南京大学

本 科 毕 业 设 计

院　　系　软件学院

专　　业　软件工程

题　　目　高考志愿填报推荐系统

　　爬虫子系统的设计与实现

年　　级　2019　学 号　191250022

学生姓名　单金明

指导教师　葛季栋　职 称　副教授

提交日期　2023 年 6 月 2 日

南京大学本科毕业论文（设计）
诚信承诺书

本人郑重承诺：所呈交的毕业论文（设计）（题目：高考志愿填报推荐系统爬虫子系统的设计与实现）是在指导教师的指导下严格按照学校和院系有关规定由本人独立完成的。本毕业论文（设计）中引用他人观点及参考资源的内容均已标注引用，如出现侵犯他人知识产权的行为，由本人承担相应法律责任。本人承诺不存在抄袭、伪造、篡改、代写、买卖毕业论文（设计）等违纪行为。

作者签名：

学号：191250022

日期：2023 年 5 月 27 日

南京大学本科生毕业论文（设计、作品）中文摘要

题目：高考志愿填报推荐系统爬虫子系统的设计与实现

院系：软件学院

专业：软件工程

本科生姓名：单金明

指导教师（姓名、职称）：葛季栋 副教授

摘要：

普通高等学校招生统一考试（下文简称高考） 是我国特色鲜明的性统一选拔考试，其重要性越发显现。随着我国教育事业的蓬勃发展，高校数量、开设专业、院校新闻、招生信息、评估方式都出现了井喷式的增长。二十一世纪以来，互联网逐渐深入介入社会与生活，在高考信息汇总和志愿推荐方面都衍生了一些网络产品或应用。然而，经过初步的市场调研，我们团队认为目前市面存在的产品要么信息单一、功能简略，无法满足用户多元需求；要么受限于付费，对非关注用户和频繁使用的用户不甚友好。基于以上信息，我们团队决定开发一款信息健全、功能高效、推荐结果多元精确的高考志愿填报推荐系统。

本文的高考志愿填报推荐系统使用 Kmeans++作为专业推荐，使用聚类算法进行院校推荐，通过 MongoDB 进行数据的存储，采用 SpringBoot+Vue 的前后端分离框架实现界面交互和逻辑处理。该项目的爬虫子系统使用 scrapy 框架进行构建，通过请求异步处理、本地和服务器部署相结合的运行方式，从而实现高效快速的获取数据。此外，爬虫子系统还增加了对数据的清洗、格式化以及归纳统计，增强了数据的可靠性，完善了系统功能。

经过实际测试和使用，该系统能够实现考生检索信息和院校推荐的功能，并自动化更新院校、专业信息，从而满足考生的实际需求，实现高效、便捷的志愿填报。

关键词：爬虫；高考；志愿推荐；MongoDB ；Scrapy

南京大学本科生毕业论文（设计、作品）英文摘要

THESIS：Design And Realization of Data System in College Preference Intelligent Recommendation System

DEPARTMENT：Software Institution

SPECIALIZATION: Software Engineering

UNDERGRADUATE: Shan Jin

MENTOR: Professor Ge Jidong

ABSTRACT： The National College Entrance Examination (hereinafter referred to as the “GaoKao”) is a unified selection examination with distinctive characteristics, and its importance is becoming more and more apparent. With the rapid development of our education cause, the number of colleges and universities, majors, news of colleges and universities, enrollment information, evaluation way all appear to have a blowout growth. Since the 21st century, the Internet has gradually been deeply involved in society and life, and some network products or applications have been derived in the aspects of college entrance examination information summary and voluntary recommendation. However, after preliminary market research, our team believes that the existing products on the market either have single information and simple functions, which cannot meet the diversified needs of users; Or it is limited to paying and not very friendly to non-following users and frequent users. Based on the above information, our team decided to develop a college entrance examination voluntary filling recommendation system with sound information, efficient functions and multiple and accurate recommendation results.

