|  |  |  |
| --- | --- | --- |
| 南农大  **本科生毕业论文（设计）**  南农校徽 | | |
|  | 题 目: | 大学生兼职信息管理系统的 |
|  |  | 设计与实现 |
|  | 姓 名: | 钟黎阳 |
|  | 学 院: | 信息科技学院 |
|  | 专 业: | 计算机科学与技术 |
|  | 班 级: | 计科142班 |
|  | 学 号: | 19214219 |
|  | 指导教师: | 赵力 职称: 副教授 |
| 2018 年5 月 5 日 | | |
| 南京农业大学教务处制 | | |

**目 录**

[摘要 1](#_Toc514747644)

[关键词 1](#_Toc514747645)

[**Abstract** 1](#_Toc514747646)

[**Key words** 2](#_Toc514747647)

[1 绪论 2](#_Toc514747648)

[1．1研究背景及意义 2](#_Toc514747649)

[1．1．1 国内研究现状 2](#_Toc514747650)

[1．1．2 国外研究现状 3](#_Toc514747651)

[1．2研究目的和内容 3](#_Toc514747652)

[2 开发环境和应用技术 3](#_Toc514747653)

[2．1开发环境概述 3](#_Toc514747654)

[2．1．1 MyEclipse概述 3](#_Toc514747655)

[2．1．2 MySQL数据库概述 4](#_Toc514747656)

[2．1．3 Tomcat服务器 4](#_Toc514747657)

[2．2相关技术概述 4](#_Toc514747658)

[2．2．1 SSM技术 4](#_Toc514747659)

[2．2．2 Spring技术 4](#_Toc514747660)

[2．2．3 Spring MVC技术 4](#_Toc514747661)

[2．2．4 MyBatis技术 4](#_Toc514747662)

[3 大学生兼职信息管理系统的需求分析与设计 4](#_Toc514747663)

[3．1需求分析 4](#_Toc514747664)

[3．1．1 业务需求 4](#_Toc514747665)

[3．1．2 功能需求分析 4](#_Toc514747666)

[3. 2 面向对象需求建模 5](#_Toc514747667)

[3. 2. 1 分析构建用例图 5](#_Toc514747668)

[3. 2. 2 分析构建活动图 6](#_Toc514747669)

[3. 3 面向对象设计建模 8](#_Toc514747670)

[3. 3. 1 分析构建设计类 8](#_Toc514747671)

[3. 3. 2 分析构建时序图 9](#_Toc514747672)

[3．4系统总体设计 10](#_Toc514747673)

[3．4．1 系统总体架构 10](#_Toc514747674)

[3．4．2 功能模块设计 11](#_Toc514747675)

[3．4．3 数据库设计 13](#_Toc514747676)

[4 大学生兼职信息管理系统的功能实现 21](#_Toc514747677)

[4．1通用功能 21](#_Toc514747678)

[4. 1. 1登录功能的实现 21](#_Toc514747679)

[4. 1. 2忘记密码重置密码功能 22](#_Toc514747680)

[4. 1. 3获取职位类别功能 24](#_Toc514747681)

[4．2大学生用户功能 25](#_Toc514747682)

[4．2．1 职位搜索功能 25](#_Toc514747683)

[4. 2. 2职位筛选功能 26](#_Toc514747684)

[4．3招聘者用户功能 27](#_Toc514747685)

[4. 3. 1 查看评价功能 27](#_Toc514747686)

[4．4 管理员用户功能 28](#_Toc514747687)

[4. 4. 1 查看发布职位统计功能 28](#_Toc514747688)

[4. 4. 2 投递简历统计 29](#_Toc514747689)

[5 系统配置及测试 30](#_Toc514747690)

[5. 1 系统配置 30](#_Toc514747691)

[5．1．1 系统环境构成 30](#_Toc514747692)

[5．1．2 Web运行环境配置步骤 31](#_Toc514747693)

[5．2 测试目的 31](#_Toc514747694)

[5．3 测试过程 31](#_Toc514747695)

[5．3．1 大学生模块功能测试 31](#_Toc514747696)

[5．3．2 招聘者模块功能测试 31](#_Toc514747697)

[5．3．3 管理员模块功能测试 32](#_Toc514747698)

[5．4测试结论 32](#_Toc514747699)

[6 总结与展望 32](#_Toc514747700)

[致谢 33](#_Toc514747701)

[参考文献： 33](#_Toc514747702)

大学生兼职信息管理系统的设计与实现

计科专业学生 钟黎阳

指导教师 赵力

摘要：兼职已经成为现代大学生大学生活中不可或缺的一部分，针对传统的找兼职方式信息不安全，针对性差，效率低下等缺点，本文设计了一种方便、快捷、安全、低成本、针对性强的找兼职方式——大学生兼职信息管理系统。本系统采用当前流行的SSM框架开发，使用HTML语言进行前端页面的搭建，Vue.js进行数据的渲染以及页面的交互，分别对大学生、招聘者和管理员三类用户的各项功能进行了设计和实现。 大学生能查找适合自己的职位进行申请，并且举报虚假信息。招聘者能发布兼职信息，对收到的简历进行管理并且可以通知投递者面试。管理员能对兼职信息进行审核、管理黑名单、统计职位信息等。通过该系统，基本满足了各类用户的不同需求，切实提高了招聘效率，信息的安全性。

关键词：大学生兼职；管理信息系统；SSM；Vue.js

Design and Implementation of College Student Part-time Information Management System

Student majoring in computer science Liyang Zhong

Tutor Li Zhao

**Abstract**：Part-time job has become an indispensable part of college students' life in modern university. For the disadvantages of traditional information, such as unsafe information, targeted investigation, and low efficiency, this study has designed a convenient, fast, and safe method to find part-time students, which is Part-time Information Management System. The system applied the current popular SSM framework for the program development, HTML language for front-end page setup, Vue.js for data rendering and page interaction. Additionally, it was designed with the functions of college students, recruiters and administrators respectively. Students can find applications that suit their position and inform false information. Recruiters can post part-time information, manage resumes they receive and notify the interviewers. Administrators can audit part-time job information, manage blacklists, and analyze position-information. It basically satisfies the different needs of various users and effectively improves the efficiency of recruitment and the security of personal information.

**Key words:** Part-time job；information management system；SSM；Vue.js

# 绪论

## 1．1研究背景及意义

对大学生而言，谋求兼职不仅能给自己带来一定程度的经济收入，还可以锻炼自己的社会能力。怎么才能方便快捷地得到有关兼职信息，对大学生来说显得尤为关键。当前大学生获取兼职信息的渠道较为分散，沟通效率较低，例如填表格等待通知、从QQ兼职群获取信息、以及电话和邮件等。招聘者发布兼职信息的渠道也比较少，如通过人工宣传、发布QQ消息等。并且整个过程缺乏管理，也存在很多虚假兼职信息缺乏审核的问题。

近年来，在互联网上发布求职和招聘信息双向选择的新型方式迅速流行起来，网络的众多优势例如，快速的信息传递、较低的空间距离影响、廉价的成本等，都有效地解决了这些问题。因此，互联网成为大学毕业生最青睐、最普及的一种求职途径[1]。为了解决大学生谋求兼职的传统问题，我们可以借鉴毕业生求职的方式，借助现有的网络技术开发一个大学生兼职信息管理系统，以改变大学生谋求兼职的传统模式。基于信息管理系统的大学生兼职管理系统的开发，具有非常重要的现实意义[2]。

### 国内研究现状

前程无忧和智联招聘是现在做的比较好的两家网站。这两家招聘网站有相似的地方，也有不同的地方。它们都是利用网络进行职位信息的发布与传播。它们在赢利手段上具有共同点，主要通过收取会员费、广告费等。它们所面向的对象没有针对性，用户范围广。它们的名气不相上下，而不同点也很鲜明，智联招聘重点面向科技方面，而前程无忧则主要面向传统企业，两个招聘网站的关注点不相同。两个都有显著的劣势，会员多。同时它们也有相同的缺点，即虚假信息多。

此外还有一些相对小一些平台，如赶集网，58同城等。这两个网站所涉及到的业务范围比较广泛，主打几个方面，如工作招聘、房屋出租与二手房买卖、二手物品买卖等。由于业务非常混杂，所以没有自己的特色。工作类比不上智联招聘，前程无忧；房产类比不上链家；二手物品买卖又比不上咸鱼、转转。并且可以免费在上面发布招聘信息，吸引了很多不法分子在其平台上发布很多虚假信息，导致其消息鱼龙混杂。信息真实性没有得到审核，安全性非常差。并且没有给用户提供有效的申诉途径，用户体验很差。

### 1．1．2 国外研究现状

上世纪九十年代初，美国率先实施了Nii计划，大量的美国公民开始接触互联网。因此美国互联网行业迅速发展起来，成为世界上互联网发展普及的开端，利用互联网进行招聘求职工作的形势也开始流行起来 [3]。1997年逐步出现了一些以招聘为主要服务内容的网站，主要分布在金融、银行、投资和高新技术产业上，具体为网民提供一些空缺的技术岗位。随着计算机产业的发展，芯片技术的更新以及网络的普及，初期的招聘网站逐步发展形成了许多具有较大规模的招聘网站，如雅虎（Yahoo）、Moster等，都是使用频次较高的大型网站。在此之后，博客和社区等小型网站开始兴起，许多网站开始探索提供专业化服务，更加贴合求职者和招聘人的需求，例如应聘者和招聘公司的双向选择，这就为现在一些大型的专业招聘网站和小型的行业型招聘网站之间激烈地相互竞争[4]。目前，很多国家和企业使用招聘网站引进人才。

## 1．2研究目的和内容

本研究的目的是开发一个大学生兼职信息管理系统，打通招聘者和大学生之间沟通的桥梁。为大学生提供一个快速获取兼职信息的渠道，同时也为招聘者提供了发布兼职信息的渠道。为了保证兼职信息的真实性，为管理员提供了对兼职信息审核的功能。与大学生传统找兼职的方式相比，本系统存在如下优势：

（1）成本低廉

已互联网为载体的兼职管理系统，使大学生不必到处打听兼职消息，在本系统即能快速的获取自己想要的兼职信息，并且快速的投递简历。同时也为招聘者提供了跟大学生进行沟通的渠道，解决了招聘者不知如何将信息提供给大学生以及通过大学组织发送兼职信息需要缴费等问题。

（2）信息安全

传统的兼职信息缺乏审核，导致大学生收到许多虚假信息，严重损害了大学生的利益。本系统提供管理员审核兼职信息功能，对兼职信息的安全性有保证。

（3）针对性强

传统信息类别混杂，本系统将职位的类别、种类存在数据库中，可根据管理员的需求手动添加和删除职位类别。只保留一些真实的职位，增强针对性。

# 开发环境和应用技术

## 2．1开发环境概述

### 2．1．1 MyEclipse概述

以eclipse为基础的企业级工作平台MyEclipse，是一个功能更为全面的工具。与使用eclipse开发J2EE相比， MyEclipse的开发效率提高了很多。MyEclipse支持等各种开源产品 [5]。本系统使用MyEclipse作为服务器开发工具。

### 2．1．2 MySQL数据库概述

MySQL具有安全性高、用法简单、速度快、兼容性强等优点，近年来，被开发人员广泛使用[6] [7]。虽然MySQL也有很多缺陷，例如，它不支持自定义数据类型，对存储过程和触发器的支持不够良好，但用于本系统足够了[8]。本系统使用MySQL作为数据库。

