**丽 水 学 院**

**毕业设计（论文）**

**（2018届）**

**题 目 基于SSM框架的实习生招聘网开发**

**指导教师 沈伟华**

**院 系 工学院**

**专 业 计算机科学与技术**

**班 级 计142**

**学 号 14103330129**

**姓 名 巫坤**

2018年4月13日

# 摘 要

实习生招聘网站是提供职位发布，职位搜索，简历投递一系列服务的平台。同时为了让用户有更多的自主性，提供了实习生用户模块、HR用户模块、后台管理系统三大系统，以供三类用户使用。为了保证以后网站运行的稳定性，选取合适的运行服务器则显得非常重要。Spring Mvc是一款优秀的MVC的开发框架，强大的性能和易用性，为网站在服务器端稳定安全地运行提供了保障；Spring对所有bean进行管理，并且与Mybatis整合。Require.js前端框架则实现了前端js模块化管理，有效解决js引用代号冲突问题，更加清晰统一化管理依赖js。

本论文介绍了基于spirng mvc+Spring+mybatis+redis框架开发的实习生招聘网设计与实现，本项目以IDEA+Jetty+Mysql5.0为开发平台。

**关键词：Spring Mvc；Spring；Mybatis；Redis；框架**

Development of intern recruitment network based on SSM framework

# Abstract

The intern job site is a platform for job Posting, job search, and resume delivery. At the same time, in order to give users more autonomy, three systems of intern user module, HR user module and background management system are provided for the use of three types of users. In order to ensure the stability of the website, it is very important to select the proper running server. Spring Mvc is a good Mvc development framework with strong performance and ease of use, which provides guarantee for the stable and safe operation of the website on the server side. Spring manages all beans and integrates with Mybatis. The front-end framework of Require. Js implements the front-end js modularization management, effectively solves the conflict problem of js reference code, and makes it more clear and unified management dependence js.

This paper introduces the design and implementation of intern recruitment network based on spirng MVC +Spring+mybatis+redis framework. This project is a development platform with IDEA+Jetty+Mysql5.0.

**Key Words：Spring Mvc；Spring；Mybatis；Redis；Framework**

目 录

[摘 要 I](#_Toc511519716)

[Abstract II](#_Toc511519717)

[引 言 V](#_Toc511519718)

[第一章 概述 1](#_Toc511519719)

[1.1 开发的背景 1](#_Toc511519720)

[1.2 系统功能概述 1](#_Toc511519721)

[1.3开发环境介绍 2](#_Toc511519722)

[第二章 需求分析 4](#_Toc511519723)

[2.1 用户需求分析 4](#_Toc511519724)

[2.1.1 实习生用户模块需求 4](#_Toc511519725)

[2.1.2 HR用户模块需求 5](#_Toc511519726)

[2.1.2 后台管理系统需求 5](#_Toc511519727)

[2.2 系统的特点 5](#_Toc511519728)

[2.3 数据流程图 6](#_Toc511519729)

[第三章 系统概要设计 8](#_Toc511519730)

[3.1 数据库概念设计 9](#_Toc511519731)

[3.2 数据库逻辑设计 9](#_Toc511519732)

[3.2 系统功能模块设计 16](#_Toc511519733)

[3.2.1 实习生功能模块 16](#_Toc511519734)

[3.2.2 HR功能模块 17](#_Toc511519735)

[3.2.3 后台管理系统模块 19](#_Toc511519736)

[3.3界面设计 20](#_Toc511519737)

[3.3.1实习生用户模块界面设计 20](#_Toc511519738)

[3.3.2 HR用户模块界面设计 25](#_Toc511519739)

[3.3.3后台管理系统界面设计 28](#_Toc511519740)

[3.3.4公共页面界面设计 28](#_Toc511519741)

[第四章 详细设计与实现 30](#_Toc511519742)

[4.1系统文件结构的设计 30](#_Toc511519743)

[4.1.1 java类的组织 30](#_Toc511519744)

[4.2系统架构的设计 31](#_Toc511519745)

[4.2.1表示层的设计 32](#_Toc511519746)

[4.2.2业务逻辑层的设计 40](#_Toc511519747)

[4.2.3数据库持久层的设计 43](#_Toc511519748)

[第五章 测试与分析 48](#_Toc511519749)

[5.1 测试环境 48](#_Toc511519750)

[5.2 实例测试 48](#_Toc511519751)

[第六章 结论 49](#_Toc511519752)

[参考文献 51](#_Toc511519753)

[致谢 52](#_Toc511519754)

# 引 言

实习是大学面临毕业的同学必须经历的阶段，如何快速找到可靠有意义的实习工作对每一位大学生来说都尤为重要。对于目前常规的找工作方式，一般是要选择一个可靠的招聘网站，然后注册账号，按自己的条件去搜索实习工作，但是一般的招聘网站不仅仅面向大学生实习这一群体，还有其他各种形式的工作。而对于经验匮乏的毕业生来说，如何在信息量大而杂中找到合适并且可靠的职位，如何辨别这个公司是否是诈骗或者培训公司是非常困难的。

为此本论文阐述的就是专注于实习生招聘网的开发，服务对象就是即将毕业的大学生。目的是让实习生用户可以在更加简洁的环境下，从已发布的职位理性搜索实习职位，而职位对应的信息更加全面与透彻，工资、实习时间、工作地点等要素都会完整的展示。实习公司都会实名化并进行注册认证,增加了平台服务的安全性。

本论文将会剖析当下企业级应用开发最流行的SSM架构在本系统中的应用，去感受spring框架、mybatis持久层框架、spring mvc web框架的实用价值。

# 第一章 概述

本人所修专业是计算机科学与技术专业，这门专业主要学习计算机科学与技术方面的基本理论和基本知识。

网站开发伴随着这些年出现的spring mvc,Spring,Mybaits框架，Ajax技术，XML等新技术使一个单纯的网站项目开发越来越容易。本次项目采用SSM架构，后台数据库选用MySQL5.0。

## 1.1 开发的背景

伴随着中国互联网的迅猛发展，如今越来越多的领域朝向互联网靠近，自然而然“互联网+”成了国家战略的一部分。而就招聘领域来说，网上招聘成为当下最流行的应聘方式。国内最早的传统三大招聘网站有前程无忧，智联招聘以及中华英才网，他们的服务对象类似，对所有大众开放，涉及领域宽泛。针对上述传统网上招聘虽然内容多，由此也衍生出没有对某一特殊对象的精准化服务。由此产生了本论文的研究课题，即针对大学生群体的精细化就业服务平台。

毕业实习是我们大学学习阶段重要的实践性教学环节之一，是理论与实践相结合的重要方式，是每一位大学生必须经历的阶段。实习环节是毕业生未来职业生涯起点和校园生涯结束点的有效承接，而一份好的实习工作对未来的发展有极大的促进作用。

围绕着如何实现毕业生及时接受就业信息，引导毕业生快速抉择就业方向，根据个人专业情况以及未来规划提供更多的职位选择，毕业生精准投递简历，公司对简历的有效筛选以及更有效率的网罗到人才，为此通过spring,spring mvc,mybatis整合框架技术开发这个实习生招聘网。本平台的使用对象有三类，分别是实习生，公司hr，平台管理员。实习生可以使用平台提供的简历模板制作简历，选择心仪的职位投递。公司hr根据所发布职位获取对应投递的简历，进行条件筛选过滤部分简历，提高浏览简历决定面试对象的效率。

该平台系统由于只针对实习职位，可以让处于实习阶段的毕业生获得更加准确的信息，针对性有助于提高实习生寻找实习工作的效率以及公司网罗人才的效率。

## 1.2 系统功能概述

本系统主要是服务于实习生找实习工作以及企业寻找合适的可培养人才。本网站平台主要包括实习生用户模块、HR用户模块、后台管理模块等几大部分，本平台具有良好的用户接口，用户交互，使用方便，具有完善的浏览已发布职位信息，实习生可以综合筛选出合适的职位进行投递简历，实习生可以对简历进行修改。HR用户所具有的注册认证公司，发布职位，简历管理等功能是本平台的重要组成部分。后台管理系统的作用相当于维护辅助，能及时、方便、灵活地进行查询、修改、删除等维护性操作。后台管理系统有对于实习生，HR用户有一定的管理，并对管理员用户的权限有一定的设置。系统主要实现以下几个功能模块：实习生用户模块, HR用户模块, 后台管理模块。

(1). 实习生用户模块

当实习生用户登录以后，在公共访问页面的顶端会显示用户登录的信息。个人中心则需要判断当前用户是否登录，登录后才允许访问个人管理界面。职位投递按钮点击后也需要验证实习生用户是否登录。

(2).HR用户模块

HR用户模块要有基本信息管理，发布职位功能，职位管理以及简历管理功能。HR用户只能注册一家公司，并且审核通过后才有发布职位的权限。

(3).后台管理模块

后台管理系统的服务对象主要是网站管理者，具有职位类别设置，企业增值服务提供，企业信息管理，实习生用户信息管理，HR信息管理。

(4).公共页面模块

实习页面部分：提供招聘职位展示多条件搜索服务，公司详细信息，职位详细信息查看。

校招页面部分：提供校招资讯，企业宣讲会信息。

朵喵学院页面部分：提供相关面试笔试资料和技能，为实习生用户提供帮助。

## 1.3开发环境介绍

本项目采用的开发组合是:IDEA + Maven + JDK1.8 + Jetty + MYSQL5.0 + Redis

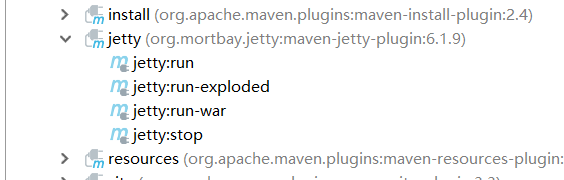
+Git+Github+Windows10

IDEA是目前市面上最流行的java开发工具之一，其本身是用java语言开发的集成环境，界面舒适功能丰富，同时携带很多开发需要的插件譬如版本管理工具git,svn、spring相关、maven插件、Web服务器Jboss,Tomcat支持等。编码过程中会有智能代码提示，对于开发者来说可以提高写代码的效率。总之这是一款十分强大java开发软件。

本项目采用Maven工具进行管理依赖包，严格来说就是一个maven项目。Maven是Apache旗下的产品，它的定义是一个项目管理以及综合的工具，专业角度阐述提供了一个项目对象模型来管理项目构造。它可以帮助开发人员快速的构建一个具有完整生命周期。Maven具有三套相互独立的生命周期，而最重要的就是Default Lifecycle，其核心部分包括常用的编译（compile），测试(test)，打包(package),部署(site)命令。使用maven前需要下载该工具，并配置相应的本地仓库以及环境变量，为了提升依赖包下载速度在配置远程中央仓库时可以将阿里仓库作为首选，当找不到依赖包再向maven官方的中央仓库下载相关依赖包。

Java的开发环境采用JDK1.8版本。首先介绍jdk，jdk的中文释义是java语言的软件开发包工具，顾名思义可以看出它是整个java开发的核心。Jdk内包括了JRE,即java的运行环境。JRE内含JVM,JVM是java虚拟机，它是实现java语言一次编译到处运行的关键。JDK包含的基本组件有javac,javadoc,java,jar.javap..，常用的包有lang,io,nio,net,util,sql...。

Jetty是开发时用到的服务器，其特点在于可以通过XML来对它进行设置，体现其易用性。Jetty本身也是用java语言编写的，可以说他就是一个servlet容器。相比于Tomcat，Jetty显得更加轻量级。Jetty在IDEA中的结构如下图所示：



Jetty:run命令用于开发时启动项目在本地localhost:8083/duomiao/测试查看项目。

MYSQL5.0是关系型数据库开发工具，其可以支持大型的数据库可以处理上千万条记录。由于其快速轻巧，它多被应用于许多小型企业。开发环境下采用windows版本的mysql，部署环境则采用linux版的mysql。

Redis数据库作为本次项目的mybatis的二级缓存，它是基于内存亦可持久化的Key-Value数据库，即非关系型数据库。读写速度快，性能极高等特点使其在消息队列，热数据（经常读，但不经常修改的数据）领域很受欢迎。当然redis也支持数据备份，同时还有list,hash,set等存储结构。

