

上海第二工业大学

# 本科毕业设计（论文）



题    目：基于协同过滤的互联网岗位推荐系统的设计与实现

英文题目：DESIGN AND IMPLEMENTATION OF  
INTERNET JOB RECOMMENDATION  
SYSTEM BASED ON COLLABORATIVE  
FILTERING

学    号：20201111012

姓    名：王耀彬

班    级：20 计科 C2

专    业：计算机科学与技术

学部(院)：计算机与信息工程学院

入学时间：2020 级

指导教师：石林祥

日    期：2024 年 5 月 1 日

# 毕业设计（论文）独创性声明

本人所呈交的毕业论文是在指导教师指导下进行的工作及取得的成果。除文中已经注明的内容外，本论文不包含其他个人已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中作了明确说明并表示谢意。

作者签名：王耀彬

日期：2024.5.1

# 基于协同过滤的互联网岗位推荐系统的设计与实现

## 摘要

本文以 Python 程序设计语言为基础，利用网络爬虫技术实现了对于互联网招聘岗位信息的采集、解析、清洗、处理与存储。同时，通过 Web 应用对爬取得到的数据进行组织管理与展示。最终通过推荐算法对数据进行分析计算，并将推荐结果反馈给用户。本文通过需求分析明确了系统应具备的基本功能，包括互联网岗位信息获取、岗位数据的修改、维护、展示及推荐。采用自顶向下、总体到局部的设计思想最终实现了互联网岗位推荐系统的主要功能。

本文设计与实现的互联网岗位推荐系统包含三个主要模块：网络爬虫模块、Web 模块和推荐算法模块。网络爬虫模块采用 Scrapy 框架，能准确地获取每日变化的互联网岗位信息。Web 模块中前端使用 Vue 框架，后端使用 Django 框架，实现了用户友好的界面和稳定的服务。推荐算法模块使用基于热度的推荐与基于协同过滤的推荐，为用户提供了个性化的推荐服务。数据库采用阿里云 RDS MySQL 云数据库。

目前，系统已经进入运营维护阶段，经过系统测试，能够稳定高效地进行互联网岗位信息的收集与推荐。

**关键词：**网络爬虫；B/S 架构；Django；Vue；推荐算法；协同过滤；

# **DESIGN AND IMPLEMENTATION OF INTERNET JOB RECOMMENDATION SYSTEM BASED ON COLLABORATIVE FILTERING**

## **ABSTRACT**

This paper, based on the Python programming language, utilizes web crawling technology to collect, parse, clean, process, and store internet job information. At the same time, it uses web applications to organize and manage the data and display it. Finally, it uses recommendation algorithms to analyze the data and provide recommendation results to users. This paper clarifies the basic functions that the system should have through a demand analysis, including internet job information acquisition, modification, maintenance, display, and recommendation. The self-top-down, overall-to-local design philosophy is used to finally achieve the main functions of the internet job recommendation system.

The internet job recommendation system designed and implemented in this paper consists of three main modules: the web crawler module, the web module, and the recommendation algorithm module. The web crawler module uses the Scrapy framework to accurately obtain daily changing internet job information. The web module uses the Vue framework for the front end and the Django framework for the back end to provide a user-friendly interface and stable services. The recommendation algorithm module uses the hot-based recommendation and the collaborative filtering recommendation to provide personalized recommendation services to users. The database uses the Alibaba Cloud RDS MySQL cloud database.

At present, the system has entered the operation and maintenance stage. After system testing, it can collect and recommend Internet job information stably and efficiently.

**Key words:** Web crawler; B/S architecture; Django; Vue; Collaborative filtering

# 目录

1 绪论.....	1
1.1 研究背景与意义 .....	1
1.2 国内外研究现状简述 .....	1
2 互联网岗位推荐系统的相关理论与技术概述 .....	3
2.1 互联网岗位推荐系统概述 .....	3
2.2 网络爬虫模块概述 .....	3
2.3 Web 模块概述 .....	3
2.3.1 B/S 架构概述 .....	3
2.3.2 Django 概述 .....	3
2.3.3 Vue 概述 .....	4
2.4 基于用户的协同过滤推荐算法概述 .....	4
3 互联网岗位推荐系统的需求分析 .....	6
3.1 系统可行性分析 .....	6
3.2 网络爬虫模块需求分析 .....	6
3.3 Web 模块需求分析 .....	6
3.3.1 管理员端需求分析 .....	7
3.3.2 普通用户端需求分析 .....	7
3.4 推荐算法模块需求分析 .....	8
3.5 系统数据库需求分析 .....	8
4 互联网岗位推荐系统的设计与实现 .....	11
4.1 系统的总体功能设计 .....	11
4.2 系统网络爬虫模块的设计与实现 .....	11

4.3 系统 Web 模块的设计与实现 .....	13
4.3.1 基于 Django 的 Web 后端的设计与实现 .....	14
4.3.2 基于 Vue 的 Web 前端的设计与实现 .....	17
4.4 系统推荐算法模块的设计与实现 .....	23
4.4.1 基于热度的推荐 .....	23
4.4.2 基于用户的协同过滤推荐 .....	24
4.5 系统数据库的设计与实现 .....	24
4.5.1 数据库概念结构设计 .....	24
4.5.2 数据库物理结构设计 .....	27
5 系统运行测试及分析 .....	32
5.1 网络爬虫模块功能测试 .....	32
5.2 Web 模块功能测试 .....	32
5.2.1 用户操作测试 .....	32
5.2.2 管理员操作测试 .....	33
5.3 推荐算法模块功能测试 .....	34
6 结论 .....	36
致谢 .....	37
参考文献 .....	38

# 1 绪论

## 1.1 研究背景与意义

随着时代的进步与科技的日新月异，数字服务走进了千家万户，在当今数字化时代，互联网招聘已经成为人才招聘领域的主流招聘方式之一<sup>[1]</sup>。然而，随着互联网信息的爆炸式增长，互联网招聘平台逐渐出现大量的信息不对称与信息过载问题。这使得求职者与招聘方在选择称心如意的岗位与候选人时面临较大挑战。在此背景下，基于协同过滤的互联网岗位推荐系统应运而生。

传统的招聘系统往往依赖于简单关键词匹配或基于简单规则过滤的方法，这些方法存在着严重的局限性，无法有效的捕捉到岗位需求与求职者技能之间的复杂关系，将求职者与招聘方都淹没在浩如烟海的信息当中<sup>[2]</sup>，求职者与招聘方难以从中梳理出自己中意的岗位与候选人信息。而基于协同过滤的推荐系统能够通过协同过滤技术对海量数据进行学习和挖掘，从而实现精准、高效的招聘匹配。

研究基于协同过滤的互联网岗位推荐系统具有重要意义。首先，对于求职者而言，它可以帮助求职者更快速、更准确的找到符合自己技能和兴趣的岗位，提高求职效率；与此同时，对于招聘方而言，借助智能化的推荐系统，能够更好在成千上万的简历中筛选匹配的候选人，降低招聘成本，提高招聘效率。此外，研究基于协同过滤的岗位推荐系统还可以推动人才招聘领域的技术创新和发展，为推动更加智能化、个性化的招聘平台奠定基础。更进一步，随着人工智能技术的不断发展和应用，基于协同过滤的岗位推荐系统也具有更大的潜力和优势。协同过滤模型能够从海量的招聘数据中学习到更加丰富、抽象的特征表示，进而提高招聘匹配的准确性和个性化程度。总而言之，深入探究本推荐系统对于提升招聘效率、促进人才流动具有重要的现实意义和应用前景。

## 1.2 国内外研究现状简述

国内外在基于协同过滤的互联网招聘岗位推荐系统领域已经展开了广泛而深入的研究。在国内研究领域，主要有以下几点研究：

（1）基于内容的推荐方法：通过分析岗位描述和求职者简历等文本信息，利用协同过滤技术提取文本特征，从而实现岗位与求职者之间的匹配。

(2) 基于协同过滤的推荐方法：通过挖掘用户行为数据，如求职者的点击、收藏等行为，利用协同过滤模型学习用户的兴趣偏好，从而将不同用户的兴趣偏好信息相互推荐，为求职者推荐合适的岗位。

(3) 融合多模态信息的推荐方法<sup>[3]</sup>：融合多模态信息，如文本、图片、视频等，以提升推荐系统的效果。利用深度学习技术实现对多种类型信息的有效建模和融合，从而更全面地捕捉岗位需求和求职者特征之间的关系。

而国外研究领域的研究主要是以下几点：

(1) 协同过滤在招聘领域的应用落地：利用协同过滤技术实现更精准的职业匹配、个性化推荐以及用户满意度的提升。

(2) 推荐系统的可解释性和公平性。

(3) 跨领域研究融合：将招聘领域的深度学习技术与其他领域的技术相结合，如自然语言处理、计算机视觉等，以进一步提升招聘系统的效果和用户体验。

尽管国内外在基于协同过滤的互联网招聘岗位推荐系统领域已经取得了一定的进展，但仍然存在一些不足之处：首先是招聘信息数据问题，招聘数据通常存在质量参差不齐和稀疏性问题，这会影响协同过滤模型的训练效果和推荐结果的准确性。其次是基于协同过滤技术产生的模型的可解释性不足，其往往以黑盒的形式呈现，缺乏对结果的解释能力。最后是数据隐私与安全性，招聘数据涉及个人隐私和敏感信息，因此数据的安全性和隐私保护成为一个重要的问题。



## 2 互联网岗位推荐系统的相关理论与技术概述

### 2.1 互联网岗位推荐系统概述

基于协同过滤的互联网岗位推荐系统主要由网络爬虫模块、Web 模块与推荐算法模块三大模块组成，每个模块的相关理论与技术如下所述：

- 1.网络爬虫模块：网络爬虫技术、Scrapy 框架。
- 2.Web 模块：B/S 架构、Django 框架、Vue 框架。
- 3.推荐算法模块：基于热度的简易推荐算法，基于用户的协同过滤算法。

### 2.2 网络爬虫模块概述

网络爬虫技术是一种自动获取互联网信息的技术，它通过模拟人类浏览互联网的行为，自动访问网页并提取其中的数据。这项技术在大数据信息收集、搜索引擎优化与数据分析等领域被广泛应用。