### 2．1．3 Tomcat服务器

Tomcat可以作为web系统的容器，可以响应静态页面。Tomcat具有技术先进、性能稳定、开源等优点。所以开发人员会经常选用Tomcat来部署项目进行测试。本系统采用Tomcat8作为系统部署的服务器。

## 2．2相关技术概述

### 2．2．1 SSM技术

SSM框架为Spring、Spring MVC和MyBatis的缩写，是目前比较主流的Java EE企业级框架，它被应用于搭建各种大型应用系统[9]。

### 2．2．2 Spring技术

Spring是一个开源框架，能够解决复杂性问题，具有简单、可测试和松耦合的特点，是一个轻量级的控制反转（IoC）和面向切面（AOP）的容器框架[10]。使用Spring容器可以减少系统耦合，简化开发。

### 2．2．3 Spring MVC技术

Spring MVC支持的Spring特性，让开发过程变得非常简单和规范。Spring MVC分离了控制器、模型对象、分派器以及处理程序对象的角色，这种分离让它们更容易进行定制[11] [12]。

### 2．2．4 MyBatis技术

传统的开发模式在对数据库操作时需要开发人员手写JDBC实现数据库的连接，然后才能进行操作。MyBatis几乎消除了所有的JDBC代码和参数的手工设置以及结果集的检索。使用者可以通过直接在XML配置文件中配置好POJO类与数据库的映射关系，就可以操作数据库以进行增删改查。简单地说，MyBatis是一个管理数据增删改查的框架[13]。

# 大学生兼职信息管理系统的需求分析与设计

## 3．1需求分析

### 3．1．1 业务需求

大学生兼职信息管理系统通过互联网进行招聘，其主要功能是帮助大学生找到兼职工作以及帮助招聘者找到兼职人员。本系统需要给大学生提供搜索职位功能和投递简历功能；需要给招聘者提供发布职位和简历管理等功能。通过互联网进行招聘具有简单、便捷、时效性非常强等优点，可以解决大学生在找兼职过程中的烦恼。

### 3．1．2 功能需求分析

本文设计的大学生兼职信息管理系统所涉及的用户可以分为三类，即大学生用户、招聘者用户和管理员用户。每一类用户的需求如下：

（1）大学生：注册、登入、修改个人信息、修改密码、修改简历、预览简历、查看职位、搜索职位、收藏职位、投递简历、举报虚假信息、查看投递简历的状态等功能。

（2）招聘者：注册、登入、修改密码、修改个人信息、发布职位、管理职位、管理收到的简历等功能。

（3）管理员：审核招聘者发布的兼职信息、处理大学生用户的举报信息、拉黑用户、黑名单管理、查看统计等功能。

## 3. 2 面向对象需求建模

### 3. 2. 1 分析构建用例图

针对大学生用户而言，使用大学生兼职信息管理系统主要是账号信息的注册，以及登入后个人信息的填写、完善简历、浏览此平台上的各类职位信息、查看职位具体信息。对自己需要的职位投递简历，并且查看自己的历史投递记录以及简历被处理的情况。在经过面试后可以对面试过的职位填写自己的评价，如果职位信息存在虚假的可能性，大学生用户可以对职位进行举报。

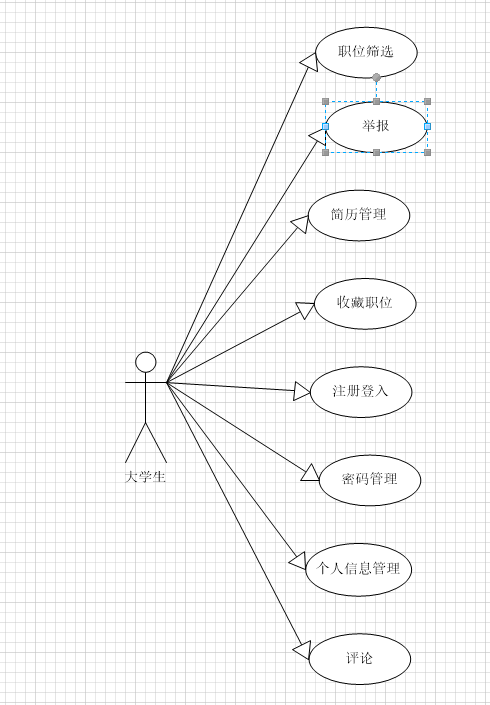


图 3-1 大学生用例图

针对招聘者而言，其使用此系统主要是账号注册和登入。登入之后对个人信息的填写以及发布职位信息。对收到的简历进行管理以及通知面试等。在进行面试之前可以查看投递简历者的个人评价，并且在面试结束之后可以对参加面试者进行评价。

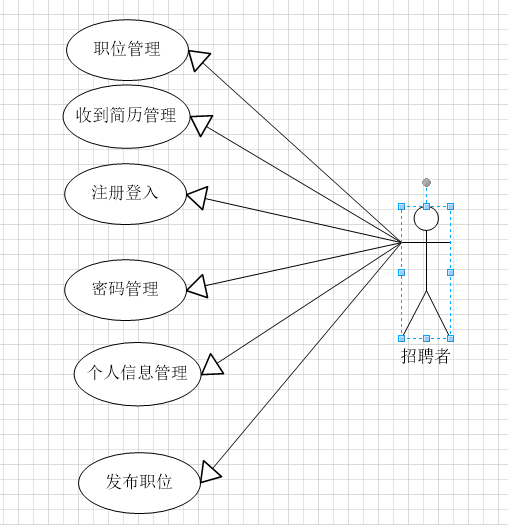


图3-2 招聘者用例图

针对管理员而言，其使用此系统主要为了审核招聘者发布的职位信息，对大学生举报的非法信息进行处理、黑名单的管理以及查看发布职位统计和投递简历统计。

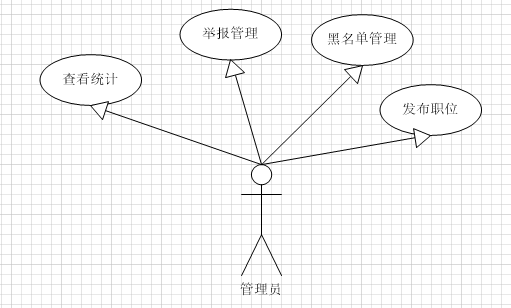


图3-3 管理员用例图

分析本系统主要业务功能，得出如下所示系统用例图：

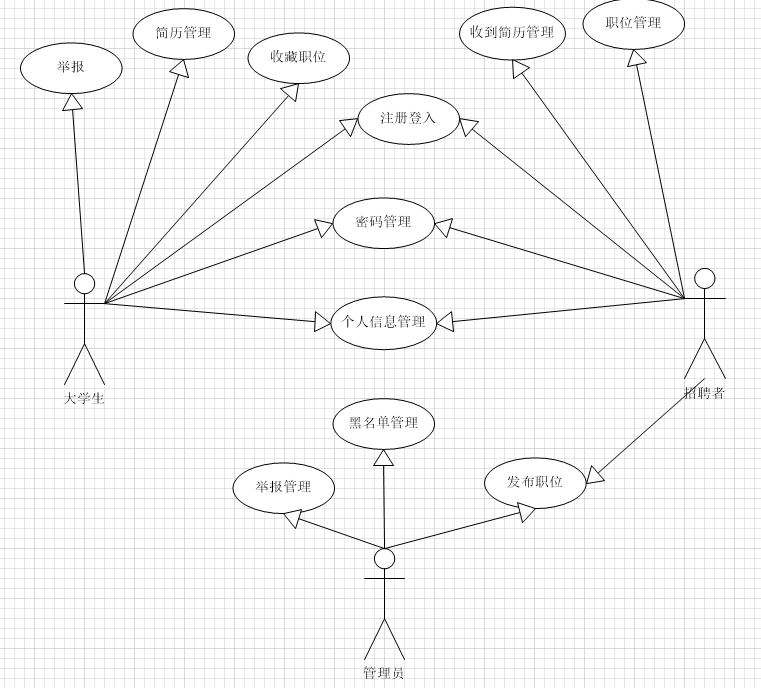


图3-4 系统用例图

### 3. 2. 2 分析构建活动图

本系统活动图较多，以下选取具有代表性的三个活动图进行展示：

（1）招聘者登录本系统之后，需要填写个人信息。在填写完个人信息之后可以进行发布职位信息，发布的职位都会显示在管理员的待审核界面上，通过管理员审核的职位信息才能显示出来，被大学生用户浏览。

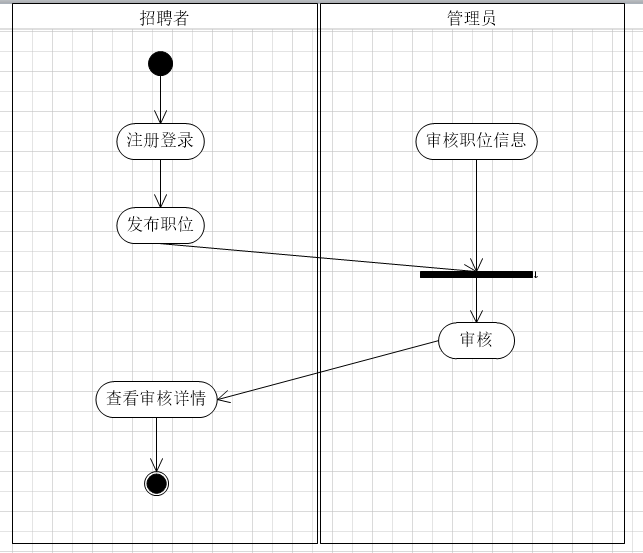


图3-5 招聘者发布职位活动图

（2）大学生用户登录本系统后，需要填写个人信息，完善个人简历。之后大学生用户可以根据自己的喜好查看职位具体信息，并且向想应聘的职位投递简历。投递简历后招聘者可以在简历管理模块对简历进行操作，大学生用户可以在简历管理模块看到自己简历的实时状态。