In this thesis, the college entrance examination voluntary filling recommendation system uses Kmeans ++ as major recommendation, uses clustering algorithm for college recommendation, uses MongoDB for data storage, and uses SpringBoot+Vue front-end separation framework to realize interface interaction and logical processing. The crawler subsystem of this project is constructed by scrapy framework, and through the operation mode combining asynchronous request processing, local and server

deployment, it achieves efficient and fast data acquisition. In addition, the crawler subsystem also increases the data cleaning, formatting and inductive statistics, which enhances the reliability of data and improves the system function.

Through practical testing and application, the system can realize the function of examinee retrieval information and college recommendation, and automatically update the college and major information, so as to meet the actual needs of examinee and realize the efficient and convenient voluntary filling.

KEYWORDS: crawler; College Entrance Examination; Recommend; MongoDB; Scrapy

目　录

插图目录

表格目录

第一章 导 论

1.1 研究背景及意义

普通高等学校招生统一考试（下文简称高考） 我国特色鲜明的性统一选拔考试，其重要性越发显现。随着我国教育事业的蓬勃发展，高校数量、开设专业、院校新闻、招生信息、评估方式都出现了井喷式的增长。同时，随着新高考改革，“高考政策的核心目的在于实现国家人才选拔和满足个体受教育意愿。有关高考的政策组合是公共教育政策的核心部分，对教育内部影响着基础教育评价导向和高校人才选拔模式，承载着百姓对社会公平正义的期待”[1] ，高考在百姓心中的重要程度也越发增加，如何在浩如烟海的信息中选择最适合自己的学校和专业成为越来越多考生和家庭的问题。基于以上两点原因，收集信息和进行志愿填报成为考生的痛点和难点。

二十一世纪以来，互联网逐渐深入介入社会与生活，在高考信息汇总和志愿推荐方面都衍生了一些网络产品或应用，典型的服务平台有掌上高考、爱高考和阳光高考网等。然而，经过初步的市场调研，我们团队认为目前市面存在的产品均有其不足之处，要么信息单一、功能简略，无法满足用户多元需求；要么受限于付费，对非关注用户和频繁使用的用户不甚友好；此外，各个平台的院校专业推荐也均基于考生分数，缺少针对不同考生不同个人情况的个性化推荐。基于以上信息，我们团队决定开发一款信息健全、功能高效、推荐结果多元精确的高考志愿填报推荐系统。

该系统通过提供完整详尽的信息资源、全面客观的分析对比，很好的解决了考生信息获取渠道的问题，同时将考生自身特点纳入个人信息系统，通过考生职业兴趣和人格特征，对考生当前分数进行合适的院校推荐和专业推荐。

在互联网蓬勃发展的二十一世纪，网络爬虫又称网络蜘蛛、网页机器人，已然成为从浩如烟海的因特网获取信息的重要手段。作为一类自动从互联网中筛选信息、采集信息的应用程序或脚本，网络爬虫高效、准确、自动化的优点

是人工筛选信息所不具备的，并随着需求发展爬虫也逐渐衍生出了众多方向、技术和框架，更进一步完善、拓展了网络爬虫的功能和结构。

1.2 国内外研究现状

1.2.1 爬虫技术的研究现状

解放军理工大学的孙立伟等人[2]对网页爬行的概念和类别、可能遇到的困难以及未来发展方向提出独到见解。针对逆向爬虫，李玺[3]总结了爬虫的架构体系、主流框架和未来发展，包括各种自动化工具、抓包工具、逆向工具的使用；来自西安邮电大学的潘晓英[4]团队则回顾近年来的爬虫现状，分析了各种主题相似度的方法以及搜索策略，比较了主题网络爬虫两种动态搜索策略并指出了未来研究方向。来自清华大学的周立柱等人[5]对聚焦爬虫当前的关键技术和未来研究方向做出深入探讨。来自福建师范大学的陈丛[6]等人针对爬虫过程中的虚假数据处理方法进行研究，通过随机取样、过滤器设计、密匙检查等方法实现虚假数据的溯源与途中过滤。电子科技大学的孙川钰[7]等人针对传统关键词检索的缺点，提出改进算法，增加了检索精度。电子科技大学的罗春[8]针对网络中冗余信息干扰，设计了基于网络爬虫技术的大数据采集系统，通过软件硬件结合完成对数据的筛选和采集。而北京交通大学的郭一峰[9]将 scrapy+MongoDB 在图书爬虫系统中实际应用，为该类框架和技术的实现提供了一个真实的案例以供参考。此外，来自北京理工大学的胡博[10]对基于网络信息的评价模式进行分析，并对评价模式中的网络数据抓取和文本分析技术给出自己的研究和设计实现；来自西南交通大学的赵鹏程[11]设计并实现了一个基于 Scrapy 框架的分布式书籍网络爬虫系统 DScrapy，实现了对互联网上书籍信息与书籍文件的下载，对爬取得到的数据进行分布式存储。