Scrapy 是一个使用 Python 编写的网络爬虫框架，它提供了一套简单且灵活的工具以定义爬取和处理目标网页的规则，从而可以快速、高效的开发与部署爬虫<sup>[4]</sup>。Scrapy 框架主要由 Spider、Item、Pipeline、Downloader 与 Middleware 几个组件构成。Scrapy 的设计理念是“简单、灵活且高效”，其在爬取大规模数据、处理动态网页与自动化部署等方面表现出色。

### 2.3 Web 模块概述

#### 2.3.1 B/S 架构概述

B/S (Browser/Server Architecture) 架构，它将 Web 应用程序分为两个核心部分，即：浏览器端 (Browser) 和服务器端 (Server)。在 B/S 架构中，用户通过浏览器与服务器进行交互，用户通过浏览器发送请求，服务器接收请求并进行处理，然后将相应的数据和内容发送回浏览器进行展示。

B/S 架构由前端浏览器 (Client-Side) 和后端服务器 (Server-side) 构成，其具有良好的跨平台兼容性、无需安装客户端软件、易于部署、安全性强，被广泛应用于各种类型的 Web 项目中。

#### 2.3.2 Django 概述

Django 是用 Python 语言编写的高级 Web 后端框架，其致力于简化 Web 开

发过程，提供了一系列强大的工具组件使开发者可以快速高效的构建安全、扩展性强的 Web 应用程序[5]。

Django 采用经典的模型-视图-模板(MVT)架构，将 Web 应用程序分为模型、视图和模板三个组件，令开发者可以简单高效的组织管理代码，实现逻辑、数据和界面的分离。Django 提供自动生成的功能丰富的管理后台，这允许开发者基于模型快速的创建管理页面。与此同时，Django 还提供强大的对象-关系映射(ORM)系统，允许开发者使用 Python 来定义数据模型，并通过简单的 API 操作数据库，而不必专门编写细致的 SQL 文件。除此之外，Django 使用 URLconf 自动将 URL 请求映射到相应的视图函数或类，开发者只需简单地配置 URL 模式，就能实现 URL 的分发和处理，使得 URL 管理更加简单和灵活。

综上所述，Django 以其简单、高效、安全的特性，成为了开发者构建 Web 应用的首选框架之一，被广泛的应用于各种 Web 项目中。

### 2.3.3 Vue 概述

Vue.js 是当下流行的 JavaScript 前端框架，用于构建交互式的 Web 界面。Vue 是渐进式的框架，可逐步应用到 Web 项目中，也可以轻松的与其他库或现有项目集成[6]。

Vue 采用数据驱动的理念，通过将数据于 DOM 进行绑定，实现了数据的响应式更新。与此同时，Vue 采用了组件化的开发方式。将界面分解为独立的组件，每个组件包含自己的 HTML 模板、JavaScript 逻辑和样式，这使得 Vue 界面结构更加清晰，同时提高了代码的可复用性。更进一步，Vue 提供了简洁、直观的模板语法，允许开发者将 JavaScript 表达式嵌入到 HTML 模板中，实现动态数据绑定和逻辑控制。此外，Vue 利用虚拟 DOM 技术来提升性能，它通过在内存中维护一个虚拟 DOM 树，实现了对实际 DOM 的高效更新。

总而言之，Vue 以其简介、灵活、高效的特性，成为开发者构建 Web 前端的热门选择，被广泛应用于各类 Web 项目中。

## 2.4 基于用户的协同过滤推荐算法概述

协同过滤（Collaborative Filtering）是推荐系统中常用的技术之一，它利用系统用户之间的历史行为数据，例如用户的评分、搜索历史、浏览记录以及收藏记录等，通过分析用户或物品之间的相似性，来预测用户对位置物品的喜好。

协同过滤有如下表 2-1 几种类型：

表 2-1 协同过滤推荐方法分类

协同过滤方法	基于记忆的协同过滤(Memory-based CF)	基于用户(User-based)
		基于物品(Item-based)
	基于模型的协同过滤 <sup>[7]</sup> (Model-based CF)	隐因子模型、朴素贝叶斯等

基于用户的协同过滤推荐算法的计算流程为：为用户 **A** 进行推荐，通过基于用户/物品的协同过滤算法计算与 **A** 最相似的前  $K$  个用户，取前  $TopN$  个物品进行找回去重并排序，最后推荐给 **A** 用户。计算相似度的方法一般有：杰卡德(Jaccard)相似系数、余弦相似度与皮尔逊(Pearson)相关系数等<sup>[8]</sup>。

(1) Jaccard系数是衡量两个集合相似度的指标，计算公式如下：

$$sim_{uu} = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|} \quad (2-1)$$

其中 $N(u), N(v)$ 分别表示用户 $u$ 和用户 $v$ 交互物品的集合，对于用户 $u$ 和 $v$ ，该公式描述了两个用户交互的物品中相同物品的数量与它们总交互物品数量的比例。它是用于处理隐士反馈数据。

(2) Pearson相关系数，计算公式如下：

$$sim(u, v) = \frac{\sum_{i \in I} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{vi} - \bar{r}_v)^2}} \quad (2-2)$$

其中， $r_{ui}, r_{vi}$ 分别表示用户 $u$ 和用户 $v$ 对物品 $i$ 是否有交互（或具体评分值）， $\bar{r}_u, \bar{r}_v$ 分别表示用户 $u$ 和用户 $v$ 交互的所有物品交互数量或者具体评分的平均值。适合判断两变量之间是否同增减。

(3) 余弦相似度：衡量了个向量的夹角，夹角越小越相似。其计算如下：

$$sim_{uu} = \frac{|N(u) \cap N(v)|}{\sqrt{|N(u)| \cdot |N(v)|}} \quad (2-3)$$

适用于度量文本相似度、用户相似度、物品相似度等场景。

本文采取余弦相似度以计算两用户之间的相似程度。

## 3 互联网岗位推荐系统的需求分析

### 3.1 系统可行性分析

基于协同过滤的互联网岗位推荐系统的可行性分析基于当下的互联网时代背景，分别在经济、技术与用户等角度进行调查，以验证本互联网岗位推荐系统的系统可行性，下面分别从上述几个角度进行可行性分析.:

(1) 经济可行性分析：本系统使用开源框架 Scrapy、Django、Vue，使用了阿里云 RDS MySQL Serverless 云数据库试用版，参加了 Pycharm、Webstorm 学生计划，项目的人力成本、软硬件成本、硬件成本与市场推广等开发成本均可忽略不计。总成本远远小于系统带来的社会收益，在经济角度，系统是完全可行的。

(2) 技术可行性分析：在技术层面，技术难点在于网络爬虫获取信息的时效性与真实性、Web 应用的可靠性与稳定性、推荐算法的准确性等。随着时代的发展，上述技术难点都已经发展出了成熟可靠的解决方案，每个模块所用技术在其领域内都已经存在稳定正确的实际部署落地，因此本系统在技术上是可行的。

(3) 用户可行性分析：本系统的目标用户定位为应届生等各类互联网岗位求职者，直接面向用户的是 Web 模块，此模块分为管理员端与普通用户端，以‘设计简洁清晰、操作简单明了’为宗旨，旨在提升用户体验，同时设计意见反馈机制，允许用户通过留言板板块与管理员交互。在用户使用可行性上没有问题。

### 3.2 网络爬虫模块需求分析

从系统功能性的角度出发，网络爬虫模块需要具有以下几点功能：

(1) 信息抓取：网络爬虫模块网络爬虫模块应具备高效的网页抓取功能，能够按照预设的规则和策略从互联网上抓取各类网页内容

(2) 数据处理与数据存储：将信息抓取取得的数据进行清洗和去重，确保数据的准确性和一致性，并将处理完毕的格式化数据存入数据库。

(3) 实时调度：支持即时使用，随时响应 Web 模块的爬虫请求。

### 3.3 Web 模块需求分析

Web 模块是与用户进行直接交互的模块，分为管理员端与普通用户端，下面分别就管理员端与普通用户端进行需求分析。

### 3.3.1 管理员端需求分析

管理员负责管理网站后台，其具备的关键功能应包括：管理员登陆、个人中心、用户管理、招聘信息管理、留言板管理与系统管理，其中系统管理又包括系统简介、轮播图管理、新闻资讯管理与查看系统数据可视化。“管理”应具备对管理目标的基本增删改查功能。**Web 模块**的管理员用例图如下图 3-1：

