘与简历信息的管理与维护。
3. 会员单位能够进入会员单位后台，实现对招聘信息、资源共享信息、技术转让等信息进行管理与维护。
4. 系统的响应速度合理，具有较高的安全性运行稳定，且应易于后期的维护与升级。

### 2.1.1主要业务流程

1. 新闻展示活动图

游客进入协会网站前台，浏览协会网站的新闻通知、协会概况、会员风采等信息。活动流程图如图2-1所示。

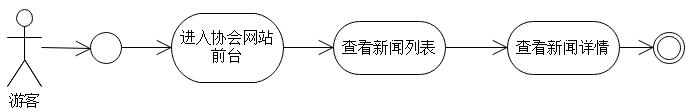


图2-1 新闻展示活动图

1. 简历投递活动图

游客进入协会网站前台后，浏览人才招聘页面查看招聘信息列表。游客点击查看详情进入特定岗位的招聘信息详情页面，并点击投递简历跳转至简历投递页面填写个人简历信息。填写完个人简历信息后，点击提交完成简历投递活动。活动流程图如图2-2所示。

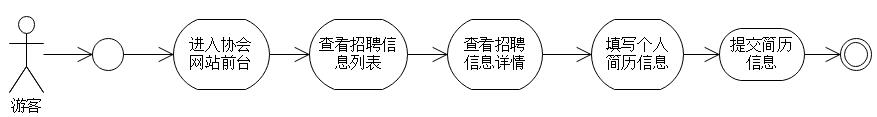


图2-2 简历投递活动图

1. 协会管理员后台管理活动

协会管理员登录管理员后台，选择信息管理模块。在信息管理模块中点击新增，进入前台信息维护页面填写要发布的信息。填写完信息后，点击提交完成前台信息发布活动。活动流程图如图2-3所示。

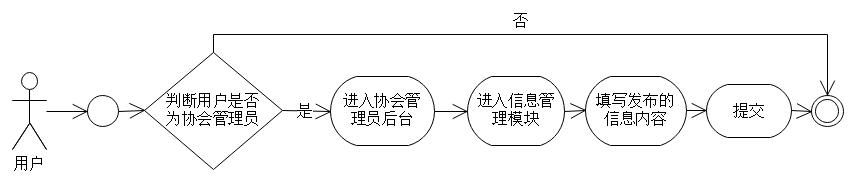


图2-3 协会管理员后台管理活动图

1. 会员单位后台管理活动图

会员单位登录会员单位后台，选择项目合作模块查看项目合作列表。点击新增按钮添加新项目合作信息。在项目合作信息维护页面填写要发布的项目合作信息，完成后点击提交按钮完成项目合作发布活动。活动图如图2-4所示。

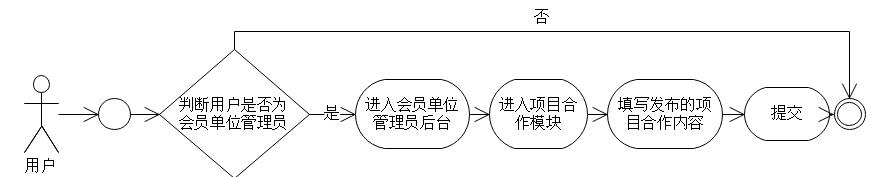


图 2-4 会员单位资源共享活动图

## 2.2 功能需求

1. 会员单位管理：管理员对协会成员信息进行维护，包括成员账号信息的新增、修改、删除、和查询操作以及成员发布信息权限的新增、修改、删除、和查询操作。
2. 信息管理：主要是管理本协会官网首页所展示的各种信息，对信息实行分类且实现信息的发布、修改、删除和查找。例如，管理员对新闻信息进行分类，且在发布新闻时选择相应类别信息类型。政策、通知、协会成员风采、协会成员人才招聘信息、协会成员合作、技术转让信息和专家信息等都是通过这个功能模块实现维护。
3. 专家库管理：管理员对专家信息进行维护，包括专家类别的新增、修改、删除、和查询操作以及专家详细信息的新增、修改、删除、和查询操作。
4. 人才招聘管理：会员单位根据自身的人才需求，可以通过平台将人才要求发布出去，发布后的信息会在前端人才招聘板块予以显示，游客或注册用户都可以查看到该信息，并投送简历。会员单位可以实现人才招聘信息的发布、修改、删除、检索、审核。
5. 项目合作管理：会员单位可能有一些项目需要协作完成，在没有合适的资源的前提下，会员单位可以将这种需求发布出去，有符合要求的会员单位就可以双方洽谈。会员单位可以实现合作项目信息的发布、撤销、查找、修改等操作。
6. 技术转让管理：会员单位若有一些需要转让的技术，可以通过这个网站平台寻找转让企业。会员单位可以实现技术转让信息的发布、修改、撤销、终止、查找等操作。
7. 收入报送管理：会员单位将每个月的收支报告上传至网站，供管理员查看业绩情况信息。

### 2.2.1 角色分析

以下从大数据及信息技术产业协会网站的实际需求分析，网站系统涉及到以下角色：

表 2-1 系统角色表

|  |  |
| --- | --- |
| 角色 | 职责或功能 |
| 协会管理员 | 负责网站信息维护，监督会员单位发布的招聘、合作等信息 |
| 会员单位 | 发布人才招聘、项目合作、技术转让、合作撮合等信息 |
| 游客 | 查看网站信息，投递简历等 |

### 2.2.2 业务功能

以下从业务角度出发，给出了系统的总体用例图，包含会员单位管理、信息管理、专家库管理、收入报送管理、技术转让管理、项目合作管理、招聘管理、简历管理。总用例图如图2-5所示。

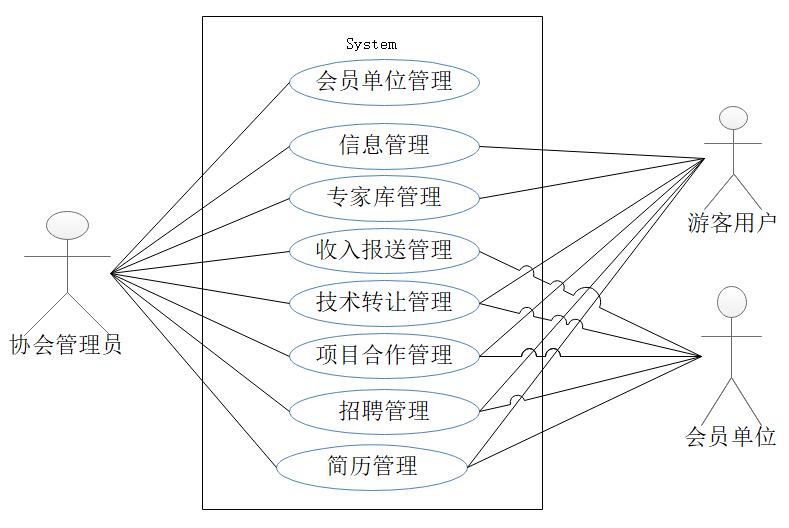


图2-5 系统总体用例图

1. 会员单位管理

会员单位管理包含协会管理员对协会会员单位信息的添加、查找、修改、删除操作。用例图如图2-6所示。

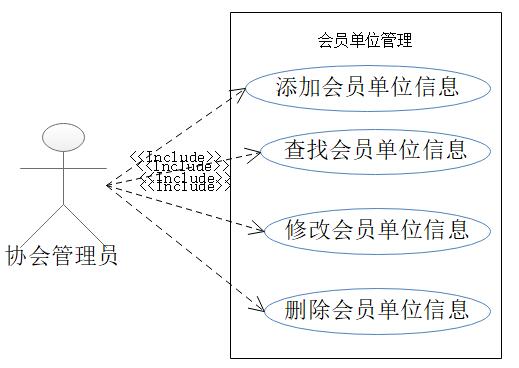


图2-6 会员单位管理用例图

表2-2 添加会员单位信息用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 添加会员单位信息 |
| 标识符\* | YL01 |
| 用例描述 | 添加会员单位管理描述了协会管理员通过协会管理员后台添加协会会员单位信息的操作过程。 |
| 参与者表 | 协会管理员 |
| 优先级 | 1 |
| 状态\* | 进行中 |
| 前置条件 | 协会管理员进入协会管理员后台 |
| 后置条件 | 成功添加会员单位信息 |
| 基本操作流 | 1. 协会管理员进入协会管理员后台 2. 进入会员单位管理模块 3. 添加会员单位信息 |

1. 信息管理

信息管理子系统包含了协会管理员对前台信息添加、查找、修改、删除操作，同时也包含了游客从网站前台对信息的查找访问。用例图如图2-7所示。

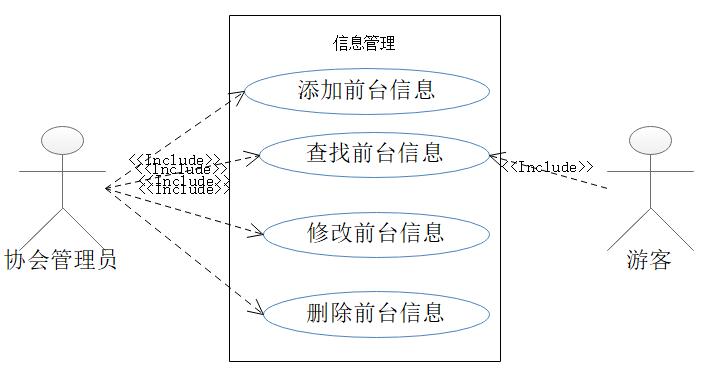


图2-7 信息管理用例图

表2-10 删除前台信息用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 删除前台信息 |
| 标识符\* | YL02 |
| 用例描述 | 删除前台信息描述了协会管理员进入协会管理员后台后，删除前台信息的操作过程。 |
| 参与者表 | 协会管理员 |
| 优先级 | 1 |
| 状态\* | 进行中 |
| 前置条件 | 协会管理员进入协会管理员后台 |
| 后置条件 | 前台信息删除 |
| 基本操作流 | 1. 协会管理员进入协会管理员后台 2. 进入信息管理模块 3. 删除前台信息 |

1. 专家库管理

专家库管理子系统包含了协会管理员对专家信息添加、查找、修改、删除操作，同时也包含了游客从网站前台对专家信息的查找访问。用例图如图2-8所示。

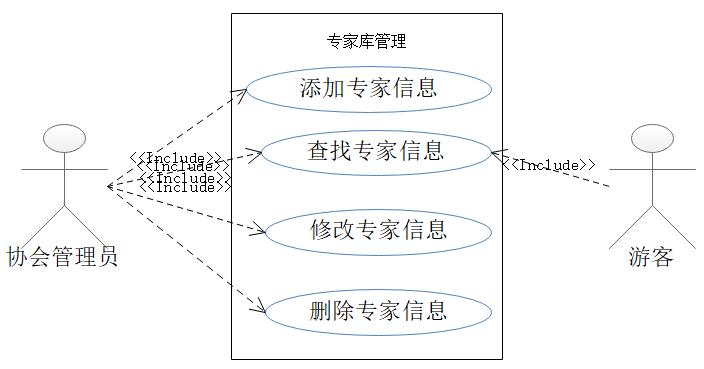


图2-8 专家库管理用例图

表2-14 修改专家信息用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 修改专家信息 |
| 标识符\* | YL03 |
| 用例描述 | 修改专家信息描述了协会管理员进入协会管理员后台后，修改专家信息的操作过程。 |
| 参与者表 | 协会管理员 |
| 优先级 | 1 |
| 状态\* | 进行中 |
| 前置条件 | 协会管理员进入协会管理员后台 |
| 后置条件 | 专家信息完成修改 |
| 基本操作流 | 1. 协会管理员进入协会管理员后台 2. 进入信息管理模块 3. 修改专家信息 |

1. 收入与报送管理

收入与报送管理子系统包含了会员单位在会员单位后台对收入与报送信息添加、查找、修改、删除操作，同时也包含了协会管理员从协会管理员后台对收入与报送信息的查找与删除。用例图如图2-9所示。

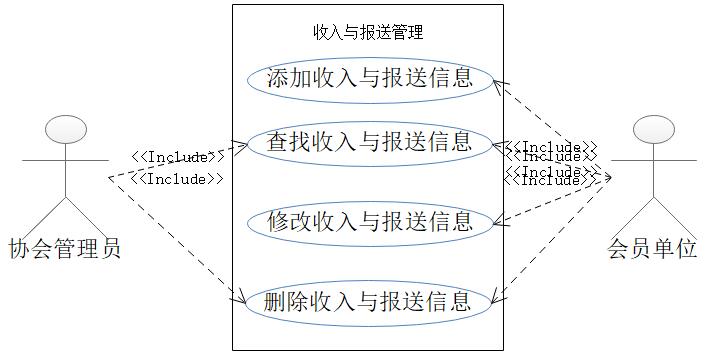


图2-9 收入与报送用例图

表2-17 协会管理员查找收入与报送信息用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 协会管理员查找收入与报送信息 |
| 标识符\* | YL04 |
| 用例描述 | 协会管理员查找收入与报送信息描述了协会管理员进入协会管理员后台后，查找收入与报送信息的操作过程。 |
| 参与者表 | 协会管理员 |
| 优先级 | 1 |
| 状态\* | 进行中 |
| 前置条件 | 协会管理员进入协会管理员后台 |
| 后置条件 | 协会管理员可以访问到收入与报送信息 |
| 基本操作流 | 1. 协会管理员进入协会管理员后台 2. 进入收入与报送信息管理模块 3. 查找收入与报送信息 |

1. 技术转让管理

技术转让管理子系统包含了会员单位在会员单位后台对技术转让信息添加、查找、修改、删除操作，同时也包含了协会管理员从协会管理员后台对技术转让信息的查找与删除。以及游客在协会网站前台对技术转让信息的查询访问。用例图如图2-10所示。

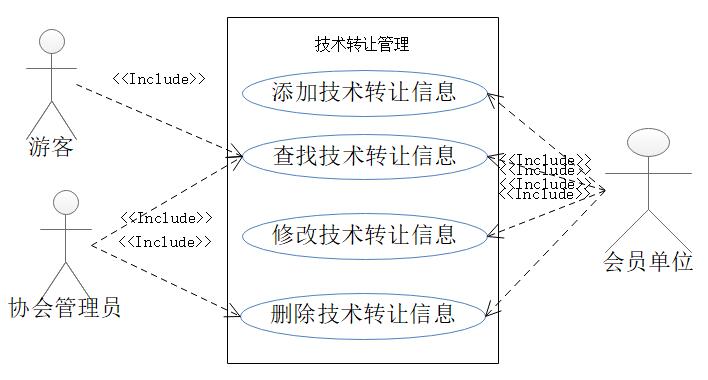


图2-10 技术转让管理用例图

表2-22 添加技术转让信息用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 添加技术转让信息 |
| 标识符\* | YL05 |
| 用例描述 | 添加技术转让信息描述了会员单位进入会员单位后台后，添加技术转让信息的操作过程。 |
| 参与者表 | 会员单位 |
| 优先级 | 1 |
| 状态\* | 进行中 |
| 前置条件 | 会员单位进入会员单位后台 |
| 后置条件 | 协会管理员和会员单位以及游客可以访问到新添加的技术转让信息 |
| 基本操作流 | 1. 会员单位进入会员单位后台 2. 进入技术转让管理模块 3. 添加技术转让信息 |

1. 项目合作管理

项目合作管理子系统包含了会员单位在会员单位后台对项目合作信息添加、查找、修改、删除操作，同时也包含了协会管理员从协会管理员后台对项目合作信息的查找与删除。以及游客在协会网站前台对项目合作信息的查询访问。用例图如图2-11所示。

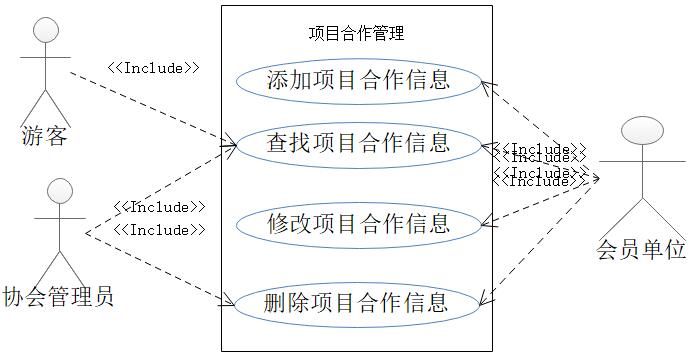
****

图2-11 项目合作管理用例图

表2-34 协会管理员删除项目合作信息用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 协会管理员删除项目合作信息 |
| 标识符\* | YL06 |
| 用例描述 | 协会管理员删除项目合作信息描述了协会管理员进入协会管理员后台后，删除项目合作信息的操作过程。 |
| 参与者表 | 协会管理员 |
| 优先级 | 1 |
| 状态\* | 进行中 |
| 前置条件 | 协会管理员进入协会管理员后台 |
| 后置条件 | 项目合作信息删除 |
| 基本操作流 | 1. 协会管理员进入协会管理员后台 2. 进入项目合作管理模块 3. 删除项目合作信息 |

1. 招聘管理

招聘管理管理子系统包含了会员单位在会员单位后台对招聘信息添加、查找、修改、删除操作，同时也包含了协会管理员从协会管理员后台对招聘信息的添加、查找与删除。以及游客在协会网站前台对招聘信息的查询访问。用例图如图2-12所示。

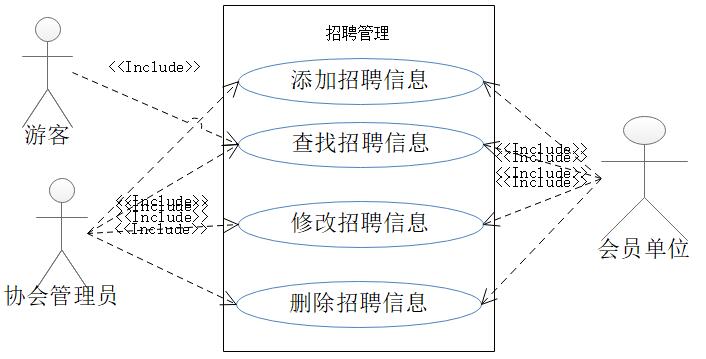


图2-12 招聘管理管理用例图

表2-41 协会管理员修改招聘信息用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 协会管理员修改招聘信息 |
| 标识符\* | YL07 |
| 用例描述 | 修改招聘信息描述了协会管理员进入协会管理员后台后，修改招聘信息的操作过程。 |
| 参与者表 | 协会管理员 |
| 优先级 | 1 |
| 状态\* | 进行中 |
| 前置条件 | 协会管理员进入协会管理员后台 |
| 后置条件 | 协会管理员和会员单位以及游客可以访问到新修改后的项目合作信息 |
| 基本操作流 | 1. 协会管理员进入协会管理员后台 2. 进入招聘管理模块 3. 修改招聘信息 |