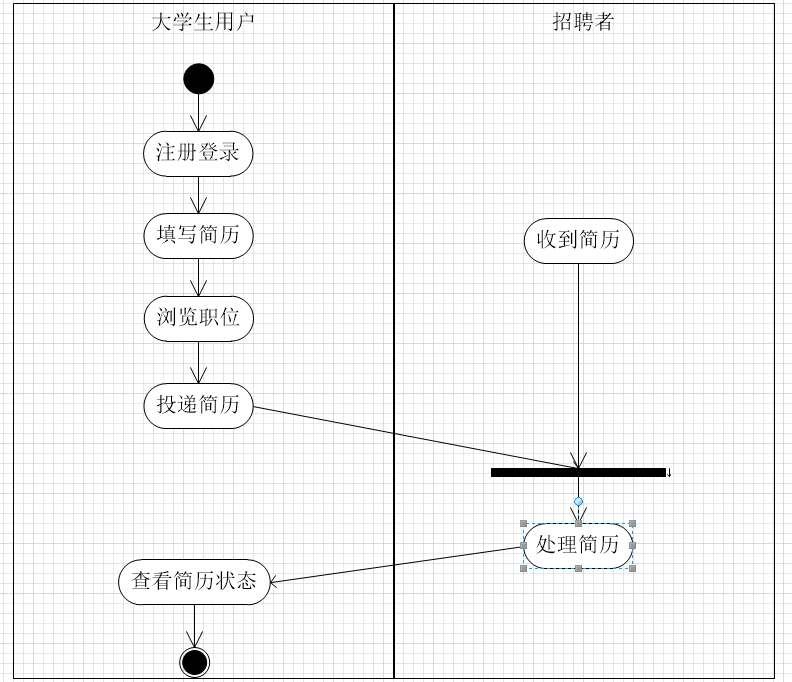


图3-6 投递简历活动图

（3）大学生用户可以举报虚假的职位信息，举报后管理员能收到举报信息并对举报信息进行处理。核实情况后可以对发布虚假信息的用户进行拉黑，不让此用户再登录本系统。

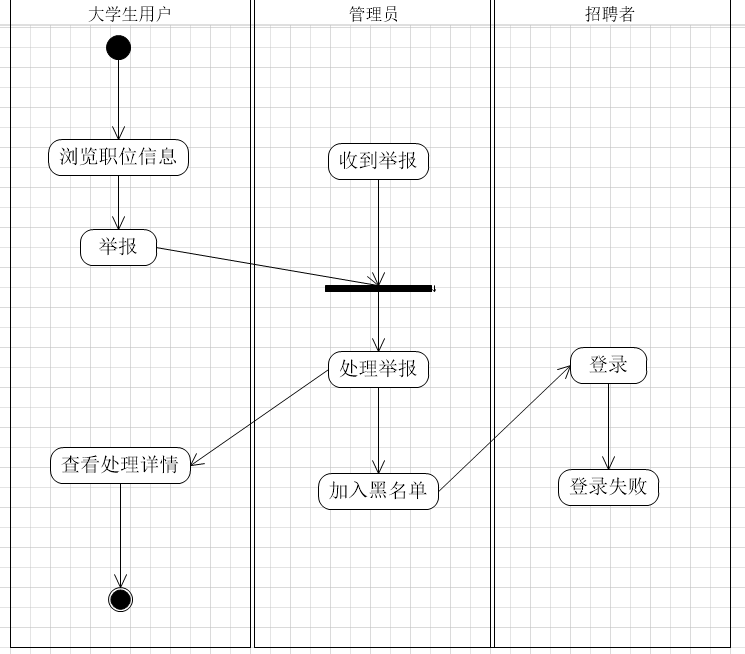


图3-7 举报活动图

## 3. 3 面向对象设计建模

### 3. 3. 1 分析构建设计类

（1）大学生用户模块设计

经过以上对系统的分析，了解了大学生用户所要进行的操作，现在进入设计阶段。大学生用户模块设计涉及到账户信息、个人信息、简历信息、已投递简历、已收藏简历、评论信息和举报信息。

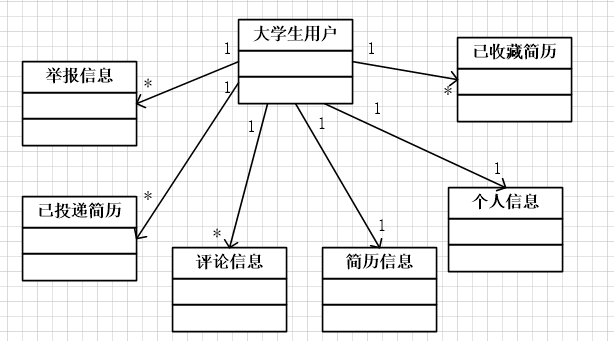


图 3-8 大学生用户类图

（2） 招聘者用户模块设计

招聘者用户设计到账户信息、个人信息、收到简历、发布职位信息、黑名单信息、审核信息。

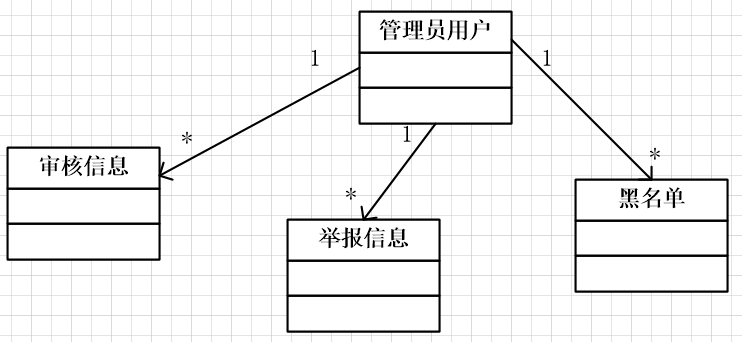


图 3-9 招聘者用户类图

（3）管理员用户模块设计

管理员用户涉及到审核信息、举报信息和黑名单。

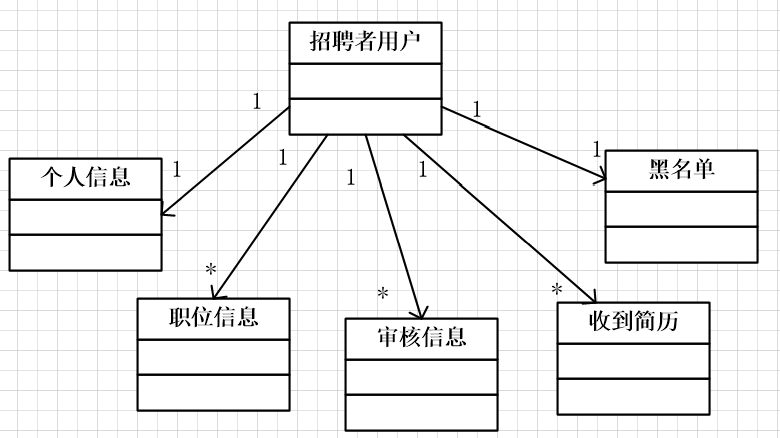


图 3-10 管理员用户类图

### 3. 3. 2 分析构建时序图

目前，所有模块的具体设计已经实现，接下来需要对每个功能模块建立时序图，时序图可以帮助开发者描述每个功能在时间和业务逻辑上的顺序，与活动图相比信息含量更丰富，此处选取登录模块进行展示说明。

3.3.2.1用户登录时序图

用户在使用系统时，如果没有账号则可以注册账号。账号注册成功以后可以登录系统，在登录系统时系统会自动判断登录用户的用户类型，从而显示不同用户的不同模块。

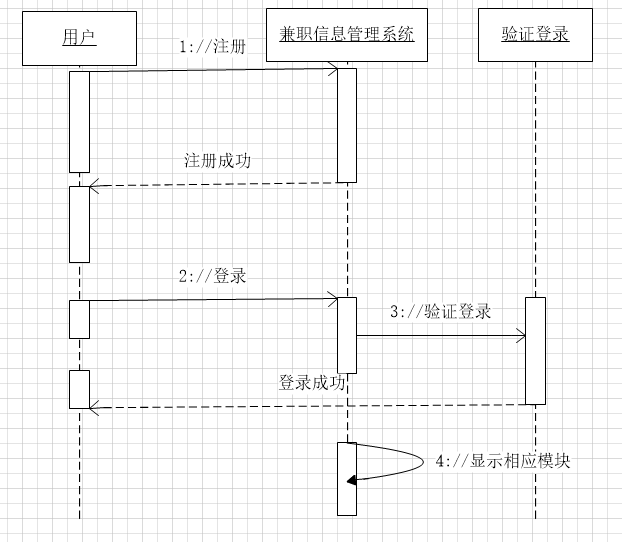


图3-11 用户登录时序图

## 3．4系统总体设计

### 3．4．1 系统总体架构

本系统使用B/S架构进行开发，使用SpringMVC模式，利用Model层处理逻辑、View层显示界面，Controller层进行前两层之间关系的控制。具体实现的时候，Service层执行数据的处理和封装，Mybatis进行系统与数据库的交互，html和js进行数据的展示与页面的交互。系统整体结构如图3-12所示：

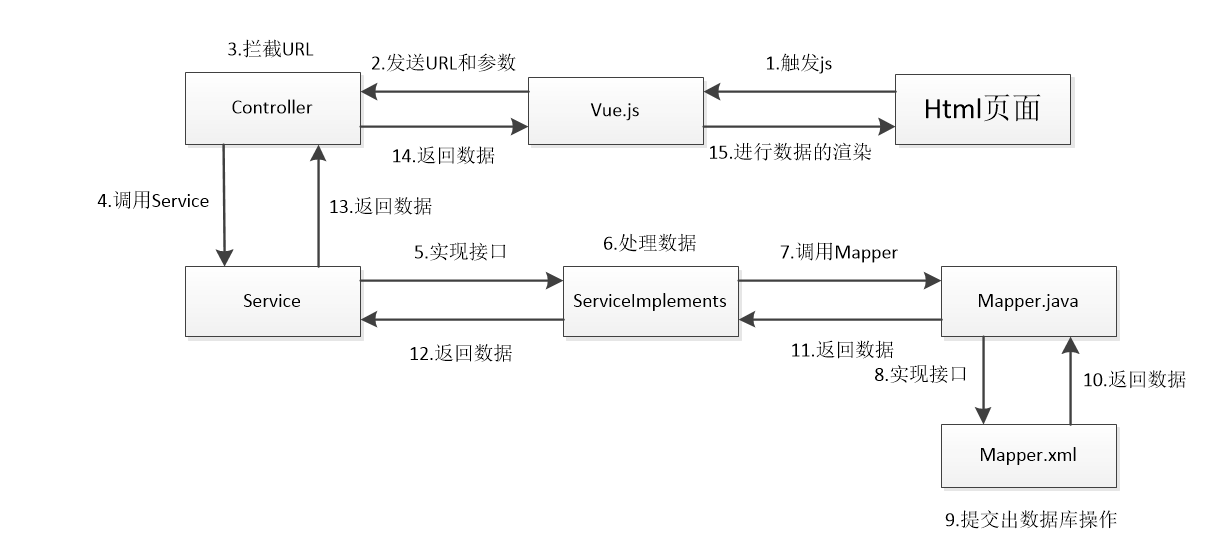


图 3-12 系统整体结构图

### 3．4．2 功能模块设计

根据对本系统功能的分析，本系统可以分为前、后台两部分。前台包括大学生（求职者）模块和招聘者模块，求职者模块的主要功能有注册、登录、密码修改、个人信息填写、简历编写、简历预览、职位搜索和查看、收藏职位、投递简历、已投递简历管理等；招聘者模块的主要功能有注册、登录、密码修改、个人信息填写、发布职位、职位管理、收到的简历管理等。后台为管理员用户，其主要功能有对招聘者发布的职位信息进行审核、对求职者举报的用户进行处理、对黑名单的管理、查看统计。以下是本系统的功能体系结构图：