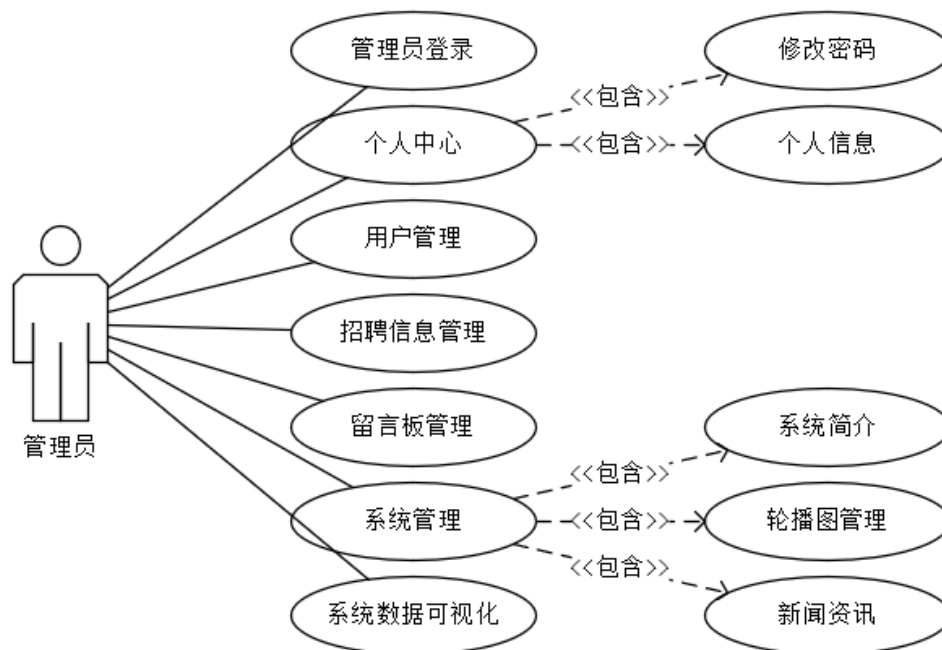


图 3-1 Web 模块管理员用例图

### 3.3.2 普通用户端需求分析

普通用户是系统的主要使用者，普通用户端具备的关键功能应包括：首页、招聘信息、新闻资讯、个人中心。普通用户用例图如下图 3-2：

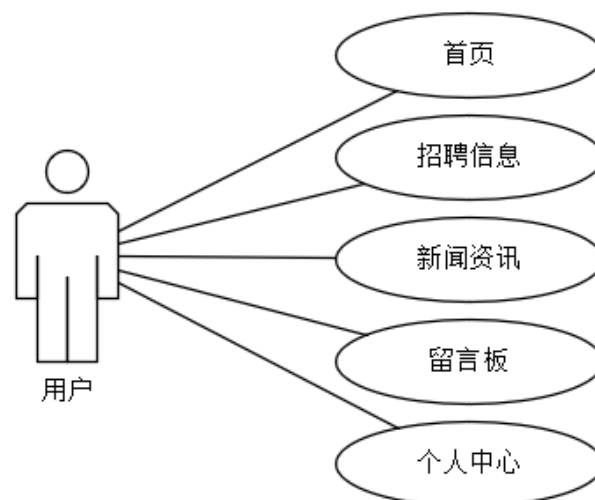


图 3-2 Web 模块用户用例图

3.4 推荐算法模块需求分析

推荐算法模块应能提供个性化、多样化且实时高效的岗位推荐服务。系统能够根据用户的浏览历史等行为数据，为用户推荐相关岗位。推荐结果应能够根据用户最新的行为或岗位的热度动态调整，保持推荐的实时性和准确性。与此同时，推荐算法的实现过程中，需要确保用户数据的安全性和隐私保护，遵守相关法律法规，保护用户个人信息和数据隐私。

3.5 系统数据库需求分析

根据上述对网络爬虫模块、Web 模块与推荐算法模块的需求分析，对系统数据库进行需求分析：在本互联网岗位推荐系统中，存在着六种不同的数据形式：用户信息、管理员信息、招聘信息、新闻资讯、留言板信息与历史浏览信息。

（1）用户信息：用户信息是系统使用者的基本信息，包括用户名、密码、用户姓名、用户性别、头像、邮箱、手机号、用户注册时间。用户信息<sup>[9]</sup>的数据字典如下表 3-1 所示：

表 3-1 用户信息数据字典

用户信息 = {用户名+密码+姓名+性别+头像+邮箱+手机号+注册时间}
用户名 = 密码 = 姓名 = 头像 = 0{字母   十进制数字}255
字母 = "a" .. "z"   "A" .. "Z"
十进制数字 = "0" .. "9"
性别 = [男   女]
邮箱 = 0{字母   十进制数字}255 + "@" + 0{字母   十进制数字}255 + ".com"
手机号 = 11{十进制数字}11
注册时间 = 年 + 月 + 日 + 时 + 分 + 秒
年 = "1970" .. "2099"
月 = "01" .. "12"
日 = "01" .. "31"
时 = "00" .. "23"
分 = 秒 = "00" .. "59"

十进制数字、字母、时间等基本数据类型后续将不再赘述。

（2）管理员信息：管理员信息由管理员账号、密码、用户类型、注册时间

组成，管理员信息的数据字典如下表 3-2 所示：

表 3-2 管理员数据字典

管理员信息 = {账号 + 密码 + 用户类型 + 注册时间}
账号 = 密码 = 0{字母   十进制数字}255
用户类型 = ["role"   "yonghu"]
注册时间 = 年 + 月 + 日 + 时 + 分 + 秒

（3）招聘信息：招聘信息由添加时间、学历要求、公司性质、公司名称、公司规模、发布时间、来源、标题、工作地址、薪资、工作经验、点击时间、点击次数构成，招聘信息数据字典如下表 3-3 所示：

表 3-3 招聘信息数据字典

招聘信息 = {添加时间+学历要求+公司性质+公司名称+公司规模+发布时间+来源+标题+工作地址+薪资+工作经验+点击时间+点击次数}
添加事件 = 发布时间 = 点击时间 = 年 + 月 + 日 + 时 + 分 + 秒
学历要求 = 公司性质 = 公司名称 = 公司规模 = 标题 = 工作地址 = 工作经验 = 0{字母   十进制数字}255
来源 = ["http"   "https"] + "://" + 0{字母}255
薪资 = 点击次数 = 0{十进制数字}255

（4）新闻资讯：新闻资讯由添加时间、标题、介绍、图片、正文构成，新闻资讯数据字典如下表 3-4 所示：

表 3-4 新闻资讯数据字典

新闻资讯 = {添加时间 + 标题 + 介绍 + 图片 + 正文}
添加时间 = 年 + 月 + 日 + 时 + 分 + 秒
标题 = 介绍 = 图片 = 正文 = 0{字母}255

（5）留言板信息：留言板信息由添加时间、留言用户名、留言图片、留言正文、回复内容、回复图片构成，留言板信息数据字典如下表 3-5 所示：

表 3-5 留言板信息数据字典

留言板信息={添加时间+留言用户名+留言图片+留言正文+回复内容+回复图片}
添加时间 = 年 + 月 + 日 + 时 + 分 + 秒
留言用户名=留言图片=留言正文=回复内容=回复图片= 0{字母}255

(6) 历史浏览信息：基于协同过滤算法的需求，历史浏览信息由浏览用户、历史浏览 1、历史浏览 2、历史浏览 3...历史浏览 8 组成，历史浏览信息数据字典如下表 3-6 所示：

表 3-6 历史浏览信息数据字典

历史浏览信息={浏览用户+历史浏览 1+历史浏览 2+历史浏览 3+...+历史浏览 8}
浏览用户 = 历史浏览 = 0{字母   十进制数字}255



## 4 互联网岗位推荐系统的设计与实现

### 4.1 系统的总体功能设计

本文设计与实现的基于协同过滤的互联网岗位推荐系统的主要功能模块分为：网络爬虫模块、Web 模块和推荐算法模块。本系统采用自顶向下、从总体到局部的设计思想，结合第三章中对三大模块的需求分析，确定每个模块在系统中的作用，确定每个模块的具体功能。系统的总体功能模块如下图 4-1 所示：

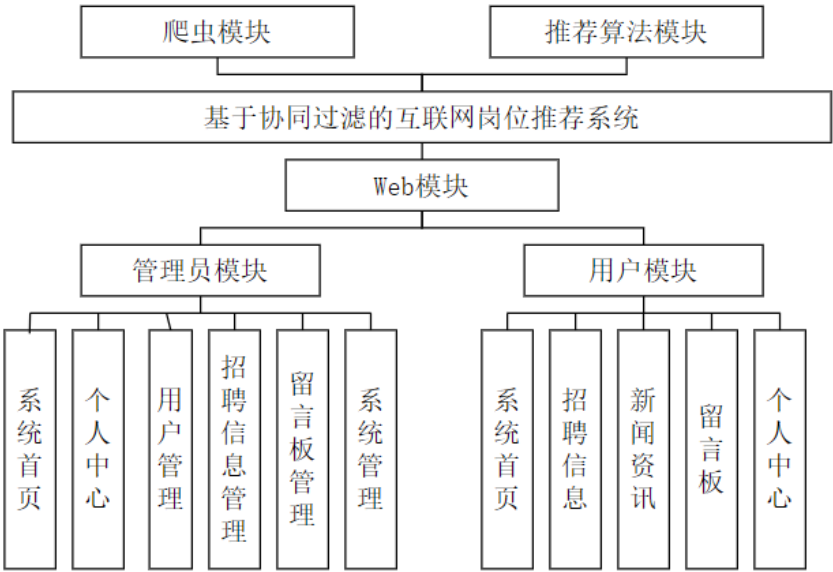


图 4-1 系统总体功能模块图

### 4.2 系统网络爬虫模块的设计与实现

网络爬虫是一种从互联网上抓取和收集网页信息和自动化工具，其工作原理类似于一只“虚拟”的爬行动物，沿着互联网的链接爬行，从一个页面到另一个页面，不断地收集和索引网页内容，为用户提供便捷的信息检索和浏览体验。<sup>[10]</sup> 本系统采用的网络爬虫模块运行流程如下图 4-2 所示：

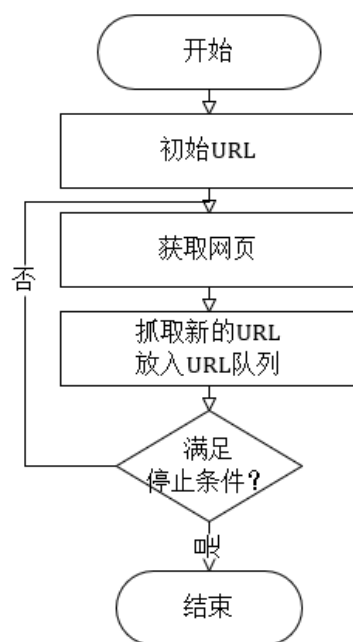


图 4-1 系统总体功能模块图

本系统的网络爬虫模块基于 Scrapy 框架实现，如第二章“网络爬虫模块概述”中所描述，Scrapy 框架主要由 Spider、Item、Pipeline、Downloader 与 Middleware 几个组件构成<sup>[11]</sup>。下面是各个组件的功能介绍以及功能实现。

(1) Item 组件：Item 组件定义了爬虫抓取数据时所使用的数据结构，它充当着保存爬取数据的容器。在本系统中，根据爬虫目标“51 前程无忧网”的信息结构，设计数据结构“ZhaopinxinxiItem”，其中定义了岗位信息字段，包括：添加时间、工作地址、薪资、工作经验、来源、标题、公司名称、公司性质、公司规模、学历要求和信息发布时间。系统将爬取的数据暂时存入“ZhaopinxinxiItem”数据结构，并通过后续操作将数据存入数据库中。

(2) Pipelines 组件：Pipelines 即数据管道，是处理数据的组件，可用于清洗、验证与存储数据。在本系统中设计了“Pipelines.py”文件，实现了如下表 4-1 中所示的方法：

表 4-1 数据管道组件方法

open_spider	Spider 启动时被调用，执行初始化操作。
close_spider	在 Spider 结束时被调用，进行清理操作，如关闭数据库
process_item	每当 Spider 抓取并解析完一个界面后调用一次，进行数据处理
insert_db	自定义函数，将爬取数据插入数据库，被 process_item 调用