1. 简历管理

简历管理管理子系统包含了会员单位在会员单位后台对简历信息的查找与删除操作。以及游客在协会网站招聘页面对个人简历信息的添加。用例图如图2-13所示。

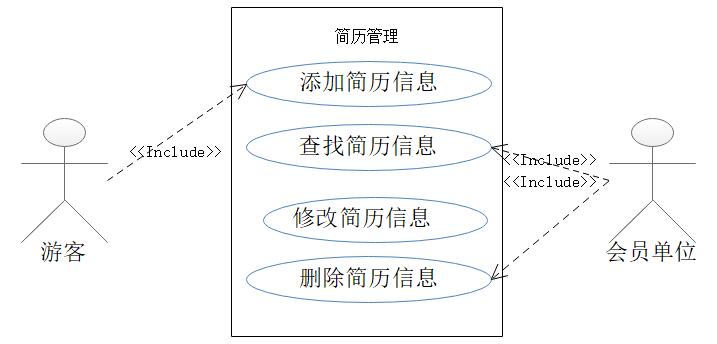


图2-13 简历管理用例图

表2-46 查找简历信息用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 查找简历信息 |
| 标识符\* | YL08 |
| 用例描述 | 查找简历信息描述了会员单位管理员在会员单位管理员管理后台对简历信息的查找访问。 |
| 参与者表 | 会员单位 |
| 优先级 | 1 |
| 状态\* | 进行中 |
| 前置条件 | 会员单位进入会员单位后台 |
| 后置条件 | 会员单位可以访问到简历信息 |
| 基本操作流 | 1. 会员单位进入会员单位后台 2. 进入简历管理模 3. 查找简历信息 |

## 2.3 非功能需求

非功能需求包括产品必须遵从的标准、规范和合约；外部界面的具体细节以及性能要求；关于设计或实现的约束条和产品的质量属性。

### 2.3.1环境需求

服务器端环境需求如表3-1所示：

表 3-1 服务器端环境

|  |  |
| --- | --- |
| 系统所使用数据库 | MySQL 5.7 |
| 操作系统 | Windows 10 |
| Web服务器 | Tomcat 8.5 |

客户端环境需求如表3-2所示：

表 3-2 客户端环境

|  |  |
| --- | --- |
| 操作系统 | Windows 7及其以上版本的操作系统 |
| 浏览器 | Firefox、Chrome、IE10.0等 |
| 分辨率 | 最佳效果为1920\*1080像素 |

### 2.3.2 性能需求

1. 响应速度

一般情况下，协会网站系统的响应时间通常不超过1.5秒，在多用户访问的高峰时段下响应时间不应当超过4秒。在推荐的环境下访问协会网站系统时，用户登入注册、刷新页面、更新资源信息的响应时间均在2秒内。

在非高峰时间，在协会网站中实现信息查询，删除，修改资源信息等功能在2秒内完成。上传添加资源信息和下载资源文件功能的运行时间，视资源大小而确定，一般响应时间不超过4秒。

1. 可靠性

在用户使用的时间内都能够保持协会网站系统性能稳定。在软件故障的情况下能够维持规定的性能水平，并能够正确的处理错误信息。在用户出现不当操作时，能够正确处理错误，并提示并指导用户进行正确的操作。

1. 吞吐量

协会网站系统中能够承载大量的用户资源信息和后台管理信息。协会网站可在给定的时间内处理一定数量的用户并发的提交请求，并保证事务计算和请求通信的稳定高效传输。

1. 可扩展性

协会网站系统的设计过程中，应当要具有一定的前瞻性。系统设计时采用“增量开发模式”，在系统设计时考虑未来可能拓展的功能，为以后添加功能模块提供接口，方便系统拓展，满足系统可扩展性。

1. 可移植性

良好的可移植性可以提高软件的生命周期。协会网站系统应具有良好的结构、灵活的体系、不应当依赖于某一特定[计算机](https://wiki.mbalib.com/wiki/%E8%AE%A1%E7%AE%97%E6%9C%BA)或操作系统。在设计与实现中采用标准化的文档资料，采用可维护的程序语言设计。

### 2.3.3 安全需求

1. 权限控制

根据不同的用户角色，设置不同的权限以达到功能的合理分配。游客权限较低，只能浏览前台页面信息或投递简历。单位会员登录会员单位账号后，可实现对单位信息管理、发布招聘信息、发布项目合作和技术转让等信息，管理员权限最高，可对前台页面数据的管理，及对会员单位所发布信息进行查阅审核删除等操作。

1. 记录日志

协会网站系统能够记录用户的关键性操作记录，同时记录协会网站系统在运行时发生的所有错误信息。以便后续协会网站系统出现故障时能够回到系统故障之前的状态。

1. 用户安全

协会网站用户设置密码需进行加密处理，不能从数据库中直接明文读取用户填入的密码数据。对用户的密码信息进行密码隐藏保护，对协会网站网站上用户登录信息的获取采用post方法传输以保证用户信息的安全。

# 3 总体设计

## 3.1系统设计的原则

在充分考虑到大数据及信息技术产业协会网站的总体规划目标，以及网站的应用环境前提下，本网站的设计将严格遵循如下设计原则：

1. 合理性原则。

在协会网站设计时，应当充分考虑协会网站的业务需求及功能的扩充，方便协会网站后期升级与改造。协会网站系统对运行环境具有较好的适应性，不依赖于特定的操作系统或特定版本的软件。

1. 经济性原则。

在满足协会网站系统功能及性能要求的前提下，尽量降低协会网站系统的设计与实现的成本，采用经济且实用的技手段与设备，综合考虑系统的建设、升级和维护成本达到最高的经济效益。

## 3.2 系统体系结构设计

南昌高新区大数据及信息技术产业协会提出，希望系统的维护简单、方便，不需要安装客户端安装就可以使用。根据客户要求和南昌高新区大数据及信息技术产业协会网站本身的特点，系统结构如图3-2所示。

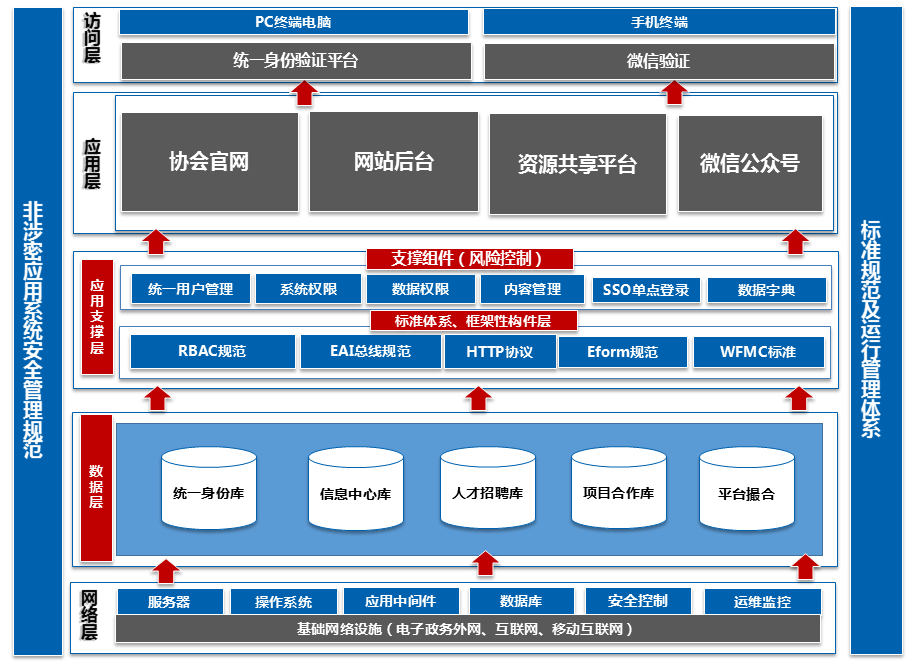


图3-2 大数据及信息技术产业协会网站部署结构图

系统一共分为五层：访问层、应用层、应用支撑层、数据层、网络层。其中数据层主要是对原始数据的操作，它的操作对象是数据库中的数据，具体为业务逻辑层或表示层提供数据服务。访问层主要表示网页，接受用户的请求，以及数据的返回，为客户端提供应用程序的访问。

项目采用SSM框架实现，并用maven统一配置。Spring作为容器属于协调上下文，管理对象间的依赖，提供事务机制。使用Spring MVC分离模型、视图、控制器以及处理程序对象的角色。利用MyBatis的XML来配置和映射数据库中的记录。

## 3.3 系统功能模块设计

系统功能架构设计定义软件系统各主要部件及成分之间的关系。系统总包图如图3-3所示。

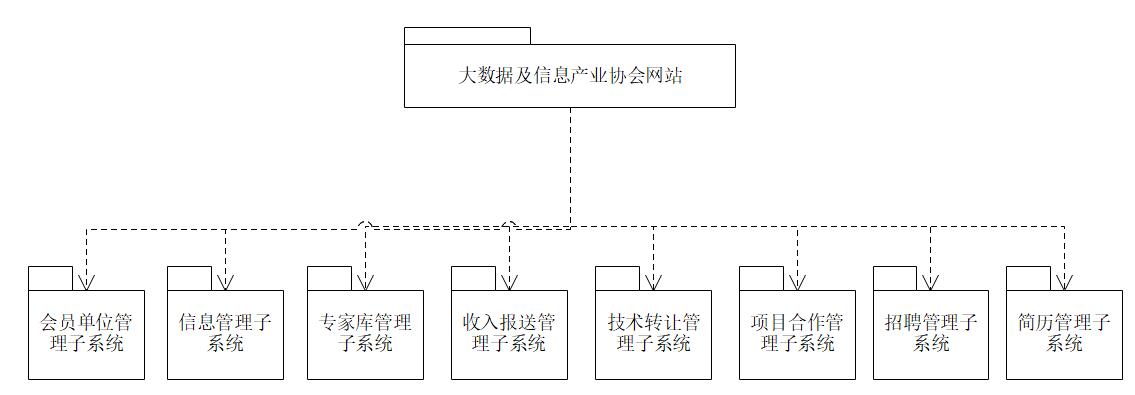


图3-3 系统总包图

协会网站系统一共分为八个子系统，分别完成协会网站的不同功能。

1. 会员单位管理子系统实现协会管理员对协会会员单位信息的添加、删除、修改、查找操作。
2. 信息管理子系统实现协会管理员对网站前台信息的添加、删除、修改、查找操作以及游客对网站前台信息的查找访问。
3. 专家库管理子系统现协会管理员对专家信息的添加、删除、修改、查找操作以及游客对专家信息的查找访问。
4. 收入报送管理子系统实现会员单位对收入报送信息的添加、删除、修改、查找操作以及协会管理员对收入报送信息的查找与删除操作。
5. 技术转让管理子系统实现会员单位对技术转让信息信息的添加、删除、修改、查找操作以及协会管理员对技术转让信息的查找与删除操作。
6. 项目合作管理子系统实现会员单位对项目合作信息信息的添加、删除、修改、查找操作以及协会管理员对项目合作的查找与删除操作。
7. 招聘管理子系统实现会员单位以及协会管理员对招聘信息的添加、删除、修改、查找操作以及游客对招聘信息的查找操作。
8. 简历管理子系统实现会员单位对简历的查找和删除操作，以及游客对简历的添加操作。

# 4 数据库设计

## 4.1概念结构设计

### 4.1.1设计思路

以游客和会员单位以及协会管理员的不同角色出发，考虑到不同信息实体所包含的属性。依据网站不同用户对各种信息实体的操作权限不同，来设定相应的实关系。

游客和协会管理员以及会员单位不同用户的角度出发，考虑每种用户所能产生的信息实体。游客用户可以投递简历信息实体，而简历实体又相应的对应着特定的招聘信息实体。招聘信息实体又来自于协会管理员用户和会员单位用户发布得到。协会管理员用户能管理专家实体以及信息实体，此外协会管理员还能管理会员单位实体。会员单位用户能提交收入报送信息实体，还能发布技术转让信息实体和项目合作实体。技术转让信息实体和项目合作实体组合对应成了平台撮合实体。

### 4.1.2 E-R图

系统总E-R图如图4-1所示：

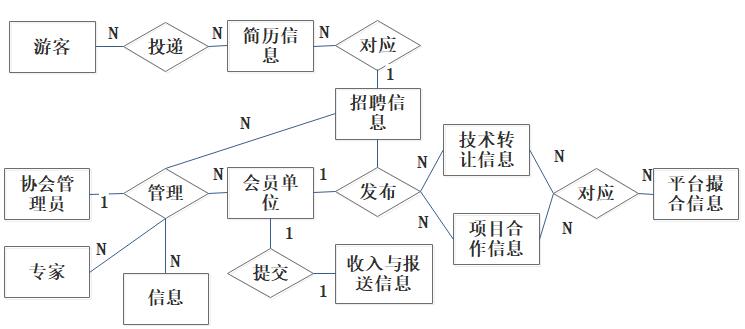


图4-1 系统总E-R图

1. 管理员实体图如图4-2所示：

实体：管理员

属性：管理员账号，管理员编号，管理员名字，密码，电话，邮箱

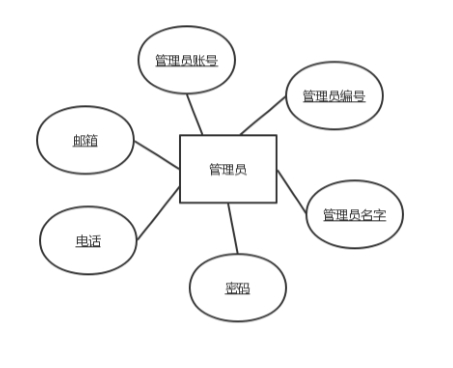


图4-2 管理员实体图

1. 会员单位实体图如图4-3所示：

实体：会员单位

属性：会员单位编号，会员单位名字，会员单位账号，会员单位密码，会员单位地址，会员单位图标，会员单位电话，会员单位介绍详情

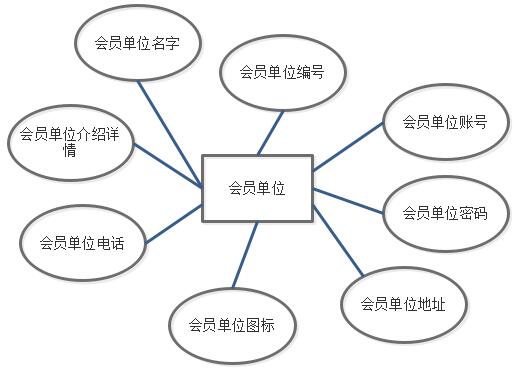


图4-3 会员单位实体图

1. 信息实体图如图4-4所示：

实体：信息

属性：信息编号，信息标题，时间，发布者编号，信息内容，信息类型，图片

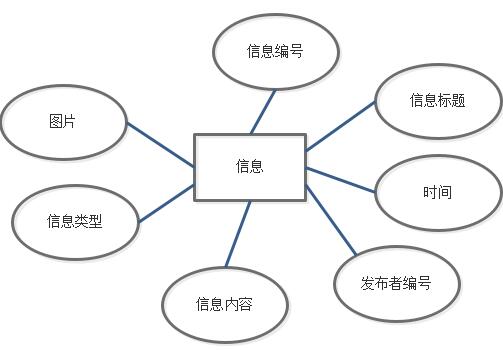


图4-4 信息实体图

1. 专家实体图如图4-5所示：

实体：专家

属性：专家名字，专家编号，专家性别，年龄，专家个人详细信息，所属领域，图片地址

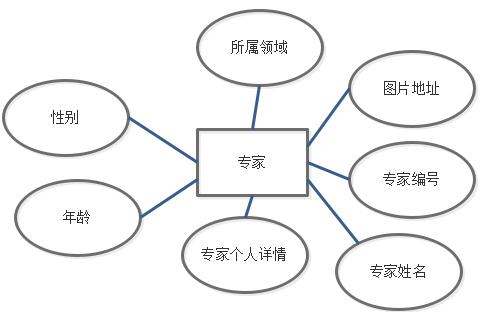


图4-5 专家实体图

1. 收入报送实体图如图4-6所示：

实体：会员单位收入

属性：收入名称，收入编号，会员单位编号，金额，时间，介绍，附件

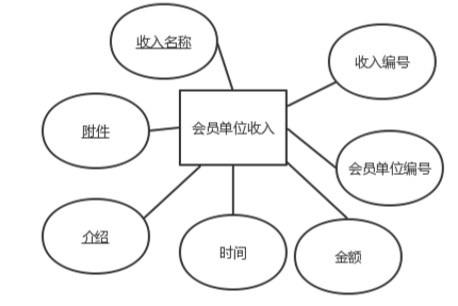


图4-6 收入报送实体图

1. 技术转让实体图如图4-7所示：

实体：技术转让

属性：类型，编号，名称，详细介绍，会员单位编号，状态，发布状态，转让公司编号

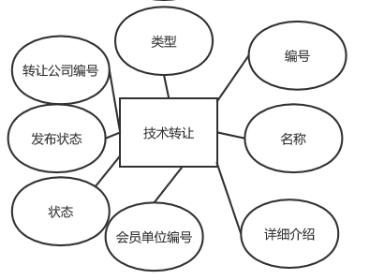


图4-7 技术转让实体图

1. 项目合作实体图如图4-8所示：

实体：项目合作

属性：类型，编号，名称，详细介绍，会员单位编号，发布状态，状态，合作公司编号

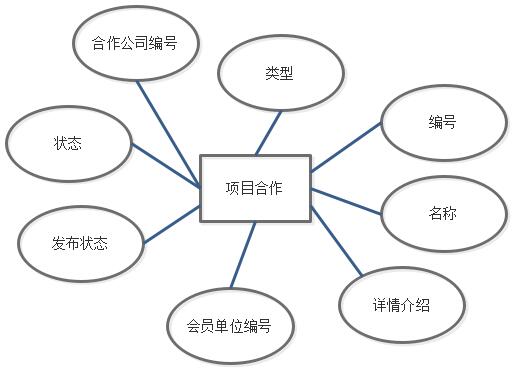


图4-8 项目合作实体图

1. 招聘信息实体图如图4-9所示：

实体：招聘信息

属性：招聘编号，岗位类型，工资，发布者，会员单位编号，状态

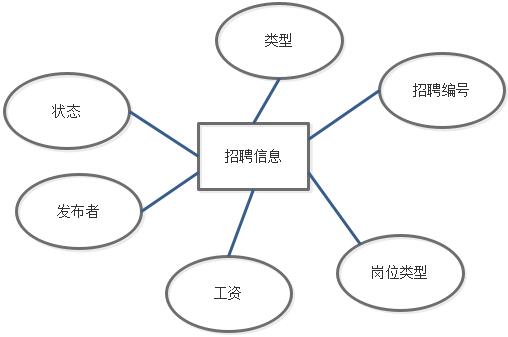


图4-9 招聘信息实体图

1. 简历实体图如图4-10所示：

实体：个人简历信息

属性：编号，名字，年龄，学历，电话，招聘类型，状态，简历附件



图4-10 简历实体图

1. 平台撮合实体图如图4-11所示：

实体：平台撮合

属性：编号，名称，详细介绍，时间，技术转让编号，项目合作编号，状态

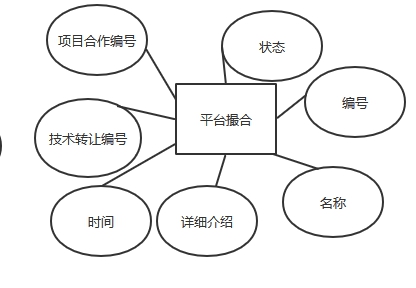


图4-11 平台撮合实体图

## 4.2 逻辑结构设计

### 4.2.1 设计思路

首先将概念结构设计中协会网站系统总E-R图转换为一般的关系模型，再根据具体DBMS的特点和限制将转换来的关系模型向特定的DBMS支持下的数据模型转换，最后进行数据模型的优化。