Git是分布式版本管理工具，基于linux内核开发，所以其终端类似于shell。本项目用git进行版本管理，在项目根目录下运行 git init命令将项目纳入版本管理。Git由于是分布式，所以没有中心服务器，可以在本地建立本地仓库，当然也可以创建远程仓库当做中央仓库。Git常用命令有git init,git clone,git status,git log,git add,git diff,git commit,git revert,git rm,git rm –cached(不清除工作目录),git clean(清除未track的文件)。

Github是全球最大的开源项目托管平台，其原理也是将git作为版本管理的工具。

此次项目采用github和git结合管理。本地git仓库和github远程仓库采用ssh通讯方式（添加同一公钥），借助IDEA将创建好的项目纳入本地仓库，然后将仓库提交到github远程仓库，代码提交时同时向本地和远程commit push,这样既能方便项目更新又能很好的安全保存。

# 第二章 需求分析

## 2.1 用户需求分析

### 2.1.1 实习生用户模块需求

实习生用户注册账户后，有权自己的profile进行操作，但是为了防止用户恶意注册，账户默认状态是未激活。为此本项目只允许采用邮箱注册，并通过邮箱激活后账户才生效。实习生的个人信息，譬如提供密码展示修改，真实姓名修改，这些对于用户来说是常规功能。所以本项目提供基本信息模块对这些重要信息进行统一管理。

在实习生角度出发，个人简历如何创建，如何填写，如何修改都要从系统的完整性以及用户体验度出发。简历内容需要包括姓名，性别，年龄，籍贯，手机号码，邮箱，专业，学校，学历，意向职位，工作经验，项目经验等等。实习生只有简历的修改查看权限。

实习生用户在浏览职位或企业的同时，可能有心仪的职位或企业出现，这时收藏功能尤为重要。当然公司收藏，职位收藏需要独立开来。我只需要建立起收藏对象与实习生的联系即可，并提供其对收藏内容的查看，删除权限。

求职者之间信息的沟通也是非常重要的，实习生之间需要一个交流平台，为此本系统提供一个名为“猫侍社区”论坛平台。实习生可以在这个平台上进行发帖或评论回复。针对实习生的求职技巧薄弱，专业性的就业辅导很有必要，继而需要对课程进行管理。为此本系统提供了朵喵学院，这一板块除了有课程体系之外还有一些文章资料信息。

### 2.1.2 HR用户模块需求

HR即招聘方，作为企业的人力资源管理者需要在最短的时间内找到符合要求的职员并不是一件容易的事情。如何找到优质实习生源，让实习生关注到自己的公司以及招聘岗位，这些都是HR最大的需求。为此本平台将实习生，HR两方资源汇聚在一起，作为两者连接的桥梁。HR需要发布职位，并且对已发布的职位进行管理，包括投递过来的简历进行筛选。

### 2.1.2 后台管理系统需求

平台运营方需要对平台进行必要的管理，比如职位类别的管理，这部分数据会影响到前端页面的展示。动态数据的建立，提高了系统的灵活度。后台管理最核心的需求就是对三类用户的信息管理，即实习生用户管理，企业信息管理（HR用户），后台管理员用户，当然密码部分没有查看权限。

企业增值服务需求，有首页企业logo展示栏的展示机制以及职位置顶，这些都是运营方盈利的要素。

## 2.2 系统的特点

系统需要实现以下特点：

1. 安全性，杜绝恶意注册，未认证HR的权限控制；数据存储方面的id采用UUID避免破解，持久层sql代码防止sql注入。
2. 高效性，精简代码，服务器优化，数据库连接池采用druid。
3. 交互良好，前端页面尽量简洁，按钮或者模块清晰，对应的业务逻辑流畅。
4. 扩展性高，代码采用面向接口的方式编程，尽量提高系统的可维护，可扩展性，满足项目的二次开发。

## 2.3 数据流程图

全局数据流程图，如图2-1所示

HR

模块用户

实习生

模块用户

注册，录入

注册，录入

管理

管理

浏览

图2-1 全局数据流程图

实习生模块数据流程图，如图2-2所示

管理请求

登录请求

注册请求

退出

图2-2 实习生数据流程图

HR模块数据流程图，如图2-3所示

管理请求

登录请求

注册请求

退出

图2-3 HR数据流程图

后台管理系统模块数据流程图，如图2-4所示

管理员管理

管理请求

登录请求

退出

图2-4 管理员数据流程图

# 第三章 系统概要设计

## 3.1 数据库概念设计

本系统主要的实体包括实习生用户和HR用户，他们之间的关系主要体现在职位&简历模块，我们列出系统表中所体现的关系图，关系图如图3-1所示。

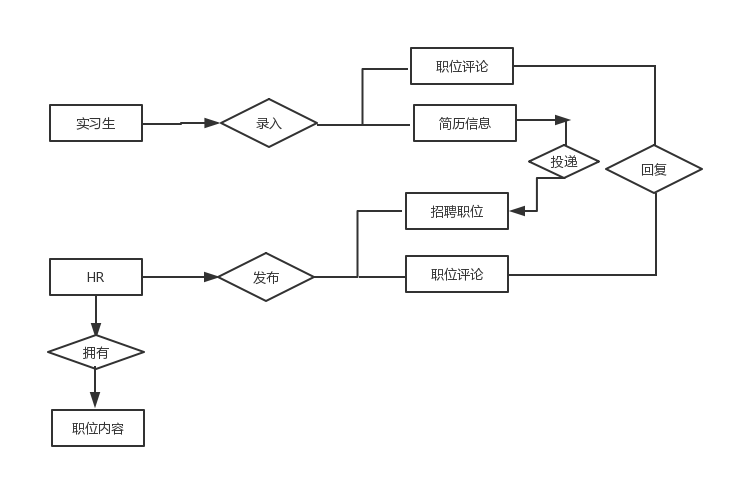


图3-1 平台ER图

## 3.2 数据库逻辑设计

本项目使用Mysql5.0数据库,考虑到其轻巧功能强大稳定的特性。构建数据库名称为duomiaodb，该数据库共有十几张表。为了方便以后开发查看表信息，以及数据库的维护，采用一定的规约设计，数据库的表名大写表示方式为“DM\_tablename”,即以“DM\_”为前缀,也有部分表采用“MS\_”为前缀；除了ID以外的每个表字段命名限定为“前缀\_字段名”，前缀为当前数据表名字的缩写；ID由java代码方式生成UUID，避免ID被破解或者降低重复率。

具体表说明如下表3-1所示：

表3-1数据库表的详细设计

|  |  |  |
| --- | --- | --- |
| 表 名 | 中文名 | 功 能 说 明 |
| DM\_INTERN\_INFO | 实习生信息表 | 保存实习生账号信息的数据表，包括登录信息，相关状态 |
| DM\_RESUME\_INFO | 简历信息表 | 存储实习生简历内容的数据表，包括基本信息，工作经历，项目经验.. |
| DM\_ENTERPRISE\_INFO | 公司信息表 | 学生学习课程时，对课程的学习感受等内容保存在这张表里，主要对应课程笔记模块 |
| DM\_HR\_INFO | HR信息表 | 保存HR账户信息，包括其联系方式注册时间等等，在后台对其进行管理 |
| DM\_JOB\_REPLY | 评论回复存储表 | 存储评论信息，该表的主要字段是hr ID,实习生ID，职位ID |
| DM\_POSITION\_CATEGORY | 职位类别表 | 保存职位大类和小类信息，用于管理员在后台管理系统中对职位类别的设置 |
| DM\_PUBLISH\_JOB | 已发布职位表 | 本张表保存了已发布职位的基本信息，在HR用户模块里是最重要的表之一 |
| DM\_ENT\_RESUME | 企业与简历关系表 | 保存了企业所受的简历信息，通过企业ID，职位ID，简历ID三者简历联系。使HR能够获得相关简历的查看权限 |
| DM\_ENT\_COLLECT | 公司收藏表 | 保存了实习生用户所收藏所有的公司基本信息，主要对应实习生模块，本表有实习生ID，公司ID以及公司名称等字段 |
| DM\_JOB\_COLLECT | 职位收藏表 | 保存了实习生用户所有收藏的心仪职位的基本信息，由实习生用户来进行管理（增加、删除、查看），主要字段有实习生ID，收藏职位ID，职位名称，公司名称等等 |
|  |  |  |
|  |  |  |
|  |  |  |

在DM\_INTERN\_INFO（实习生信息表）中，存放了实习生用户的基本信息。详细信息如表3-2所示。

表3-2 DM\_INTERN\_INFO表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实习生信息表(DM\_INTERN\_INFO） | | | | | |
| 字段 | 中文名 | 类型 | 约束 | 备注 | |
| ID | ID标识 | varchar(40) | 主键 | UUID(通用唯一识别码) |  |
| INT\_LOGIN\_NAME | 登录名 | varchar(20) | unique | 使用邮箱注册 |  |
| INT\_LOGIN\_PWD | 密码 | varchar(20) |  | 6-10位数字和字母混合 |  |
| INT\_REAL\_NAME | 真实姓名 | varchar(20) |  |  | |
| INT\_IMG\_URL | 头像路径 | varchar(255) |  |  | |
| INT\_DB | 朵币 | varchar(20) |  | 数字 | |
| INT\_FLAG | 唯一标识 | char(1) |  | 默认为I | |
| INT\_STATE | 状态 | char(1) |  | 0为无效，1为有效 | |
| INT\_REGISTER\_TIME | 注册时间 | datetime |  | 24小时制，精确到秒 | |

在 DM\_RESUME\_INFO（简历信息表）中，存放了实习生用户简历的基本信息。详细信息如表3-3所示。

表3-3 DM\_RESUME\_INFO表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 简历信息表（DM\_RESUME\_INFO） | | | | | |
| 字段 | 中文名 | 类型 | 约束 | 备注 | |
| ID | ID标识 | varchar(40) | 主键 | UUID(通用唯一识别码) |  |
| RI\_NAME | 姓名 | varchar(20) |  |  | |
| RI\_MOTTO | 人生格言 | varchar(20) |  | 字数限制在12个字以内 |  |
| RI\_SEX | 性别 | varchar(20) |  | 0位女，1为男 |  |
| RI\_AGE | 年龄 | varchar(5) |  |  | |
| RI\_HP | 籍贯，出生地 | varchar(10) |  | 格式 |省份+城市 | |
| RI\_TEl | 手机号码 | varchar(20) |  | 必填 | |
| RI\_EMAIL | 邮箱 | varchar(22) |  |  | |
| RI\_MAJOR | 专业 | varchar(20) |  |  | |
| RI\_COLLOGE | 学校 | varchar(20) |  |  | |
| RI\_ACADEMIC | 学历 | varchar(5) |  |  | |
| 求职意向（部分） | | | | | |
| RI\_JOB\_NAME | 职位 | varchar(20) |  |  |  |
| RI\_WORKPLACE | 期望工作地点 | varchar(10) |  | 不能超过10个字 |  |
| RI\_EDUCATION | 教育背景 | longtext |  |  |  |
| RI\_CERT\_REWARD | 奖励证书 | longtext |  |  |  |
| RI\_SKILL | 专业技能 | longtext |  |  |  |
| RI\_WORK\_EXPERIENCE | 工作经验，实习经验 | longtext |  |  |  |
| RI\_PRO\_EXPERIENCE | 项目经验 | longtext |  |  |  |
| RI\_VALUATION | 自我评价 | longtext |  |  |  |

在 DM\_ENTERPRISE\_INFO（公司信息表）中，存放了公司相关的基本信息。详细信息如表3-4所示。

表3-4 DM\_ENTERPRISE\_INFO表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 公司信息表（DM\_ENTERPRISE\_INFO） | | | | | |
| 字段 | 中文名 | 类型 | 约束 | 备注 | |
| ID | ID标识 | varchar(40) | 主键 | UUID(通用唯一识别码) |  | |
| EI\_ENINAME | 企业名称（全称） | varchar(20) | unique,非空 |  |  | |
| EI\_RESUME | 简要概述,一句话介绍 | varchar(20) |  | 字数不能超过20字 |  | |
| EI\_IMG\_URL | 公司图像 | varchar(255) |  |  |  | |
| EI\_INTRODUCTION | 介绍 | longtext |  |  |  | |
| EI\_ADDRESS | 地址 | varchar(30) |  |  |  | |
| EI\_TEL | 公司联系方式 | varchar(20) | 不为空 |  |  | |
| EI\_EMPLOYEENUM | 员工数 | int |  |  |  | |
| EI\_INDUSTRY | 行业方向，领域 | varchar(15) |  |  |  | |
| EI\_UNISCID | 统一社会信用代码 | varchar(30) |  |  |  | |
| EI\_REGTIME | 成立日期 | date |  |  |  | |
| EI\_REG\_CAPITAL | 注册资本 | varchar(15) |  |  |  | |
| EI\_WELFARE | 企业福利 | longtext |  |  |  | |
| EI\_STATUS | 企业审核状态 | char(1) |  | 0为无效，1为有效 |  | |
|  |  |  |  |  |  | |