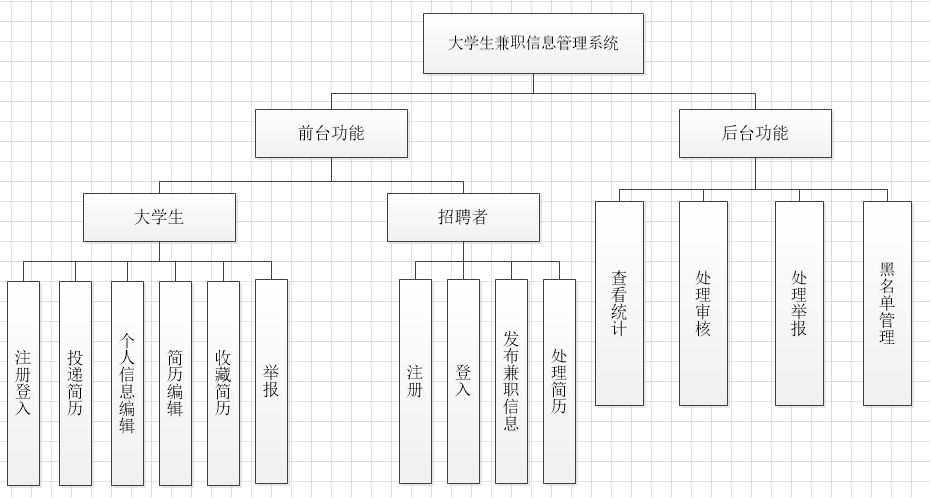


图3-13 系统功能体系结构图

3.4.2.1大学生模块功能设计

大学生需要注册一个账号以登入系统，然后才能享受到系统提供的各种功能。大学生需要使用自己常用的邮箱进行注册，此邮箱为之后忘记密码从而重置密码接收验证码用。大学生首次登入后进入个人信息编辑页面，填写个人信息保存。之后可使用各种服务。在首页查看各类职位信息、搜索职位、查看职位具体信息等功能。查看具体职位信息时，可以收藏心仪的职位，也可以向招聘公司投递简历。可以在收藏职位模块查看收藏的职位信息，在我投递的简历模块可以查看自己已投递的简历，以及目前简历的投递状态。以下为大学生模块功能图：

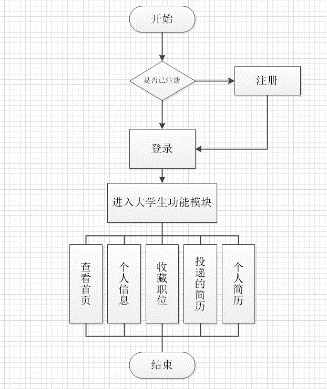


图3-14大学生功能流程图

3.4.2.2 招聘者模块功能设计

招聘者可以在本系统注册账号，注册后登录系统即可使用系统提供的各种功能。可以发布兼职信息，对自己发布的信息进行管理。查看收到的简历，并且对简历进行相应的处理。同时招聘者也可以对自己的个人信息进行填写。以下为招聘者模块功能图：

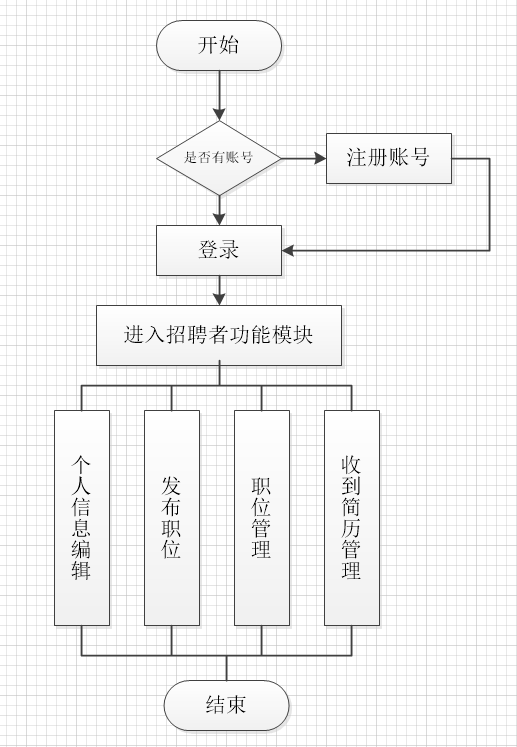


图3-15 招聘者功能流程图

3.4.2.3管理员模块功能设计

管理员的账号是一开始系统分配的，不允许自己创建管理员的账号。管理员成功登录后才能对招聘者发布的兼职信息进行审核、对大学生所举报的用户进行处理、对黑名单进行管理。以下为管理员模块功能图：

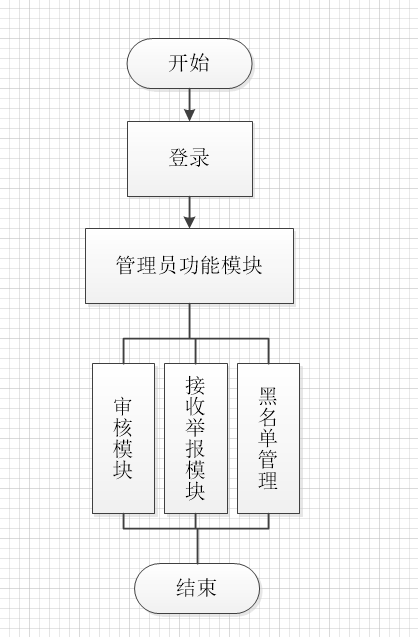


图3-16 管理员功能流程图

3.4.2.4登录功能设计

本系统将三类用户的登录功能集成在一起，通过后台来判断登录用户的信息来判断登录用户的类型，从而达到进去不同模块的功能。登录时首先判断用户名是否存在，如不存在则提示用户用户名不存在。如果用户名存在，则判断此用户名对应的密码与用户输入的密码是否匹配。如密码不匹配则提示用户密码不正确，如匹配则读取用户信息，判断用户类型，跳转到不同的功能模块。如果是首次登入的用户则会跳转到个人信息页面，提示用户先填写个人信息。以下为登录功能流程图：

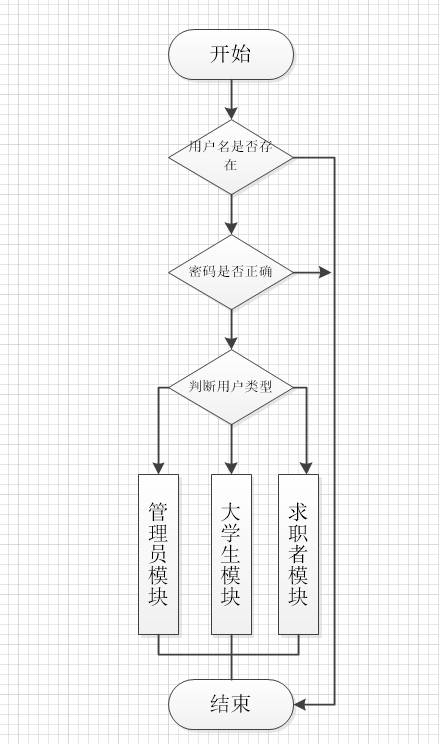


图3-17 登录功能流程图

### 3．4．3 数据库设计

信息管理系统的主要功能就是给用户提供信息，所以应该特别注意数据的存储，本系统数据库设计的主要任务就是设计一个合理的符合要求的数据库模型。

通过上述需求分析可知，本系统所需要设计的表如下：账号信息、个人信息、简历信息、投递简历信息、工作信息、工作种类信息、黑名单信息、审核信息、举报信息和收藏职位信息等数据信息。以上各个信息数据之间都有关联，有利于减少冗余、方便信息的获取。

3.4.3.1数据库设计概要

通过对本系统的需求分析和功能分析，可以得到本系统的数据库实体如下：个人信息实体、账号信息实体、简历信息实体、大学生收藏的职位信息实体、大学生投递的简历信息实体、大学生举报的信息实体、工作种类信息实体、职位信息实体、求职者收到的简历信息实体、黑名单信息实体、管理员审核信息实体、评价信息实体等。账号信息存储了所有用户的账号和密码以及用户类型，个人信息存储了各用户的基本信息。账号信息与个人信息为一对一的关系。设计满足不同用户需求的实体是实现数据库设计的基础[12]。以下为本系统数据库E-R图：

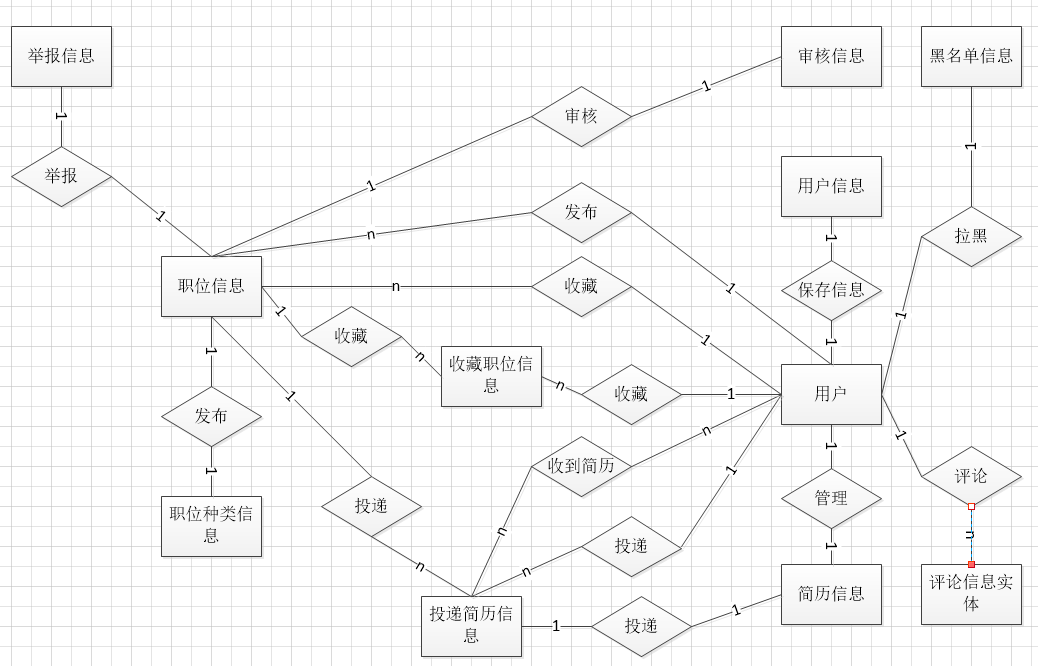


图3-18 数据库全局E-R图

（1）账号信息实体

账号信息实体包含：用户id、登录邮箱、登录密码、用户类型。以下为账号信息实体属性图：

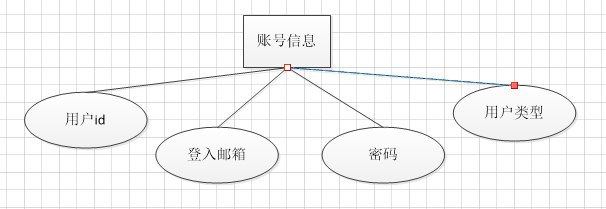


图3-19 账号信息实体

（2）用户信息实体

用户信息实体包含：用户信息id、用户id、用户姓名、用户手机号码、用户头像、学校名称、专业名称、身份证号码和身份证学生证图片。以下为用户信息实体属性图：

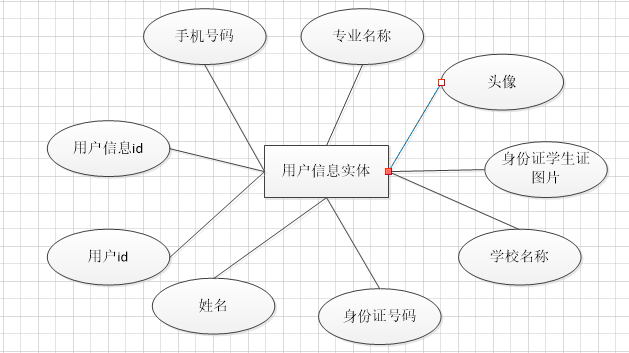


图3-20 用户信息实体

（3）职位种类实体

职位种类实体包含：职位种类id、大类名称、小类名称、类别名称和职位代码。以下为职位种类实体属性图：

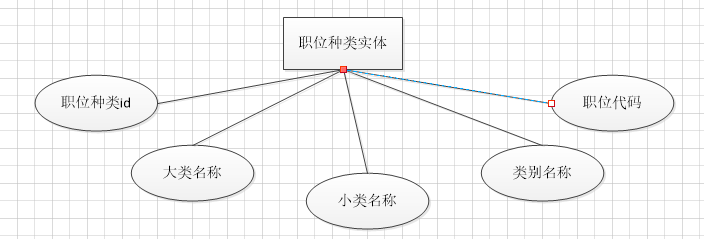


图3-21 职位种类实体

（4）职位信息实体

职位信息实体包含：职位id、职位代码、职位名称、工作性质、月薪、工作经验要求、学历要求、职位描述、工作地址、接收简历邮箱、发布者用户id、浏览次数、审核状态、发布时间、有效状态。以下为职位信息实体属性图如：