### 4.2.2 逻辑模型

由E-R图得到逻辑模型如下所示

1. 管理员（管理员编号、管理员账号、管理员密码、管理员名字、管理员电话、管理员邮箱）
2. 会员单位（会员单位编号、会员单位账号、会员单位密码、会员单位名字、会员单位地址、会员单位图标、会员单位电话、会员单位介绍详情）
3. 信息（信息编号、信息标题、发布者编号、发布时间、信息内容、信息类型、信息图片地址）
4. 专家（专家编号、专家名字、专家图片地址、专家详情介绍、专家年龄、专家性别、专家所属领域）
5. 收入报送（收入编号、收入名称、收入时间、金额、收入介绍、附件地址、会员单位编号）
6. 技术转让（技术转让编号、技术转让名称、技术转让介绍、会员单位编号、技术转让状态、转让公司编号、技术转让信息发布状态、技术转让类型）
7. 项目合作（项目合作编号、项目合作名称、项目合作介绍、会员单位编号、项目合作状态、合作伙伴编号、项目合作信息发布状态、项目合作类型）
8. 招聘信息（招聘信息编号、岗位类型、工资、招聘介绍、招聘状态、会员单位编号）
9. 个人简历（个人简历编号、个人名字、年龄、电话、招聘类型标号、附件地址、学历、个人简历状态）
10. 平台撮合（平台撮合编号、时间、平台撮合介绍、项目合作编号、技术转让编号、平台撮合状态）

## 4.3 物理结构设计

### 4.3.1 存取方式

根据以上的分析，本系统采用索引的存取方式对数据库进行存取，在每个表中设置一个主键的B+树索引，提高数据库的查找效率。本系统采用MySQL数据库进行存储。数据类型整形确定数据类型具体化为int类型，字符串类型确定数据类型具体化为varchar，日期时间类型确定数据类型具体化为datetime。

### 4.3.2 存储结构

1. 管理员表

管理员表的物理表设计如表4-11所示：

表4-11管理员表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键否 | 约束条件 | 中文说明 |
| id | int | 10 | 是 | 自增 | 管理员编号 |
| account | varchar | 20 | 否 | 非空 | 管理员账号 |
| psd | varchar | 20 | 否 | 非空 | 管理员密码 |
| name | varchar | 20 | 否 | 非空 | 管理员名字 |
| tel | varchar | 20 | 否 |  | 管理员电话 |
| email | varchar | 20 | 否 |  | 管理员邮箱 |

1. 会员单位表

会员单位表的物理表设计如表4-12所示：

表4-12 会员单位表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键否 | 约束条件 | 中文说明 |
| id | int | 10 | 是 | 自增 | 会员单位编号 |
| account | varchar | 20 | 否 | 非空 | 会员单位账号 |
| psd | varchar | 20 | 否 | 非空 | 会员单位密码 |
| com\_name | varchar | 20 | 否 | 非空 | 会员单位名字 |
| com\_address | varchar | 20 | 否 |  | 会员单位地址 |
| logo | varchar | 255 | 否 |  | 会员单位图标 |
| tel | varchar | 20 | 否 |  | 会员单位电话 |
| introduction | text | 1000 | 否 |  | 会员单位介绍详情 |

1. 信息表

信息表的物理表设计如表4-13所示：

表4-13 信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键否 | 约束条件 | 中文说明 |
| id | int | 10 | 是 | 自增 | 信息编号 |
| title | varchar | 225 | 否 | 非空 | 信息标题 |
| publish\_id | int | 50 | 否 | 非空 | 发布者编号 |
| time | datetime | 15 | 否 | 非空 | 发布时间 |
| context | text | 1000 | 否 |  | 信息内容 |
| Pic | varchar | 255 | 否 |  | 信息图片地址 |
| type | varchar | 10 | 否 |  | 信息类型 |

1. 专家表

专家表的物理表设计表4-14所示：

表4-14专家表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键否 | 约束条件 | 中文说明 |
| id | int | 10 | 是 | 自增 | 专家编号 |
| name | varchar | 20 | 否 | 非空 | 专家名字 |
| pic | varchar | 50 | 否 | 非空 | 专家图片地址 |
| introduction | varchar | 255 | 否 | 非空 | 专家详情介绍 |
| age | int | 10 | 否 |  | 专家年龄 |
| sex | tinyint | 1 | 否 | 默认值为0 | 性别0:未知1:男2:女 |
| field | varchar | 10 | 否 |  | 专家所属领域编号 |

1. 收入报送表

收入报送表的物理表设计如表4-15所示：

表4-15 收入报送表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键否 | 约束条件 | 中文说明 |
| id | int | 10 | 是 | 自增 | 收入编号 |
| name | varchar | 20 | 否 | 非空 | 收入名称 |
| time | datetime | 20 | 否 | 非空 | 收入时间 |
| money | decimal | 20 | 否 |  | 金额 |
| introduction | varchar | 255 | 否 |  | 收入介绍 |
| url | varchar | 255 | 否 |  | 附件地址 |
| com\_id | int | 10 | 否 | 非空 | 会员单位编号 |

1. 技术转让表

技术转让表的物理表设计如表4-16所示：

表4-16 技术转让表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键否 | 约束条件 | 中文说明 |
| id | int | 10 | 是 | 自增 | 编号 |
| name | varchar | 20 | 否 | 非空 | 技术转让名称 |
| introduction | text | 1000 | 否 |  | 技术转让介绍 |
| com\_id | int | 10 | 否 | 非空 | 会员单位编号 |
| status | varchar | 10 | 否 |  | 技术转让状态 |
| transfer\_id | int | 10 | 否 |  | 转让公司编号 |
| check\_status | varchar | 10 | 否 | 非空 | 发布状态， |

续表4-16

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| type | varchar | 20 | 否 |  | 技术转让类型 |

1. 项目合作表

项目合作表的物理表设计如表4-17所示：

表4-17 项目合作表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键否 | 约束条件 | 中文说明 |
| id | int | 10 | 是 | 自增 | 编号 |
| name | varchar | 20 | 否 | 非空 | 项目合作名称 |
| Introduction | varchar | 255 | 否 |  | 项目合作介绍 |
| com\_id | int | 10 | 否 | 非空 | 会员单位编号 |
| status | varchar | 10 | 否 |  | 项目合作状态 |
| partner\_id | int | 10 | 否 |  | 合作伙伴编号 |
| check\_status | varchar | 10 | 否 | 非空 | 发布状态 |
| type | varchar | 20 | 否 |  | 项目合作类型 |

1. 招聘信息表

招聘信息表的物理表设计如表4-18所示：

表4-18 招聘信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键否 | 约束条件 | 中文说明 |
| id | int | 10 | 是 | 非空，自增 | 招聘编号 |
| job\_type | vachar | 255 | 否 | 非空 | 岗位类型 |
| salary | decimal | 20 | 否 |  | 工资 |
| introduction | text | 100 | 否 |  | 招聘介绍 |
| check\_status | tinyint | 1 | 否 | 非空，默认值为0 | 状态 0:正常1:禁用 |
| com\_id | int | 10 | 否 | 非空 | 会员单位编号 |

1. 个人简历表

个人简历表的物理表设计如表4-19所示：

表4-19 个人简历表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键否 | 约束条件 | 中文说明 |
| id | int | 10 | 是 | 非空，自增 | 编号 |
| Name | varchar | 20 | 否 | 非空 | 个人名字 |
| age | varchar | 3 | 否 | 非空 | 年龄 |
| Tel | varchar | 20 | 否 | 非空 | 电话 |
| recruit\_id | int | 10 | 否 | 非空 | 招聘类型标号 |

续表4-19

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| url | varchar | 255 | 否 |  | 附件地址 |
| degree | varchar | 255 | 否 | 非空 | 学历 |

1. 平台撮合表

平台撮合表的物理表设计如表4-20所示：

表4-20 项目合作表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键否 | 约束条件 | 中文说明 |
| id | int | 10 | 是 | 非空，自增 | 编号 |
| status | varchar | 255 | 否 |  | 状态 |
| introduction | varchar | 50 | 否 |  | 介绍 |
| project\_id | int | 10 | 否 |  | 项目合作编号 |
| technology\_id | int | 10 | 否 |  | 技术转让编号 |
| Time | datetime | 20 | 否 |  | 时间 |

# 5 界面设计

## 5.1 界面关系图或工作流图

用户首先访问协会官网，可以查看新闻列表信息以及新闻详情等前台页面信息。如果用户是协会管理员并登录了协会管理员后台，则可以访问到协会管理员后台并访问到专家库管理等页面。如果用户是会员单位并登录了会员单位后台，则可以访问到会员单位后台并访问到招聘管理等页面。界面关系图如图5-1所示。

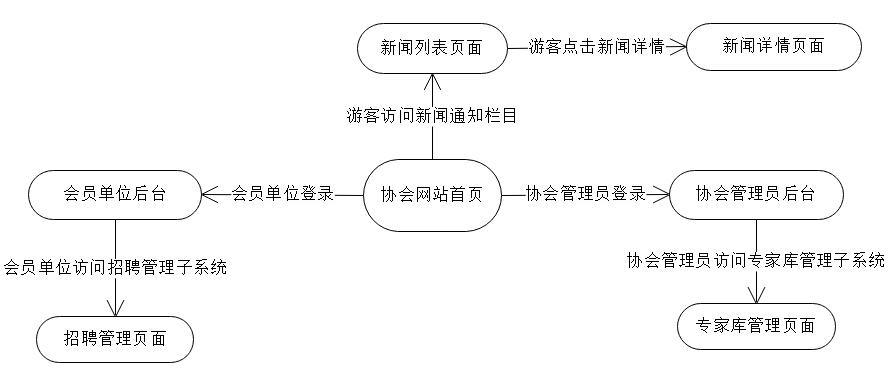


图5-1界面关系图

## 5.2 界面设计成果

### 5.2.1 主界面

1. 协会网站首页

界面以蓝白相衬作为背景，界面上部采用导航栏分隔不同栏目，界面中部以轮播图作为新闻信息的展示，下部以块状组件的方式分别展示不同栏目的信息简介。协会网站首页主界面如图5-2所示。



图5-2 协会网站首页主界面

1. 协会网站后台登录界面

协会网站后台登录界面为协会管理员和会员单位提供登录到协会管理员或会员单位后台的窗口。界面保留了蓝色的背景色，界面左部是大数据的形象图，界面右部是登录窗口用于账号密码的验证。协会网站后台登录界面如图5-3所示。



图5-3 协会网站后台登录界面

1. 协会管理员后台

协会管理员后台界面展示了以协会概况为首页的欢迎界面，以及协会管理员后台的各个模块的菜单列表展示。后台使用easyui完成设计，界面左部通过竖型导航栏展示不同管理模块，界面右部展示模块的具体细节。协会管理员后台界面如图5-4所示。



图5-4 协会管理员后台界面

1. 会员单位后台

会员单位后台界面展示了以名言“天时不如地利，地利不如人和”修饰的欢迎界面，以及会员单位后台的各个模块的菜单列表展示。界面左部通过竖型导航栏展示不同管理模块，界面右部展示模块的具体细节。会员单位后台界面如图5-4所示。



图5-5 会员单位后台界面

### 5.2.2 子界面

1. 新闻通知列表

新闻通知列表展示了游客在协会网站前台访问新闻通知栏目后，查看到的新闻通知列表信息。保留了界面上部的导航栏以及蓝色的背景色不变，不同的新闻通知以不同的块状信息表示。新闻通知列表界面图如图5-6所示。



图5-6 新闻通知列表界面图

1. 新闻通知详情

新闻通知详情展示了游客在新闻通知列表信息总，选取一条新闻通知跳转后查看到的新闻通知详情页面。先展示了新闻图片，接着大字号展示新闻标题，最后放入新闻详情在界面下方。新闻通知详情界面图如图5-7所示。



图5-7 新闻通知详情界面图

1. 专家库管理

专家库管理界面展示了协会管理员进入协会管理员后台后，选取专家库管理模块后查看到的页面。详情页面在界面右部，位于页面左上方的新增和刷新按钮便于用户操作。专家信息以列表的形式展现，并在每一行最后附上具体操作按钮。专家库管理界面图如图5-8所示。



图5-8 专家库管理界面图

1. 招聘管理

招聘管理界面展示了会员单位进入会员单位后台后，选取招聘管理模块后查看到的页面。详情页面在界面右部，位于页面左上方的新增和刷新按钮便于用户操作。招聘信息以列表的形式展现，并在每一行最后附上具体操作按钮。招聘管理界面图如图5-9所示。



图5-9 招聘管理界面图

# 6 详细设计

## 6.1系统主要功能模块介绍

协会网站系统一共分为八个子系统，分别完成协会网站的不同功能。

1. 会员单位管理子系统实现协会管理员对协会会员单位信息的添加、删除、修改、查找操作。
2. 信息管理子系统实现协会管理员对网站前台信息的添加、删除、修改、查找操作以及游客对网站前台信息的查找访问。
3. 专家库管理子系统现协会管理员对专家信息的添加、删除、修改、查找操作以及游客对专家信息的查找访问。
4. 收入报送管理子系统实现会员单位对收入报送信息的添加、删除、修改、查找操作以及协会管理员对收入报送信息的查找与删除操作。
5. 技术转让管理子系统实现会员单位对技术转让信息信息的添加、删除、修改、查找操作以及协会管理员对技术转让信息的查找与删除操作。
6. 项目合作管理子系统实现会员单位对项目合作信息信息的添加、删除、修改、查找操作以及协会管理员对项目合作的查找与删除操作。
7. 招聘管理子系统实现会员单位以及协会管理员对招聘信息的添加、删除、修改、查找操作以及游客对招聘信息的查找操作。
8. 简历管理子系统实现会员单位对简历的查找和删除操作，以及游客对简历的添加操作。

## 6.2会员单位管理

### 6.2.1 功能结构设计

会员单位管理功能主要有四项操作：对会员单位用户信息的增删改查。增加操作即当有新的会员单位加入协会时即插入一条该单位的信息。删除操作即当会员单位离开协会，则删除该单位的账号信息。管理员和会员单位用户同时拥有修改会员单位信息的权限。查询操作指管理员可通过id查询到会员单位信息。会员单位管理包图如图6-1所示。

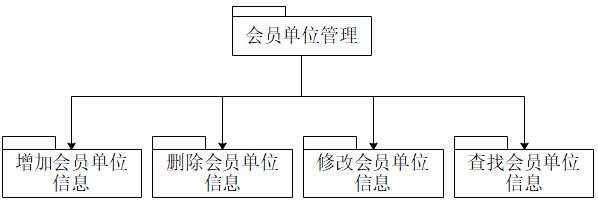


图6-1 会员单位管理包图

### 6.2.2 类图设计

会员单位管理类设计如图6-2所示。

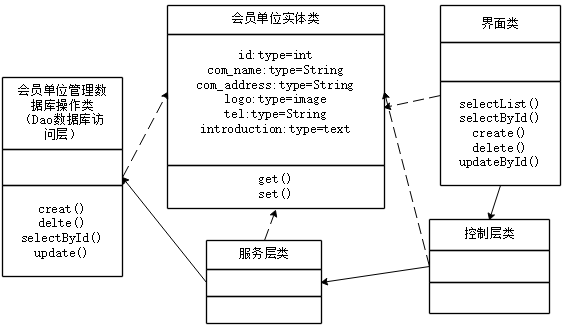


图6-2 会员单位管理类图

会员单位用户实体类描述如表6-1所示。

表6-1 会员单位用户实体类

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 01 | | 类名称（中文） | | 会员单位用户实体类 | | |
| 类功能描述： | 会员单位的实体类对象 | | | | | | |
| 类属性 | 属性类型 | 属性名称 | | 属性描述 | | 其他 |  |
|  | Int | id | | 用户id | |  |  |
|  | String | account | | 账户用户名 | |  |  |
|  | String | Psd | | 密码 | |  |  |
|  | String | Com\_name | | 公司名字 | |  |  |
|  | String | Com\_addres | | 公司地址 | |  |  |
|  | image | Logo | | 公司logo | |  |  |
|  | String | Tel | | 公司电话 | |  |  |
|  | text | Introduction | | 公司介绍 | |  |  |
| 主要实现方法 | getter();setter() | | | | | | |

会员单位管理界面类描述如表6-2所示。

表6-2 会员单位管理界面类

|  |  |  |  |
| --- | --- | --- | --- |
| 类代号（code） | 02 | 类名称（中文） | 会员单位管理界面类 |
| 类功能描述： | 负责会员单位信息的管理操作 | | |

续表6-2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方法属性 | 方法名 | 输入 | 输出 | 方法功能描述 |
|  | selectList() |  |  | 列表展示会员单位信息 |
|  | create() |  |  | 创建会员单位信息 |
|  | delete() |  |  | 删除会员单位信息 |
|  | Select() |  |  | 查询会员单位信息 |
|  | update() |  |  | 更新会员单位信息 |