其中，“open\_spider”、“close\_spider”与“process\_item”均是 Scrapy 自动触

发的，是框架的内置机制，这也是 Scrapy 框架的便捷之处。

（3）middlewares 组件：middlewares 组件定义了 Spider 中间件（Spider Middleware）和下载器中间件（Downloader Middleware），在本系统中，对应这两个中间件，设计“SpiderSpiderMiddleware”与“SpiderDownloaderMiddleware”类，“SpiderSpiderMiddleware”类定义了处理 Spider 输入和输出的方法。“SpiderDownloaderMiddleware”类定义了处理请求和响应的方法。

（4）Spider 组件：Spider 组件即爬虫主体项目文件，本系统设计了“ZhaopinxinxiSpider.py”，本系统目标在对“51job 前程无忧网”进行爬虫，在“ZhaopinxinxiSpider”类中设计并实现了以下表 4-2 中方法：

表 4-2 Spider 组件设计方法

start_request	爬虫入口类，查看数据暂存表“05zp2_zhaopinxinxi”是否为空，不空则调用 temp_data 方法，否则遍历 start_urls 中每个 URL，创建 Scrapy 请求，开始爬虫。最后再次调用 temp_data 方法。
parse	解析爬虫界面，为每个界面创建“ZhaopinxinxiItem”实例
datail_pares	解析页面详情
remove_html	利用正则表达式去除 HTML 标签
db_connect	连接数据库的方法，被其他方法调用。
table_exists	检查表是否为空
temp_data	将数据暂存表“05zp2_zhaopinxinxi”中数据存入“zhaopinxinxi”表中，检查是否存在相同数据并进行插入

网络爬虫应具有时效性，这意味着爬虫代码将被多次执行，为了避免数据库中出现大量重复数据，设计数据暂存表“05zp2\_zhaopinxinxi”，其详细设计与实现见下“（五）系统数据库的设计与实现”。

### 4.3 系统 Web 模块的设计与实现

系统 Web 模块分为两个部分：Web 前端与 Web 后端。Web 前端是用户直接接触和操作的界面，包括用户界面的设计、内容呈现和交互功能。而 Web 后端则是幕后运行的部分，主要负责处理 Web 应用的业务逻辑、服务端任务并与数据库进行交互。在本文设计与实现的岗位推荐系统中，前端采用 Vue 框架，后端采用 Django 框架，下面分开介绍前端与后端的设计与实现。

4.3.1 基于 Django 的 Web 后端的设计与实现

Django 是一个高级的 Python Web 框架，其遵循 MVC（模型-视图-控制器）设计模式<sup>[12]</sup>。与设计模式对应，Django 分为三层：数据访问层、业务逻辑层与表示层。下面分别就三个不同层次介绍本系统 Web 后端的设计与实现。

（1）数据访问层：负责处理所有与数据相关的任务，包括：定义数据模型、设置数据库接口及配置数据库信息等操作。下面分别就这三种操作进行介绍。

① 定义数据载体：定义数据在前后端流转时，数据在后端的载体。在 Django 框架中，数据载体的设计是：继承 django.db 提供的子类 models.Model。在数据在前后端流转的过程中，数据的基本形式不变，这意味着数据载体形式与数据库列名完全一致，因此数据载体详细信息见本章第五节“系统数据库的设计与实现”。数据载体的基本定义如下表 4-3 所示：

表 4-3 Web 后端数据载体

user_model.py	管理员信息模型
config_model.py	系统介绍轮播图信息模型
models.py	模型集合类，定义了：普通用户、招聘信息、新闻与系统介绍的信息模型，其中“系统介绍”是前端模块中，数据可视化界面的界面介绍。

② 定义数据库接口：定义数据库接口指的是定义后端对数据库的增删改查等操作接口，将数据库操作与后端业务逻辑分开处理，抽象出 model.py 文件，定义 BaseModel 类，定义相应接口，在业务逻辑层涉及到数据库相关操作时，可以直接调用相关函数。BaseModel 类定义的主要方法如下表 4-4 所示：

表 4-4 BaseModel 所定义函数

retrieve	获取所有数据
page	分页获取数据
getbyColumn	获取某个字段的所有不重复值
createbyreq	根据请求参数创建新的记录
getbetweenparams	获取某个字段在某个范围内的所有数据
updatebyparams	根据参数更新数据
deletes	根据 id 列表删除数据
getallcolumn	获取模型的所有字段

③ 配置数据库信息：设计 `config.ini` 文件，写入数据库配置信息，详细数据库配置见本章第五节“系统数据库的设计与实现”。

(2) 业务逻辑层：在本文设计与实现的互联网岗位推荐系统中，存在用户、管理员、招聘信息、系统介绍、新闻资讯、系统轮播图和留言板对象，每个对象都有对应的功能，根据业务逻辑需求，分别对应的设计了：`Yonghu_v.py`、`user_v.py`、`Zhaopinxinxi_v.py`、`systeminfo_v.py`、`news_v.py`、`config_v.py`、`messages_v.py`，下面分别就每个对象对应的业务逻辑的设计与实现做介绍。

① 用户：设计 `Yonghu_v.py` 文件，用户的业务需求有：用户注册、用户登录、用户登出、用户修改密码、处理用户详细信息、更新用户信息、删除用户信息等，详细实现的业务逻辑接口见下表 4-5：

表 4-5 用户对象的业务逻辑接口

yonghu_register	处理用户注册请求
yonghu_login	处理用户登录请求
yonghu_logout	处理用户登出请求
yonghu_resetPass	处理用户重置密码请求
yonghu_session	处理与用户会话相关的请求
yonghu_list	处理前台分页请求，在管理员界面查看所有用户信息
yonghu_save	处理后台新增请求
yonghu_add	处理前台新增请求
yonghu_detail	处理获取用户详细信息的请求

② 管理员：管理员的业务需求与用户的需求并无本质的差别，管理员只是具有权限的用户，二者在业务逻辑层面并无显著区别，这里不再赘述。

③ 招聘信息、系统介绍、新闻资讯、系统轮播图和留言板对象：同管理员与普通用户的关系类似，上述对象本身都是以不同组织方式存储的不同信息，归根结底都是信息，在业务逻辑层面并无显著区别，这里以招聘信息对象为例，招聘信息对应的业务逻辑接口设计见下表 4-6

表 4-6 招聘信息对象的业务逻辑接口

zhaopinxinxi_page	处理招聘信息页面的请求。根据会话中参数获取分页数据
zhaopinxinxi_autoSort	基于热度的智能推荐，实现见本章第四节。

续表 4-6 招聘信息对象的业务逻辑接口

zhaopinxxi_list	处理前台分页的请求。
zhaopinxxi_add	处理前台新增的请求
zhaopinxxi_update	新增招聘信息
zhaopinxxi_delete	批量删除招聘信息
zhaopinxxi_autoSort2	基于协同过滤的推荐，实现见本章第四节

（3）表示层：Django 提供简单的表示层实现，而在本系统中，基于 Vue 框架开发前端，因此后端表示层中只定义了前后端相关接口：urls.py 与 views.py，下面对表示层接口分别介绍。

① views.py: views.py 是 Django 框架中的一个核心组件，它承担着处理用户请求和返回响应的任务。在 views.py 中，可以定义多种视图函数，每个函数对应着一个或多个 URL 路由。实际上，views 是后端对前端的请求可视化。针对前端的所有请求，设计相应的方法。如：“index(request)”方法处理对/index/的请求，“check\_suffix(fileName,path1)”方法根据文件名的后缀返回不同类型的 HTTP 响应。views.py 在系统中的运行效果如下图 4-2：

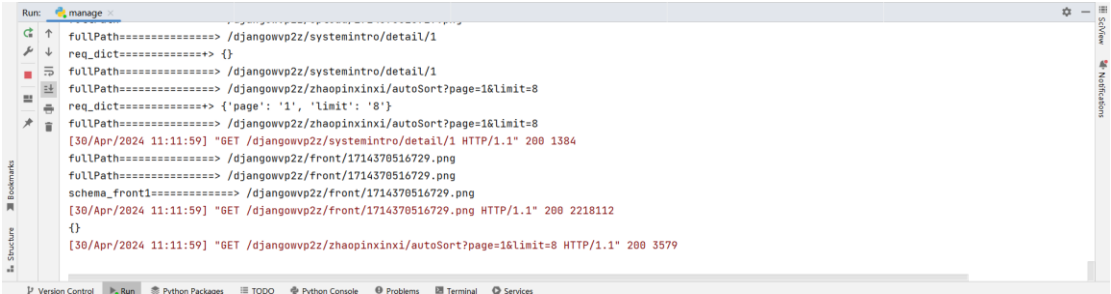


图 4-2 views.py 运行效果图

② urls.py: urls.py 文件是 URL 配置文件，它的主要作用是将 URL 路径映射到相应地视图函数上，当 Django 收到一个 HTTP 请求时，其会首先检查 urls.py，根据请求的 URL 路径找到对应的视图函数。在本系统中，包含两个 urls.py 分别是全局 URL 映射与其他 URL 映射。部分 URL 映射如下图 4-3 所示：

```
11 urlpatterns = [  
12     path(r'users/register', users_v.users_register),  
13     path(r'users/login', users_v.users_login),  
14     path(r'users/logout', users_v.users_logout),  
15     path(r'users/session', users_v.users_session),  
16     path(r'users/page', users_v.users_page),  
]
```

图 4-3 部分 URL 映射

(4)其他文件: 其余文件有: settings.py wsgi.py xmiddleware init.py manage.py admin.py, 基础配置文件 settings.py 都按照 Django 框架提供的格式书写, 这里不再赘述, 下面分别介绍这些文件的作用。

① wsgi.py: wsgi 是 Web 服务器网关接口 (Web Server Gateway Interface) 的配置文件, 定义 Python web 应用和 Web 服务器之间的标准接口。

② xmiddleware: xmiddleware 是中间件模块, 由 Django 框架自行调用, 在文件中设计 xauth.py、xparam.py 文件, 分别起到了处理用户认证、处理和修改 HTTP 请求的作用。

③ admin.py: admin.py 文件是 Django 框架中的一个重要组成部分, 它用于配置 Django 的自动管理界面。在本文件中, 调用 django 内置的“get\_app\_config”方法, 将主文件中所有模型在 Django 中进行注册。