在 DM\_HR\_INFO（hr信息表）中，存放了HR用户相关的基本信息。详细信息如表3-5所示。

表3-5 DM\_HR\_INFO表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| hr信息表（DM\_HR\_INFO） | | | | | |
| 字段 | 中文名 | 类型 | 约束 | 备注 | |
| ID | ID标识 | varchar(40) | 主键 | UUID(通用唯一识别码) |  |
| HI\_HR\_NAME | 账户 | varchar(20) | unique,非空 | 任意注册 |  |
| HI\_PASSWORD | 密码 | varchar(32) |  | 6-10位数字和字母混合 |  |
| HI\_IMG\_URL | hr头像路径 | varchar(255) |  | 绝对路径 |  |
| HI\_ENTID | ID标识 | varchar(40) | null | 企业ID |  |
| HI\_FLAG | 唯一标识 | char(1) |  | 默认为H |  |
| HI\_TEL | hr联系方式 | varchar(20) |  |  |  |
| HI\_REGTIME | 注册时间 | datetime |  |  |  |

在 DM\_JOB\_REPLY（评论回复信息表）中，存放了评论回复相关的基本信息。详细信息如表3-6所示。

表3-6 DM\_JOB\_REPLY表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 评论回复储存表（DM\_JOB\_REPLY） | | | | | |
| 字段 | 中文名 | 类型 | 约束 | 备注 | |
| ID | ID标识 | varchar(40) | 主键 | UUID(通用唯一识别码) |  |
| JR\_HR\_ID | hr ID | varchar(40) | 非空 | 邮箱注册，验证码，格式 |  |
| JR\_INT\_ID | 实习生ID | varchar(40) |  |  |  |
| JR\_MAIN\_INFO | 评论主体内容 | varchar(52) |  | 不能超过50字 |  |
| JR\_REPLY\_TIME | 回复时间 | datetime |  |  |  |
| JR\_JOB\_ID | 职位ID | varchar(40) |  |  |  |
| JR\_PARENT\_ID | 父ID | varchar(40) |  | 0为无 |  |
| JR\_REPLY\_NUM | 评论数量 | varchar(10) |  |  |  |
| JR\_AGREE\_NUM | 赞同数量 | varchar(10) |  |  |  |
| JR\_CLICK\_USER | 点过赞的用户UUID拼接在一起 | longtext |  | 初始值为""逗号作为分隔符 |  |

在 DM\_POSITION\_CATEGORY（职位类别信息表）中，存放了职位类别相关的基本信息。详细信息如表3-7所示。

表3-7 DM\_POSITION\_CATEGORY表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 职位类别表（DM\_POSITION\_CATEGORY） | | | | |
| 字段 | 中文名 | 类型 | 约束 | 备注 |
| ID | ID标识 | varchar(40) | 主键 | UUID(通用唯一识别码) |
| PC\_TYPE\_NAME | 类别名称 | varchar(20) | 非空，unique | 不能重名 |
| PC\_FATHER\_ID | 父ID | varchar(40) |  | 空默认为0 |
| PC\_URL | 链接 | varchar(255) |  |  |

在 DM\_PUBLISH\_JOB（已发布职位信息表）中，存放了已发布职位相关的基本信息。详细信息如表3-8所示。

表3-8 DM\_PUBLISH\_JOB表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 已发布职位表（DM\_PUBLISH\_JOB） | | | | | |
| 字段 | 中文名 | 类型 | 约束 | 备注 | |
| ID | ID标识 | varchar(40) | 主键 | UUID(通用唯一识别码) |  |
| PJ\_JOB\_NAME | 职位名称 | varchar(20) | 非空 |  | |
| PJ\_ENT\_ID | 所属公司 | varchar(40) |  | 空默认为0 | |
| PJ\_PC\_ID | 所属职位的类别 | varchar(40) |  |  | |
| PJ\_MIN\_SALARY | 最低薪水 | varchar(10) |  |  | |
| PJ\_MAX\_SALARY | 最高薪水 | varchar(10) |  |  | |
| PJ\_MIN\_EDUCATION\_LEV | 最低学位要求 | varchar(5) |  | 大专，本科，博士，硕士 四选一 |  |
| PJ\_WORK\_PLACE | 工作地点 | varchar(10) |  | 选择 市一级 下拉框 |  |
| PJ\_DESCRIBE | 职位描述 | longtext |  |  | |
| PJ\_WORK\_DAYS | 每周工作天数 | varchar(2) |  |  | |
| PJ\_WORK\_MONTH | 工作月数 | varchar(2) |  |  | |
| PJ\_UPDATE\_TIME | 刷新时间 | datetime |  |  | |
| PJ\_CUT\_DATE | 截止日期 | datetime |  | 2017-12-30 | |
| PJ\_URGENCY | 急招标志 | char(1) |  | 1代表急招，0代表默认，需要付钱 |  |
| PJ\_SEND\_NUM | 热门职位标志HOT | int |  | 投递次数最多 |  |

在DM\_ENT\_RESUME（企业与简历关系信息表）中，存放了企业与简历关系相关的基本信息。详细信息如表3-9所示。

表3-9 DM\_ENT\_RESUME表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 企业与简历关系表（DM\_ENT\_RESUME） | | | | | |
| 字段 | 中文名 | 类型 | 约束 | 备注 | |
| ID | ID标识 | varchar(40) | 主键 | UUID(通用唯一识别码) |  |
| ER\_ENT\_ID | 企业ID | varchar(40) | 非空 |  | |
| ER\_PJ\_ID | 所属职位ID | varchar(40) |  |  | |
| ER\_RI\_ID | 简历ID | varchar(40) |  | 解决重复投递，唯一标识 |  |
| ER\_ADD\_TIME | 投递时间 | datetime |  |  | |
| ER\_SEX | 性别 | char(1) |  |  | |
| ER\_AGE | 年龄 | varchar(3) |  |  | |
| ER\_ACADEMIC | 学历 | varchar(5) |  | 大专，本科，博士，硕士 四选一 | |
| ER\_ISDEL | 是否删除 | char(1) |  | 默认为0没有删除 1为删除 | |
| ER\_DEL\_TIME | 删除时间 | datetime |  | 数据恢复后删除时间为null | |

在DM\_ENT\_COLLECT（公司收藏信息表）中，存放了所有收藏公司相关的基本信息。详细信息如表3-10所示。

表3-10 DM\_ENT\_COLLECT表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 公司收藏表（DM\_ENT\_COLLECT） | | | | | |
| 字段 | 中文名 | 类型 | 约束 | 备注 | |
| ID | ID标识 | varchar(40) | 主键 | UUID(通用唯一识别码) |  |
| EC\_INT\_ID | 实习生ID | varchar(40) | 非空 |  | |
| EC\_ENT\_ID | 公司ID | varchar(40) |  |  | |
| EC\_URL | 链接 | varchar(255) |  |  | |
| EC\_ENT\_NAME | 公司名称 | varchar(20) |  |  | |
| EC\_INPUT\_TIME | 录入时间 | datetime |  |  | |

在DM\_JOB\_COLLECT（职位收藏信息表）中，存放了实习生用户所有对应收藏职位相关的基本信息。详细信息如表3-11所示。

表3-11 DM\_JOB\_COLLECT表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 职位收藏表（DM\_JOB\_COLLECT） | | | | | |
| 字段 | 中文名 | 类型 | 约束 | 备注 | |
| ID | ID标识 | varchar(40) | 主键 | UUID(通用唯一识别码) |  |
| JC\_INT\_ID | 实习生ID | varchar(40) | 非空 |  | |
| JC\_PJ\_ID | 收藏职位ID | varchar(40) |  |  | |
| JC\_PJ\_NAME | 职位名称 | varchar(20) |  |  | |
| JC\_GO\_URL | 去向路径 | varchar(255) |  |  | |
| JC\_ENT\_NAME | 公司名称 | varchar(20) |  |  | |
| JC\_INPUT\_TIME | 录入时间 | datetime |  |  | |

## 3.2 系统功能模块设计

### 3.2.1 实习生功能模块

实习生用户功能模块实现以下功能模块，如图3-2所示。

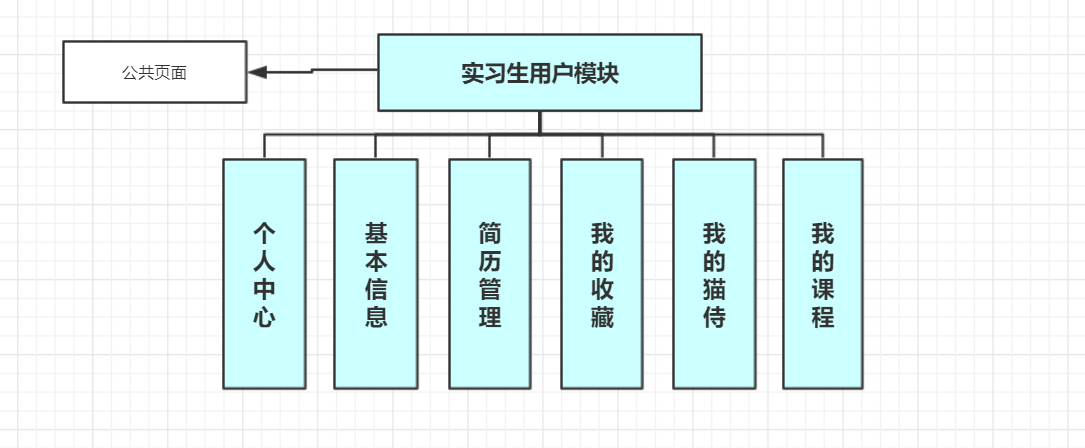


图3-2 实习生功能模块功能划分图

(1) 个人中心，是整个实习生功能模块的中枢神经，有点类似于用户的后台管理界面；在个人中心模块下，包含基本信息模块、简历管理模块、我的收藏模块、我的猫侍模块以及我的课程模块共5个子模块，实习生用户的profile配置或者核心的简历填写生成以及简评组件都可以在这些子模块功能区中完成。

(2) 基本信息：该模块只在个人中心里展示，主要显示的信息有实习生用户的用户名，密码，真实姓名，头像图片。其中用户名只读不可修改，密码可以修改但不能为空，真实姓名可以为空，可以上传头像（不上传时采用默认头像显示）。

(3) 简历管理：本子模块用户展示实习生用户的简历，实习生可以对简历进行初始化创建，后期修改，但不具备删除简历的权限；关于简历内容的填写限制，姓名、性别、年龄、籍贯、手机号码、专业、学校、学历为必填项，此外可选填的有人生格言、邮箱、期望职位、期望工作地点、教育背景、奖励证书、专业技能、工作经验、项目经验、自我评价部分。

(4) 我的收藏：我的收藏主要是展示实习生用户所收藏的企业和职位的信息；实习生用户对每个对象只能收藏一次，可以对已收藏对象进行删除操作，但不具备修改操作，对于收藏对象的数量也会有一定的限制。

(5) 我的猫侍：该模块针对猫侍社区论坛实习生对应的账户管理，包括初始化（不可修改）的代号名字显示；对评论的管理以及对回复信息的查看，评论可以删除，评论相关的回复也会一并删除。

(6) 我的课程：在这里展示的是对应朵喵学院已添加的课程，用户可以点击这些课程进行学习相关的就业知识；实习生用户可以删除已添加的课程，总之实习生用户只具有添加，删除，查看权限，没有修改权限。