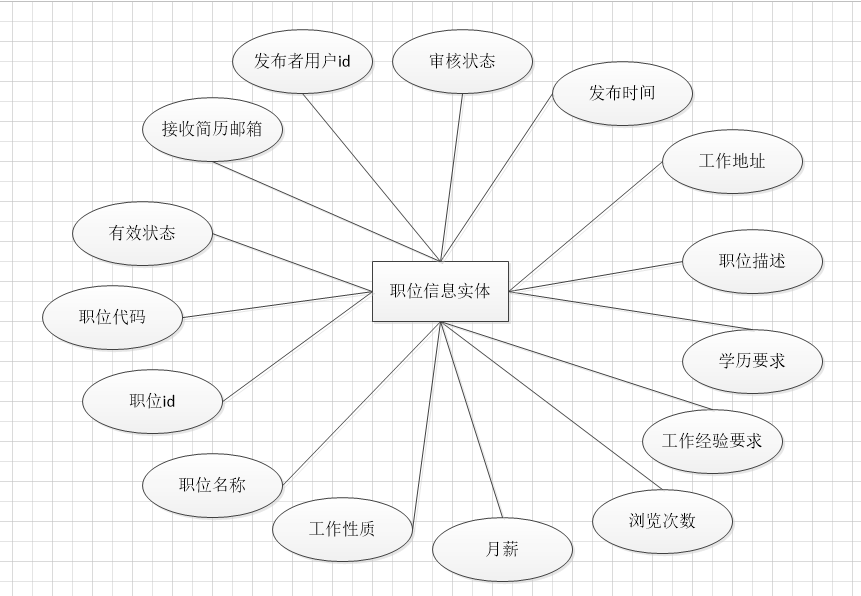


图3-22 职位信息实体

（5）简历信息实体

简历信息实体包含：简历id、姓名、学历、性别、工作经验、手机号码、邮箱、期望工作地址、期望工作性质、期望职位、期望月薪、公司名称、职位名称、学校名称、专业名称、学业开始时间、学业结束时间、自我描述、简历名称、简历照片、简历所有者id。以下为简历信息实体属性图：

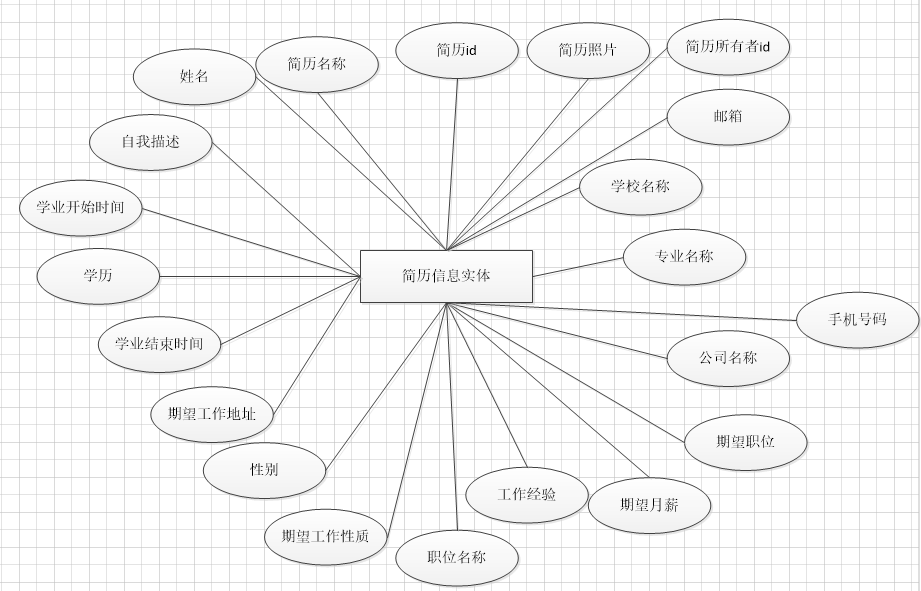


图3-23 简历信息实体

（6）收藏职位信息实体

收藏职位信息实体包含：记录id、收藏职位id、收藏者id。以下为收藏职位信息实体属性图：

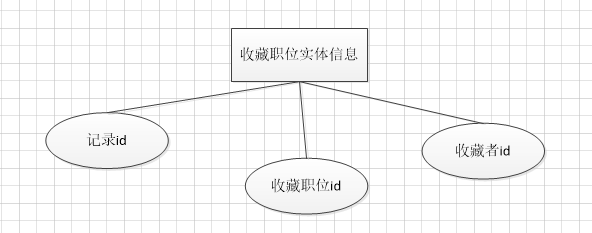


图3-24 收藏职位信息实体

（7）投递简历信息实体

投递简历信息实体包含：记录id、简历id、职位id、简历状态、投递时间、查看时间。以下为投递简历信息实体属性图：

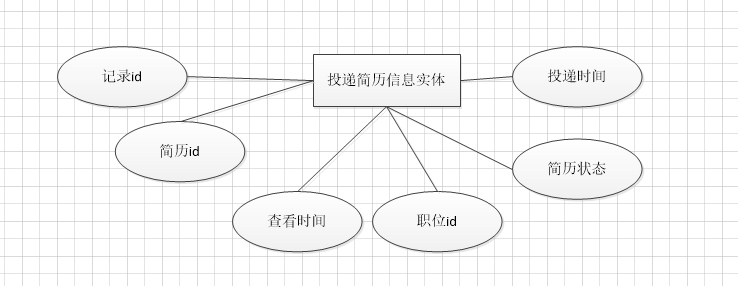


图3-25 投递简历信息实体

（8）举报信息实体

举报信息实体包含：记录id、举报类型、举报人id、被举报人id、举报时间、处理时间、处理结果、举报内容。以下为举报信息实体属性图：

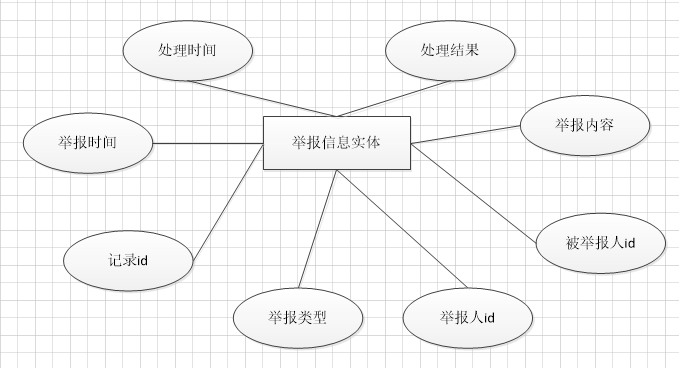


图3-26 举报信息实体

（9）审核信息实体

审核信息实体包含：记录id、审核类型、发起时间、审核时间、状态、审核信息id、账户。以下为审核信息实体属性图：

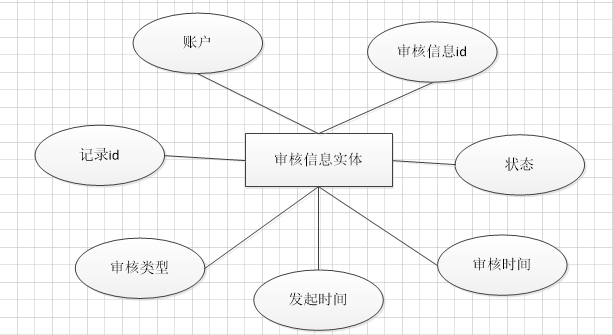


图3-27 审核信息实体

（10） 黑名单信息实体

黑名单信息实体包含：记录id、用户id、拉黑时间、举报信息id、拉黑原因。以下为黑名单信息实体属性图：

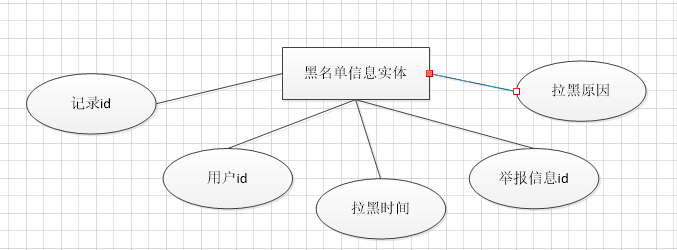


图3-28 黑名单信息实体

（11）评论信息实体

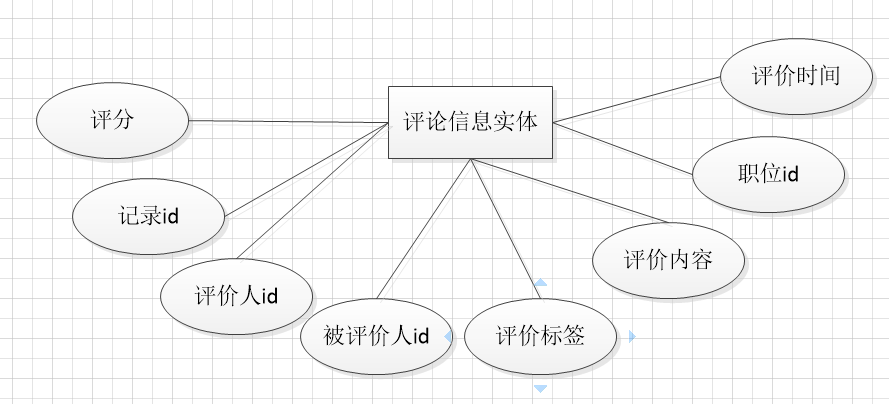
评论信息实体包含：记录id、评价人id、被评价人id、标签、内容、时间、职位id、评分。以下为评论信息实体属性图：

图3-29 评论信息实体

3.4.3.2 数据库详细设计

本系统的核心内容是进行数据库设计，本系统使用的是MySQL数据库。以下是本系统的数据表。

数据表t\_user用来存储所有用户的账号基本信息，主键为c\_id。其内容如下：

表3-1 t\_user表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 注释 |
| c\_id | VARCHAR(32) | 否 | 记录id |
| c\_password | VARCHAR(300) | 否 | 密码 |
| c\_email | VARCHAR(300) | 否 | 邮箱账号 |
| n\_type | INT | 是 | 用户类型 |