### 4.3.2 基于 Vue 的 Web 前端的设计与实现

本系统采用 Vue 框架设计 Web 模块的前端。据本文第三章第三节“Web 模块需求分析”, Web 前端分为管理员端和普通用户端, 为方便管理员分析岗位数据信息, 从而更好的对系统进行维护, 系统还设计了数据可视化界面。下面分别就普通用户端、管理员端与数据可视化界面进行介绍。

(1) 普通用户端 (前台): 前台是普通用户注册后能够直接看见的界面, 根据第三章第三节第二小节“普通用户端需求分析”, 普通用户端应具备的关键功能应包括: 首页、招聘信息、新闻资讯、个人中心。下面分别介绍这几个部分的设计与实现。

① 首页应包括对其他功能 (详细招聘信息、新闻资讯、留言板、个人中心) 的链接跳转, 同时首页可以显示推荐算法给出招聘岗位信息推荐, 同时为用户提供跳转功能, 方便用户浏览详细信息。在首页下方, 用户还可以查看新闻咨询信息。根据 Vue 框架的结构, 设计 home.vue。首页运行状态如下图 4-4、4-5、4-6:



图 4-4 登陆界面



图 4-5 系统首页（1）



图 4-6 系统首页（2）

② 招聘信息：点击首页招聘信息推荐的“查看更多”或首页上部导航栏的“招聘信息”，会进入招聘信息界面。招聘信息前端设计 `add.vue`、`detail.vue`、`list.vue`，其中 `add.vue` 用于后台，`detail.vue` 负责用户查看详细招聘信息，`list.vue` 绘制招聘信息界面的招聘信息显示框架，其中，`list.vue` 绘制出的框架是链接形式的，点击即可直接查看详细招聘信息，点击则进入 `detail.vue` 设计的。此外，详细招聘信息中的“来源”一栏，指的是招聘信息的发布界面，同样是链接形式，点击即可直接进入信息发布网站。详细设计如下图 4-7、4-8 所示：





图 4-7 招聘信息



图 4-8 详细招聘信息

③ 新闻资讯：在首页下方可以查看到新闻资讯，新闻资讯可以帮助使用者了解互联网招聘行情，新闻资讯的查看设计了 newsdetail.vue 与 news-list.vue，查看详细新闻界面的实现如下图 4-9 所示：



图 4-9 详细新闻资讯

④ 留言板：为方便系统的升级维护，使管理员与普通用户有交流的通道，设计留言板模块 `list.vue`。由用户留言。管理员进行回复，由后端进行权限验证，普通用户只有发布留言的权利，而留言的回复、修改、删除均在管理员端进行修改。留言板板块的详细设计与实现见下图 4-10：

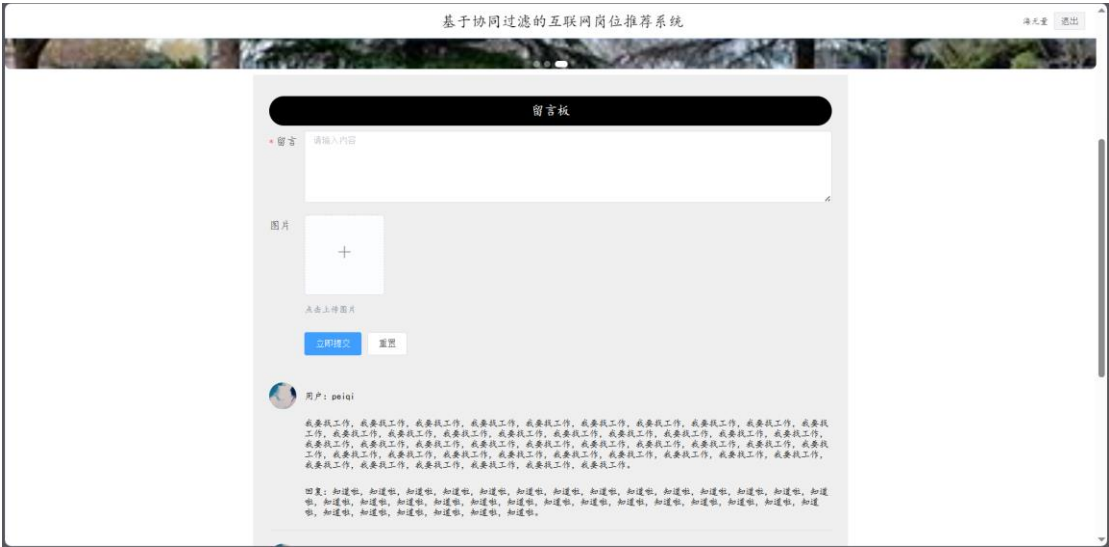


图 4-10 留言板

⑤ 个人中心：为方便用户修改该个人信息，设置个人中心板块，在个人中心板块中，用户可以查看并修改自己的所有个人数据，包括用户名、密码、姓名、性别、头像、邮箱与手机号。其中头像支持在本地图片中选择，系统会将图片自动下载至数据库并保存。个人中心的详细设计与实现见下图 4-11：

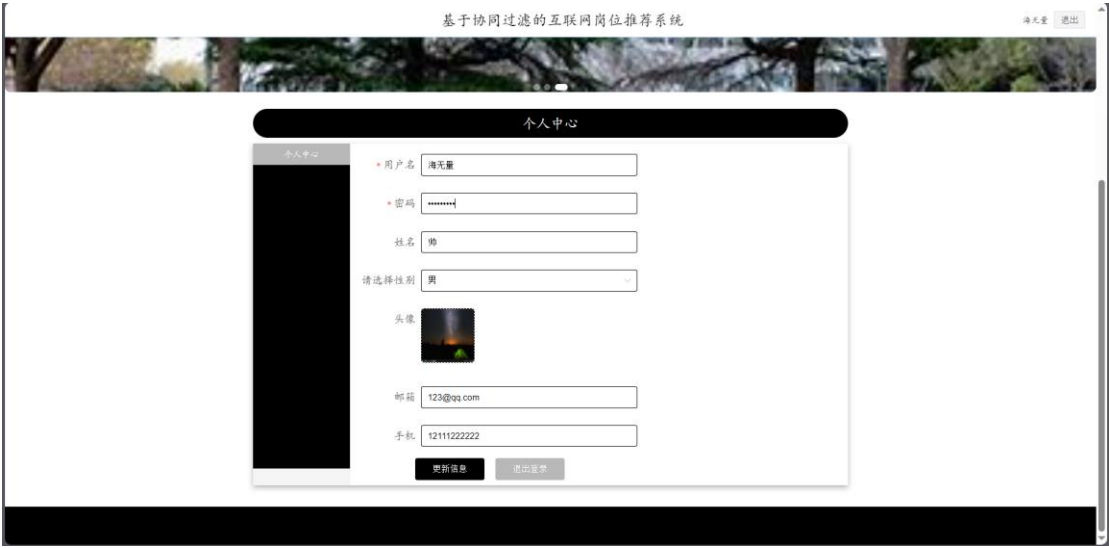


图 4-11 个人中心

(2) 管理员端（后台）：根据第三章第三节第一小节“管理员端需求分析”，管理员负责管理网站后台，其具备的关键功能应包括：管理员登陆、个人中心、用户管理、招聘信息管理、留言板管理与系统管理，其中系统管理又包括系统简介、新闻资讯管理、轮播图管理与查看系统数据可视化。下面分别介绍各功能的详细介绍与实现。

①系统首页：为保证系统安全性，管理员账号等信息只能通过直接对数据库进行操作来添加，测试数据为管理员账号密码为：“admin”、“admin”。管理员登陆界面以及登陆成功进入系统首页界面如下图 4-12、4-13 所示：



图 4-12 管理员登陆



图 4-13 管理员端系统首页

② 个人中心：为处理管理员职位人员更换的情况，允许管理员有权限对自己的账户信息进行修改，但修改密码时必须输入原密码，不提供除此之外的任何认证手段。个人中心功能简单这里不再予以展示。

③用户管理：管理员可以对用户信息进行查询、修改、删除。详细设计与实现见下图 4-14：



图 4-14 用户管理界面

④招聘信息管理：管理员可以对招聘信息进行查询、修改、删除还可以调用爬虫进行招聘信息获取。详细设计与实现见下图 4-15：



图 4-15 招聘信息管理

④留言板管理：管理员可以对留言板进行查询、修改、删除与回复。设计与用户管理、招聘信息管理类似，这里不再赘述。

⑤系统管理：系统管理包括系统简介、轮播图管理、新闻资讯管理。这三种管理界面与用户管理、招聘信息管理本质相同，不过是换了数据对象，这里也不再赘述。

(3) 数据可视化界面：为方便管理员分析岗位数据信息，从而更好的对系统进行维护，系统设计了数据可视化界面。可视化界面的设计与实现见下图 4-16：



图 4-15 数据可视化界面

## 4.4 系统推荐算法模块的设计与实现

系统推荐算法模块实现集成在了系统后端中，表现为 Zhaopinxinxi\_v.py 的 autoSort 与 autoSort2 方法，算法在前端请求招聘信息时自动执行，将后端返回的信息进行排序后再返回给前端，基于对互联网岗位推荐系统的需求分析，系统推荐算法模块设计了基于热度的推荐与基于用户的协同过滤推荐。下面分别介绍这两种推荐方法的设计与实现。

### 4.4.1 基于热度的推荐

基于热度的推荐主要实现依赖于对各岗位的热度了解，在本系统将热度量化为点击次数<sup>[13]</sup>，将其作为表“zhaopinxinxi”的一个字段“clicknum”，在基于热度的推荐中，首先检查“clicknum”字段是否存在，存在则将其作为排序字段令 Django 自行排序，将排序方式设置为降序并返回排序后的结果给前端。其中“clicknum”字段随着“ZhaopinxinxiDetail”、“ZhaopinxinxiInfo”的调用而递增。基于热度的推荐详细设计与实现见下图 4-16：

```

237 def zhaopinxinxi_autoSort(request):
238     ...
239     . 智能推荐功能(属性: [IntelRecom (是/否)], 新增clicktime[前端不显示该字段]字段(调用info/detail接口的时候更新), 按clicktime排序查询)
240     主要信息列表(如商品列表, 新闻列表)中使用, 显示最近点击的或最新添加的5条记录就行
241     ...
242     if request.method in ["POST", "GET"]:
243         msg = {"code": normal_code, "msg": mes.normal_code, "data":{"currPage":1,"totalPage":1,"total":1,"pageSize":10,"list":{}}}
244         req_dict = request.session.get("req_dict")
245         if "clicknum" in zhaopinxinxi.getallcolumn(zhaopinxinxi, zhaopinxinxi):
246             req_dict['sort']='clicknum'
247         elif "browseduration" in zhaopinxinxi.getallcolumn(zhaopinxinxi, zhaopinxinxi):
248             req_dict['sort']='browseduration'
249         else:
250             req_dict['sort']='clicktime'
251         req_dict['order']='desc'
252         msg['data']['list'], msg['data']['currPage'], msg['data']['totalPage'], msg['data']['total'], \
253         msg['data']['pageSize'] = zhaopinxinxi.page(zhaopinxinxi, zhaopinxinxi, req_dict)
254
255     return JsonResponse(msg)
256