(7) 公共页面：本项目的首页属于公共页面，该页面展示常规的导航栏进行跳转选择，页面包含有职位分类展示，banner企业或职位企划活动轮播图展示，按照职位名称或公司名称搜索框功能；该页面还具有实习生登录和公司登录注册入口。除了首页，实习模块旗下所含的所有页面，校招下的所有页面，朵喵学院的页面，猫侍社区的所有页面。

### 3.2.2 HR功能模块

HR用户功能模块主要实现了以下几个功能模块，如图3-3所示：

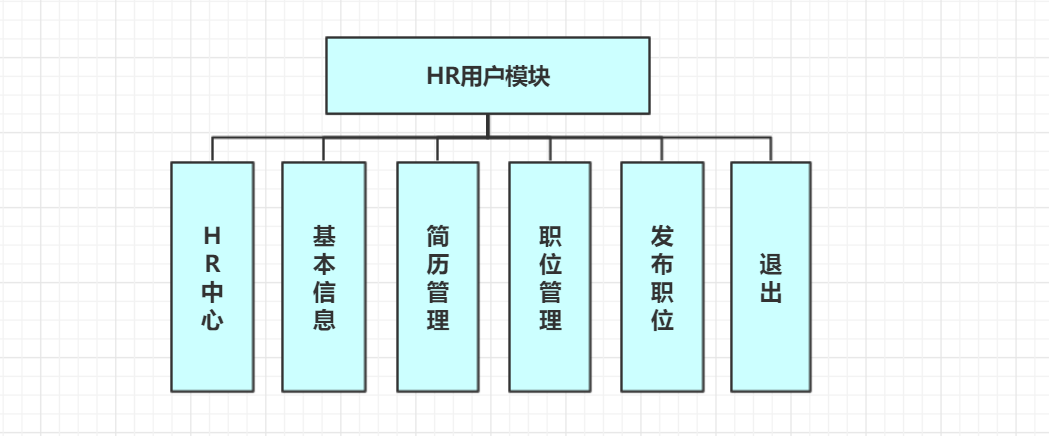


图3-3 HR用户模块功能划分图

(1) HR中心

同样HR中心就是HR用户功能模块的中枢神经，HR中心负责调度以下基本信息、发布职位、职位管理、简历管理、退出功能一共6个子模块。本项目的公共首页有HR中心的链接，可以直接跳转到HR用户的管理中心主界面。

(2) 基本信息

在这个子模块中可以浏览到HR用户的账户信息，包括账号密码、联系方式等等；其中HR用户可以对密码、头像、联系方式进行修改，而账户名，关联企业ID，注册时间都是一经初始化就不可以修改的。

(3) 发布职位

该子模块功能是HR用户管理中心的核心功能，HR用户经过注册企业通过审核有效后才可以发布职位；发布职位过程中可填写项有职位名称，所属类别（选择），最低薪水，最高薪水，最低学位要求（限制填写范围和格式），工作地点（必填），职位描述，每周工作天数，工作月数，截止日期，急招标志；此外的ID标识，所属公司，刷新时间，热门职位标志都是有系统完成填写；急招标志需要支付一定代价才有机会设置。

(4) 职位管理

HR用户可以在这里浏览已经发布的职位信息，会将重要的信息缩减展示；HR用户觉得需要更新已发布职位时，可以对选定的职位进行修改；当然HR用户也具有对职位的删除权限，添加权限则体现在发布职位功能模块中。

(5) 简历管理

此子功能模块用于显示实习生用户所投递过来的简历信息，HR用户可以查看这些简历，并通过多条件进行筛选浏览，快速选择最优实习生简历进行评估；HR用户也可以对不满意的简历进行删除操作，删除后也可以对简历在一定的时间内进行恢复;简历的添加体现在实习生用户的投递，投递限制为一个职位只能投一次。

(6) 退出

退出操作，会将服务器上HR用户对应的session进行删除操作，页面会返回项目的公共主页面；如果长时间不操作，系统默认执行session清空操作。

### 3.2.3 后台管理系统模块

后台管理系统功能模块主要实现了以下几个功能模块，如图3-4所示：

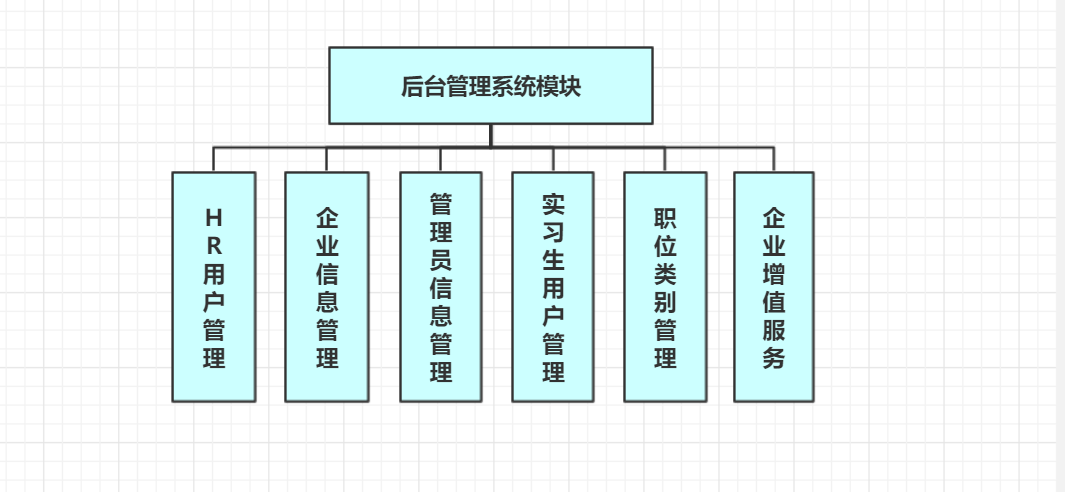


图3-4 后台管理系统模块功能划分图

(1) HR用户信息管理：在这里可以查看和删除HR用户。可以查看的内容有账户名称，密码非明文显示，头像路径，关联企业，联系方式，注册时间信息。删除功能一般用不到，只有存在特殊情况才会使用此功能。

(2) 企业信息管理：在此处可以查看，删除企业的信息。

(3) 实习生用户管理：本功能包含可以查看实习生用户的部分信息，按条件筛选查询，可以在特殊情况下动用删除功能。

(4) 管理员信息管理：本子模块提供对管理员用户的增加、删除、修改、多条件查询功能。

(5) 职位类别管理：可以对职位类别进行添加，修改，删除以及查看。

(6) 企业增值服务统计：录入报名加入首页企业logo展示的HR用户以及所对应的企业信息，然后通知前端人员对页面进行实时更新。

(7) 退出：删除session中所存储有关管理员账户的信息。

## 3.3界面设计

下面罗列本系统其中一些代表性的设计。

### 3.3.1实习生用户模块界面设计

(1) 实习生模块个人中心首页显示效果如图3-5所示。



图3-5主页

(2) 实习生用户注册页面如图3-6所示。

![C:\Users\哥斯达黎加\AppData\Roaming\Tencent\Users\3073415886\TIM\WinTemp\RichOle\LU1RZT~6%C]5(3]((8_SYEP.png](data:image/png;base64,)