会员单位管理控制器类描述如表6-3所示。

表6-3 会员单位管理控制器类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 03 | | 类名称（中文） | | | 会员单位管理控制器类 |
| 类功能描述： | 负责会员单位信息相关请求的处理和转发 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | selectList() |  | |  | 列表展示会员单位信息 | | |
|  | create() |  | |  | 创建会员单位信息 | | |
|  | delete() |  | |  | 删除会员单位信息 | | |
|  | Select() |  | |  | 查询会员单位信息 | | |
|  | update() |  | |  | 更新会员单位信息 | | |

会员单位管理服务层类描述如表6-4所示。

表6-4 会员单位管理服务层类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 04 | | 类名称（中文） | | | 会员单位管理服务层类 |
| 类功能描述： | 负责会员单位信息管理的服务层方法封装 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | selectList() |  | |  | 列表展示会员单位信息 | | |
|  | create() |  | |  | 创建会员单位信息 | | |
|  | delete() |  | |  | 删除会员单位信息 | | |
|  | Select() |  | |  | 查询会员单位信息 | | |
|  | update() |  | |  | 更新会员单位信息 | | |

会员单位信息数据库操作类描述如表6-5所示。

表6-5 会员单位信息数据库操作类

|  |  |  |  |
| --- | --- | --- | --- |
| 类代号（code） | 05 | 类名称（中文） | 会员单位信息数据库操作类 |
| 类功能描述： | 会员单位信息数据库的数据进行管理操作 | | |

续表6-5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方法属性 | 方法名 | 输入 | 输出 | 方法功能描述 |
|  | create() |  |  | 创建用户信息 |
|  | delete() |  |  | 创建会员单位信息 |
|  | Select() |  |  | 删除会员单位信息 |
|  | update() |  |  | 查询会员单位信息 |

### 6.2.3 顺序图设计

协会管理员添加会员单位信息活动顺序图如图6-3所示。

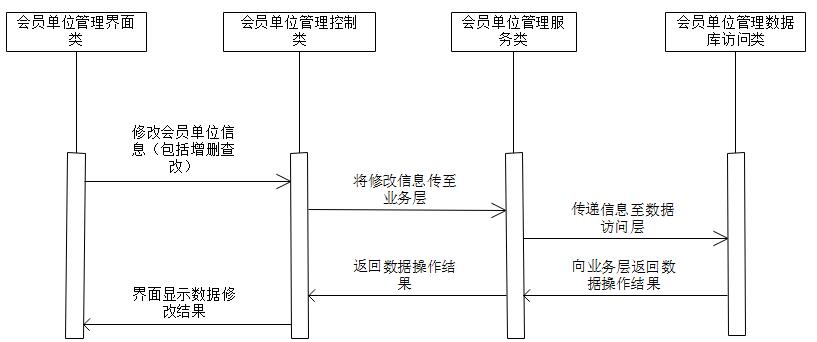


图6-3 会员单位管理顺序图

### 6.2.4 核心处理流程设计

会员单位管理子系统中添加会员单位信息功能流程图如图6-4所示。

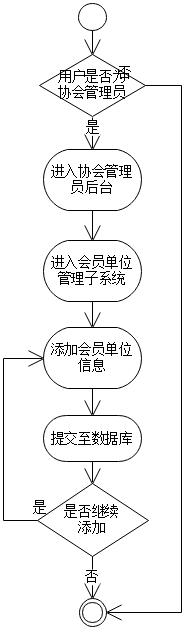


图6-4 添加会员单位信息流程图

## 6.3信息管理

### 6.3.1 功能结构设计

信息管理功能主要有四项操作：对网站前台信息的增删改查。增加即当有新的前台信息要发布时即插入新发布的前台信息。删除即当前台信息失效时，则删除该前台信息。当发布的前台信息有误时，可以对前台信息完成修改操作。查询指管理员以及游客可查询前台信息。信息管理包图如图6-5所示。

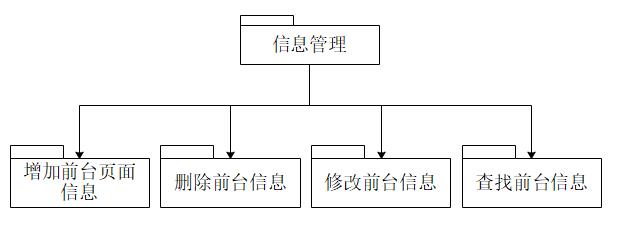


图6-5 信息管理包图

### 6.3.2 类图设计

信息管理类设计如图6-6所示。

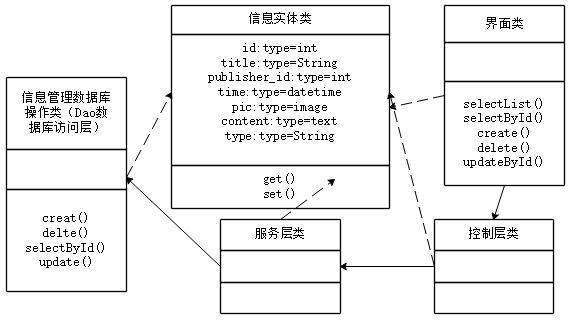


图6-6 信息管理类图

网站前台信息实体类描述如表6-6所示。

表6-6 网站前台信息实体类

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 06 | | 类名称（中文） | | 网站前台信息实体类 | | |
| 类功能描述： | 负责对网站前台信息实体类对象 | | | | | | |
| 类属性 | 属性类型 | 属性名称 | | 属性描述 | | 其他 |  |
|  | Int | id | | Id | |  |  |
|  | String | title | | 标题 | |  |  |
|  | Int | publisher\_id | | 信息发布者id | |  |  |
|  | datetime | time | | 发布时间 | |  |  |
|  | image | pic | | 信息附属图片 | |  |  |
|  | Text | content | | 信息文字 | |  |  |
|  | String | type | | 信息类型 | |  |  |
| 主要实现方法 | getter();setter() | | | | | | |

网站前台信息管理界面类描述如表6-7所示。

表6-7 网站前台信息管理界面类

|  |  |  |  |
| --- | --- | --- | --- |
| 类代号（code） | 07 | 类名称（中文） | 网站前台信息管理界面类 |
| 类功能描述： | 负责对网站前台信息进行管理操作 | | |

续表6-7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方法属性 | 方法名 | 输入 | 输出 | 方法功能描述 |
|  | selectList() |  |  | 列表展示页面信息 |
|  | create() |  |  | 创建页面信息 |
|  | delete() |  |  | 删除页面信息 |
|  | Select() |  |  | 查询页面信息 |
|  | update() |  |  | 更新页面信息 |

网站前台信息管理控制器类描述如表6-8所示。

表6-8 网站前台信息管理控制器类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 08 | | 类名称（中文） | | | 网站前台信息管理控制器类 |
| 类功能描述： | 负责对网站前台信息请求的处理和转发 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | selectList() |  | |  | 列表展示页面信息 | | |
|  | create() |  | |  | 创建页面信息 | | |
|  | delete() |  | |  | 删除页面信息 | | |
|  | Select() |  | |  | 查询页面信息 | | |
|  | update() |  | |  | 更新页面信息 | | |

网站前台信息服务层类描述如表6-9所示。

表6-9 网站前台信息服务层类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 09 | | 类名称（中文） | | | 网站前台信息服务层类 |
| 类功能描述： | 负责对网站前台信息管理操作的服务层方法的封装 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | selectList() |  | |  | 列表展示页面信息 | | |
|  | create() |  | |  | 创建页面信息 | | |
|  | delete() |  | |  | 删除页面信息 | | |
|  | Select() |  | |  | 查询页面信息 | | |
|  | update() |  | |  | 更新页面信息 | | |

网站前台信息数据库操作类描述如表6-10所示。

表6-10 网站前台信息数据库操作类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 10 | | 类名称（中文） | | | 网站前台信息数据库操作类 |
| 类功能描述： | 负责对网站前台信息数据库的数据进行管理操作 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |

续表6-10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | create() |  |  | 创建页面信息 |
|  | delete() |  |  | 删除页面信息 |
|  | Select() |  |  | 查询页面信息 |
|  | update() |  |  | 更新页面信息 |

### 6.3.3 顺序图设计

协会管理员添加网站前台信息活动顺序图如图6-7所示。

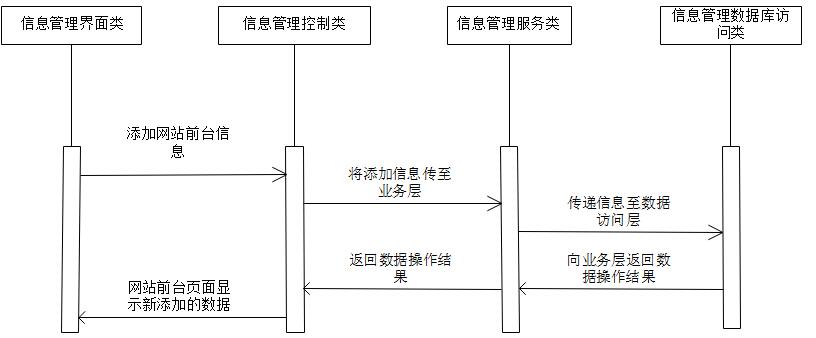


图6-7 信息管理顺序图

### 6.3.4 核心处理流程设计

信息管理子系统中添加网站前台信息功能流程图如图6-8所示。

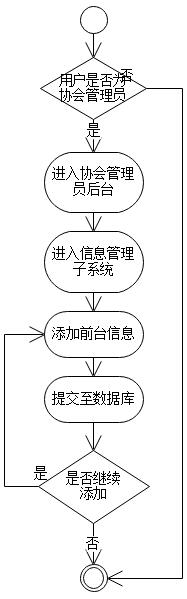


图6-8 信息管理流程图

## 6.4专家库管理

### 6.4.1 功能结构设计

专家库管理功能主要有四项操作：协会管理员对专家库信息的增删改查。增加即当有新专家作为协会顾问时，则插入新专家信息至专家库中。删除即当专家辞去协会顾问时，则在专家库中删除该专家信息。当发布专家信息有误时，可以对专家库中的专家信息完成修改操作；查询指管理员以及游客可查询专家库中的专家息。专家管理包图如图6-9所示。

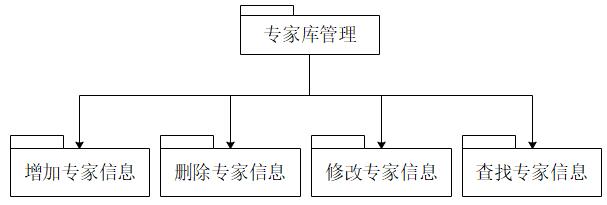


图6-9 专家库管理包图

### 6.4.2 类图设计

专家库管理类设计如图6-10所示。

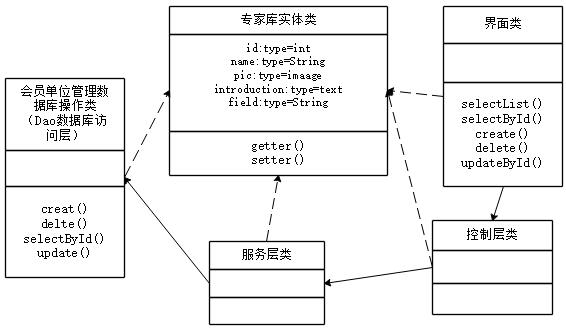


图6-10 专家库管理类图

专家实体类描述如表6-11所示。

表6-11专家实体类

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 11 | | 类名称（中文） | | 专家实体类 | | |
| 类功能描述： | 专家库实体类对象 | | | | | | |
| 类属性 | 属性类型 | 属性名称 | | 属性描述 | | 其他 |  |
|  | Int | id | | 识别专家id | |  |  |
|  | String | Name | | 专家名字 | |  |  |
|  | Image | Pic | | 专家照片 | |  |  |
|  | Text | Introduction | | 专家简介 | |  |  |
|  | String | Field | | 所属领域 | |  |  |
| 主要实现方法 | getter();setter() | | | | | | |

专家库管理界面类描述如表6-12所示。

表6-12 专家库管理界面类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 12 | | 类名称（中文） | | | 专家库管理界面类 |
| 类功能描述： | 负责专家库信息的管理操作 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | selectList() |  | |  | 列表展示专家信息 | | |
|  | create() |  | |  | 创建专家信息 | | |
|  | delete() |  | |  | 删除专家信息 | | |
|  | Select() |  | |  | 查询专家信息 | | |
|  | update() |  | |  | 更新专家信息 | | |

专家库管理控制器类描述如表6-13所示。

表6-13 专家库管理控制器类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 13 | | 类名称（中文） | | | 专家库管理控制器类 |
| 类功能描述： | 负责专家库信息相关请求的处理和转发 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | selectList() |  | |  | 列表展示专家信息 | | |
|  | create() |  | |  | 创建专家信息 | | |
|  | delete() |  | |  | 删除专家信息 | | |
|  | Select() |  | |  | 查询专家信息 | | |
|  | update() |  | |  | 更新专家信息 | | |

专家库管理服务层类描述如表6-14所示。

表6-14 专家库管理服务层类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 14 | | 类名称（中文） | | | 专家库管理服务层类 |
| 类功能描述： | 负责专家库信息管理的服务层方法封装 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | selectList() |  | |  | 列表展示专家信息 | | |
|  | create() |  | |  | 创建专家信息 | | |
|  | delete() |  | |  | 删除专家信息 | | |
|  | Select() |  | |  | 查询专家信息 | | |
|  | update() |  | |  | 更新专家信息 | | |

专家库信息数据库操作类描述如表6-15所示。

表6-15 专家库信息数据库操作类

|  |  |  |  |
| --- | --- | --- | --- |
| 类代号（code） | 15 | 类名称（中文） | 专家库信息数据库操作类 |

续表6-15

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类功能描述： | 专家库信息数据库的数据进行管理操作 | | | |
| 方法属性 | 方法名 | 输入 | 输出 | 方法功能描述 | |
|  | create() |  |  | 创建专家信息 | |
|  | delete() |  |  | 删除专家信息 | |
|  | Select() |  |  | 查询专家信息 | |
|  | update() |  |  | 更新专家信息 | |

### 6.4.3 顺序图设计

协会管理员添加专家信息活动顺序图如图6-11所示。

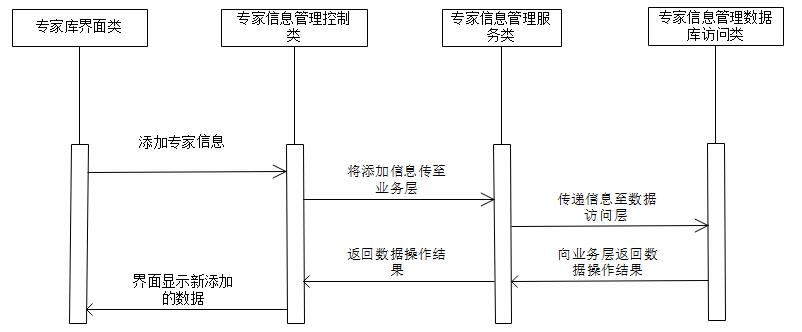


图6-11 专家库管理顺序图

### 6.4.4 核心处理流程设计

专家库子系统中添加专家信息功能流程图如图6-12所示。

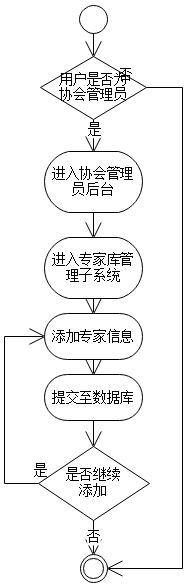


图6-12 专家库管理流程图

## 6.5收入与报送管理

### 6.5.1 功能结构设计

收入与报送管理功能主要有三项操作：对收入与报送信息的增删查。增加操作只能由会员单位在会员单位后台完成。会员单位按期在后台提交收入与报送信息。协会管理员和会员单位都具有删除收入与报送信息的权限，会员单位只能删除本单位提交的收入与报送信息，而协会管理员能删除协会全体会员单位提交的收入与报送信息。查询操作指的是会员单位只能查询到本单位提交的收入与报送信息，而协会管理员能查到协会全体会员单位提交的收入与报送信息。收入报送管理包图如图6-13所示。

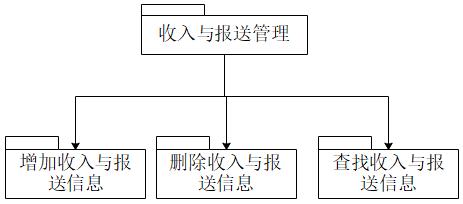


图6-13 收入与报送管理包图

### 6.5.2 类图设计

收入与报送管理类设计如图6-14所示。

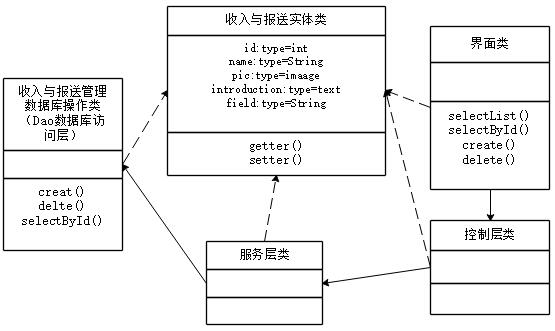


图6-14 收入与报送管理类图

收入与报送实体类描述如表6-16所示。

表6-16 收入与报送实体类

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 16 | | 类名称（中文） | | 收入与报送实体类 | | |
| 类功能描述： | 收入与报送实体类对象 | | | | | | |
| 类属性 | 属性类型 | 属性名称 | | 属性描述 | | 其他 |  |
|  | int | id | | 收入报表编号 | |  |  |
|  | String | Name | | 报表标题名字 | |  |  |
|  | String | Time | | 时间 | |  |  |
|  | String | Money | | 收入 | |  |  |
|  | String | Introduction | | 详情简报 | |  |  |
|  | image | url | | 上传 | |  |  |
|  | String | Com\_id | | 对应公司id | |  |  |
| 主要实现方法 | getter();setter() | | | | | | |

收入与报送管理界面类描述如表6-17所示。

表6-17 收入与报送管理界面类

|  |  |  |  |
| --- | --- | --- | --- |
| 类代号（code） | 17 | 类名称（中文） | 收入与报送管理界面类 |
| 类功能描述： | 收入与报送信息的管理操作 | | |