数据表t\_userinfo用来存储用户的具体个人信息，主键为c\_id。其内容如下：

表3-2 t\_userinfo表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 注释 |
| c\_id | VARCHAR(32) | 否 | 记录id |
| c\_bh | VARCHAR(32) | 否 | 账号编号 |
| c\_xm | VARCHAR(300) | 是 | 姓名 |
| c\_sjhm | VARCHAR(300) | 是 | 手机号码 |
| c\_txzp | VARCHAR(300) | 是 | 头像照片 |
| c\_xxmc | VARCHAR(300) | 是 | 学校名称 |
| c\_zymc | VARCHAR(300) | 是 | 专业名称 |
| C\_sfzhm | VARCHAR(300) | 是 | 身份证号码 |
| C\_tplj | VARCHAR(300) | 是 | 图片路径 |

数据表t\_jobinfo用来存储职位具体信息，主键为c\_id。其内容如下：

表3-3 t\_jobinfo表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 注释 |
| c\_id | VARCHAR(32) | 否 | 记录id |
| N\_code | INT | 是 | 职位类别代码 |
| c\_zwmc | VARCHAR(300) | 是 | 职位名称 |
| n\_gzxz | INT | 是 | 工作性质 |
| n\_yx | INT | 是 | 月薪 |
| N\_gzjy | INT | 是 | 工作经验 |
| N\_xlyq | INT | 是 | 学历要求 |
| C\_zwms | VARCHAR(300) | 是 | 职位描述 |
| C\_gzdz | VARCHAR(300) | 是 | 工作地址 |
| C\_jsjlyx | VARCHAR(300) | 是 | 接收简历邮箱 |
| C\_fbzid | VARCHAR(32) | 是 | 发布简历用户id |
| N\_llcs | INT | 是 | 浏览次数 |
| N\_zt | INT | 是 | 审核状态 |
| Dt\_fbsj | TIMESTAMP | 是 | 发布时间 |
| N\_yxzt | INT | 是 | 有效状态 |

数据表t\_jianliinfo用来存储用户具体的简历信息，主键为c\_id。其内容如下：

表3-4 t\_jianliinfo表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 注释 |
| c\_id | VARCHAR(32) | 否 | 记录id |
| c\_name | VARCHAR(300) | 是 | 姓名 |
| N\_xl | INT | 是 | 学历 |
| n\_xb | INT | 是 | 性别 |
| N\_gzjy | INT | 是 | 工作经验 |
| C\_sjhm | VARCHAR(300) | 是 | 手机号码 |
| C\_yx | VARCHAR(300) | 是 | 邮箱 |
| C\_qwgzdz | VARCHAR(300) | 是 | 期望工作地址 |
| N\_gzxz | INT | 是 | 工作性质 |
| C\_qwzw | VARCHAR(300) | 是 | 期望职位 |
| N\_qwyx | INT | 是 | 期望月薪 |
| C\_gsmc | VARCHAR(300) | 是 | 公司名称 |
| C\_zwmc | VARCHAR(300) | 是 | 职位名称 |
| C\_xxmc | VARCHAR(300) | 是 | 学校名称 |
| C\_zymc | VARCHAR(300) | 是 | 专业名称 |
| Dt\_kssj | TIMESTAMP | 是 | 开始时间 |
| Dt\_jssj | TIMESTAMP | 是 | 结束时间 |
| C\_zwms | VARCHAR(300) | 是 | 自我描述 |
| C\_jlmc | VARCHAR(300) | 是 | 简历名称 |
| C\_zp | VARCHAR(300) | 是 | 头像照片 |
| C\_syzid | VARCHAR(300) | 是 | 简历所有者id |

数据表t\_jobtype用来存储职位的种类信息，主键为c\_id。其内容如下：

表3-5 t\_jobtype表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 注释 |
| c\_id | VARCHAR(32) | 否 | 记录id |
| c\_jobtype1 | VARCHAR(300) | 否 | 工作第一大类 |
| c\_jobtype2 | VARCHAR(300) | 否 | 工作第二大类 |
| C\_jobtype3 | VARCHAR(300) | 否 | 职位类别 |
| N\_code | INT | 否 | 职位代码 |

数据表t\_receivejianliinfo表用来存储招聘者收到的简历信息以及大学生投递的简历信息，主键为c\_id。其内容如下：

表3-6 t\_receivejianliinfo表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 注释 |
| c\_id | VARCHAR(32) | 否 | 记录id |
| c\_jlic | VARCHAR(32) | 否 | 简历id |
| c\_jobid | VARCHAR(32) | 否 | 职位id |
| N\_zt | INT | 否 | 状态 |
| Dt\_tdsj | TIMESTAMP | 否 | 投递时间 |
| Dt\_cksj | TIMESTAMP | 是 | 查看时间 |

数据表t\_shenhelist表用来存储管理员需要审核的信息，主键为c\_id。其内容如下：

表3-7 t\_shenhelist表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 注释 |
| c\_id | VARCHAR(32) | 否 | 记录id |
| N\_shlx | INT | 是 | 审核类型 |
| Dt\_fqsj | TIMESTAMP | 是 | 发起时间 |
| Dt\_shsj | TIMESTAMP | 是 | 审核时间 |
| N\_zt | INT | 是 | 状态 |
| C\_jlid | VARCHAR(32) | 是 | 审核信息id |
| C\_zh | VARCHAR(300) | 是 | 账号 |

数据表t\_reportinfo表用来存储管理员需要处理的举报信息，主键为c\_id。

其内容如下：

表3-8 t\_reportinfo表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 注释 |
| c\_id | VARCHAR(32) | 否 | 记录id |
| c\_jbrid | VARCHAR(32) | 是 | 举报人id |
| c\_bjbrid | VARCHAR(32) | 是 | 被举报人id |
| n\_jblx | INT | 是 | 举报类型 |
| Dt\_jbsj | TIMESTAMP | 是 | 举报时间 |
| Dt\_clsj | TIMESTAMP | 是 | 处理时间 |
| N\_cljg | INT | 是 | 处理结果 |
| C\_jbnr | VARCHAR(300) | 是 | 举报内容 |

数据表t\_wscjob表存储了大学生用户收藏的职位信息，主键为c\_id。其内容如下：

表3-9 t\_wscjob表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 注释 |
| c\_id | VARCHAR(32) | 否 | 记录id |
| c\_userid | VARCHAR(32) | 否 | 用户id |
| c\_jobid | VARCHAR(32) | 否 | 职位id |

数据表t\_blacklis表存储了被管理员拉黑的用户信息，主键为c\_id。其内容如下：

表3-10 t\_blacklist表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 注释 |
| c\_id | VARCHAR(32) | 否 | 记录id |
| c\_yhid | VARCHAR(32) | 否 | 用户id |
| Dt\_lhsj | TIMESTAMP | 是 | 拉黑时间 |
| C\_reportid | VARCHAR(32) | 是 | 举报的记录id |
| N\_hlyy | INT | 是 | 拉黑原因 |

数据表t\_commentinfo表存储了用户之间的评价的信息，主键为c\_id。其内容如下：

表3-11 t\_commentinfo表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 注释 |
| c\_id | VARCHAR(32) | 否 | 记录id |
| c\_password | VARCHAR(300) | 否 | 密码 |
| c\_email | VARCHAR(300) | 否 | 邮箱账号 |
| n\_type | INT | 是 | 用户类型 |

# 大学生兼职信息管理系统的功能实现

## 4．1通用功能

### 4. 1. 1登录功能的实现

本系统将三类用户的登录集成在一个方法之中，通过判断用户类型的不同，实现进入不同系统模块的功能。实现登入功能的流程图如下所示：

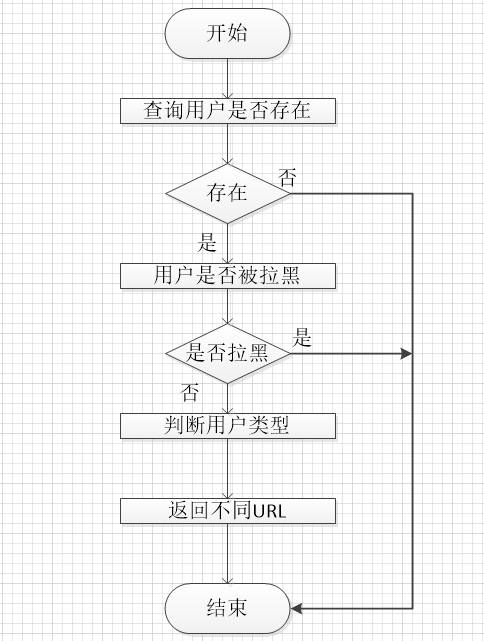


图4-1 登录流程图

S1：首先后台拦截前端发送的请求，然后通过JSONObject.fromObject()方法解析JSON的方式获取参数中的账号和密码。然后通过MyBatis查询账号信息表。

S2：查看账户信息是否存在。如果存在则转S3，否则转S4。

S3：查看用户是否被拉黑。如果被拉黑则转S5，否则转S6。

S4：提示用户用户名不存在。

S5：提示用户已被拉黑。

S6：判断用户类型。如果是学生用户则转S7，如果是招聘者则转S8，否则转S9。

S7：跳转到学生用户界面。

S8：跳转到招聘者用户界面。

S9：跳转到管理员用户界面。

### 4. 1. 2忘记密码重置密码功能

本系统实现了忘记密码用户可以通过注册邮箱实现邮箱验证重置密码的功能。为经常忘记密码的用户提供了方便。具体实现流程图如下：

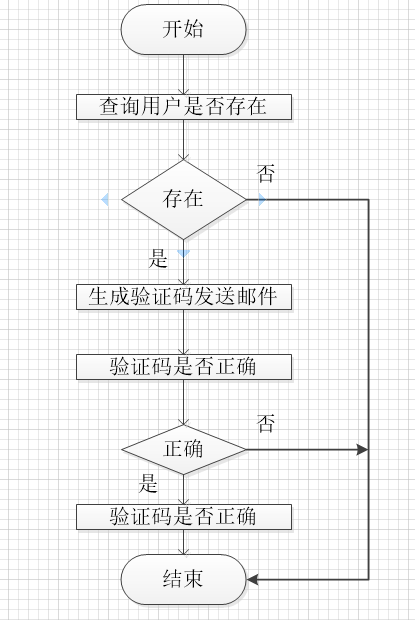


图4-2 邮箱验证修改密码流程

具体实现代码如下：

//根据邮箱查找用户信息，如不存在则返回

LoginEntity log=loginMapper.selectUser1(*to*);

**if**(如果用户不存在){

result.put("success", **false**);

result.put("message", "用户名不存在");

**return** GyUtils.*returnResult*(**true**, "成功", result);

}

//通过随机生成uuid的方法生成一组验证码保存

**public** **static** String getUUid(){

String uuid=UUID.*randomUUID*().toString();

**return** uuid.replace("-", "");

}

//调用封装好的发送邮件方法，设置基本信息与内容

String user = "zly75509@163.com";

String password = "sbsb75509";