图3-6实习生用户注册页面

(3) 实习生用户登录页面如图3-7所示。



图3-7实习生登录页面

(4)基本信息子功能模块管理页面，显示效果如图3-8所示。

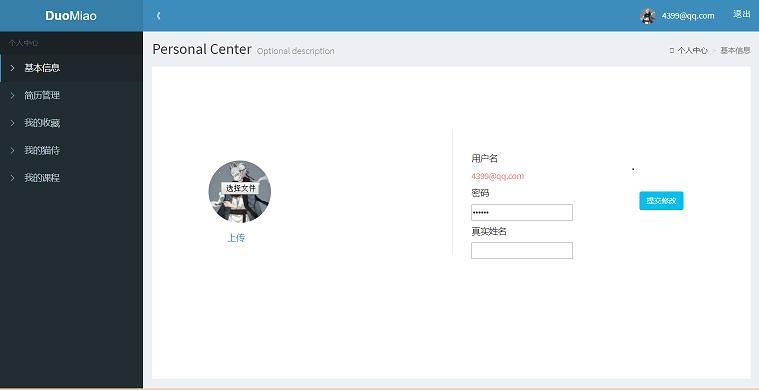


图3.8基本信息页面

(5)简历管理模块页面，显示效果如图3-9所示。

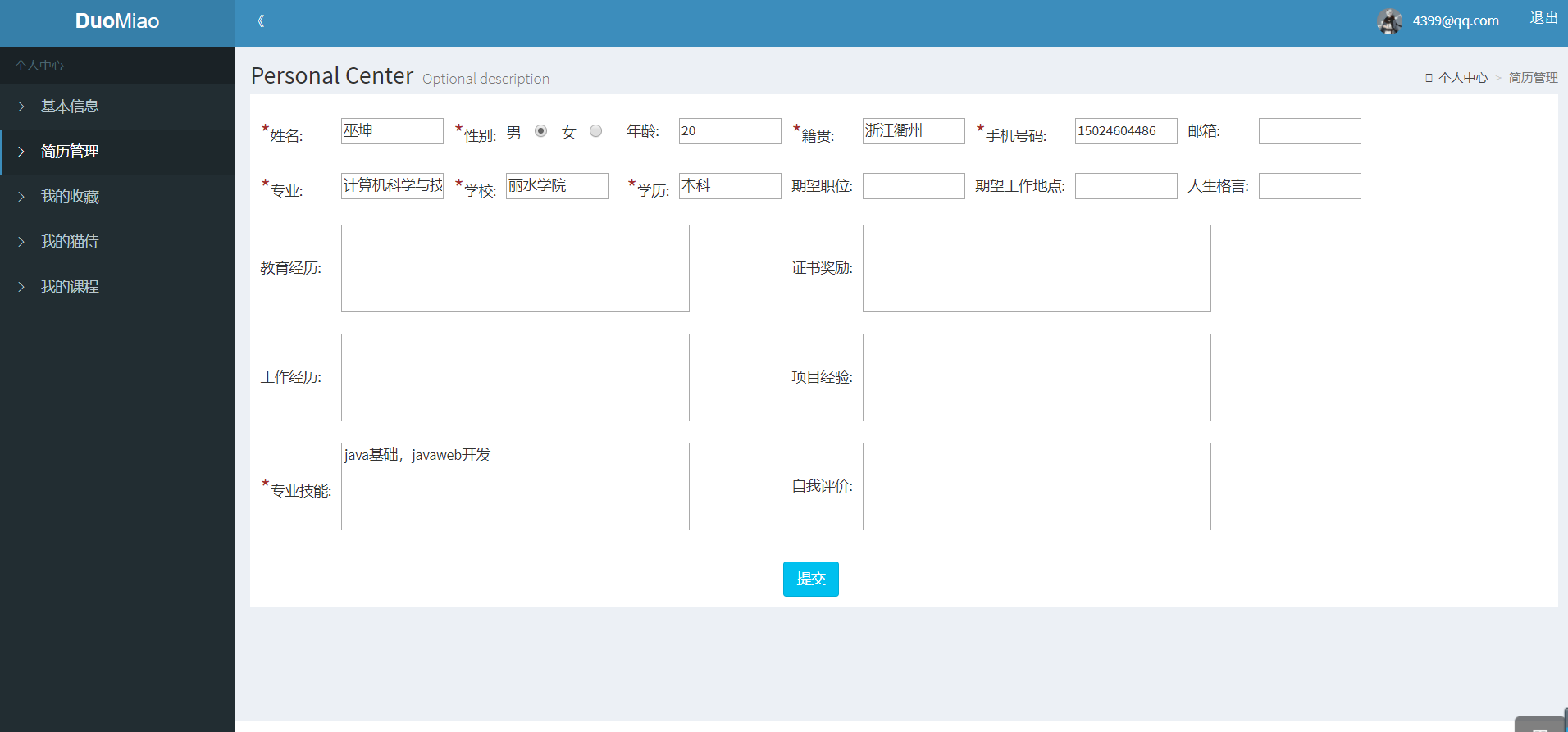


图3-9简历管理页面

(6)我的收藏，我的猫侍，我的课程等子模块页面

这三个页面的样式类似，在这里进行统一展示，显示效果如图3-10，3-11，3-12所示。

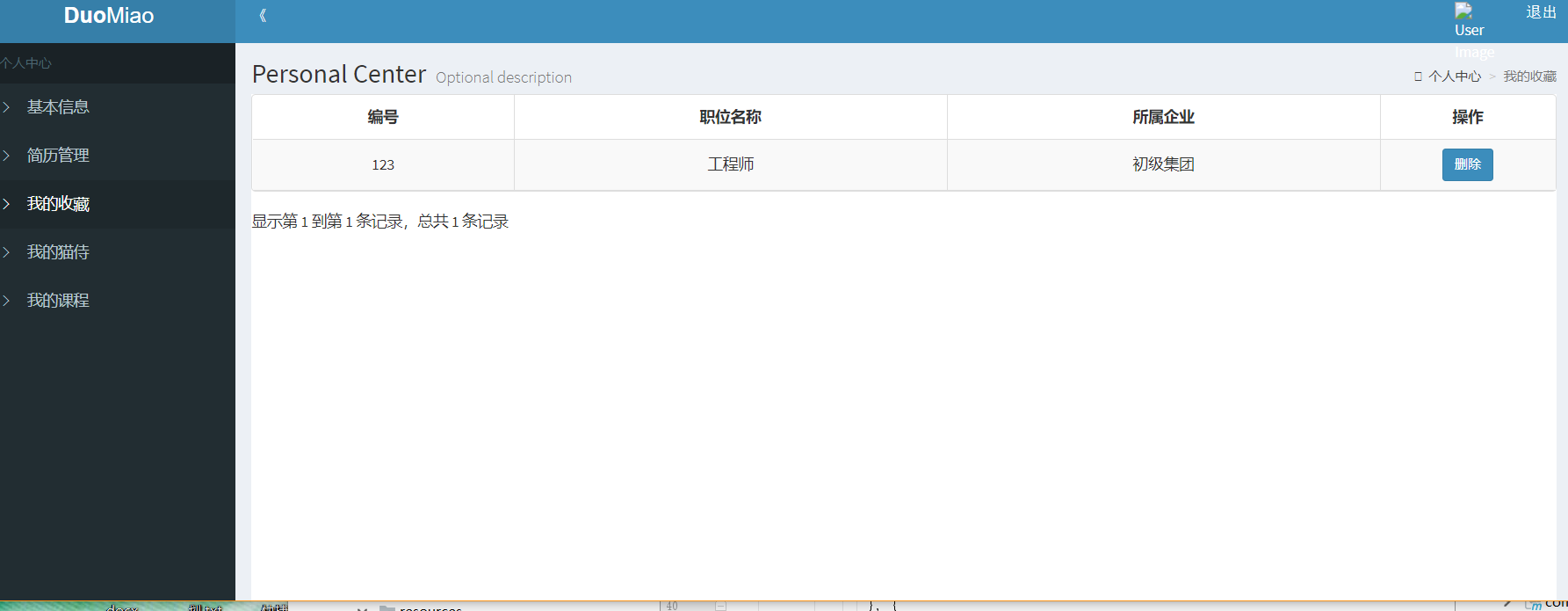


图3-10我的收藏页面

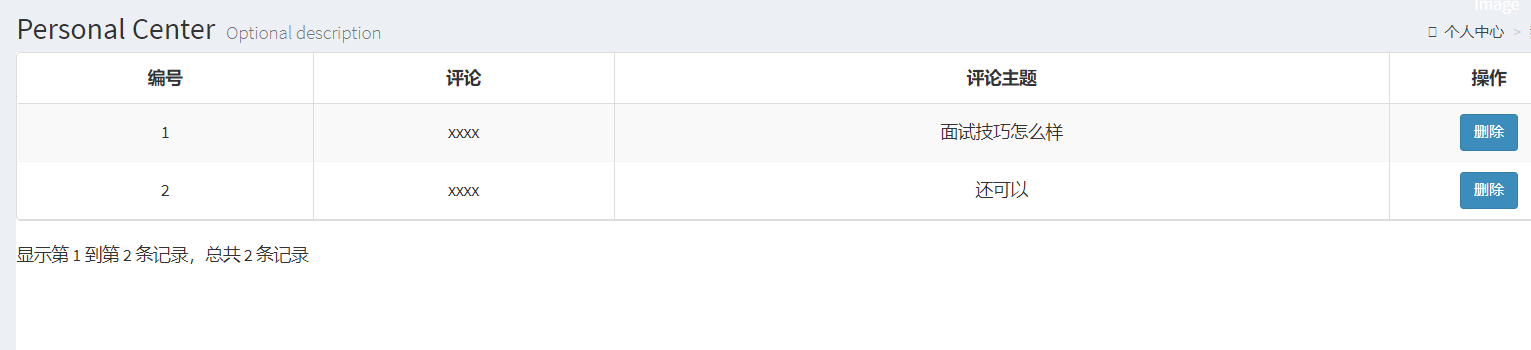


图3-11我的猫侍页面



图3-12我的课程页面

### 3.3.2 HR用户模块界面设计

(1)HR用户模块界面HR中心首页如图3-13所示。



图3-13 HR中心首页

(2) 基本信息界面，与实习生模块的基本信息界面样式一致，如图3-14所示。

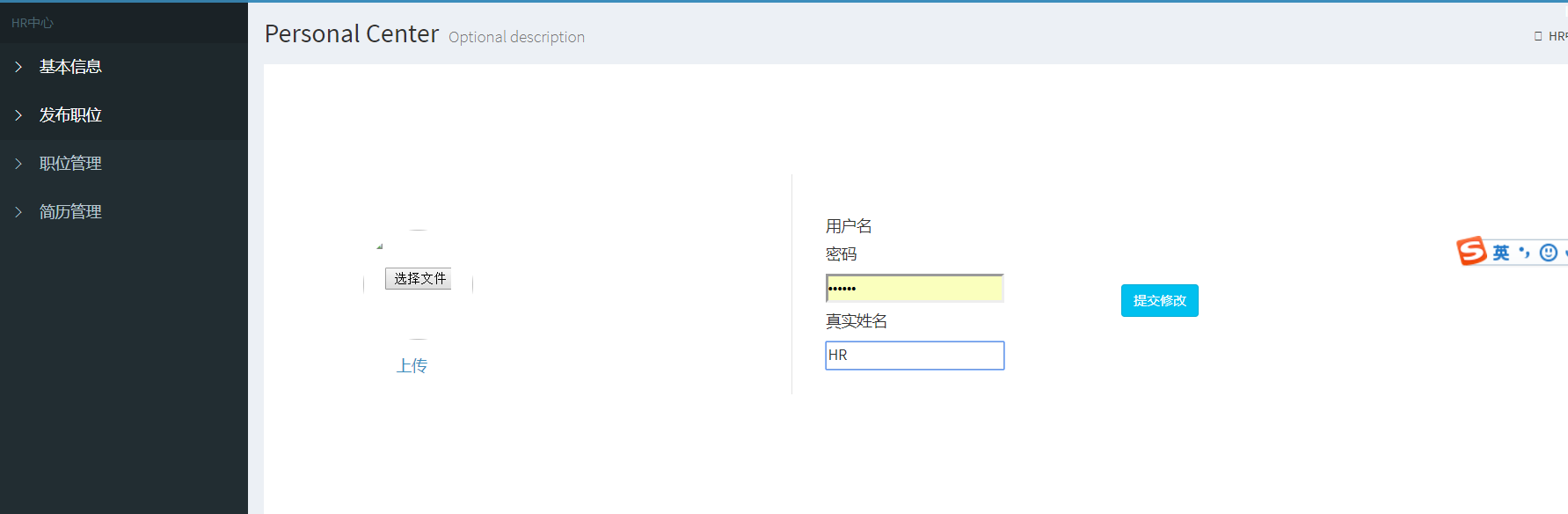


图3-14 HR的基本信息页面

(3) 发布职位的相关页面如图3-15所示。

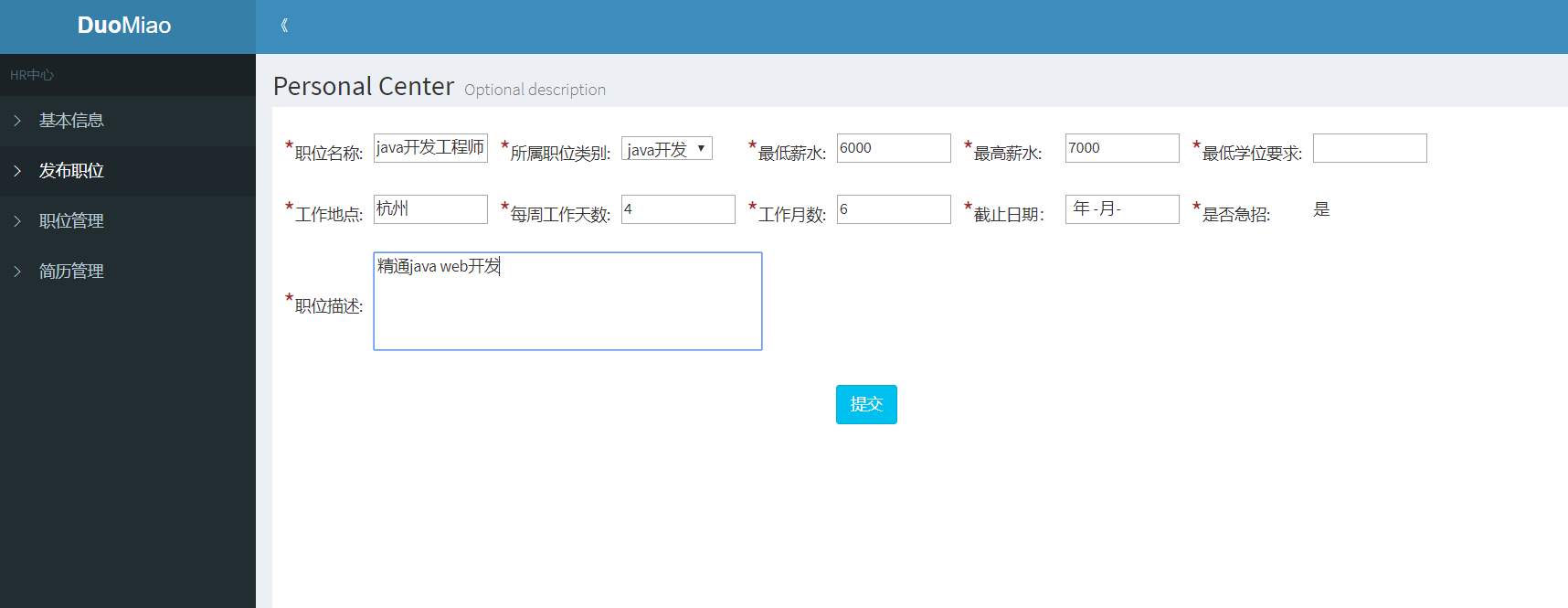


图3-15发布职位页面

(4) 职位管理页面如图3-16所示。



图3-16职位管理页面

(5) 简历管理页面与上面的职位管理页面样式类似，页面设计效果如图3-17所示。



图3-17简历管理页面

### 3.3.3后台管理系统界面设计

本平台的后台管理系统首页设计效果如图3-18所示。

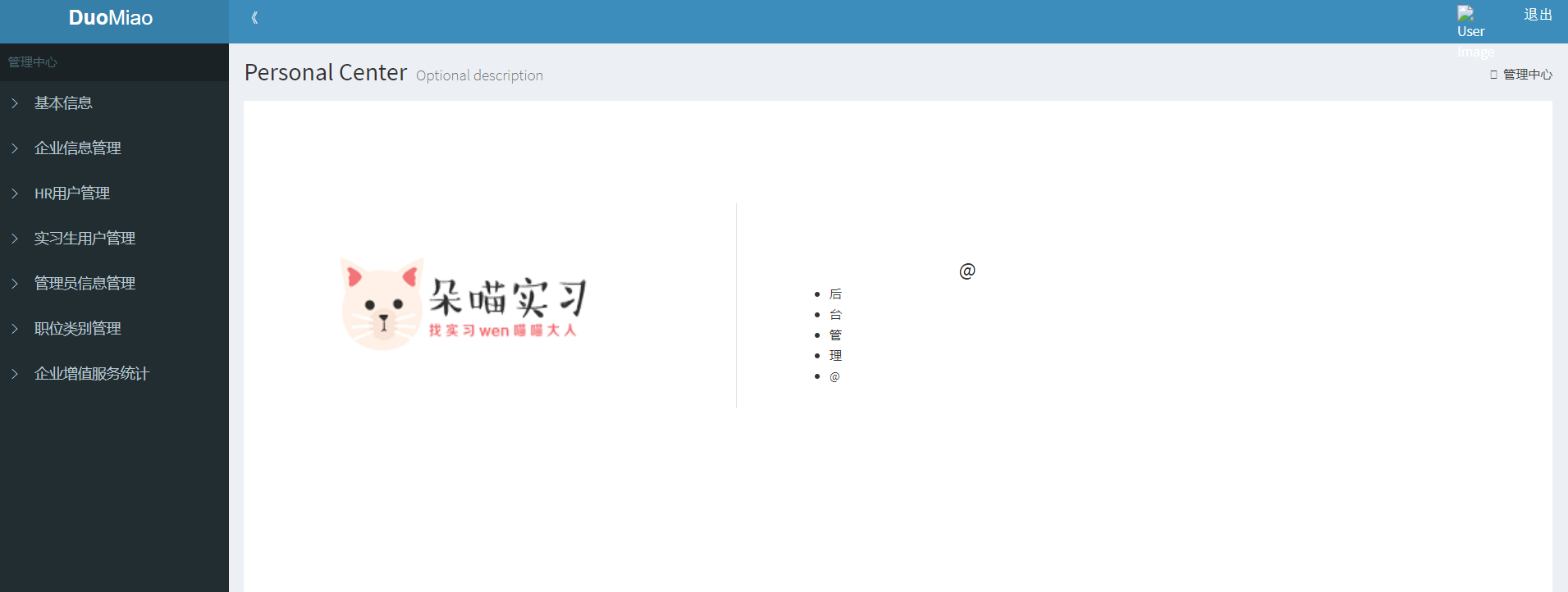


图3-18后台管理系统首页页面

### 3.3.4公共页面界面设计

这里只展示部分公共页面效果图，本平台的公共主页如下图3-19，图3-20所示。

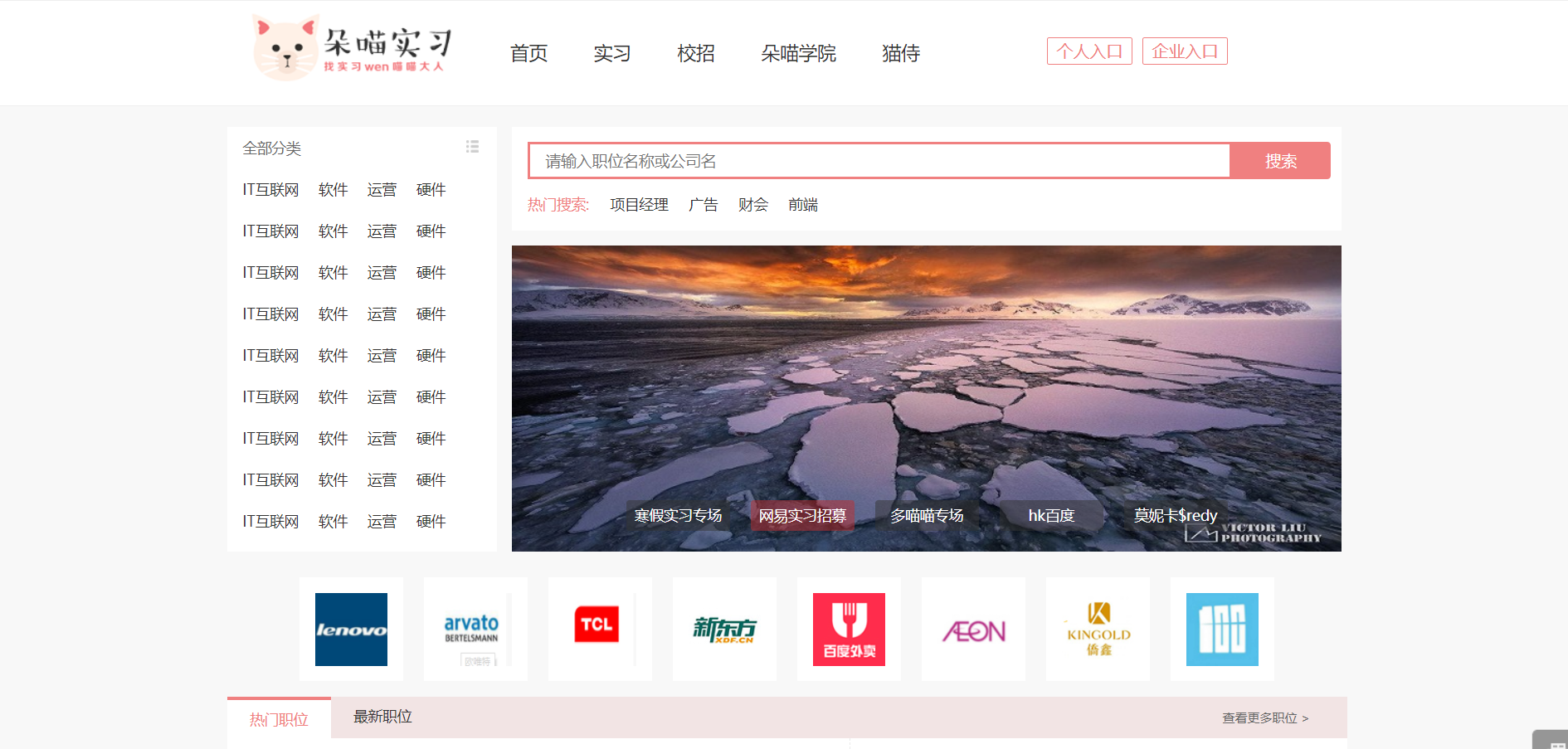


图3-19项目公共首页页面

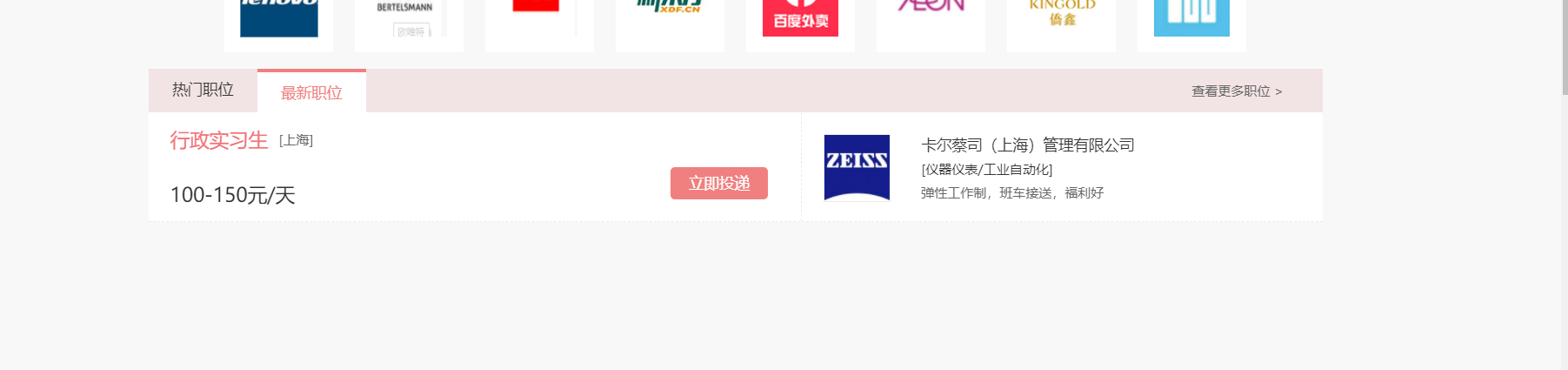


图3-20项目公共首页页面

# 第四章 详细设计与实现

## 4.1系统文件结构的设计

### 4.1.1 java类的组织

本项目的部分关键java类信息，java接口信息如下表4-1所示。

表4-1 java类或接口信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 类名 | 说明 | 类名 | 说 明 |
| IndexControl | 公共页面访问类 | InternInfo | 实习生实体类 |
| InternInfoControl | 实习生相关访问控制类 | ResumeInfo | 简历实体类 |
| RedisControl | Redis测试控制类 | EntCollect | 收藏企业实体类 |
| ResumeInfoControl | 简历访问控制类 | EnterpriseInfo | 企业信息实体类 |
| EntCollectControl | 企业收藏访问类 | EntResume | 企业与简历关系实体类 |
| EnterpriseInfoControl | 企业信息访问类 | HrInfo | HR实体类 |
| EntResumeControl | 企业简历关系访问类 | JobCollect | 收藏职位实体类 |
| HrInfoControl | HR用户访问类 | JobReply | 职位评论实体类 |
| JobCollectControl | 职位收藏访问类 | PositionCategory | 职位类别实体类 |
| JobReplyControl | 评论访问控制类 | PublishJob | 职位实体类 |