续表6-17

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方法属性 | 方法名 | 输入 | 输出 | 方法功能描述 |
|  | selectList() |  |  | 列表展示收入报送信息 |
|  | create() |  |  | 创建一条收入报送信息 |
|  | delete() |  |  | 删除一条收入报送信息 |
|  | Select() |  |  | 查询一条收入报送信息 |
|  | update() |  |  | 更新一条收入报送信息 |

收入与报送管理控制器类描述如表6-18所示。

表6-18 收入与报送管理控制器类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 18 | | 类名称（中文） | | | 收入与报送管理控制器类 |
| 类功能描述： | 收入与报送信息相关请求的处理和转发 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | selectList() |  | |  | 列表展示收入报送信息 | | |
|  | create() |  | |  | 创建一条收入报送信息 | | |
|  | delete() |  | |  | 删除一条收入报送信息 | | |
|  | Select() |  | |  | 查询一条收入报送信息 | | |
|  | update() |  | |  | 更新一条收入报送信息 | | |

收入与报送管理服务层类描述如表6-19所示。

表6-19 收入与报送管理服务层类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 19 | | 类名称（中文） | | | 收入与报送管理服务层类 |
| 类功能描述： | 收入与报送信息管理的服务层方法封装 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | selectList() |  | |  | 列表展示报送信息 | | |
|  | create() |  | |  | 创建一条收入报送信息 | | |
|  | delete() |  | |  | 删除一条收入报送信息 | | |
|  | Select() |  | |  | 查询一条收入报送信息 | | |
|  | update() |  | |  | 更新一条收入报送信息 | | |

收入与报送信息数据库操作类描述如表6-20所示。

表6-20 收入与报送信息数据库操作类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 20 | | 类名称（中文） | | | 收入与报送信息数据库操作类 |
| 类功能描述： | 收入与报送信息数据库的数据进行管理操作 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | create() |  | |  | 创建一条收入报送信息 | | |

续表6-20

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | delete() |  |  | 删除一条收入报送信息 |
|  | Select() |  |  | 查询一条收入报送信息 |
|  | update() |  |  | 更新一条收入报送信息 |

### 6.5.3 顺序图设计

会员单位添加收入与报送信息活动顺序图如图6-15所示。

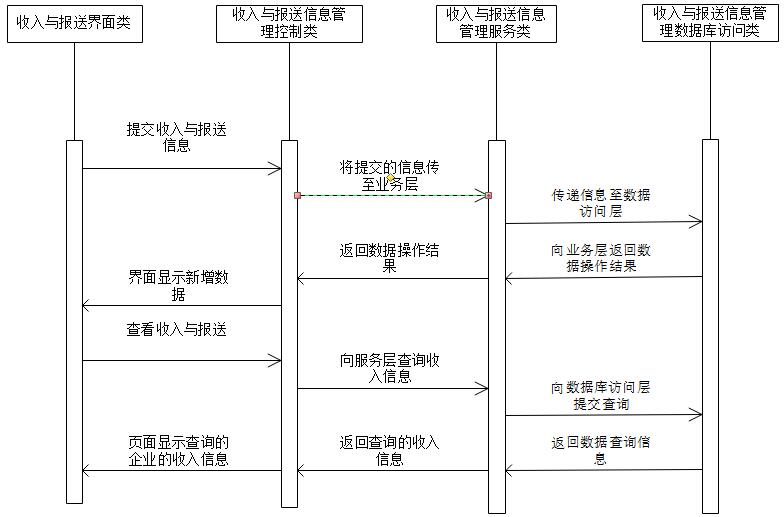


图6-15 收入与报送管理顺序图

### 6.5.4 核心处理流程设计

收入与报送管理子系统中添加收入与报送信息功能流程图如图6-16所示。

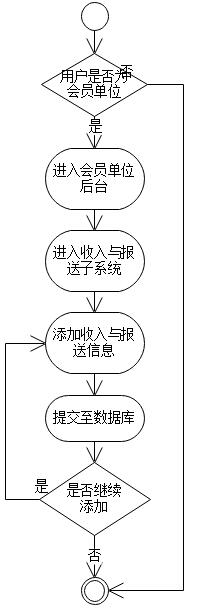


图6-16 收入与报送管理流程图

## 6.6技术转让管理

### 6.6.1 功能结构设计

技术转让管理功能主要有四项操作：对技术转让信息的增删改查。增加操作会员单位有新的技术可供转让时，即插入一条新的技术转让信息。协会管理员和会员单位都具有删除技术转让信息的权限，会员单位只能删除本单位发布的技术转让信息，而协会管理员能删除协会全体会员单位发布的技术转让信息。查询操作指的是会员单位只能查询到本单位发布的技术转让信息，而协会管理员能查到协会全体会员单位发布的技术转让信息。当发布的技术转让信息有误时，会员单位可以对本单位发布的技术转让信息完成修改操作。技术转让管理包图如图6-17所示。

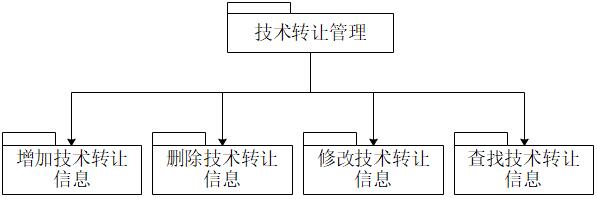


图6-17 技术转让管理包图

### 6.6.2 类图设计

技术转让管理类设计如图6-18所示。

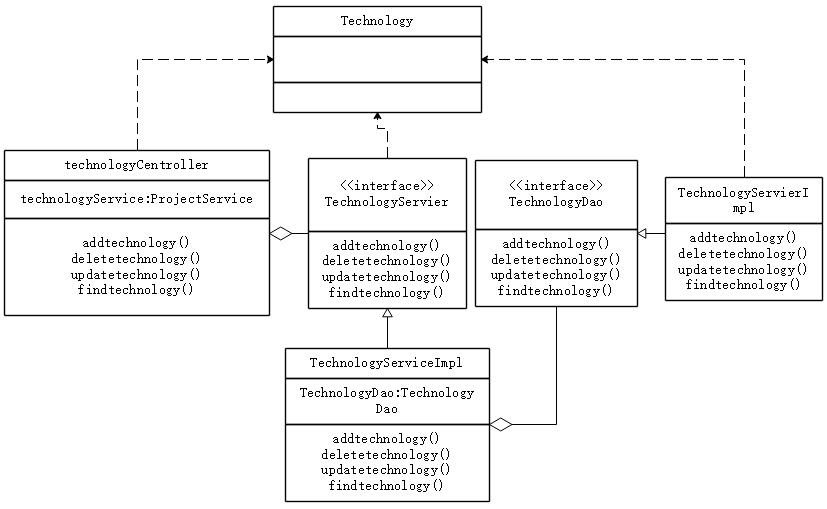


图6-18 技术转让管理类图

技术转让控制类描述如表6-21所示。

表6-21 技术转让控制类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 21 | | 类名称（中文） | | | 技术转让控制类 |
| 类功能描述： | 负责前台页面的请求，负责请求处理和转发 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | addTechnology(Technology technology) |  | |  | 将前台页面传入的技术转让实体类加入方法中，并调用服务层 | | |
|  | deleteTechnology(int id) |  | |  | 将前台页面传入的技术转让编号加入方法中，并调用服务层 | | |
|  | updateTechnologyt(Technology technology) |  | |  | 将前台页面传入的技术转让类加入方法中，并调用服务层 | | |
|  | findTechnology(Map<String,Object> map) |  | |  | 将前台页面传入的某些数据加入方法中，并调用服务层 | | |

项目合作服务类描述如表6-22所示。

表6-22 技术转让服务类

|  |  |  |  |
| --- | --- | --- | --- |
| 类代号（code） | 22 | 类名称（中文） | 技术转让服务类 |
| 类功能描述： | 负责技术转让相关操作的服务层方法封装 | | |

续表6-22

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方法属性 | 方法名 | 输入 | 输出 | 方法功能描述 |
|  | addTechnology(Technology technology) |  |  | 将控制器类传入的技术转让类加入方法中，并调用dao层 |
|  | deleteTechnology(int id) |  |  | 将控制器类传入的技术转让编号加入方法中，并调用dao层 |
|  | updateTechnologyt(Technology technology) |  |  | 将控制器类传入的技术转让类加入方法中，并调用dao层 |
|  | findTechnology(Map<String,Object> map) |  |  | 将控制器类传入的某些数据加入方法中，并调用dao层 |

技转让信息数据库操作类描述如表6-23所示。

表6-23 技术转让信息数据库操作类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 23 | | 类名称（中文） | | | 技术转让信息数据库操作类 |
| 类功能描述： | 负责技术转让信息相关操作的数据库层方法封装 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | addTechnology(Technology technology) |  | |  | 将服务器类传入的技术转让类加入方法中，并调用mapping映射，进行数据插入 | | |
|  | deleteTechnology(int id) |  | |  | 将服务器类传入的技术转让编号加入方法中，并调用mapping映射，进行数据删除 | | |
|  | updateTechnologyt(Technology technology) |  | |  | 将服务器类传入的技术转让类加入方法中，并调用mapping映射，根据项目主键，进行数据更新 | | |
|  | findTechnology(Map<String,Object> map) |  | |  | 将服务器类传入的map集合加入方法中，并调用mapping映射，进行数据查询 | | |

### 6.6.3 顺序图设计

会员单位添加技术转让信息活动顺序图如图6-19所示。

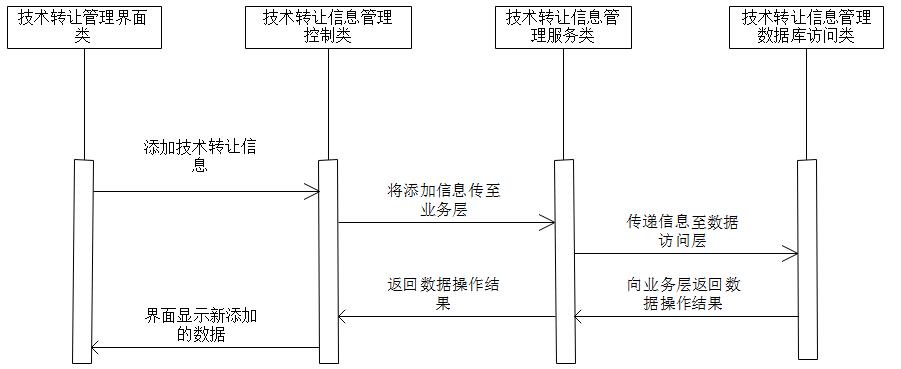


图6-19 技术转让管理顺序图

### 6.6.4 核心处理流程设计

技术转让管理子系统中添加技术转让信息功能流程图如图6-20所示。

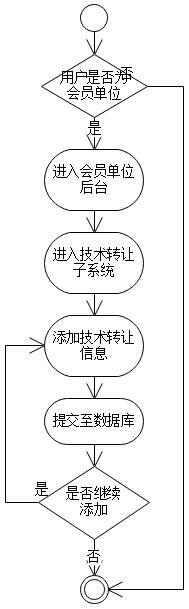


图6-20 技术转让管理流程图

## 6.7项目合作管理

### 6.7.1 功能结构设计

项目合作管理功能主要有四项操作：对项目合作信息的增删改查。增加操作会员单位有新的项目需要合作时，即插入一条新的项目合作信息。协会管理员和会员单位都具有删除项目合作信息的权限，会员单位只能删除本单位发布的项目合作信息，而协会管理员能删除协会全体会员单位发布的项目合作信息。查询操作指的是会员单位只能查询到本单位发布的项目合作信息，而协会管理员能查到协会全体会员单位发布的项目合作信息。当发布的项目合作信息有误时，会员单位可以对本单位发布的项目合作信息完成修改操作。项目合作管理包图如图6-17所示。

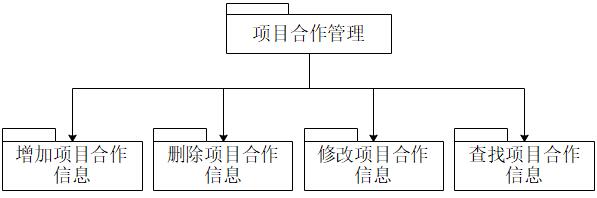


图6-21 项目合作管理包图

### 6.7.2 类图设计

项目合作管理类设计如图6-22所示。

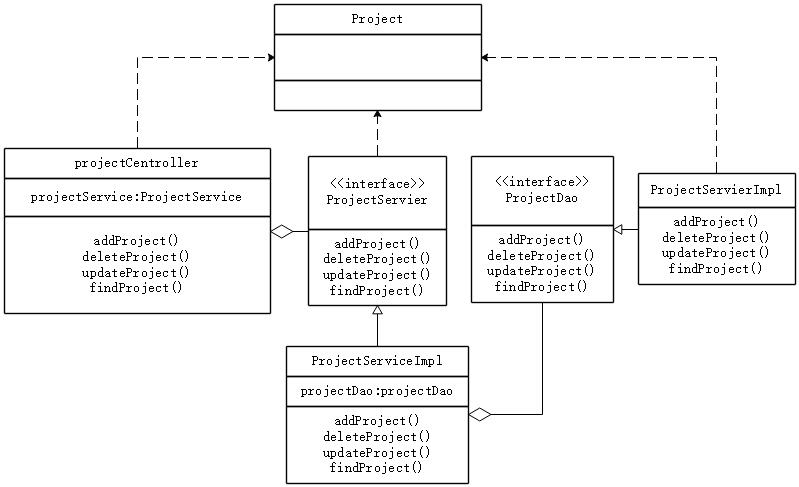


图6-22 项目合作管理类图

项目合作控制类描述如表6-24所示。

表6-24项目合作控制类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 24 | | 类名称（中文） | | | 项目合作控制类 |
| 类功能描述： | 负责前台页面的请求，负责请求处理和转发 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | addProject(Project project) |  | |  | 将前台页面传入的项目实体类加入方法中，并调用服务层 | | |
|  | deleteProject(int id) |  | |  | 将前台页面传入的项目编号加入方法中，并调用服务层 | | |
|  | updateProject(Project project) |  | |  | 将前台页面传入的项目实体类加入方法中，并调用服务层 | | |
|  | findProject(Map<String,Object> map) |  | |  | 将前台页面传入的某些数据加入方法中，并调用服务层 | | |

项目合作服务类描述如表6-25所示。

表6-25 项目合作服务类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 25 | | 类名称（中文） | | | 项目合作服务类 |
| 类功能描述： | 负责项目合作相关操作的服务层方法封装 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | addProject(Project project) |  | |  | 将控制器类传入的项目实体类加入方法中，并调用dao层 | | |
|  | deleteProject(int id) |  | |  | 将控制器类传入的项目编号加入方法中，并调用dao层 | | |
|  | updateProject(Project project) |  | |  | 将控制器类传入的项目实体类加入方法中，并调用dao层 | | |
|  | findProject(Map<String,Object> map) |  | |  | 将控制器类传入的某些数据加入方法中，并调用dao层 | | |

项目合作信息数据库操作类描述如表6-26所示。

表6-26 项目合作信息数据库操作类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 26 | | 类名称（中文） | | | 项目合作信息数据库操作类 |
| 类功能描述： | 负责项目合作信息相关操作的数据库层方法封装 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | addProject(Project project) |  | |  | 将服务器类传入的项目实体类加入方法中，并调用mapping映射，进行数据插入 | | |

续表6-26

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | deleteProject(int id) |  |  | 将服务器类传入的项目编号加入方法中，并调用mapping映射，进行数据删除 |
|  | updateProject(Project project) |  |  | 将服务器类传入的项目实体类加入方法中，并调用mapping映射，根据项目主键，进行数据更新 |
|  | findProject(Map<String,Object> map) |  |  | 将服务器类传入的map集合加入方法中，并调用mapping映射，进行数据查询 |

### 6.7.3 顺序图设计

会员单位添加项目合作信息活动顺序图如图6-23所示。

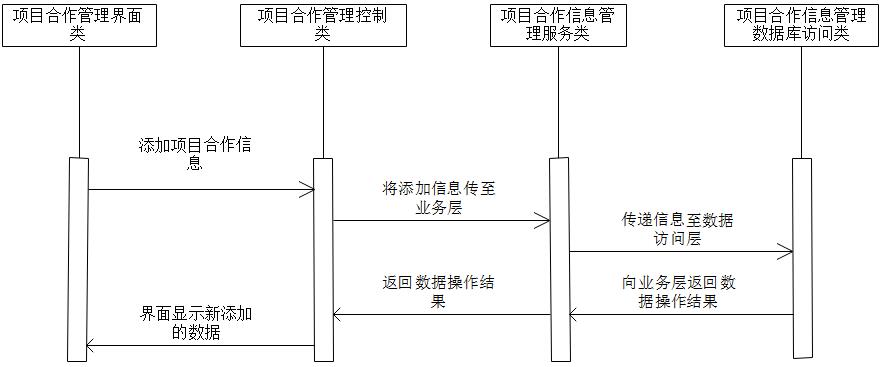


图6-23 项目合作管理顺序图

### 6.7.4 核心处理流程设计

项目合作管理子系统中添加项目合作信息功能流程图如图6-24所示。

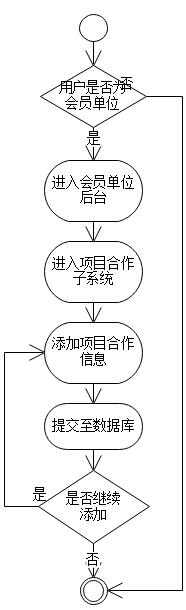


图6-24 项目合作管理流程图

## 6.8招聘管理

### 6.8.1 功能结构设计

招聘管理功能主要有四项操作：对招聘信息的增删改查。协会管理员和会员单位都具有增删改查的权限。新的岗位需要时，协会管理员或会员单位即可插入一条新的招聘信息。会员单位只能删除本单位发布的招聘信息，而协会管理员能删除协会全体会员单位发布的招聘信息。查询操作指的是会员单位只能查询到本单位发布的招聘信息，而协会管理员和游客能查到协会全体会员单位发布的招聘信息。当发布的招聘信息有误时，会员单位可以对本单位发布的招聘信息完成修改操作，而协会管理员能修改协会全体会员单位发布的招聘信息。项目合作管理包图如图6-25所示。