String host = "smtp.163.com";

String from = "zly75509@163.com";

// 收件人

*to* = param.getString("yx");

//发送邮件方法

operation.sendMail(user, password, host, from, to,subject, sb.toString());

//用户输入验证码后与之前保存的验证码比较

**if**(验证码正确){

result.put("success", **true**);

result.put("message", "验证码正确");

**return** GyUtils.*returnResult*(**true**, "成功", result);

}**else**{

result.put("success", **false**);

result.put("message", "验证码不正确");

**return** GyUtils.*returnResult*(**true**, "成功", result);

}

首先查询所要修改密码的账号是否存在，如不存在则提示用户“账号不存在”。用户存在则发送邮件至用户邮箱，邮件信息中包含本次重置密码所需要的验证码。用户将验证码提交给后台，后台判断验证码是否与发送的验证码一致，如不一致则提示用户“验证码错误”，如验证码正确则用户可以修改密码。效果展示如图4-3、图4-4、图4-5所示：



图4-3 验证码效果图



图4-4 验证码发送成功效果图



图4-5 验证码正确效果图

### 4. 1. 3获取职位类别功能

本系统实现了对职位的类别进行数据库的存储，通过查询数据库的信息即可获取职位的种类信息。封装了一个获取职位信息的方法，可以被多次调用。管理员可以通过在数据库添加职位类别的方法动态的改变职位类别，使发布者发布职位的时候选择职位类别比较灵活。具体实现代码如下：

//配置拦截器，拦截获取职位种类的URL

@RequestMapping(value="/jobtype",method=RequestMethod.*POST*)

//将大类先放入一个数组存储

List<String> jobtype1=**new** ArrayList<String>();

//去掉重复的类，等到大类数组

**for**(遍历职位信息){

**if**(jobtype1包含大类职位名称){

**continue**;

}**else**{

将大类职位加入jobtype1中;

}

}

//遍历第一大类数组

**for**(遍历职位信息){}

//将在同一大类下的信息放入第二大类数组

**if**(大类名称相同){

**if**(jobtype2包含第二类名称){

将第二类名称加入jobtype2中;

}**else**{

**continue**;

}

}

//遍历第二大类相同的数组将职位类别以json对象的形式返回

**if**(第二类名称相同){

将职位信息封装，返回给前端；

}

效果展示如图4-6所示：



图4-6 职位类别效果图

## 4．2大学生用户功能

### 4．2．1 职位搜索功能

本系统实现了对职位信息的搜索和筛选，通过前端所传的参数来判断有哪些筛选条件，将筛选筛选的结果封装好返回给前端。实现过程如下：

首先拦截前端的请求，并且解析得到前端所传的参数，其中包含用户筛选的条件以及操作分页的参数page和pagesize。根据页码和每页显示数量计算偏移量。根据MyBatis写动态SQL，此处使用Mybatis的条件判断语句<if test=*""*></if>，最后拼接limit pagesize offset offset实现分页。搜索职位功能流程图如下所示：

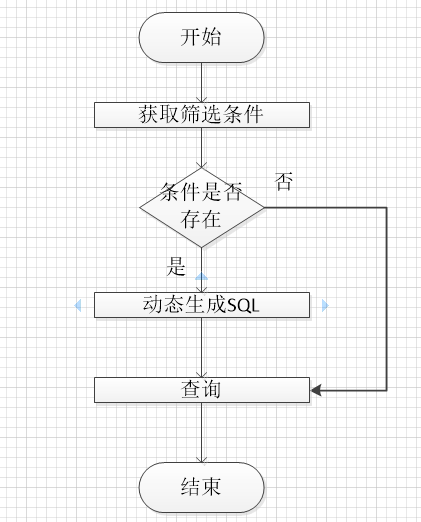


图 4-7 搜索职位流程图

效果展示如图4-8所示：



图4-8搜索职位效果图

### 4. 2. 2职位筛选功能

本系统可以根据大学生用户在填写简历时所写的期望职位，自动为用户筛选与他期望职位相关的信息展示给用户。筛选功能会优先筛选有评分的职位，并且按照评分的高低优先展示给用户。职位筛选流程图如下：

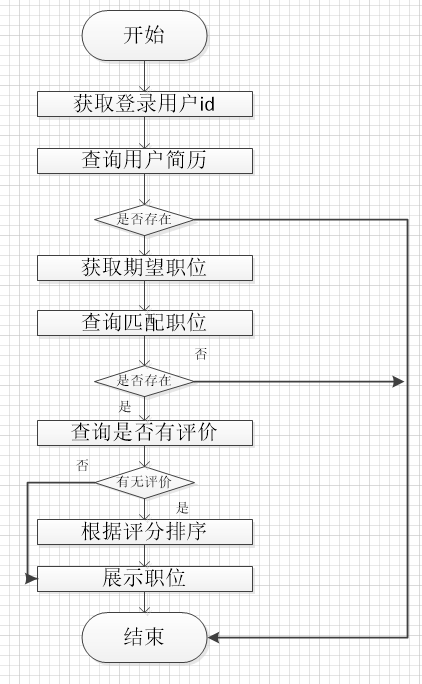


图4-9 筛选职位流程图

S1：获取登录大学生用户的用户id。

S2：查询用户简历。

S3：用户简历是否存在，若存在则转S4，否则结束。

S4：获取简历中的期望职位选项。

S5：根据期望职位到数据库中匹配职位。

S6：若存在职位则转S7，否则结束。

S7：查询评价。

S8：是否有评价，若有则转S9，否则转S10。

S9：根据评分排序。

S10：展示职位。

筛选职位效果图如下所示：



图4-10 筛选职位效果图

## 4．3招聘者用户功能

### 4. 3. 1 查看评价功能

本系统实现招聘者和大学生用户之间相互评价的功能。招聘者可以对面试过的应聘人员进行评价，大学生用户可以对面试过的职位进行评价。并且招聘者可以查看应聘人员的评价信息，帮助决策。评价功能流程图如下：

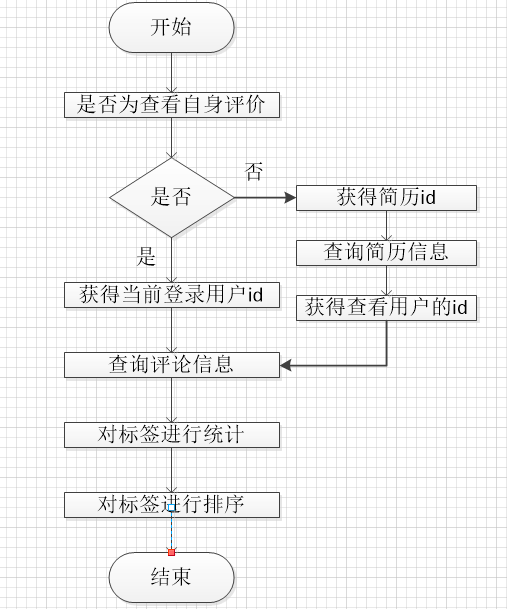


图4-11 查看评论流程图

S1：判断是否为自身评价。如果是则转S2，否则转S3。

S2：通过session获取当前登录用户的用户id，转S6。

S3：获得简历id，转S4。

S4：通过简历id查询简历信息，转S5。

S5：获得需要查看的用户id，转S6。

S6：查询评论信息。

S7：对标签进行统计算出平均分。

S8：对标签进行排序。

S9：封装好信息返回给前端。

查看收到的评论效果图如下所示：



图4-12 评价页面效果图

## 4．4 管理员用户功能

### 4. 4. 1 查看发布职位统计功能

本系统为管理员提供了查看发布职位统计功能，管理员在登录系统后可以在统计模块查看发布职位统计。管理员可以设置日期查看某一段时间发布的职位统计。发布职位统计功能流程图如下所示：

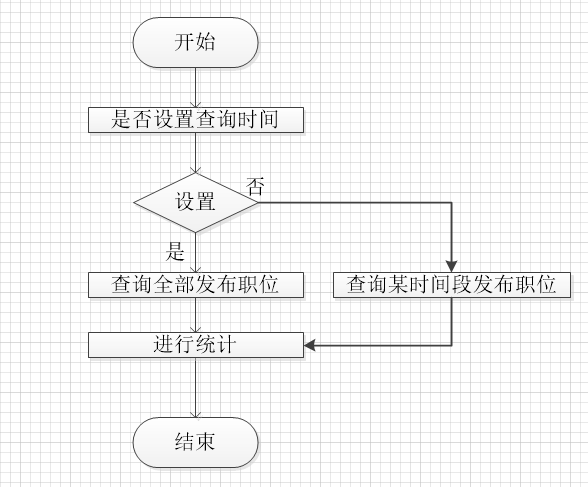


图4-13 发布职位流程图

S1：查询管理员设置的时间。

S2：管理员如果没设置时间则转S3，否则转S4。

S3：查询全部发布职位，转S5。

S4：查询某段时间发布职位。

S5：将这些信息与职位类型表匹配，统计出发布的职位有哪些类别的，发布了多少条信息。将统计好的数据返回给前端进行展示。

发布职位统计效果图如下所示：

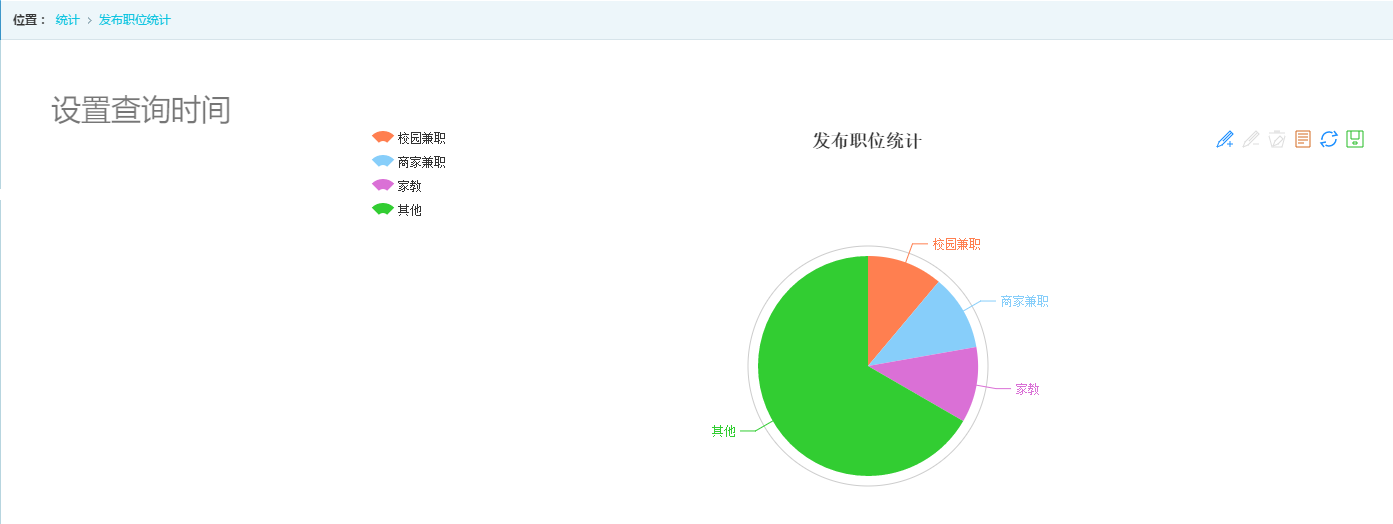


图4-14 发布职位统计效果图

### 4. 4. 2 投递简历统计

本系统为管理员提供了查看投递简历的功能，管理员登录后可在统计模块找到投递简历统计，管理员可以自己设置时间统计某段时间内投递的简历，也可以查看系统中所有投递的简历。投递简历统计流程图如下：