| PositionCategoryControl | 职位类别访问控制类 |  |  |
| PublishJobControl | 职位访问控制类 | RedisCacheUtil | 第三方缓存服务类 |
|  |  |  |  |
| InternInfoService | 实习生服务接口 | HrInfoService | HR用户服务接口 |
| InternInfoServiceImpl | 实习生服务接口实现类 | HrInfoServiceImpl | HR用户服务接口实现类 |
| PublishJobService | 职位服务接口 | EntResumeService | 企业简历关系服务接口 |
| PublishJobServiceImpl | 职位服务接口实现类 | EntResumeServiceImpl | 企业简历关系服务接口实现类 |
| JobReplyService | 职位评论服务接口 | EnterpriseInfoService | 企业信息服务接口 |
| JobReplyServiceImpl | 职位评论服务接口实现类 | EnterpriseInfoServiceImpl | 企业信息服务接口实现类 |
| JobCollectService | 收藏职位服务接口 | EntCollectService | 收藏企业服务接口 |
| JobCollectServiceImpl | 收藏职位服务接口实现类 | EntCollectServiceImpl | 收藏企业服务接口实现类 |
| PositionCategoryService | 职位类别服务接口 | ResumeInfoService | 简历服务接口 |
| PositionCategoryServiceImpl | 职位类别服务接口实现类 | ResumeInfoServiceImpl | 简历服务接口实现类 |
|  |  |  |  |
| InternInfoMapper | 实习生Dao接口 | ResumeInfoMapper | 简历Dao接口 |
| PublishJobMapper | 职位Dao接口 | HrInfoMapper | HR用户Dao接口 |
| JobReplyMapper | 职位评论Dao接口 | EntResumeMapper | 企业简历关系Dao接口 |
| JobCollectMapper | 收藏职位Dao接口 | EnterpriseInfoMapper | 企业信息Dao接口 |
| PositionCategoryMapper | 职位类别Dao接口 | EntCollectMapper | 收藏企业Dao接口 |
|  |  |  |  |
| AjaxResult | Ajax返回实体类 | Consant | 常量类 |
| DateHelper | 时间格式处理工具类 | RedisCacheTransfer | Redis静态注入属性类 |
| UUIDBuilder | UUID生成调用类 |  |  |
|  |  |  |  |

## 4.2系统架构的设计

本项目是B/S项目，底层框架采用spring+spring mvc+mybatis+redis整合架构，底层分层包括表示层、业务逻辑层和数据访问层、数据层。数据层即数据库，本项目中使用的是MySql5.0数据库。前端页面使用的插件或框架有jquery插件、bootstrap框架、require.js插件、layer.js插件。

### 4.2.1表示层的设计

本项目的表示层采用spring mvc框架，它MVC设计模式项目的web层框架。它是以DispatcherServlet为核心设计的，其工作流程大致分为4步:

1. 前端控制器DispatcherServlet接收到来自客户端的请求后，将其交给处理器映射器HandlerMapping;
2. 处理器映射器会根据请求路径去找到相对应的处理器适配器，然后选则对应的Control控制器执行对应的方法；
3. 处理完请求后，返回一个ModelandView视图模型对象；
4. 视图解析器对其进行处理，最后将数据渲染到视图上。