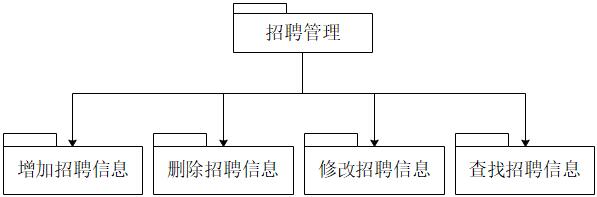


图6-25 信息管理包图

### 6.8.2 类图设计

招聘管理类设计如图6-26所示。

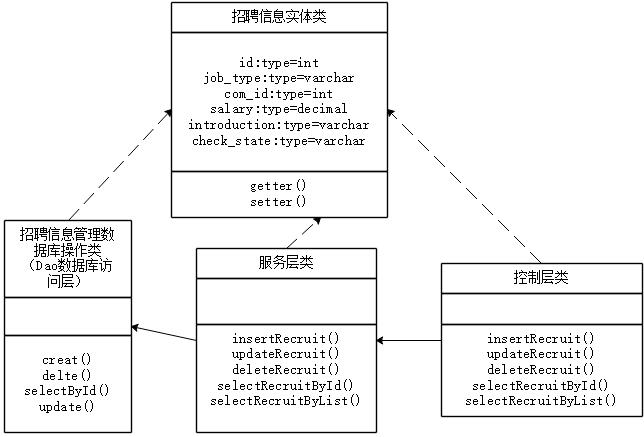


图6-26 招聘管理类图

招聘信息实体类秒速如表6-27所示。

表6-27 招聘信息实体类

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 类代号（code） | 27 | 类名称（中文） | 招聘信息实体类 | | |
| 类功能描述： | 招聘信息实体类对象 | | | | |
| 类属性 | 属性类型 | 属性名称 | 属性描述 | 其他 |  |
|  | int | id | id |  |  |
|  | varchar | job\_type | 职业类型 |  |  |
|  | int | com\_id | 招聘发布公司id |  |  |
|  | decimal | salary | 薪资 |  |  |

续表6-27

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | varchar | introduction | 工作简介 |  |  |
|  | varchar | check\_state | 管理员审查状态 |  |  |
| 主要实现方法 | setter/getter() | | | | |

招聘信息管理控制器类描述如表6-28所示。

表6-28 招聘信息管理控制器类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 28 | | 类名称（中文） | | | 招聘信息管理控制器类 |
| 类功能描述： | 招聘信息相关请求的处理和转发 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | insertRecruit () |  | |  | 响应插入一条招聘信息的请求 | | |
|  | updateRecruit () |  | |  | 响应更新一条招聘信息的请求 | | |
|  | deleteRecruitById() |  | |  | 响应根据ID删除一条招聘信息的请求 | | |
|  | selectRecruitById() |  | |  | 响应根据ID查询一条招聘信息的请求 | | |
|  | selectRecruitByList() |  | |  | 响应查询所有招聘信息的请求 | | |

招聘信息服务层类描述如表6-29所示。

表6-29 招聘信息服务层类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 29 | | 类名称（中文） | | | 招聘信息服务层类 |
| 类功能描述： | 负责招聘信息相关操作的服务层方法封装 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | insertRecruit () |  | |  | 插入一条招聘信息 | | |
|  | updateRecruit () |  | |  | 更新一条招聘信息 | | |
|  | deleteRecruitById() |  | |  | 根据ID删除一条招聘信息 | | |
|  | selectRecruitById() |  | |  | 根据ID查询一条招聘信息 | | |
|  | selectRecruitByList() |  | |  | 查询所有招聘信息 | | |

招聘信息数据库操作类描述如表6-30所示。

表6-30 招聘信息数据库操作类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 30 | | 类名称（中文） | | | 招聘信息数据库操作类 |
| 类功能描述： | 负责招聘信息相关操作的数据库层方法封装 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | insert() |  | |  | 插入一条招聘信息数据 | | |
|  | update() |  | |  | 更新一条招聘信息数据 | | |
|  | deleteById() |  | |  | 根据ID删除一条招聘信息数据 | | |
|  | selectById() |  | |  | 根据ID查询一条招聘信息数据 | | |

续表6-30

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | selectByList() |  |  | 查询所有招聘信息数据 |

### 6.8.3 顺序图设计

协会管理员添加招聘信息活动顺序图如图6-27所示。

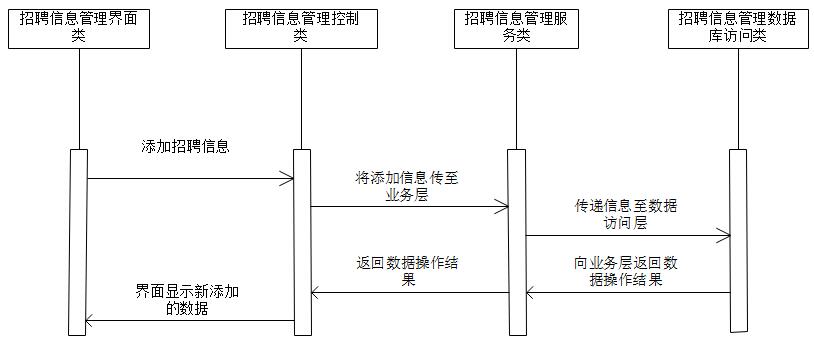


图6-27 管理顺序图

### 6.8.4 核心处理流程设计

信息管理子系统中添加网站前台信息功能流程图如图6-28所示。

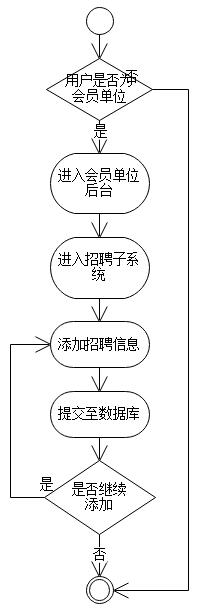


图6-28 信息管理流程图

## 6.9简历管理

### 6.9.1 功能结构设计

简历管理功能主要有三项操作：对简历信息的增删查。游客进入网站的招聘页面后，向网站系统提交个人简历。查询操作指会员单位可以查到本单位特定岗位接收到简历信息。删除操作即会员单位删除本单位接受到的简历信息。简历管理包图如图6-29所示。

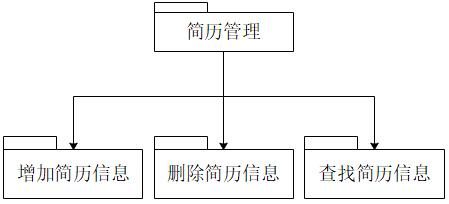


图6-29 简历管理包图

### 6.9.2 类图设计

简历管理类设计如图6-30所示。

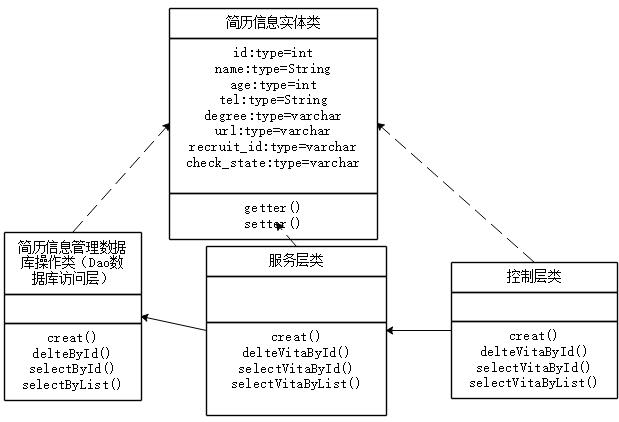


图6-30 简历管理类图

简历信息实体类描述如表6-31所示。

表6-31 简历信息实体类

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 类代号（code） | 31 | 类名称（中文） | 简历信息实体类 | | |
| 类功能描述： | 简历信息实体类对象 | | | | |
| 类属性 | 属性类型 | 属性名称 | 属性描述 | 其他 |  |
|  | Int | id | 简历信息id |  |  |
|  | String | name | 求职者姓名 |  |  |
|  | int | age | 求职者年龄 |  |  |
|  | String | tel | 求职者电话 |  |  |
|  | varchar | degree | 求职者学历 |  |  |
|  | varchar | url | 保存简历文档的地址记录 |  |  |
|  | varchar | recruit\_id | 简历所属的招聘信息id |  |  |
|  | varchar | check\_state | 管理员审查状态 |  |  |
| 主要实现方法 | setter/getter() | | | | |

简历信息管理控制器类描述如表6-32所示。

表6-32 简历信息管理控制器类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 32 | | 类名称（中文） | | | 简历信息管理控制器类 |
| 类功能描述： | 简历信息相关请求的处理和转发 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | deleteVitaById() |  | |  | 响应根据ID删除一条简历信息的请求 | | |
|  | selectVitaById() |  | |  | 响应根据ID查询一条简历信息的请求 | | |
|  | selectVitaByList() |  | |  | 响应查询所有简历信息的请求 | | |

简历信息服务层类描述如表6-33所示。

表6-33 简历信息服务层类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 33 | | 类名称（中文） | | | 简历信息服务层类 |
| 类功能描述： | 负责简历信息相关操作的服务层方法封装 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | deleteVitaById() |  | |  | 根据ID删除一条简历信息 | | |
|  | selectVitaById() |  | |  | 根据ID查询一条简历信息 | | |
|  | selectVitaByList() |  | |  | 查询所有简历信息 | | |

简历信息数据库操作类描述如表6-34所示。

表6-34 简历信息数据库操作类

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类代号（code） | 34 | | 类名称（中文） | | | 简历信息数据库操作类 |
| 类功能描述： | 负责简历信息相关操作的数据库层方法封装 | | | | | |
| 方法属性 | 方法名 | 输入 | | 输出 | 方法功能描述 | | |
|  | deleteById() |  | |  | 根据ID删除一条简历信息数据 | | |
|  | selectById() |  | |  | 根据ID查询一条简历信息数据 | | |
|  | selectByList() |  | |  | 查询所有简历信息数据 | | |

### 6.9.3 顺序图设计

游客提交个人简历信息活动顺序图如图6-31所示。

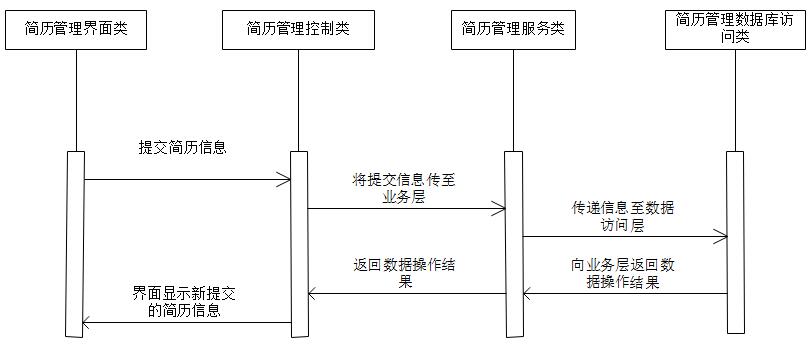


图6-31 信息管理顺序图

### 6.9.4 核心处理流程设计

简历管理子系统中游客添加个人简历信息功能流程图如图6-32所示。



图6-32 信息管理流程图

# 7 编码

## 7.1 代码实现与核心算法

### 7.1.1协会管理员登录页面

协会管理员登录页面代码实现。

|  |
| --- |
| div class="col-md-4 col-md-offset-1" style="padding-top:40px;">  <div class="form-top">  <div >  <h3 style="color:white">协会管理员登录窗口</h3>  </div>  </div>  <div class="form-bottom">  <form role="form" action="" method="post" >  <div class="form-group">  <label class="sr-only" for="txtNumber">UserName</label>  <div class="input-group">  <div class ="input-group-addon">  <span class="glyphicon glyphicon-user"></span></div>  <input type="text" name="number" placeholder="账号" class=" form-control" id="txtNumber" required>  </div>  </div>  <div class="form-group">  <label class="sr-only" for="txtPwd">PassWord</label>  <div class="input-group">  <div class="input-group-addon"><span class="glyphicon glyphicon-lock"></span></div>  <input type="password" name="password" placeholder="密码" class="form-control" id="txtPwd" required>  </div>  </div> |

### 7.1.2新闻通知查询业务

新闻通知查询业务代码实现。

|  |
| --- |
| function findNewsList() {  var con = new Object();  con.info\_type = "新闻通知";  doData("Info/findByType", con, function (result) {  console.log(result)  $("#newsList").empty();  if (result != null && result.obj != undefined && result.obj.length > 0) {  for (var i = 0; i < result.obj.length; i++) {  $("#newsList").append(  '<div class="single-post d-flex flex-row row">' +  '<div class="thumb col-lg-1">' +  '</div>'+  '<div class="details col-lg-10">'+  '<div class="title d-flex flex-row justify-content-between">'+  '<div class="titles">'+  '<a href="'+getRootPath()+"/admin/showIndex/newsDetail.jsp?id="+result.obj[i].id+'">'+'<h4>'+result.obj[i].info\_title+'</h4>'+'</a>'+  '</div>'+  '</div>'+  '<p>'+result.obj[i].info\_content.substr(0,100)+'......'+'</p>'+  '</div>'+  '</div>'  )  }  }  }, true);  } |

### 7.1.3技术转让查询业务

技术转让查询业务代码实现。

|  |
| --- |
| function find() {  var con=new Object();  doData("Technology/find",con,function (result) {  $("#tData").empty();//清空原有数据  var total=0;  if (result!=null && result.obj!=undefined && result.obj.length>0){  for(var i=0;i<result.obj.length;i++){  $("#tData").append("<tr>" +  "<td>"+ result.obj[i].name+"</td>" +  "<td>"+ result.obj[i].types+"</td>" +  "<td><img src='"+getRootPath()+"/uploadfiles/"+result.obj[i].img+"' style='width:60px;height:80px;' ></td>" +  "<td>"+ result.obj[i].power+"</td>" +  "<td>"+  // "<button class='btn-warning btn-xs' doType='edit' recordId='"+result.obj[i].id+"'>修改</button>"+  "<button class='btn-danger btn-xs' doType='del' recordId='"+result.obj[i].id+"'>删除</button></td></tr>");  }  }  },true);  } |

### 7.1.4 简历管理

简历管理业务代码实现。

|  |
| --- |
| function find() {  var con=new Object();  power=sessionStorage.getItem("power");  console.log(power);  con.power=power;  doData("Vita/findVitaByPower",con,function (result) {  $("#tData").empty();//清空原有数据  var total=0;  if (result!=null && result.obj!=undefined && result.obj.length>0){  for(var i=0;i<result.obj.length;i++){  $("#tData").append("<tr>" +  "<td>"+ result.obj[i].name+"</td>" +  "<td>"+ result.obj[i].age+"</td>" +  "<td>"+ result.obj[i].tel+"</td>" +  "<td>"+ result.obj[i].degree+"</td>" +  "<td>"+ result.obj[i].recruit\_name.name+"</td>" +  "<td>"+ result.obj[i].recruit\_name.power+"</td>" +  "<td>"+result.obj[i].vita\_files+"</td>" +  "<td>"+  // "<button class='btn-warning btn-xs' doType='edit' recordId='"+result.obj[i].id+"'>修改</button>"+  "<button class='btn-danger btn-xs' doType='del' recordId='"+result.obj[i].id+"'>删除</button></td></tr>");  }  }  },true);  } |

## 7.2 代码优化分析

1. JS代码优化

字符串的拼接在我们开发中会经常遇到，我们往往习惯的直接用+=的方式来拼接字符串，其实这种拼接的方式效率非常的低，在项目中用利用数组的join方法来实现字符串的拼接。

1. Java代码优化

使用单一职责原则来优化Java代码。对于一个java类，应该仅有一个引起它变化的原因。即一个类内部只有一组相关性很高的函数、数据的封装。模型-视图-控制器各层之间都要特定的类进行处理，不能混淆了类的功能。

# 8 测试

## 8.1 测试方案设计

本系统的测试主要按照单元测试、黑盒测试、白盒测试和系统集成测试四种测试原则进行。对每个子系统、模块、功能点进行针对性测试，以达到对系统的合理完整的测试。

### 8.1.1 测试策略

在单元测试阶段，检查系统项目中的每一个类的功能是否正确。在黑盒测试阶段，通过检测系统项目的每个功能对于外界的输入是否都能达到需求分析的要求并正常运行。在白盒测试阶段，结合每个功能点和相关代设计测试用例，用测试用例来判断程序的正确性。在系统集成测试中，将不同子系统部署到一起来测试系统集成在一起后是否能达到用户需求。

### 8.1.2 测试进度安排

本系统的测试进度计划安排如表8-1所示：

表8-1测试进度计划安排表

|  |  |
| --- | --- |
| 测试阶段 | 测试任务 |
| 第一阶段  功能测试 | 1. 会员单位管理 2. 信息管理 3. 专家库管理 4. 收入与报送管理 5. 技术转让管理 6. 项目合作管理 7. 招聘管理 8. 简历管理 |
| 第二阶段  系统测试 | 1. 完成所有模块的组合测试 2. 确保所有业务流向以及数据都是正确的 |
| 第三阶段  性能测试 | 在多用户同时访问的情况下，对系统进行负载测 |
| 第四阶段  安装部署测试 | 将项目部署后进行测试 |

续表8-1

|  |  |
| --- | --- |
| 第五阶段  兼容测试 | 测试项目用不同的浏览器访问的运行情况 |

### 8.1.3 测试资源

1. 测试环境

Windows 10操作系统 ，CPU为INTEL CORE i3处理器 ，固态硬盘18G ，内存8G ，IDEA2018 ，Tomcat8.0.37。

1. 测试管理工具

Mozilla Firefox是一个自由及开放源代码的[网页浏览器](https://baike.baidu.com/item/%E7%BD%91%E9%A1%B5%E6%B5%8F%E8%A7%88%E5%99%A8/8309940" \t "https://baike.baidu.com/item/Mozilla%20Firefox/_blank)，使用[Gecko](https://baike.baidu.com/item/Gecko)排版引擎，支持多种操作系统，如[Windows](https://baike.baidu.com/item/Windows" \t "https://baike.baidu.com/item/Mozilla%20Firefox/_blank)、[Mac OS X](https://baike.baidu.com/item/Mac%20OS%20X" \t "https://baike.baidu.com/item/Mozilla%20Firefox/_blank)及[GNU/Linux](https://baike.baidu.com/item/GNU/Linux" \t "https://baike.baidu.com/item/Mozilla%20Firefox/_blank)等。