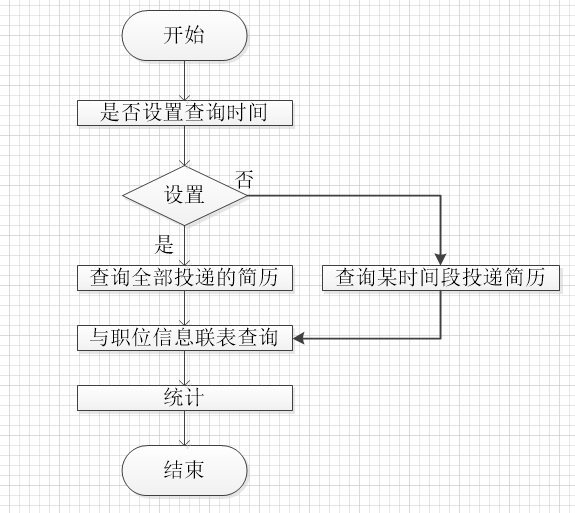


图4-15 投递简历统计流程图

S1：查询管理员设置的时间。

S2：管理员如果没设置时间则转S3，否则转S4。

S3：查询全部投递简历的信息，转S5。

S4：查询某段时间内投递简历的信息。

S5：将得到的投递简历信息与职位信息进行联表查询，得到职位信息。

S6：最后将信息进行统计，返回给前端进行展示。

投递简历统计效果图如下所示：

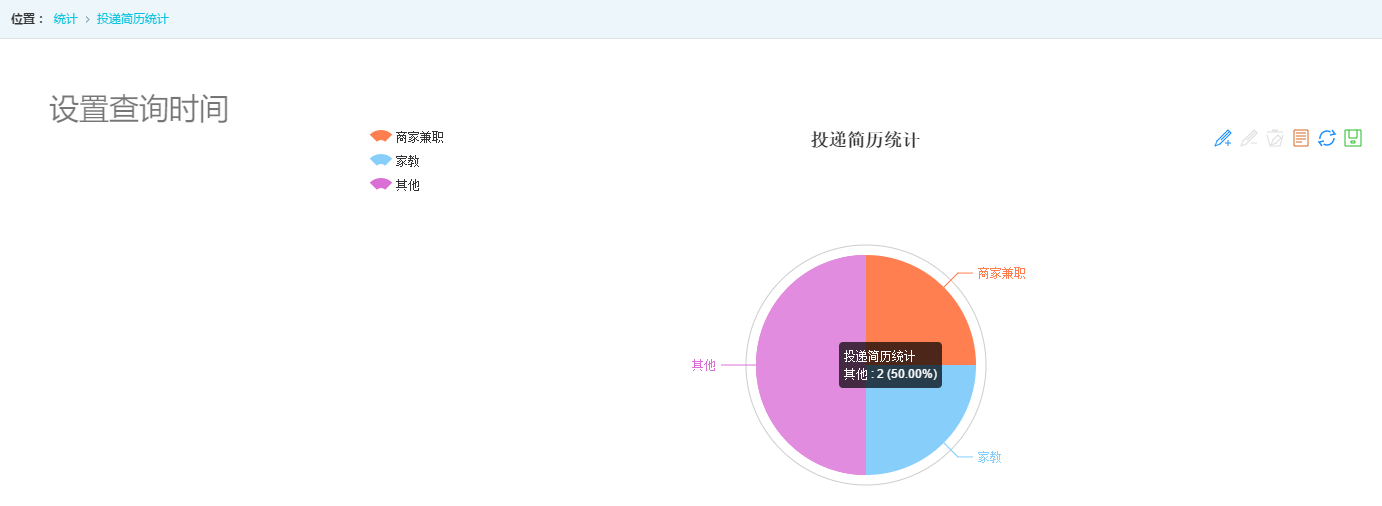


图4-16 投递简历统计效果图

# 系统配置及测试

## 5. 1 系统配置

### 5．1．1 系统环境

本系统环境配置如下所示：

•服务器：Tomcat8

•数据库：MySQL

•开发工具：My Eclipse2014

•JDK版本：JDK1.7

### 5．1．2 Web运行环境配置步骤

（1）下载相应的JDK、Tomcat、MySQL、My Eclipse2014，安装并且配置环境变量。

（2）在My Eclipse2014上配置好服务器之后把程序部署到Tomcat8下，并启动服务器。

（3）打开Google Chrome浏览器，输入项目访问网址http://localhost：8080/项目名称，就可以访问该系统了。

## 5．2 测试目的

测试系统功能是否有错误，测试系统流程是否有问题，以及页面展示效果是否美观。

## 5．3 测试过程

### 5．3．1 大学生模块功能测试

5.3.1.1测试步骤

（1）注册一个大学生账号并登录。

（2）填写个人信息、简历、查看简历。

（3）能够搜索职位、查看职位具体信息。

（4）能够收藏职位、取消收藏职位。

（5）投递简历，并且能够对投递的简历状态进行查看。

（6）能够对虚假信息进行举报。

（7）能查看系统自动筛选职位。

（8）能对面试过的职位进行评价。

（9）能查看我收到的评价。

5.3.1.2预期结果

（1）注册成功并且能够跳转到登录界面。

（2）成功进行个人信息的填写、简历的填写和预览。

（3）成功通过搜索查询到正确的职位，并且能查看职位的具体信息。

（4）成功对职位进行收藏、取消收藏等操作。

（5）能够正常投递简历并且能查看简历的状态。

（6）成功对虚假信息进行举报。

（7）成功查看系统自动筛选职位。

（8）成功对面试过的职位进行评价。

（9）成功查看我收到的评论。

5.3.1.3测试结果

测试通过。

### 5．3．2 招聘者模块功能测试

5.3.2.1测试步骤

（1）注册一个招聘者账号并跳转到登录界面。

（2）进行个人信息的编辑保存。

（3）发布职位信息、并且对职位信息的审核状态查看、已经管理职位信息实现上线下线等操作。

（4）管理收到的简历，并且通知投递者面试等。

（5）对面试过的大学生进行评价。

（6）查看需要面试的大学生的评价信息。

5.3.2.2.预期结果

（1）能够注册用户跳转到登录界面。

（2）能够对个人信息进行编辑保存。

（3）能够发布职位信息、能够查看职位信息的审核状态、上线下线已发布的职位信息。

（4）能够管理收到的简历，并且通知投递者面试。

（5）能够对面试过的大学生进行评价。

（6）能够查看需要面试的大学生的评价信息。

5.3.2.3测试结果

通过测试。

### 5．3．3 管理员模块功能测试

5.3.3.1测试步骤

（1）使用管理员账号登录系统并成功跳转到管理员模块。

（2）查看审核信息模块，对审核信息进行处理。

（3）查看举报信息模块，对举报信息进行处理。

（4）查看黑名单，对黑名单信息进行处理。

（5）查看发布职位统计和投递简历统计。

5.3.3.2预期结果

（1）成功使用管理员账号登录系统并成功跳转到管理员模块。

（2）成功查看审核信息模块，对审核信息进行处理。

（3）成功查看举报信息模块，对举报信息进行处理。

（4）成功查看黑名单，对黑名单信息进行处理。

（5）成功查看发布职位统计和投递简历统计。

5.3.3.3测试结果

通过测试。

## 5．4测试结论

本系统各项功能均可正常工作，功能没有错误，流程没有问题，页面展示效果美观。

1. 总结与展望

本课题通过对一般兼职招聘网站进行了设计与实现，实现了各项功能。但是本系统开发的时间较短，如果时间允许还可以做如下功能：

（1）可以为招聘者添加一个简历信息统计，可以让招聘者更直观的看到自己收到的简历的各类信息，有助于招聘者对自己的职位信息做出分析，提高招聘效率。

（2）可以添加用户之间的聊天功能，增强用户之间的沟通，为一些对人员需求比较紧急的招聘者提供了另一个有效的渠道，提高了招聘的效率，同时丰富了网站的功能。

（3）在本系统中目前只能对招聘者发布的信息进行审核，但是依然会有一些虚假。本系统在最初设计的时候已经包含了对用户信息进行审核的功能，但是由于时间的问题暂时没有完成，希望能在后续的工作中进行改进，以提高信息的安全性。

# 致谢

本次毕业设计是本人在南农本科学习生涯的最后一个任务，因此我以十分严谨和认真的态度完成毕业设计及论文撰写，这是对自己大学四年来所学知识和技术的综合体现。本次毕业设计首先要感谢南京农业大学信息科技学院计算机科学系提供的设计平台。感谢指导老师赵力老师的悉心指导，对本人的选题方向、研究思路和实验设计都提出了宝贵的意见和指正。还要感谢本人的毕业实习单位北京华宇信息技术有限公司对本次毕业设计的技术指导。

在这匆匆度过的时光里，要感谢班主任舒欣老师在四年里对我无微不至的照顾，以及在大学生活中给予我的帮助。也要感谢辅导员对我细心的指引，还有各任课老师的耐心教学。最后还要感谢我的父母一直以来对我的支持和鼓励。

# 参考文献：

[1] 吴昊，张东.大学生利用网络求职的调查与分析—基于全国 42 所高校大样本的实证研究[J].中国大学生就业，2007，(18)：49-51.

[2] 袁庆宏,刘昌苹.国内外网络招聘行业现状分析—兼论我国高校毕业生网络求职概况[J].中小企业管理与科技(下旬刊)，2009，07:281-282.

[3] 张萌，衣冯源.我国网络招聘研究综述[J].现代经济信息，2009，(20):138-139.

[4] 乔铁松.我国中小企业的人才招聘与管理问题研究[J].中小企业管理与科技(上旬刊)，2008，(8):16-17.

[5] 窦东阳，赵英凯，许文稼.利用Myeclipse简化基于Hibernate的数据库访问[J].计算机工程与设计，2007，28(4):773-774、788.

[6] 李荣国，王见.MySQL数据库在自动测试系统中的应用[J].计算机应用，2011，31(a02):169-171.

[7] Ioan G. Designing and interrogation a database by using MySQL, PHP and LabVIEW programmes [J]. Metalurgia International, 2008, 13:32-36.

[8] 高张，康小军.提高Tomcat服务器运行性能的研究[J].计算机与数字工程，2008，36(10):203-205.

[9] 王柄印.基于SSM技术的企业质量管理信息系统设计[J].数据库与信息管理，2015,23(33):67-68.

[1] 陈思耀. Spring技术在系统构建中的应用[J].韩山师范学院学报，2007,28(6):41-47

[11] 符红霞. Spring MVC技术分析及在实践教学系统中的应用[J].信息技术，2012,10(37):42-46.

[12] Szyperski, C. Beyond object-oriented programming [M]. New York：ACM Press，1988，7:90-97.

[13] 荣艳冬.关于Mybatis持久层框架的应用研究[J].应用成果，2015，6(12):86-88.