本项目中的所有页面的访问等都是由spring mvc的dispacher servlet来控制。

(1) 视图部分

JSP页面

本系统所有页面采用jsp形式的页面类型，对于不算太庞大的项目使用服务器页面jsp可以使动态数据展示更加容易。Jsp支持多种标签语言，这里用到c标签语言处理判断，循环数据展示问题，数据展示部分也用到el表达式；Jsp的require,session等内置对象可以存储用户相关的信息。当然大部分数据都是采用ajax方式请求后台返回json数据，然后解析json格式数据展示到界面上。

(2) 控制部分

①Control类

每一个实体对象大多数情况下都对应一个control类，这些类都存放在com.duomiao.control包下面。

每个control类都加上@Controller注解，以用来告诉spring容器这是一个控制器类，当然还可以加上一个@RequestMapping全局访问路径注解，然后每个方法也有相对应的@RequestMapping注解，浏览器可以输入对应的url来访问。

方法如果要返回数据，统一用AjaxResult类来包装，加上@ResponseBody然后序列化成json对象返回客户端。

下面展示InternInfoControl类代码:

import java.io.File;

import java.io.IOException;

import java.util.Date;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.multipart.MultipartFile;

import com.duomiao.entity.InternInfo;

import com.duomiao.service.InternInfoService;

import com.duomiao.util.AjaxResult;

import com.duomiao.util.Constant;

import com.duomiao.util.DateHelper;

import com.duomiao.util.UUIDBuilder;

@Controller

@RequestMapping("/intern")

public class InternInfoControl {

private AjaxResult ajaxResult = new AjaxResult();;

@Autowired

private InternInfoService internInfoService;

@RequestMapping("/enLoginAndRegister")

public String enLoginAndRegisterPage() {

return "interncenter/login\_register";

}

@RequestMapping("/doLogin")

@ResponseBody

public AjaxResult doLogin(@RequestParam Map<String,Object> map,HttpSession session) {

List<InternInfo> internInfos = internInfoService.getInternInfoList(map);

if(internInfos != null && internInfos.size() > 0) {

System.err.println("ok");

ajaxResult.setSuccess(true);

session.setAttribute(Constant.SESSION\_INTERN\_INFO, internInfos.get(0));

session.removeValue(Constant.SESSION\_HR\_INFO);

}else {

ajaxResult.setSuccess(false);

}

return ajaxResult;

}

@RequestMapping("/doLoginOut")

public String doLoginOut(HttpSession session) {

session.setAttribute(Constant.SESSION\_INTERN\_INFO, null);

return "redirect:/public/index";

}

@RequestMapping("/doCheckSameName")

@ResponseBody

public AjaxResult doCheckSameName(HttpServletRequest request) {

String name = request.getParameter("name");

ajaxResult.setSuccess(false);

int count = internInfoService.validateSameName(name);

if(count > 0) {

ajaxResult.setSuccess(true);//名字已存在

}

return ajaxResult;

}

@RequestMapping("/doRegister")

@ResponseBody

public AjaxResult doRegister(@ModelAttribute InternInfo internInfo) {

internInfo.setRegisterTime(DateHelper.getFormatDate("yyyy-MM-dd HH:mm:ss", new Date()));

internInfo.setId(UUIDBuilder.createUUID());

internInfo.setDb("0");

internInfo.setFlag("I");//唯一标识

internInfo.setState("0");//默认未激活

internInfo.setImgUrl("/imgs/index\_header\_bottom/default.png");//图像默认

int result = internInfoService.registerInternInfo(internInfo);

if(result>0) {

ajaxResult.setSuccess(true);

}else {

ajaxResult.setSuccess(false);

}

return ajaxResult;

}

//基本信息修改,密码，真实姓名，图像

@RequestMapping("/adminlte/doEditMainInfo")

@ResponseBody

public AjaxResult alertMainInfoByOwn(@ModelAttribute InternInfo internInfo,HttpSession session){

if(internInfo.getImgUrl().equals("")) {

internInfo.setImgUrl("/imgs/index\_header\_bottom/default.png");

}

internInfoService.updateInternInfoForOwn(internInfo);

//更新session

Map<String,Object> map = new HashMap<String,Object>();

map.put("id", internInfo.getId());

List<InternInfo> internInfos = internInfoService.getInternInfoList(map);

session.setAttribute(Constant.SESSION\_INTERN\_INFO, internInfos.get(0));

return ajaxResult;

}

//访问个人中心

@RequestMapping("/adminlte/enIndex")

public String enAdminLteIndex() {

return "interncenter/adminlte/index";

}

//访问基本信息页面

@RequestMapping("/adminlte/enBasicInfo")

public String enBasicInfo() {

return "interncenter/adminlte/basicinfo";

}

//上传图片

@RequestMapping("/adminlte/doUpImg")

@ResponseBody

public AjaxResult upImg(@RequestParam(value = "file", required = false)MultipartFile file,HttpServletRequest request) throws IllegalStateException, IOException {

//物理路径

String path = request.getSession().getServletContext().getRealPath("/imgs/imgs\_server");

Map<String,Object> map = new HashMap<String,Object>();

String fileName = file.getOriginalFilename();

File dir = new File(path,fileName);

if(!dir.exists()) {

dir.mkdirs();

}

file.transferTo(dir);

String enPath = "/imgs/imgs\_server/"+ file.getOriginalFilename();

map.put("imgPath", enPath);

ajaxResult.setMessage(map);

ajaxResult.setSuccess(true);

return ajaxResult;

}

}

②配置文件

Spirng\_mvc.xml作为spring mvc框架的配置文件，具体内容如下:

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:mvc="http://www.springframework.org/schema/mvc"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd

http://www.springframework.org/schema/mvc

http://www.springframework.org/schema/mvc/spring-mvc.xsd

">

<mvc:annotation-driven>

<!-- spring解析返回结果成json -->

<mvc:message-converters register-defaults="true">

<bean class="com.alibaba.fastjson.support.spring.FastJsonHttpMessageConverter">

<property name="supportedMediaTypes">

<list>

<value>text/html;charset=UTF-8</value>

<value>application/json</value>

<value>application/xml;charset=UTF-8</value>

</list>

</property>

<property name="features">

<list>

<value>PrettyFormat</value>

<value>WriteMapNullValue</value>

<value>WriteNullStringAsEmpty</value>

<value>WriteNullListAsEmpty</value>

<value>DisableCircularReferenceDetect</value><!-- 解决fastjson自动引用 $ref -->

</list>

</property>

</bean>

</mvc:message-converters>

</mvc:annotation-driven>

<!-- 文件上传，定义文件解释器 -->

<bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver">

<property name="defaultEncoding" value="utf-8"/>

<!-- 上传文件大于10kb时会生成临时文件 -->

<property name="maxInMemorySize" value="10240"/>

<!-- 最大文件大小，单位字节，-1为不限制大小 默认限制5M-->

<property name="maxUploadSize" value="5242440"/>

</bean>

<!-- 扫描control包 -->

<context:component-scan base-package="com.duomiao.control">

</context:component-scan>

<mvc:resources location="/dmcss/" mapping="/dmcss/\*\*"/>

<mvc:resources location="/dmjs/" mapping="/dmjs/\*\*"/>

<mvc:resources location="/imgs/" mapping="/imgs/\*\*"/>

<mvc:resources location="/NewFile.html" mapping="/NewFile.html"/>

<!-- 配置mvc的拦截器 可以配置多个 -->

<!-- <mvc:interceptors>

<mvc:interceptor>

需要被拦截的路径

<mvc:mapping path="/admin/\*\*" />

<mvc:exclude-mapping path="/admin/user/login.do"/>

<mvc:exclude-mapping path="/admin/user/register/\*"/>

拦截处理的interceptor

<bean class="com.permission.interceptor.PermissionInterceptor" />

</mvc:interceptor>

</mvc:interceptors> -->

<!--两大组件被取代 -->

<!-- <bean class="org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping"></bean>

<bean class="org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter"></bean> -->

<bean id="jspViewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<property name="viewClass" value="org.springframework.web.servlet.view.JstlView"/>

<property name="prefix" value="/WEB-INF/jsp/"/>

<property name="suffix" value=".jsp"/>

</bean>

<!-- <bean id="admin/dept/toadd.do" class="com.permission.control.DeptControl"></bean> -->

</beans>

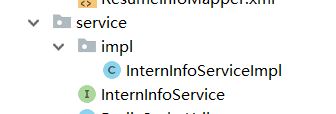
### 4.2.2业务逻辑层的设计

(1) 业务逻辑层的分析

只负责处理业务，没有与数据直接打交道。充分体现了“高内聚，低耦合”的特点，业务逻辑层负责逻辑性数据的处理，与表现层以及数据联络层相联系交互。对于数据的具体展示样式，则不在业务逻辑层考虑范围之内。

(2)业务逻辑层的实现

业务层的结构如下图4-1所示：

  
图4-1实现接口的类

每个业务都会建一个对应的服务接口，然后新建一个服务类去实现这个服务接口，提高了系统的可扩展性，可维护性。业务层的每个实现类都会加上@Service注解，将其交给spring容器管理，在初始化的时候就创建好这个bean类并且注入依赖。业务层的每个业务类对应的方法作为事务管理的切入点。这里展示InternInfoService业务接口代码:

import java.util.List;

import java.util.Map;

import com.duomiao.entity.InternInfo;

public interface InternInfoService {

//查询列表

public List<InternInfo> getInternInfoList(Map<String,Object> qryMap);

//批量删除

public int batchDelInternInfos(List<Integer> ids);

//修改,权限仅对用户自己开放

public int updateInternInfoForOwn(InternInfo internInfo);

//特殊更新,系统修改

public int specialUpdateInternInfo(Map<String,Object> qryMap);

//定时删除 无效账户 减轻数据库负担

public int specialDelInternInfoByState();

//注册

public int registerInternInfo(InternInfo internInfo);

//重名校验

public int validateSameName(String loginName);

}

其对应的InternInfoServiceImpl实现类:

import java.util.List;

import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.duomiao.entity.InternInfo;

import com.duomiao.mapper.InternInfoMapper;

import com.duomiao.service.InternInfoService;

@Service

public class InternInfoServiceImpl implements InternInfoService{

@Autowired

private InternInfoMapper internInfoMapper;

//查询

@Override

public List<InternInfo> getInternInfoList(Map<String, Object> qryMap) {

return internInfoMapper.selectInternInfoList(qryMap);

}

//批量删除

@Override

public int batchDelInternInfos(List<Integer> ids) {

return internInfoMapper.batchDelInternInfos(ids);

}

//自我更新

@Override

public int updateInternInfoForOwn(InternInfo internInfo) {

return internInfoMapper.updateInternInfoForOwn(internInfo);

}

//系统自动更新

@Override

public int specialUpdateInternInfo(Map<String, Object> qryMap) {

return internInfoMapper.specialUpdateInternInfo(qryMap);

}

//定时删除无效账户，减轻数据库负载

@Override

public int specialDelInternInfoByState() {

return internInfoMapper.specialDelInternInfoByState();

}

//注册

@Override

public int registerInternInfo(InternInfo internInfo) {

return internInfoMapper.register(internInfo);

}

//重名校验

@Override

public int validateSameName(String loginName) {

return internInfoMapper.sameName(loginName);

}

}

(3) 业务逻辑层的管理：在spring的配置文件中统一扫描后交由spring容器管理，每个类对应的bean对象采用单例模式，特殊情况下用prototype，采用原型模式的化，每次调用getBean()方法都会产生一个对象，这个对象是不受spring控制的。

### 4.2.3数据库持久层的设计

本项目使用Mybatis框架作为持久层，mybatis支持定制化Sql、高级映射以及存储过程的持久层架构。其优点有使用灵活、提供映射标签，半自动orm框架，支持编写动态sql，可以借助第三方数据库作为二级缓存，缓解关系型数据库的压力。

(1) Mybatis配置

本项目采用spring+mybatis整合方式，提供spring-mybatis.xml对关键部分进行配置。首先将dbconfig.properties数据库属性配置文件注入进来，然后配置datasource数据源，数据源采用阿里的druid开源数据源，接着配置sqlsessionFactory工厂bean,并注入datasource数据源；之后配置mapperscanner即数据联络层的自动扫描组件并且注入sqlsessionfactory单例对象供dao层实现类方法使用，component-scan扫描service层组件。相关配置代码:

!-- 添加properties文件 -->

<bean

class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">

<property name="locations">

<list>

<value>classpath:spring/dbconfig.properties</value>

</list>

</property>

<property name="ignoreUnresolvablePlaceholders" value="true" />

</bean>

<!-- <bean id="dataSource" class="org.apache.ibatis.datasource.pooled.PooledDataSource">