```

图 4-16 基于热度的推荐

#### 4.4.2 基于用户的协同过滤推荐

基于对推荐算法模块的需求分析,选择基于用户的协同过滤算法。基于用户的协同过滤算法的原理是:根据用户的历史浏览信息,通过余弦相似度计算用户之间的相似度表格,从前端获取当前用户 ID,排序出与当前用户 ID 最相似的用户,将推荐算法得出的用户的历史浏览记录推荐给当前用户。

设计“zhaopinxinxiRecomm\_calculate”函数,以实现上述算法,用户的历史浏览记录存储在表“zhaopinxinxirecomm”中,根据表中信息计算用户相似度,存至列表“list\_i”,调用“yonghuld()”函数,根据当前用户 ID,对列表做排序,查询出最大相似度的用户,并进行数据库查询,找出该用户的历史浏览记录,并最终进行推荐。

用户历史浏览信息的记录依赖于函数“zhaopinxinxi\_detail”,其功能是显示详细招聘信息,在用户点击时被调用。当用户点击时,系统自动获取当前使用者 ID,并利用此 ID 修改表“zhaopinxinxirecomm”及历史浏览数据。

### 4.5 系统数据库的设计与实现

本文设计与实现的互联网岗位推荐系统采用 MySQL 数据库,MySQL 是一种流行的关系型数据库管理系统(RDBMS),它功能强大、性能优越且易于使用,被广泛应用于 Web 应用程序和其他需要大规模数据存储和管理的场景。下面分别说明系统数据库的概念结构设计和物理结构设计。

#### 4.5.1 数据库概念结构设计

根据第三章第五节“系统数据库需求分析”得出的结论,在本系统中,存在六种不同的数据形式,分别是用户信息、管理员信息、招聘信息、新闻资讯、留

言板信息、历史浏览信息。下面分别介绍每种数据形式的概念结构设计。

（1）用户信息：用户信息是用户进入网站的凭证，与此同时，在本系统中用户的身份是求职者，用户信息还应包括用户的个人基本信息，综上所述，绘制出用户实体图如下图 4-17 所示：

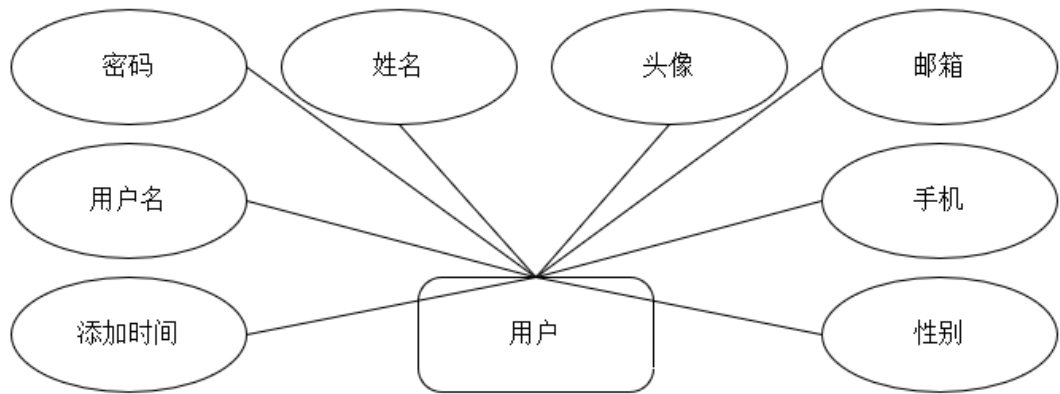


图 4-17 用户实体图

（2）管理员信息：管理员信息是管理员登陆后台的凭证，管理员信息应包括：管理员名、密码、管理员职能、管理员账户添加时间，管理员信息实体图如下图 4-18 所示：

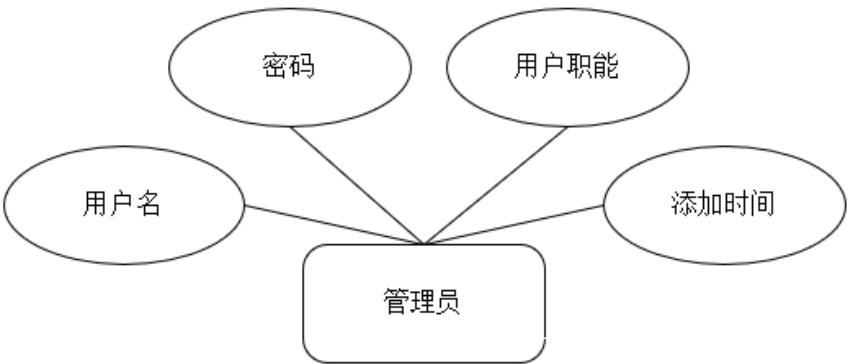


图 4-18 管理员实体图

（3）招聘信息：根据对数据库的需求分析，招聘信息应包含：添加时间、学历要求、公司性质、公司名称、公司规模、发布时间、来源、标题、工作地址、薪资、工作经验、点击时间、点击次数，招聘信息实体图如下图 4-19 所示：

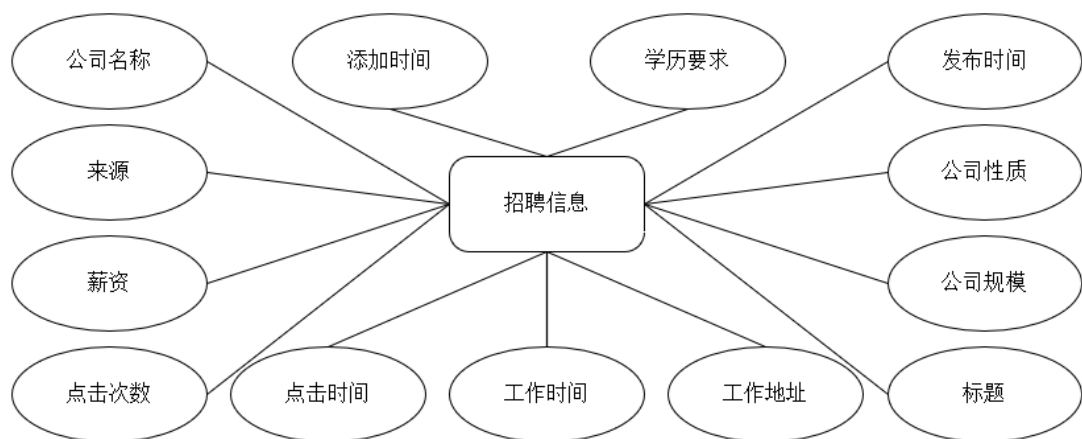


图 4-19 招聘信息实体图

(4) 新闻资讯：新闻资讯由添加时间、标题、介绍、图片、正文构成，新闻资讯的实体图如下图 4-20 所示：

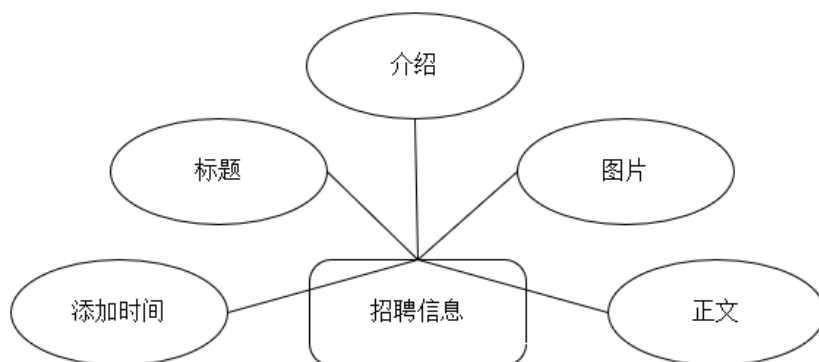


图 4-20 新闻资讯实体图

(5) 留言板信息：留言板信息由添加时间、留言用户名、留言图片、留言正文、回复内容、回复图片构成，留言板的实体图如下图 4-21 所示：

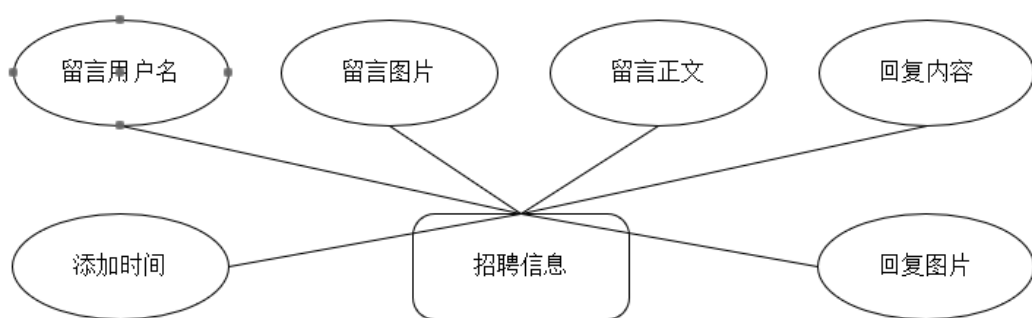


图 4-21 留言板实体图

(6) 历史浏览信息：历史浏览信息由浏览用户、浏览历史 1、浏览历史 2、浏览历史 3...浏览历史 8 构成，历史浏览信息的实体图如下图 4-22 所示：



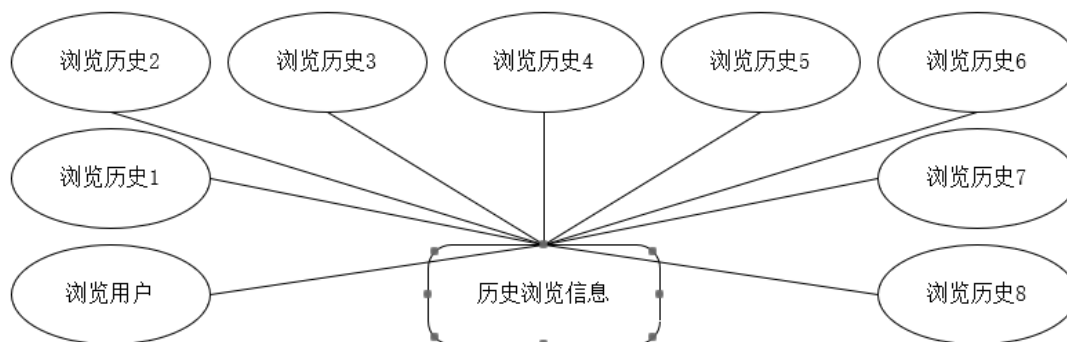


图 4-22 历史浏览信息实体图

(7) 其他信息：数据可视化介绍信息与招聘信息暂存信息，其中数据可视化介绍信息只且只有一条，由管理员直接在数据库中修改，招聘信息暂存信息是为了爬虫时不让招聘信息库中出现重复数据而设立的，因此其结构与招聘信息表完全一致，这里不再赘述。