1. 测试文档

软件测试计划文档、软件测试用例、缺陷报告文档。

### 8.1.4 关键测试点

1. 协会管理员是否能修改专家信息。
2. 会员单位用户是否能提交收入报送信息。
3. 游客是否能在简历投递页面向系统提交简历。

## 8.2 测试用例构建

本节主要介绍测试用例的设计，从编写规范到设计、到实现测试、到维护等进行完成的测试用例的构建。

### 8.2.1 测试用例编写约定

用例编号：测试用例的唯一标识

用例名称：对测试用例的描述

操作步骤：执行该测试用例需要完成的操作

预期结果：完成该步骤后程序应当得到的结果

实际结果：完成该步骤后程序实际得到的结果

问题描述：对于实际结果与预期结果不一的用例而产生的错误描述

### 8.2.2 测试用例设计

发布新闻通知信息测试用例如表8-2所示：

表8-2 发布新闻通知信息测试用例

|  |  |
| --- | --- |
| 用例编号 | 01 |
| 用例名称 | 发布新闻通知信息 |
| 操作步骤 | 1. 进入协会管理员后台 2. 进入信息模块 3. 点击“新增”按钮 4. 填入要新增的新闻通知信息 5. 点击“提交”按钮 |
| 预期结果 | 前台查看到发布的新闻通知信息 |
| 实际结果 | 与预期结果相符 |
| 问题描述 | 暂无 |

发布技术转让信息测试用例如表8-3所示：

表8-3 发布技术转让信息测试用例

|  |  |
| --- | --- |
| 用例编号 | 02 |
| 用例名称 | 发布技术转让信息 |
| 操作步骤 | 1. 进入会员单位后台 2. 进入技术转让模块 3. 点击“新增”按钮 4. 填入要新增的技术转让信息 5. 点击“提交”按钮 |
| 预期结果 | 前台查看到发布的技术转让信息 |
| 实际结果 | 与预期结果相符 |
| 问题描述 | 暂无 |

查看招聘信息测试用例如表8-4所示：

表8-4 查看招聘信息测试用例

|  |  |
| --- | --- |
| 用例编号 | 03 |
| 用例名称 | 查看招聘信息 |
| 操作步骤 | 1. 游客进入协会网站首页 2. 进入招聘页面 3. 点击“岗位详情 |

续表8-4

|  |  |
| --- | --- |
| 预期结果 | 游客可以查到岗位信息 |
| 实际结果 | 与预期结果相符 |
| 问题描述 | 暂无 |

### 8.2.3 关键测试用例

修改专家信息测试用例如表8-5 所示：

表8-5 修改专家信息测试用例

|  |  |
| --- | --- |
| 用例编号 | 04 |
| 用例名称 | 修改专家信息 |
| 操作步骤 | 1. 进入协会管理员后台 2. 进入专家库模块 3. 点击已有专家信息 4. 点击“修改”按钮 5. 填入要修改的信息 6. 点击“提交”按钮 |
| 预期结果 | 前台查看到修改后的专家信息 |
| 实际结果 | 与预期结果相符 |
| 问题描述 | 暂无 |

提交收入与报送信息测试用例如表8-6所示：

表8-6 提交收入与报送信息测试用例

|  |  |
| --- | --- |
| 用例编号 | 05 |
| 用例名称 | 提交收入与报送信息 |
| 操作步骤 | 1. 进入会员单位后台 2. 进入收入与报送模块 3. 点击“新增”按钮 4. 填入要提交的收入与报送信息 5. 点击“提交”按钮 |
| 预期结果 | 协会管理员后台可以查到单位提交的收入报送信息 |
| 实际结果 | 与预期结果相符 |
| 问题描述 | 暂无 |

投递简历测试用例如表8-7所示：

表8-7 投递简历测试用例

|  |  |
| --- | --- |
| 用例编号 | 06 |
| 用例名称 | 投递简历 |
| 操作步骤 | 1. 游客进入进入招聘页面 2. 点击“投递简历” 3. 添加个人简历信息 4. 点击“提交” |
| 预期结果 | 会员单位后台可以查到游客提交的简历信息 |
| 实际结果 | 与预期结果相符 |
| 问题描述 | 暂无 |

### 8.2.4 测试用例维护

1. 及时检测版本的变更情况，对应不同版本的每个测试用例要进行冗余管理，保证测试用例的针对性与高效性。此外对于新版本的新功能也要添加新的测试用例。
2. 及时删除已不存在功能的测试用例。部分功能会在版本变更中被删除或被修改，因此对于当前不存在的功能要删去对应的测试用例。
3. 及时根据版本改进测试用例。由于版本的不同，功能存在被优化的可能。对于被优化的功能，不能用原有的测试用例来检测其正确与否，应当更新新的测试用例。

# 9 总结与展望

## 9.1 设计工作总结

从项目选题到需求调研，网站的设计过程要详细确定每一个功能点。在保证需求分析正确的前提下，与用户多次确认之后才能开始进入系统详细设计阶段。

在协会网站设计过程中起初没有使用增量模型，导致系统功能不断增多而产生版本迭代问题。在认识到开发设计规范化的重要性后，项目按部就班的从需求到功能再到实现不断完善。得益于前期需求分析和功能设计阶段对于项目考虑的较为全面，使得项目开发代码阶段的完成效率大大提升。

在需求分析的基础上设计得到的系统框架，每个模块之间可以拆分按步骤依次开发，避免了没有单元测试就直接并入系统而导致的错误。项目的数据处理过程得益于框架的帮助，省去了不必要的繁琐过程，大大降低了开发成本，但仍有许多技术细节要处理。

## 9.2 未来工作展望

该版本的大数据及信息技术产业协会基础功能较为完善，但前台和后台页面美观性相差较大。该版本的后台开发过程主要解决了数据处理的问题，在未来的开发和维护中将完善优化后台界面设计。

此外该版本的大数据及信息技术产业协会网站，其主要任务是服务会员单位和游客。在当前数据通信快速的时代，该版本协会网站未添加实时客服系统。这一弊端使得游客无法与协会取得即时通信，将会影响协会活动的效率。在未来的开发和维护中，我们将添加实时客服系统以此来增加网站用户体验，促进协会和会员单位更高效的发展。

# 谢 辞

# 参考文献

1. 刘义忠,张伟.[基于SSM框架的后台管理系统设计与实现[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=7001297922&from=Qikan_Article_Detail).软件导刊,2019，18(2):68-71
2. 董珊,周德阳,罗荣良.[基于SSM的实验室管理系统设计[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=68789083504849575055485048&from=Qikan_Article_Detail).电脑知识与技术:学术版,2019,0(9X):45-46.
3. 贺雪梅.[web应用开发中的SSM框架设计[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=7001167514&from=Qikan_Article_Detail).电子世界,2019,0(1):206-206
4. 曾艳丽,李诺.[针对SSM框架Web系统的相关思考[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=7001325559&from=Qikan_Article_Detail).信息与电脑,2019,0(3):116-117
5. 许鹏.[基于SSM框架的高校教师业绩考核系统设计与实现[J]](http://qikan.cqvip.com/Qikan/Article/Detail?id=7001442515&from=Qikan_Article_Detail).攀枝花学院学报:综合版,2019,36(2):50-56
6. 王婧,王晓云,于波.[基于SSM框架的分布式架构二手书交易系统[J]](http://qikan.cqvip.com/Qikan/Article/Detail?id=68789083504849574851485156&from=Qikan_Article_Detail).电脑知识与技术:学术交流,2019,0(1X):86-88
7. 曹阳.[软件工程Web技术开发平台[J]](http://qikan.cqvip.com/Qikan/Article/Detail?id=68908285504849574853485151&from=Qikan_Article_Detail).电子技术与软件工程,2019,0(5):34-34
8. 刘颖.[软件工程技术在系统软件开发中的运用[J]](http://qikan.cqvip.com/Qikan/Article/Detail?id=68908285504849574853485148&from=Qikan_Article_Detail).电子技术与软件工程,2019,0(5):31-31
9. 潘光添,林建斌,梁伟玲,钟秀娟.[基于B/S的医疗器械自主采购与评审系统设计及应用[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=7100983052&from=Qikan_Article_Detail).医疗卫生装备,2020,41(2):44-47
10. 曹珍,杨帆.[基于SSM框架的商户管理平台设计与实现[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=674172581).计算技术与自动化,2017,36(4):119-121
11. 程传蕊,李娜.[基于Spring MVC＋Shiro＋EasyUI的市场信息管理系统[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=675090955).漯河职业技术学院学报,2018,17(2):50-53
12. 李宇,刘彬.[基于SSM框架的学术成果展示平台设计与实现[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=675692624).信息与电脑,2018,0(14):70-71
13. 苏庭波,王世权.[基于SSM的品优购后台管理系统的设计与实现[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=676384997).科学,2018,36(5):866-870
14. 陈培龙,仁增多杰.[基于SSM框架旅游网站后台系统设计与实现[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=676425543).电脑编程技巧与维护,2018,0(10):28-29
15. 唐权,韩文智.[基于SpringMVC框架文件上传技术应用研究[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=676874745).信息通信,2018,0(11):188-189
16. 唐权.[SSM框架条件分页查询研究与实现[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=7001440173).四川职业技术学院学报,2019,29(1):139-143
17. 刘欣,汪洋,韦俊,邓勇军.[基于B/S结构的核燃料软件模型评估数据管理平台研制[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=7001708236).软件,2019,40(3):126-129
18. 谢涵风,蔡国宝,夏曦,王秀友.[基于SSM的寻亲互联网络平台的设计与实现[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=7001845567).电脑知识与技术:学术版,2019,15(2):85-88
19. 曹鹏.[Spring MVC框架下电信用户信息分析大数据可视化研究[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=7100804104&from=Qikan_Article_Detail).电脑知识与技术:学术版,2019,15(11X):253-254
20. 孙旸.[基于SSM框架的高校信息采集管理系统设计与实现[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=7100811841&from=Qikan_Article_Detail).无线互联科技,2019,16(23):48-50
21. 章胜江,刘萍.[基于SSM的网上评教系统的设计与实现[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=68789083504849564955485352&from=Qikan_Article_Detail).电脑知识与技术：学术交流,2018,0(6Z):132-134.
22. 李秀明,武怀生.[基于MVC模式架构的中小学教务管理系统的设计与实现[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=68898885504849574854484856).电子元器件与信息技术,2019,0(6):28-31
23. 李明,李长昱,杨艳梅,吴楠,陈丽珠.[基于B/S架构的企业科技信息管理系统设计[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=7001499159).软件导刊,2019,18(3):100-103
24. 张鲁.[基于SSM架构的高校智能报修管理系统设计与实现[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=81899090504849574855485654).侨园,2019,0(7):112-112
25. 刘旺.[MVC架构下的教培信息发布系统的设计与实现[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=7100118713&from=Qikan_Article_Detail).电脑知识与技术:学术版,2019,15(8Z):111-113
26. 刘振宇.[数字化校园统一信息门户规划与实现[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=7100709845).科教导刊(电子版),2019(35):80-80
27. 张玮.[从自然人到“数据人”——大数据带来的自然人税收治理新思考[J]](http://lib.cqvip.com/Qikan/Article/Detail?id=7100925027).税收经济研究,2019,24(6):69-74
28. Volunteer Involvement and Organizational Performance: The Use of Volunteer Officers in Public Safety[J]. Seong C. Kang. Public Performance & Management Review. 2019(3)
29. Police volunteers in the Netherlands: a study on policy and practice[J]. Ronald van Steden,Shanna M. Mehlbaum. Policing and Society. 2019(4)
30. Voluntary sensemaking: the identity formation process of volunteers in sport organizations[J]. Christine E. Wegner,Gareth J. Jones,Jeremy S. Jordan. European Sport Management Quarterly. 2019(5)

# 附录A 外文翻译—原文部分

The Source of Article：《Spring in Action, 5th Edition》

**What is Spring?**

I know you’re probably itching to start writing a Spring application, and I assure you that before this chapter ends, you’ll have developed a simple one. But first, let me set the stage with a few basic Spring concepts that will help you understand what makes Spring tick.

Any non-trivial application is composed of many components, each responsible for its own piece of the overall application functionality, coordinating with the other application elements to get the job done. When the application is run, those components somehow need to be created and introduced to each other.

At its core, Spring offers a container, often referred to as the Spring application context,that creates and manages application components. These components, or beans,are wired together inside the Spring application context to make a complete application,much like bricks, mortar, timber, nails, plumbing, and wiring are bound together to make a house.

The act of wiring beans together is based on a pattern known as dependency injection (DI). Rather than have components create and maintain the lifecycle of other beans that they depend on, a dependency-injected application relies on a separate entity (the container) to create and maintain all components and inject those into the beans that need them. This is done typically through constructor arguments or property accessor methods.

For example, suppose that among an application’s many components, there are two that you’ll address: an inventory service (for fetching inventory levels) and a product service (for providing basic product information). The product service depends on the inventory service to be able to provide a complete set of information about products. Figure 1.1 illustrates the relationships between these beans and the Spring application context.

On top of its core container, Spring and a full portfolio of related libraries offer a web framework, a variety of data persistence options, a security framework, integration with other systems, runtime monitoring, microservice support, a reactive programming model, and many other features necessary for modern application development.

Historically, the way you would guide Spring’s application context to wire beans together was with one or more XML files that described the components and their relationship to other components. For example, the following XML declares twobeans, an InventoryService bean and a ProductService bean, and wires the InventoryService bean into ProductService via a constructor argument:

<bean id="inventoryService"

class="com.example.InventoryService" />

<bean id="productService"

class="com.example.ProductService" />

<constructor-arg ref="inventoryService" />

</bean>

In recent versions of Spring, however, a Java-based configuration is more common.The following Java-based configuration class is equivalent to the XML configuration:

@Configuration

public class ServiceConfiguration {

@Bean

public InventoryService inventoryService() {

return new InventoryService();

}

@Bean

public ProductService productService() {

return new ProductService(inventoryService());

}

}

The @Configuration annotation indicates to Spring that this is a configuration class that will provide beans to the Spring application context. The configuration’s class methods are annotated with @Bean, indicating that the objects they return should be added as beans in the application context (where, by default, their respective bean IDs will be the same as the names of the methods that define them).

Java-based configuration offers several benefits over XML-based configuration,including greater type safety and improved refactorability. Even so, explicit configuration with either Java or XML is only necessary if Spring is unable to automatically configure the components.

Automatic configuration has its roots in the Spring techniques known as autowiring and component scanning. With component scanning, Spring can automatically discover components from an application’s classpath and create them as beans in the Springapplication context. With autowiring, Spring automatically injects the components with the other beans that they depend on.

More recently, with the introduction of Spring Boot, automatic configuration has gone well beyond component scanning and autowiring. Spring Boot is an extension of the Spring Framework that offers several productivity enhancements. The most well-known of these enhancements is autoconfiguration, where Spring Boot can make reasonable guesses of what components need to be configured and wired together,based on entries in the classpath, environment variables, and other factors.

I’d like to show you some example code that demonstrates autoconfiguration. But I can’t. You see, autoconfiguration is much like the wind. You can see the effects of it, but there’s no code that I can show you and say “Look! Here’s an example of autoconfiguration!” Stuff happens, components are enabled, and functionality is provided without writing code. It’s this lack of code that’s essential to autoconfiguration and what makes it so wonderful.

Spring Boot autoconfiguration has dramatically reduced the amount of explicit configuration (whether with XML or Java) required to build an application. In fact, by the time you finish the example in this chapter, you’ll have a working Spring application that has only a single line of Spring configuration code!

Spring Boot enhances Spring development so much that it’s hard to imagine developing Spring applications without it. For that reason, this book treats Spring and Spring Boot as if they were one and the same. We’ll use Spring Boot as much as possible, and explicit configuration only when necessary. And, because Spring XML configuration is the old-school way of working with Spring, we’ll focus primarily on Spring’s Java-based configuration.

**The core Spring Framework**

As you might expect, the core Spring Framework is the foundation of everything else in the Spring universe. It provides the core container and dependency injection framework. But it also provides a few other essential features.

Among these is Spring MVC, Spring’s web framework. You’ve already seen how to use Spring MVC to write a controller class to handle web requests. What you’ve not yet seen, however, is that Spring MVC can also be used to create REST APIs that produce non-HTML output. We’re going to dig more into Spring MVC in chapter 2 and then take another look at how to use it to create REST APIs in chapter 6.

The core Spring Framework also offers some elemental data persistence support,specifically template-based JDBC support. You’ll see how to use JdbcTemplate in chapter 3.

In the most recent version of Spring (5.0.8), support was added for reactive-style programming, including a new reactive web framework called Spring WebFlux that borrows heavily from Spring MVC. You’ll look at Spring’s reactive programming model in part 3 and Spring WebFlux specifically in chapter 10.

**Spring Boot**

We’ve already seen many of the benefits of Spring Boot, including starter dependencies and autoconfiguration. Be certain that we’ll use as much of Spring Boot as possible throughout this book and avoid any form of explicit configuration, unless it’s absolutely necessary. But in addition to starter dependencies and autoconfiguration, Spring Boot also offers a handful of other useful features:

* The Actuator provides runtime insight into the inner workings of an application,including metrics, thread dump information, application health, and environment properties available to the application.
* Flexible specification of environment properties.
* Additional testing support on top of the testing assistance found in the core framework.

What’s more, Spring Boot offers an alternative programming model based on Groovy scripts that’s called the Spring Boot CLI (command-line interface). With the Spring Boot CLI, you can write entire applications as a collection of Groovy scripts and run them from the command line. We won’t spend much time with the Spring Boot CLI,but we’ll touch on it on occasion when it fits our needs.

Spring Boot has become such an integral part of Spring development; I can’t imagine developing a Spring application without it. Consequently, this book takes a Spring Boot–centric view, and you might catch me using the word Spring when I’m referring to something that Spring Boot is doing.

**Spring Data**

Although the core Spring Framework comes with basic data persistence support,Spring Data provides something quite amazing: the ability to define your application’s data repositories as simple Java interfaces, using a naming convention when defining methods to drive how data is stored and retrieved.

What’s more, Spring Data is capable of working with a several different kinds of databases, including relational (JPA), document (Mongo), graph (Neo4j), and others.You’ll use Spring Data to help create repositories for the Taco Cloud application in chapter 3.

**Spring Security**

Application security has always been an important topic, and it seems to become more important every day. Fortunately, Spring has a robust security framework in Spring Security.Spring Security addresses a broad range of application security needs, including authentication, authorization, and API security. Although the scope of Spring Security is too large to be properly covered in this book, we’ll touch on some of the most common use cases in chapters 4 and 12.

**Spring Integration and Spring Batch**

At some point, most applications will need to integrate with other applications or even with other components of the same application. Several patterns of application integration have emerged to address these needs. Spring Integration and Spring Batch provide the implementation of these patterns for Spring-based applications.