<property name="driver" value="com.mysql.jdbc.Driver"/> <property name="url"

value="jdbc:mysql://localhost:3306/testdb"/> <property name="username" value="root"/>

<property name="password" value="tkmr93820"/> </bean> -->

<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource"

destroy-method="close">

<!-- 数据库基本信息配置 -->

<property name="url" value="${url}" />

<property name="username" value="${username}" />

<property name="password" value="${password}" />

<property name="driverClassName" value="${driverClassName}" />

<property name="filters" value="${filters}" />

<!-- 最大并发连接数 -->

<property name="maxActive" value="${maxActive}" />

<!-- 初始化连接数量 -->

<property name="initialSize" value="${initialSize}" />

<!-- 配置获取连接等待超时的时间 -->

<property name="maxWait" value="${maxWait}" />

<!-- 最小空闲连接数 -->

<property name="minIdle" value="${minIdle}" />

<!-- 配置间隔多久才进行一次检测，检测需要关闭的空闲连接，单位是毫秒 -->

<property name="timeBetweenEvictionRunsMillis" value="${timeBetweenEvictionRunsMillis}" />

<!-- 配置一个连接在池中最小生存的时间，单位是毫秒 -->

<property name="minEvictableIdleTimeMillis" value="${minEvictableIdleTimeMillis}" />

<property name="validationQuery" value="${validationQuery}" />

<property name="testWhileIdle" value="${testWhileIdle}" />

<property name="testOnBorrow" value="${testOnBorrow}" />

<property name="testOnReturn" value="${testOnReturn}" />

<property name="maxOpenPreparedStatements" value="${maxOpenPreparedStatements}" />

<!-- 打开 removeAbandoned 功能 -->

<property name="removeAbandoned" value="${removeAbandoned}" />

<!-- 1800 秒，也就是 30 分钟 -->

<property name="removeAbandonedTimeout" value="${removeAbandonedTimeout}" />

<!-- 关闭 abanded 连接时输出错误日志 -->

<property name="logAbandoned" value="${logAbandoned}" />

</bean>

<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">

<property name="dataSource" ref="dataSource"></property>

<property name="mapperLocations" value="classpath:com/duomiao/mapper/\*Mapper.xml"></property>

<!--开启缓存支持，mybatis和redis整合 -->

<property name="configurationProperties">

<props>

<prop key="cacheEnabled">true</prop>

<!-- 查询时，关闭关联对象即时加载以提高性能 -->

<prop key="lazyLoadingEnabled">false</prop>

<!-- 设置关联对象加载的形态，此处为按需加载字段(加载字段由SQL指定)，不会加载关联表的所有字段，以提高性能 -->

<prop key="aggressiveLazyLoading">true</prop>

<!-- 对于未知的SQL查询，允许返回不同的结果集以达到通用的效果 -->

<prop key="multipleResultSetsEnabled">true</prop>

<!-- 允许使用列标签代替列名 -->

<prop key="useColumnLabel">true</prop>

<!-- 允许使用自定义的主键值(比如由程序生成的UUID 32位编码作为键值)，数据表的PK生成策略将被覆盖 -->

<prop key="useGeneratedKeys">true</prop>

<!-- 给予被嵌套的resultMap以字段-属性的映射支持 -->

<prop key="autoMappingBehavior">FULL</prop>

<!-- 对于批量更新操作缓存SQL以提高性能 -->

<prop key="defaultExecutorType">BATCH</prop>

<!-- 数据库超过25000秒仍未响应则超时 -->

<prop key="defaultStatementTimeout">25000</prop>

</props>

</property>

<property name="typeAliasesPackage" value="com.duomiao.entity"></property>

</bean>

<!-- 自动扫描 -->

<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">

<property name="basePackage" value="com.duomiao.mapper"></property>

<property name="sqlSessionFactoryBeanName" value="sqlSessionFactory"></property>

</bean>

<!-- <bean id="redisCacheUtil" class="com.duomiao.service.RedisCacheUtil">

<constructor-arg value="id"/>

</bean> -->

<!-- 扫描control包 -->

<context:component-scan base-package="com.duomiao.service">

</context:component-scan>

<!-- 业务层事务管理解决方案 回滚-->

<bean id="transactionManager"

class="org.springframework.jdbc.datasource.DataSourceTransactionManager">

<property name="dataSource" ref="dataSource"></property>

</bean>

<!-- 如何用 -->

<tx:advice id="myadvice" transaction-manager="transactionManager">

<tx:attributes>

<tx:method name="add\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="del\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="update\*" propagation="REQUIRED"

rollback-for="java.lang.Exception" />

<tx:method name="insert\*" propagation="REQUIRED"

rollback-for="java.lang.Exception" />

<tx:method name="edit\*" propagation="REQUIRED"

rollback-for="java.lang.Exception" />

</tx:attributes>

</tx:advice>

(2)映射配置

数据库表与对象通过xml映射文件进行映射，对mapper方法的访问就可以访问数据库。

(3) DAO层的设计

Dao层包含一个接口和一个xml文件，spring容器会自动给这个dao接口创建实现类，所以不需要自己写实现类。Dao接口的方法名和xml文件中每个Sql的id对应，通过执行方法发送Sql语句到数据库，最后执行。

# 第五章 测试与分析

## 5.1 测试环境

这次测试在本机的虚拟机上进行。

硬件环境:

处理器I7-6700HQ

内存:4GB

硬盘:1TB

软件环境:

Windows10环境下Vmware安装的linux虚拟机

Linux CentOs

Tomcat7.0

Mysql

## 5.2 实例测试

首先运行项目:

(1)项目打成war包部署到linux服务器下tomcat的wepapp文件夹，解压。

(2)将数据库文件导入到linux服务器下的mysql数据库中。

(3)启动Tomcat服务器

(4)浏览器输入http:// 192.168.190.129:8080/duomiao/public/index

开始测试：

测试对象是针对实习生用户模块、HR用户模块、后天管理系统三大系统。

部分测试用例如下表5-1所示

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试项目 | 测试方法 | 测试结果 |
| 1 | 实习生注册 | 进入注册界面，填写不同的格式账号和密码，点击注册 | 不符条件的无法注册，符合条件的注册成功 |
| 2 | 实习生基本信息操作 | 修改用户名 | 只读无法修改 |
| 上传图像 | 当上传非图像文件时无效 |
| 3 | 实习生简历填写 | 名字不输，学历不输 | 提醒用户不能为空，并无法提交 |
| 按照要求范填写 | 提交后，提醒成功 |
| 4 | 删除 | 删除企业信息 | 提醒删除成功，数据库数据更新 |
| 5 | 查询HR用户 | 多条件查询 | 展示筛选出的数据 |
| 6 | 数据导入到录入窗体 | 导入每项数据 | 显示并计算出当前工作量 |
| 7 | 发布职位 | 乱填写 | 提示输入不符条件，无法提交 |
| 按照规定填写 | 提交成功，数据库添加数据 |
| 8 | 简历管理 | 删除职位的情况下 | 对应简历也被删除 |
| 职位保留的情况下 | 简历正常展示 |
| 9 | 职位管理 | 查询，删除职位 | 能删除数据 |

# 第六章 结论

通过这次毕业设计，深刻体会到采用spring mvc + spring + mybatis 的模式开发的优势，更低的成本，更快的效率，更清晰的项目架构这些都是在这个开发过程中切身的体会。以往我并未使用过该类型的MVC框架从头到尾的开发一个项目，最初只是采用原始的servlet+jsp模式开发一个简单的后台管理系统。由于之后自学这些框架后，发现其的使用广泛性和高效性后，决定将SSM框架应用到这次毕设中。

在做毕设前，一直在公司实习，感受到学校和公司对学习方式的极大不同。在公司通常都是通过不断在工作中去实现复杂需求，去攻克技术难点来提升自己，而这种方式恰恰是最锻炼人的，边实战边学习通常比学校的理论学习更加的快速且有意义，但是学校学的理论基础对日后的进一步提升也是非常重要的。这段是间，接触了struts2+spring+ibatis，spring+spring mvc+mybatis等模式的开发方式，在应用方面更加的熟悉了。也学习到团队开发的经验，知道更多的联合开发规范。

之所以会想到实习生招聘网站这个毕业设计主题，是自己本身就处于这个实习找工作阶段，自己也切身体会到工作不易。由于时间所限，本项目还有诸多瑕疵，因为项目本身比庞大所以需求方面也未考虑的十分细致，包括技术方面遇到的问题。不过经过这此项目的锤炼，遇到问题解决问题，学到了很多实际性的技术。

未来这类型的网站开发趋势都会朝彻底前后端分离方向发展，前端和后台互不干涉开发，后台只需要提供规范的接口即可，这样分工开来对项目的快速开发和部署尤为重要。当访问量大到一定程度时，通常需要解决高并发问题，这时候就会用到分布式架构，数据库更多优化。

。

# 参考文献

[1] 陈恒,楼偶俊,巩庆志,林徐. Spring MVC开发技术指南[M].北京:清华大学出版社, 2017：17-30.

[2] Josiah L. Carlson. Redis in Action [M].北京: 人民邮电出版社, 2015：20-30.

[3] 徐郡明. MyBatis技术内幕[M].北京：电子工业出版社, 2017:40-50.

[4]李杉,贾彦平,达虎.Mybatis逆向工程在JavaEE中的应用[J].通讯世界,2017(24):342.

[5]王鹤琴,汪炜玮,朱珍元.基于SSM框架技术的办公管理系统的研究[J].安徽警官职业学院学报,2017,16(03):118-122.

[6]吴东明,王丽娟.基于MySQL的中文全文搜索研究[J].电脑知识与技术,2017,13(32):7-8.

[7]宋成明.基于Shiro的某高校科研信息管理系统的设计与实现[J].智能计算机与应用,2017,7(04):62-63+66.

[8]张光平.Java多线程的分析[J].科技风,2018(01):71.

[9]荣艳冬.关于Mybatis持久层框架的应用研究[J].信息安全与技术,2015,6(12):86-88.

[10]Salvador Martínez,Valerio Cosentino,Jordi Cabot. Model-based analysis of Java EE web security misconfigurations[J]. Computer Languages, Systems & Structures,2017,49.

[11]Guido Barbaglia,Simone Murzilli,Stefano Cudini. Definition of REST web services with JSON schema[J]. Software: Practice and Experience,2017,47(6).

[12]Wang Bing,Dobosh Paul A,Chalk Stuart,Sopek Mirek,Ostlund Neil S. Computational Chemistry Data Management Platform Based on the Semantic Web.[J]. The journal of physical chemistry. A,2016.

[13]Aibek Sarimbekov,Lukas Stadler,Lubomír Bulej,Andreas Sewe,Andrej Podzimek,Yudi Zheng,Walter Binder. Workload characterization of JVM languages[J]. Software: Practice and Experience,2016,46(8).

[14]罗贵木. 基于微服务化的Web后台系统架构优化及实现[D].北京邮电大学,2017.

[15]北京市工商局网监处. 招聘网站履行审查义务相关问题探析[N]. 中国工商报,2017-09-05(003).

[16]Xi-mei ZHANG,Ying LI,Yi-Zhen Cao,Yan-fang LI. Research on System LoginSecurity Encryption Method Based on MD5[P]. DEStech Transactions on Engineering and Technology Research,2016.

# 致谢

毕业设计的结束也预示着大学生涯即将结束，我也将会投入到工作岗位中迈入新的生活，同时我想说在这次毕业设计中学到了很多以往未接触过的珍贵经验。

在这里我要感谢我的指导老师，在毕设开题的时候为我拟定毕业设计课题，并为我指定好设计方向和规划。开发过程中，老师对系统的见解和想法，都灌输到我的项目当中。最后在老师的帮助下完成这个项目以及毕业论文，整个过程学到了老师德严谨态度、创新意识等等。

同时要感谢平时遇到问题，帮助我解决的同学们，让我的项目开发变得更加顺畅。

其次要感谢我实习公司的前辈们，是他们让我感受到真实的企业项目开发，让我从一个懵懂的菜鸟在历练中成长。

最后感谢丽水学院机损及科学与技术的全体老师，四年来对我的教导和培育。