系统数据库总体实体关系图如下图 4-23 所示：

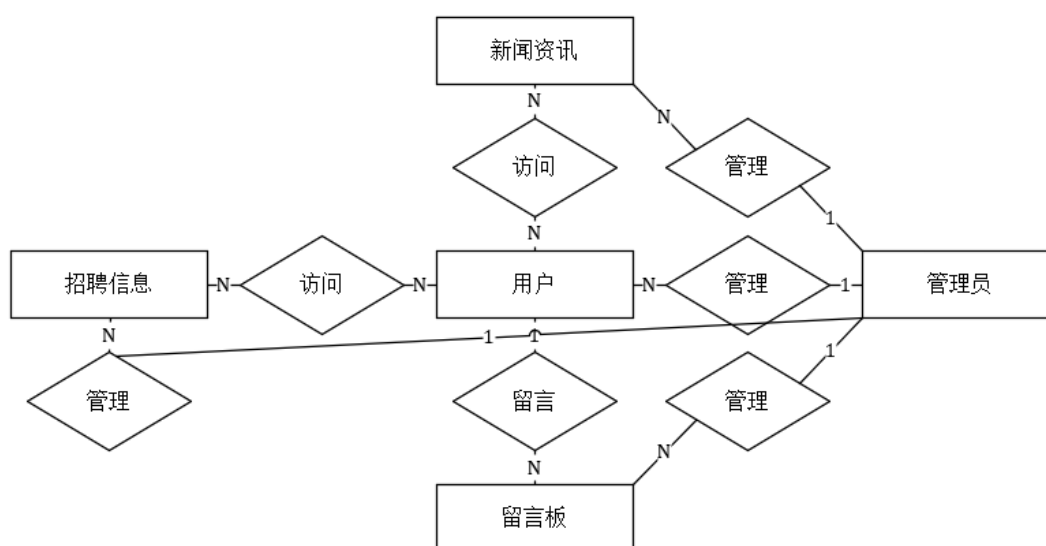


图 4-23 系统总体实体图

#### 4.5.2 数据库物理结构设计

基于系统数据库需求分析与数据库概念结构设计，创建数据库物理结构如下：

(1) 用户信息表：

表 4-7 用户信息表

用户信息表				
列名	数据类型	描述	是否为空	是否主键
id	bigint	用户编号	No	Yes

续表 4-7

addtime	timestamp	创建时间	No	No
yonghuming	varchar	用户名	No	No
mima	varchar	密码	No	No
xingming	varchar	姓名	Yes	No
xingbie	varchar	性别	Yes	No
touxiang	longtext	头像	Yes	No
shouji	varchar	邮箱	Yes	No

(2) 招聘信息表:

表 4-8 招聘信息表

招聘信息表				
列名	数据类型	描述	是否为空	是否主键
id	bigint	招聘信息编号	No	Yes
addtime	timestamp	创建时间	No	No
xlyq	varchar	学历要求	Yes	No
gsxz	varchar	公司性质	Yes	No
gsmc	varchar	公司名称	Yes	No
gsgm	varchar	公司规模	Yes	No
fabushijian	varchar	发布时间	Yes	No
laiyuan	varchar	来源	Yes	No
biaoti	varchar	标题	No	No
gzdz	varchar	工作地址	Yes	No
xinzi	varchar	薪资	Yes	No
gzjy	varchar	工作经验	Yes	No
clicktime	datetime	最近点击时间	Yes	No
clicknum	int	点击次数	Yes	No

(3) 管理员信息:

表 4-9 管理员信息表

管理员信息表
--------

续表 4-9

列名	数据类型	描述	是否为空	是否主键
id	bigint	管理员编号	No	Yes
username	varchar	用户名	No	No
password	varchar	密码	No	No
role	varchar	角色	Yes	No
addtime	timestamp	添加时间	No	No

(4) 新闻资讯信息:

表 4-10 新闻资讯表

新闻资讯表				
列名	数据类型	描述	是否为空	是否主键
id	bigint	新闻编号	No	Yes
addtime	timestamp	创建时间	No	No
title	varchar	标题	No	No
introduction	longtext	简介	Yes	No
picture	longtext	图片	No	No
content	longtext	内容	No	No

(5) 留言板

表 4-11 留言板表

留言板表				
列名	数据类型	描述	是否为空	是否主键
id	bigint	管理员编号	No	Yes
addtime	timestamp	创建时间	No	No
userid	bigint	留言人 id	No	No
username	varchar	用户名	Yes	No
avatarurl	longtext	头像	Yes	No
content	longtext	留言内容	No	No
cpicture	longtext	留言图片	Yes	No
reply	longtext	回复内容	Yes	No

续表 4-11

rpicture	longtext	回复图片	Yes	No
----------	----------	------	-----	----

(6) 数据可视化界面介绍信息表:

表 4-12 数据可视化界面介绍信息表

数据可视化界面介绍信息表				
列名	数据类型	描述	是否为空	是否主键
id	bigint	信息编号	No	Yes
addtime	timestamp	创建时间	No	No
title	varchar	标题	No	No
subtitle	varchar	副标题	Yes	No
content	longtext	内容	No	No
picture1	longtext	图片 1	Yes	No
picture2	longtext	图片 2	Yes	No
picture3	longtext	图片 3	Yes	No

(7) 用户浏览信息表:

表 4-13 用户浏览信息表

用户浏览信息表				
列名	数据类型	描述	是否为空	是否主键
id	bigint	信息编号	No	Yes
username	varchar	用户名	No	No
Like1	varchar	浏览记录 1	Yes	No
Like2	varchar	浏览记录 2	Yes	No
Like3	varchar	浏览记录 3	Yes	No
⋮	⋮	⋮	⋮	⋮
Like8	varchar	浏览记录 8	Yes	No

为提高系统稳定性、扩展性与灵活性,并进一步保障用户的数据安全,本系统采用了阿里云 RDS 云数据库。阿里云关系型数据库 RDS (Relational Database Service) 是一种稳定可靠、可弹性伸缩的在线数据库服务<sup>[14]</sup>。

RDS MySQL 云数据库的参数配置如下表 4-13 所示:

表 4-13 云数据库参数配置

参数类型	参数值
实例名称	rm-cn-adn3prwom0001m
地域及可用区	华东 2（上海） 可用区 M
网络类型	专有网络
系统数据库字符集	Chinese_PRC_CI_AS
数据库类型	MySQL5.7
数据库内存	[1024 M - 4096 M]

云数据库连接参数如下表 4-14 所示：

表 4-14 云数据库连接参数

参数类型	参数值
网络类型	专有网络（ VPC： - / vpc-uf67y4bzwh9jovf4egs7u， 网 段：172. 16. 0. 0/12 ）
内网连接方式	rm-cn-adn3prwom0001m.rwlb.rds.aliyuncs.com,3306
外网连接方式	rm-cn-adn3prwom0001mo.rwlb.rds.aliyuncs.com,3306

## 5 系统运行测试及分析

### 5.1 网络爬虫模块功能测试

网络爬虫功能测试主要验证爬虫模块能否正常运行，爬虫模块运行过程与结果如下图 5-1、5-2 所示：

```
2024-04-30 16:19:06 [scrapy.core.engine] INFO: Spider opened
2024-04-30 16:19:07 [scrapy.extensions.logstats] INFO: Crawled 0 pages (at 0 pages/min), scraped 0 items (at 0 items/min)
2024-04-30 16:19:07 [scrapy.extensions.telnet] INFO: Telnet console listening on 127.0.0.1:6023
2024-04-30 16:19:07 [scrapy.core.engine] INFO: Closing spider (finished)
2024-04-30 16:19:07 [scrapy.statscollectors] INFO: Dumping Scrapy stats:
{'elapsed_time_seconds': 0.107863,
 'finish_reason': 'finished',
 'finish_time': datetime.datetime(2024, 4, 30, 8, 19, 7, 123194),
 'log_count/DEBUG': 1,
 'log_count/INFO': 10,
 'start_time': datetime.datetime(2024, 4, 30, 8, 19, 7, 15331)}
2024-04-30 16:19:07 [scrapy.core.engine] INFO: Spider closed (finished)
```

图 5-1 爬虫模块运行图

序号	姓名	性别	年龄	学历	公司	职位	入职时间	联系电话	电子邮箱	所在城市	薪资范围	工作年限	期望薪资	期望职位	期望城市	操作
6	本科	男	25	本科	无业	无业	2023-04-03 0 03:57:20	https://www.rv.cn/	游戏开发	上海市	8千-1.3万	5-7年	1	详情	修改	删除
7	本科	男	25	本科	无业	无业	2023-09-29 17:09:33	https://www.yan.cn/	信息安全 工程师	北京市	2.5-5万	5-7年	4	详情	修改	删除
8	中技/中专	男	25	中技/中专	无业	无业	2022-11-05 08:24:54	https://www.na.cn/	数据运营 运营	北京市	8千-1.5万 + 13薪	3-4年	4	详情	修改	删除
9	本科	男	25	本科	无业	无业	2022-11-05 09:07:20	https://www.ss.me.cn/	系统分析 员	北京市	7千-1万	3年	1	详情	修改	删除
10	本科	男	25	本科	无业	无业	2023-03-05 17:28:34	https://www.l.cn/	信息安全 工程师	上海市	4-8千	2年	4	详情	修改	删除