Spring Integration addresses real-time integration where data is processed as it’s made available. In contrast, Spring Batch addresses batched integration where data is allowed to collect for a time until some trigger (perhaps a time trigger) signals that it’s time for the batch of data to be processed. You’ll explore both Spring Batch and Spring Integration in chapter 9.

**Spring Cloud**

As I’m writing this, the application development world is entering a new era where we’ll no longer develop our applications as single deployment unit monoliths and will instead compose applications from several individual deployment units known as microservices.

Microservices are a hot topic, addressing several practical development and runtime concerns. In doing so, however, they bring to fore their own challenges. Those challenges are met head-on by Spring Cloud, a collection of projects for developing cloud-native applications with Spring.

Spring Cloud covers a lot of ground, and it’d be impossible to cover it all in this book. We’ll look at some of the most common components of Spring Cloud in chapters 13, 14, and 15. **Summary**

* Spring aims to make developer challenges easy, like creating web applications,working with databases, securing applications, and microservices.
* Spring Boot builds on top of Spring to make Spring even easier with simplified dependency management, automatic configuration, and runtime insights.
* Spring applications can be initialized using the Spring Initializr, which is webbased and supported natively in most Java development environments.
* The components, commonly referred to as beans, in a Spring application context can be declared explicitly with Java or XML, discovered by component scanning, or automatically configured with Spring Boot autoconfiguration.

# 附录B 外文翻译—译文部分

文章来源：《Spring in Action, 第五版》

**什么是Spring？**

我知道你可能很想开始编写 Spring 应用程序，我向你保证，在本章结束之前，你将开发一个简单的应用程序。但是首先，我得介绍一些 Spring 的基本概念，以帮助你了解 Spring 的变化。

任何不平凡的应用程序都由许多组件组成，每个组件负责自己的在整体应用程序中的那部分功能，并与其他应用程序元素协调以完成工作。在运行应用程序时，需要以某种方式创建这些组件并相互引用。

Spring 的核心是一个 容器，通常称为 Spring 应用程序上下文，用于创建和管理应用程序组件。这些组件（或 bean）在 Spring 应用程序上下文中连接在一起以构成一个完整的应用程序，就像将砖、灰浆、木材、钉子、管道和电线绑在一起以组成房屋。

将 bean 连接在一起的行为是基于一种称为 依赖注入（DI）的模式。依赖项注入的应用程序不是由组件自身创建和维护它们依赖的其他 bean 的生命周期，而是依赖于单独的实体（容器）来创建和维护所有组件，并将这些组件注入需要它们的 bean。通常通过构造函数参数或属性访问器方法完成此操作。

例如，假设在应用程序的许多组件中，要处理两个组件：inventory service（用于获取库存级别）和 product service（用于提供基本产品信息）。product service 取决于 inventory service，以便能够提供有关产品的完整信息。

除了其核心容器之外，Spring 和完整的相关库产品组合还提供 Web 框架、各种数据持久性选项、安全框架与其他系统的集成、运行时监视、微服务支持、响应式编程模型以及许多其他功能，应用于现代应用程序开发。

从历史上看，引导 Spring 应用程序上下文将 bean 连接在一起的方式是使用一个或多个 XML 文件，这些文件描述了组件及其与其他组件的关系。例如，以下 XML 声明两个 bean，一个 InventoryService bean 和一个 ProductService bean，然后通过构造函数参数将 InventoryService bean 注入到 ProductService 中：

<bean id="inventoryService" class="com.example.InventoryService" />

<bean id="productService" class="com.example.ProductService" >

<constructor-arg ref="inventoryService" />

</bean>

但是，在最新版本的 Spring 中，基于 Java 的配置更为常见。以下基于 Java 的配置类等效于 XML 配置：

@Configurationpublic class ServiceConfiguration {

@Bean

public InventoryService inventoryService() {

return new InventoryService();

}

@Bean

public ProductService productService() {

return new ProductService(inventoryService());

}

}

@Configuration 注释向 Spring 表明这是一个配置类，它将为 Spring 应用程序上下文提供 beans。 配置的类方法带有 @Bean 注释，指示它们返回的对象应作为 beans 添加到应用程序上下文中（默认情况下，它们各自的 bean IDs 将与定义它们的方法的名称相同）。

与基于 XML 的配置相比，基于 Java 的配置具有多个优点，包括更高的类型安全性和改进的可重构性。即使这样，仅当 Spring 无法自动配置组件时，才需要使用 Java 或 XML 进行显式配置。

自动配置起源于 Spring 技术，即 自动装配 和 组件扫描。借助组件扫描，Spring 可以自动从应用程序的类路径中发现组件，并将其创建为 Spring 应用程序上下文中的 bean。通过自动装配，Spring 会自动将组件与它们依赖的其他 bean 一起注入。

最近，随着 Spring Boot 的推出，自动配置的优势已经远远超出了组件扫描和自动装配。Spring Boot 是 Spring 框架的扩展，它提供了多项生产力增强功能。这些增强功能中最著名的就是 自动配置，在这种配置中，Spring Boot 可以根据类路径中的条目、环境变量和其他因素，合理地猜测需要配置哪些组件，并将它们连接在一起。

这里想要展示一些演示自动配置的示例代码，但是并没有这样的代码，自动配置就如同风一样，可以看到它的效果，但是没有代码可以展示。我可以说 “看！这是自动配置的示例！” 事情发生、组件启用并且提供了功能，而无需编写代码。缺少代码是自动配置必不可少的要素，这使它如此出色。

Spring Boot 自动配置大大减少了构建应用程序所需的显式配置（无论是 XML 还是 Java）的数量。实际上，当完成本章中的示例时，将拥有一个正在运行的 Spring 应用程序，该应用程序仅包含一行 Spring 配置代码！

Spring Boot 极大地增强了 Spring 开发的能力，很难想象没有它如何开发 Spring 应用程序。因此，本书将 Spring 和 Spring Boot 视为一模一样。我们将尽可能使用 Spring Boot，并仅在必要时使用显式配置。而且，由于 Spring XML 配置是使用 Spring 的老派方式，因此我们将主要关注基于 Java 的 Spring 配置。

**Spring 核心框架**

正如你所期望的，Spring 核心框架是 Spring 领域中其他一切的基础。它提供了核心容器和依赖注入框架。但它也提供了一些其他的基本特性。

其中包括 Spring MVC 和 Spring web 框架。已经了解了如何使用 Spring MVC 编写控制器类来处理 web 请求。但是，您还没有看到的是，Spring MVC 也可以用于创建产生非 HTML 输出的 REST API。我们将在第 2 章深入研究 Spring MVC，然后在第 6 章中讨论如何使用它来创建 REST API。

Spring 核心框架还提供了一些基本数据持久性支持，特别是基于模板的 JDBC 支持。将在第 3 章中看到如何使用 JdbcTemplate。

在 Spring 的最新版本（5.0.8）中，添加了对响应式编程的支持，包括一个新的响应式 web 框架 —— Spring WebFlux，它大量借鉴了 Spring MVC。将在第 3 部分中看到 Spring 的响应式编程模型，并在第 10 章中看到 Spring WebFlux。

**Spring Boot**

我们已经看到了 Spring Boot 的许多好处，包括启动依赖项和自动配置。在本书中我们确实会尽可能多地使用 Spring Boot，并避免任何形式的显式配置，除非绝对必要。但除了启动依赖和自动配置，Spring Boot 还提供了一些其他有用的特性：

 Actuator 提供了对应用程序内部工作方式的运行时监控，包括端点、线程 dump 信息、应用程序健康状况和应用程序可用的环境属性。

 灵活的环境属性规范。

 在核心框架的测试辅助之外，还有额外的测试支持

此外，Spring Boot 提供了一种基于 Groovy 脚本的替代编程模型，称为 Spring Boot CLI（命令行界面）。使用 Spring Boot CLI，可以将整个应用程序编写为 Groovy 脚本的集合，并从命令行运行它们。我们不会在 Spring Boot CLI 上花太多时间，但是当它适合我们的需要时，我们会接触它。

Spring Boot 已经成为 Spring 开发中不可或缺的一部分；我无法想象开发一个没有它的 Spring 应用程序。因此，本书采用了以 Spring Boot 为中心的观点，当我提到 Spring Boot 正在做的事情时，你可能会发现我在使用 Spring 这个词。

**Spring Data**

尽管 Spring 核心框架提供了基本的数据持久性支持，但 Spring Data 提供了一些非常惊人的功能：将应用程序的数据存储库抽象为简单的 Java 接口，同时当定义方法用于如何驱动数据进行存储和检索的问题时，对方法使用了命名约定。

更重要的是，Spring Data 能够处理几种不同类型的数据库，包括关系型（JPA）、文档型（Mongo）、图型（Neo4j）等。在第 3 章中，将使用 Spring Data 来帮助创建 Taco Cloud 应用程序的存储库。

**Spring Security**

应用程序安全性一直是一个重要的主题，而且似乎一天比一天重要。幸运的是，Spring 在 Spring security 中有一个健壮的安全框架。

Spring Security 解决了广泛的应用程序安全性需求，包括身份验证、授权和 API 安全性。尽管 Spring Security 的范围太大，本书无法恰当地涵盖，但我们将在第 4 章和第 12 章中讨论一些最常见的用例

**Spring Integration and Spring Batch**

在某种程度上，大多数应用程序将需要与其他应用程序集成，甚至需要与同一应用程序的其他组件集成。为了满足这些需求，出现了几种应用程序集成模式。Spring Integration 和 Spring Batch 为基于 Spring 的应用程序提供了这些模式的实现。

Spring Integration 解决了实时集成，即数据在可用时进行处理。相反，Spring Batch 解决了批量集成的问题，允许在一段时间内收集数据，直到某个触发器（可能是一个时间触发器）发出信号，表示该处理一批数据了。将在第 9 章中研究 Spring Batch 和 Spring Integration。

**Spring Cloud**

在我写这篇文章的时候，应用程序开发领域正在进入一个新时代，在这个时代中，我们不再将应用程序作为单个部署单元来开发，而是将由几个称为 微服务 的单个部署单元组成应用程序。

微服务是一个热门话题，解决了几个实际的开发和运行时问题。然而，在这样做的同时，他们也带来了自己的挑战。这些挑战都将由 Spring Cloud 直接面对，Spring Cloud 是一组用 Spring 开发云本地应用程序的项目。

Spring Cloud 覆盖了很多地方，这本书不可能涵盖所有的地方。我们将在第 13、14 和 15 章中查看 Spring Cloud 的一些最常见的组件。

**Summary**

Spring 的目标是让开发人员轻松应对挑战，比如创建 web 应用程序、使用数据库、保护应用程序和使用微服务。

Spring Boot 构建在 Spring 之上，简化了依赖管理、自动配置和运行时监控，让 Spring 变得更加简单。

Spring 应用程序可以使用 Spring Initializr 进行初始化，它是基于 web 的，并且在大多数 Java 开发环境中都支持它。

在 Spring 应用程序上下文中，组件（通常称为 bean）可以用 Java 或 XML 显式地声明，可以通过组件扫描进行发现，也可以用 Spring Boot 进行自动配置。

# 附录C 软件使用说明书

**1 大数据及信息技术产业协会网站**

**1.1系统概述**

本系统是一个大数据及信息技术产业协会网站，用来向社会各界人士展示当前大数据行业的发展方向与最新动态。此外连接协会大数据及信息技术产业协会内部各个成员，以增加相互的沟通交流、促进合作和资源共享，使大数据行业更快速发展。通过此网站，管理员可以发布新闻通知、协会成员信息、行业政策等各类大数据行业有关的信息。并且管理员还能监察协会会员单位招聘、协会成员之间项目合作及技术转让等情况。

**1.2功能介绍**

网站前台页面主要供用户访问，主要展示协会相关政策、通知、新闻、协会成员单位信息、协会成员人才招聘信息和专家信息等。后台管理系统分别由协会管理员和会员单位操作，二者操作权限不同，所能看到的界面操作菜单项也不同。协会管理员管理的后台系统由会员单位管理、信息管理、专家库管理、收入报送、技术转让、项目合作、平台撮合、招聘管理模块组成。会员单位后台由收入报送、项目合作、技术转让、招聘管理、简历管理组成。此外，管理员具有监督管理协会成员的功能，如监查协会成员发布的招聘信息、项目合作信息和技术转让信息是否合法。

1. 会员单位管理：管理员对协会成员信息进行维护，包括成员账号信息的新增、修改、删除、和查询操作以及成员发布信息权限的新增、修改、删除、和查询操作。
2. 信息管理：主要是管理本协会官网首页所展示的各种信息，对信息实行分类且实现信息的发布、修改、删除和查找。例如，管理员对新闻信息进行分类，且在发布新闻时选择相应类别信息类型。政策、通知、协会成员风采、协会成员人才招聘信息、协会成员合作、技术转让信息和专家信息等都是通过这个功能模块实现维护。
3. 专家库管理：管理员对专家信息进行维护，包括专家类别的新增、修改、删除、和查询操作以及专家详细信息的新增、修改、删除、和查询操作。
4. 人才招聘管理：会员单位根据自身的人才需求，可以通过平台将人才要求发布出去，发布后的信息会在前端人才招聘板块予以显示，游客或注册用户都可以查看到该信息，并投送简历。会员单位可以实现人才招聘信息的发布、修改、删除、检索、审核。
5. 项目合作管理：会员单位可能有一些项目需要协作完成，在没有合适的资源的前提下，会员单位可以将这种需求发布出去，有符合要求的会员单位就可以双方洽谈。会员单位可以实现合作项目信息的发布、撤销、查找、修改等操作。
6. 技术转让管理：会员单位若有一些需要转让的技术，可以通过这个网站平台寻找转让企业。会员单位可以实现技术转让信息的发布、修改、撤销、终止、查找等操作。
7. 收入报送管理：会员单位将每个月的收支报告上传至网站，供管理员查看业绩情况信息。

新闻通知界面如图C-1所示



图C-1 协会网站首页主界面

协会网站后台登录界面如图C-2所示



图C-2 协会网站后台登录界面

**2运行环境**

**2.1服务器端环境**

服务端环境如表C-1所示

表 C-1 服务器端环境

|  |  |
| --- | --- |
| 系统所使用数据库 | MySQL 5.7 |
| 操作系统 | Windows 10 |
| Web服务器 | Tomcat 8.5 |

**2.2客户端环境**

客户端环境如表C-2所示

表 C-2 客户端环境

|  |  |
| --- | --- |
| 操作系统 | Windows 7及其以上版本的操作系统 |
| 浏览器 | Firefox、Chrome、IE10.0等 |
| 分辨率 | 最佳效果为1920\*1080像素 |

# 附录D 主要源代码

协会管理员登录页面代码实现。

|  |
| --- |
| div class="col-md-4 col-md-offset-1" style="padding-top:40px;">  <div class="form-top">  <div >  <h3 style="color:white">协会管理员登录窗口</h3>  </div>  </div>  <div class="form-bottom">  <form role="form" action="" method="post" >  <div class="form-group">  <label class="sr-only" for="txtNumber">UserName</label>  <div class="input-group">  <div class ="input-group-addon">  <span class="glyphicon glyphicon-user"></span></div>  <input type="text" name="number" placeholder="账号" class=" form-control" id="txtNumber" required>  </div>  </div>  <div class="form-group">  <label class="sr-only" for="txtPwd">PassWord</label>  <div class="input-group">  <div class="input-group-addon"><span class="glyphicon glyphicon-lock"></span></div>  <input type="password" name="password" placeholder="密码" class="form-control" id="txtPwd" required>  </div>  </div> |

新闻通知查询业务代码实现。

|  |
| --- |
| function findNewsList() {  var con = new Object();  con.info\_type = "新闻通知";  doData("Info/findByType", con, function (result) {  console.log(result)  $("#newsList").empty();  if (result != null && result.obj != undefined && result.obj.length > 0) {  for (var i = 0; i < result.obj.length; i++) {  $("#newsList").append(  '<div class="single-post d-flex flex-row row">' +  '<div class="thumb col-lg-1">' +  '</div>'+  '<div class="details col-lg-10">'+  '<div class="title d-flex flex-row justify-content-between">'+  '<div class="titles">'+  '<a href="'+getRootPath()+"/admin/showIndex/newsDetail.jsp?id="+result.obj[i].id+'">'+'<h4>'+result.obj[i].info\_title+'</h4>'+'</a>'+  '</div>'+  '</div>'+  '<p>'+result.obj[i].info\_content.substr(0,100)+'......'+'</p>'+  '</div>'+  '</div>'  )  }  }  }, true);  } |

技术转让查询业务代码实现。

|  |
| --- |
| function find() {  var con=new Object();  doData("Technology/find",con,function (result) {  $("#tData").empty();//清空原有数据  var total=0;  if (result!=null && result.obj!=undefined && result.obj.length>0){  for(var i=0;i<result.obj.length;i++){  $("#tData").append("<tr>" +  "<td>"+ result.obj[i].name+"</td>" +  "<td>"+ result.obj[i].types+"</td>" +  "<td><img src='"+getRootPath()+"/uploadfiles/"+result.obj[i].img+"' style='width:60px;height:80px;' ></td>" +  "<td>"+ result.obj[i].power+"</td>" +  "<td>"+  // "<button class='btn-warning btn-xs' doType='edit' recordId='"+result.obj[i].id+"'>修改</button>"+  "<button class='btn-danger btn-xs' doType='del' recordId='"+result.obj[i].id+"'>删除</button></td></tr>");  }  }  },true);  } |

简历管理业务代码实现。

|  |
| --- |
| function find() {  var con=new Object();  power=sessionStorage.getItem("power");  console.log(power);  con.power=power;  doData("Vita/findVitaByPower",con,function (result) {  $("#tData").empty();//清空原有数据  var total=0;  if (result!=null && result.obj!=undefined && result.obj.length>0){  for(var i=0;i<result.obj.length;i++){  $("#tData").append("<tr>" +  "<td>"+ result.obj[i].name+"</td>" +  "<td>"+ result.obj[i].age+"</td>" +  "<td>"+ result.obj[i].tel+"</td>" +  "<td>"+ result.obj[i].degree+"</td>" +  "<td>"+ result.obj[i].recruit\_name.name+"</td>" +  "<td>"+ result.obj[i].recruit\_name.power+"</td>" +  "<td>"+result.obj[i].vita\_files+"</td>" +  "<td>"+  // "<button class='btn-warning btn-xs' doType='edit' recordId='"+result.obj[i].id+"'>修改</button>"+  "<button class='btn-danger btn-xs' doType='del' recordId='"+result.obj[i].id+"'>删除</button></td></tr>");  }  }  },true);  } |