图 5-2 爬虫模块运行结果图

### 5.2 Web 模块功能测试

#### 5.2.1 用户操作测试

对用户所有可能的操作进行测试，包括登录测试、留言测试、招聘信息测试、修改个人信息测试<sup>[15]</sup>，测试结果如下表 5-1 所示。

表 4-11 用户操作测试

测试序号	测试功能	输入/操作	输出/结果	测试说明
001	登录测试	1. 输入正确的用户名密码 2. 输入错误的用户名密码 3. 不输入用户名/密码	1. 登录成功 2. 登录失败 3. 请输入用户名密码	系统给出了设定好的提示。 测试通过
002	登出测试	1.退出登录	1.退出成功	测试通过
003	留言测试	1.正常留言 2.不输入内容	1.留言成功 2.请输入内容	系统给出了设定好的提示。

续表 4-11

004	招聘信息测试	1.正常搜索 2.按学历要求搜索 3.按公司名称搜索 4.按标题搜索 5.按工作地址搜索 6.不输入内容	1.搜索成功 2.搜索成功 3.搜索成功 4.搜索成功 5.搜索成功 6.不做反应	系统给出了设定好的提示。 测试通过
005	修改个人信息测试	1.正常更改 2.不输入密码	1.更改成功 2.请输入密码	系统给出了设定好的提示。 测试通过
006	新闻咨询测试	1.查询新闻咨询	1.查询成功	系统给出了设定好的提示。 测试通过
007	注册测试	1.按规则输入 2.不输入密码 3.两次输入密码不一致 4.不输入姓名 5.不按规律输入邮箱 6.输入错误位的手机号	1.注册成功 2.请输入密码 3.密码不一致 4.请输入姓名 5.邮箱格式错误 6.手机号格式错误	系统给出了设定好的提示。 测试通过

### 5.2.2 管理员操作测试

对管理员所有可能的操作做测试，包括登录测试、修改密码测试、用户管理测试、招聘信息管理测试、留言板管理测试、系统管理测试。

表 4-12 管理员操作测试

测试序号	测试功能	输入/操作	输出/结果	测试说明
001	登录测试	1. 输入正确的用户名密码 2. 输入错误的用户名密码 3. 不输入用户名/密码	1. 登录成功 2. 登录失败 3. 请输入用户名密码	系统给出了设定好的提示。 测试通过
002	修改密码测试	1.正常修改 2.不输入原密码 3.不输入新密码	1.修改成功 2.请输入原密码 3.请输入新密码	系统给出了设定好的提示。 测试通过

续表 4-12

003	用户管理 测试	1.正常查询/删除 2.不选中目标删除	1.查询/删除成功 2.删除失败	系统给出了设定好的提示。 测试通过
004	招聘信息 管理测试	1.正常查询/删除 2.不选中目标删除	1.查询/删除成功 2.删除失败	系统给出了设定好的提示。 测试通过
005	留言板管理 测试	1.正常查询/删除 2.不选中目标删除 3.留言空白	1.查询/删除成功 2.删除失败 3.允许空白留言	系统给出了设定好的提示。 测试通过
006	系统管理 测试	1.正常查询/删除 2.不选中目标删除	1.查询/删除成功 2.删除失败	测试通过
007	系统简介图 管理	1.正常选择图片 1-3 2.只选择一个图片 3.一个图片都不选	1.选择成功 2.选择成功 3.请选择图片	系统给出了设定好的提示。 测试通过
008	系统轮播图 管理	1.正常选择图片 1-3 2.只选择一个图片 3.一个图片都不选	1.选择成功 2.选择成功 3.请选择图片	系统给出了设定好的提示。 测试通过
009	新闻咨询 管理测试	1.正常查询/删除 2.不选中目标删除	1.查询/删除成功 2.删除失败	系统给出了设定好的提示。 测试通过

### 5.3 推荐算法模块功能测试

根据基于用户的协同过滤算法的设计与实现,不同用户的推荐结果取决于用户之间的相似度,选取随机用户 1、用户 2。观察其推荐结果是否相同,不相同即为推荐成功。用户 1、用户 2 的推荐结果如下图 5-3、5-4:





图 5-3 用户 1 推荐结果图



图 5-4 用户 2 推荐结果图

系统响应符合预期。测试通过

## 6 结论

本文设计与实现了基于协同过滤的互联网岗位推荐系统，在技术学习、需求分析、系统开发与系统测试后，可以得出结论，本系统在提高招聘效率、优化求职者体验和增加匹配度方面具有显著的潜力和优势。通过爬虫模块，系统能够实时地从互联网上获取最新的岗位信息，确保推荐系统中的岗位数据与实际情况保持同步，从而提高了推荐的全面性和准确性。同时，**Web** 模块为用户提供了友好的界面和交互平台，使用户可以轻松地搜索、浏览和申请岗位，从而增强了用户体验和参与度。给用户带来流畅舒心的用户体验。

网络爬虫模块、**Web** 模块与推荐算法模块共同作用，能够更加全面地满足用户的需求，并提供更加个性化、精准的岗位推荐服务，能够帮助用户更快地找到理想的工作机会，提高了招聘效率和质量。

然而，尽管基于协同过滤的互联网岗位推荐系统在理论上和实践中都表现出了良好的效果，但仍然存在一些挑战和改进的空间。例如，爬虫模块需要不断地优化和更新以应对网页结构的变化和反爬虫机制的挑战，目标网站的 **URL** 规则变化很快；**Web** 模块需要不断地改进和优化用户界面和交互设计，以提高用户体验和满意度；用户数据的稀疏性和冷启动问题可能会影响推荐的准确性和覆盖范围，需要进一步的研究和改进算法来解决这些问题。

综上所述，虽然仍有许多不足，但本文设计与实现的基于协同过滤的互联网岗位推荐系统较好地完成了其设计目标，系统运行稳定且流畅，能够帮助求职者更快更好的找到工作。

## 致谢

在这篇论文的最后，我想表达我最深的感激之情，我的毕业设计的完成过程中充满了挑战和收获，我想感谢所有在这个过程中支持和鼓励我的人们。

首先，我要衷心的感谢我的指导教师石林祥老师。在整个研究过程中，您的知识、指导与悉心关怀是我不断成长的动力源泉，您的耐心指导和专业建议让我受益匪浅。您不仅是我学业上的良师益友，更是我人生中的贴心朋友。您给予我的极大的信任和支持。我由衷感谢您为我付出的一切，您的教诲将铭刻在我心中，时刻激励着我勇往直前，追求卓越。

我还要感谢我的同学们，有你们的陪伴和支持，让我感到无比幸运和欣慰。在学术上，我们相互学习、相互探讨，共同成长。在生活中，你们总是给予我无私的支持。

最后，我想特别感谢我的家人朋友，他们是我在漫漫研究路上的坚强后盾，始终支持和鼓励着我。

在此，我再次表达我最诚挚的感谢和敬意。感谢每一位对我提供过帮助的人，你们的影响和帮助将永远铭刻在我的心中。

## 参考文献

- [1] 曹培, 林永意. 基于 Scrapy 的岗位推荐系统的设计与实现 [J]. 无线互联科技, 2023, 20 (24) :75–77.
- [2] 黄中毓. 招聘系统的设计与实现 [D]. 西南交通大学, 2020. DOI:10. 27414/d. cnki. gxnju. 2019. 000651.
- [3] 熊玮楠. 融合图神经网络与多视图学习的社区问答专家推荐方法研究 [J]. 现代电子技术, 2024, 47 (09) :115–118. DOI:10. 16652/j. issn. 1004–373x. 2024. 09. 021.
- [4] Addo Prince Clement. A Big Data Analytics Using Scrapy Web Crawler Mining Technique[J]. Asian Journal of Research in Computer Science. Volume , Issue . 2021. PP 52-71.
- [5] Dr. Mahdi Hosseini. Online Job Portal using Django[J].International Journal of Innovative Science and Modern Engineering (IJISME). Volume 12 , Issue 1.
- [6] 刘泓杰. 基于 Vue.js 的视频营销创作平台的研究与设计 [D]. 北京邮电大学, 2024. DOI:10. 26969/d. cnki. gbydu. 2023. 000563.
- [7] Chen Lan. Application of SVM model based on collaborative filtering hybrid algorithm in e-commerce recommendation[J].International Journal of Computers and Applications. Volume 46 , Issue 5 . 2024. PP 292-300.
- [8] 贾丽坤, 赵亚丽, 黄晓英, 等. 考虑数据稀疏性的图书推荐协同过滤算法仿真 [J]. 计算机仿真, 2024, 41 (04) :470–474.
- [9] 涂文奇. 数据库通用动态查询工具的设计与实现 [D]. 东华大学, 2024. DOI:10. 27012/d. cnki. gdhuu. 2023. 000381.
- [10] 刘晓旭. 主题网络爬虫研究综述 [J]. 电脑知识与技术, 2024, 20 (08) :97–99. DOI:10. 14004/j. cnki. ckt. 2024. 0342.
- [11] Ma Xiaojun. Design and Implementation of Crawler Based on Scrapy[J].Journal of Physics: Conference Series. Volume 2033 , Issue 1.
- [12] Zhang Huilian. Design and Application of College and University Entrepreneurship Platform Based on MVC Architecture[J].Procedia Computer Science. Volume 228 , Issue . 2023. PP 211-222.

- [13] 王建文. 基于记忆增强和项目热度的社会推荐算法研究[D]. 黑龙江大学, 2024. DOI:10. 27123/d. cnki. ghlju. 2023. 001777.
- [14] 冯嘉润, 刘颖. 基于微信云数据库的 CRUD 操作工具的设计与实现[J]. 电脑知识与技术, 2023, 19(07): 55-57+67. DOI:10. 14004/j. cnki. ckt. 2023. 0486.
- [15] 潘锐. 基于 Selenium 脚本的 Web 功能测试脚本编写中的行为分析[D]. 南京邮电大学, 2022. DOI:10. 27251/d. cnki. gnjdc. 2021. 000741.