1.2.2 志愿推荐算法的研究现状

中国科学院大学沈阳计算技术研究所、山东大学大数据技术与认知智能实验室和中国科学院沈阳计算技术研究所的团队[12]针对推荐算法，提出“针对新高考志愿填报要求，以考生分数能否被院校专业录取作为评价标准，结合专业类型、选科要求、院校属性、学费等多维数据，使用长短时记忆网络算法，可以预测出该分数在当年录取位次，再根据一定报考规则形成志愿填报方案。使用考生报考当年前 3 年的专业录取位次和当年录取位次作为输入，得到当年考生被各专业的录取概率，并基于该录取概率为考生进行志愿推荐。”；吉林大学软件学院、吉林大学原子与分子物理研究所和吉林大学计算机科学与技术学院[13]提出基于分数线预测的多特征融合高考志愿推荐算法，“该算法首先利用历年高校最低投档位次，通过 BP 神经网络预测报考年份各高校最低投档位次以及最低投档分数线，然后根据考生分数进行院校初筛，进而构建 3 种与录取分数相关的特征，结合院校软科排名，通过遗传算法进行权值寻优，得到不同院校的录取概率，并在此基础上定义推荐度实现为考生进行不同录取风险层次的高校推荐，形成完整的推荐结果。”；内蒙古大学的白俊杰[14]使用混合推荐设计推荐系统，“本文通过灰色预测理论中的 GM(1,1)预测模型和 Verhulst 预测模型对录取分数线进行预测,采用后验差法对预测结果进行检验并择优选择作为最终的录取分数线预测值。本文以 2756 所院校 2017-2020 年在内蒙古自治区的理科录取分数线为输入进行了测试,测试结果显示,97.16%的院校 2021 年预测录取分数线与实际录取分数线误差值在 10 以内。(2)本文以霍兰德职业兴趣测试和迈尔斯-布里格斯性格分类指标的测试结果为输入,通过基于内容推荐和基于用户协同过滤的混合推荐算法完成对考生的专业推荐。本文采用归一化折损累计增益对推荐结果进行评估,结果显示,混合推荐算法的推荐效果优于基于内容的推荐算法和基于用户的协同过滤算法,且混合推荐模型采用 TOP-15 的方式进行推荐。(3)本文采用 K-means++算法对各院校预测录取分数线进行聚类分析并最终实现对考生的院校推荐。本文使用轮廓系数对聚类效果进行评估,评估结果显示,当聚类数为 6 时,K-means++算法聚类效果最佳。”；中国科学院大学(中国科学院沈阳计算技术研究所) 的王柏琦[15]基于多特征权重设计推荐系统，“以考生被院校专业录取的概率作为评价标准,结合办学性质,学校类型,专业大类,院校等级等多个数据特征,对考生进行志愿推荐。文章使用报考当年前 3 年的专业录取位次和当年考

生的录取位次,以及专业的作为输入,使用长短时记忆网络(LSTM)算法,预测出该分数在当年被各个院录取的概率;再结合多个数据特征维度,通过赋予其不同的权重对各个专业的再进行排序,最终将两个数据整个排序,结合该省的报考规则形成最终的志愿填报方案。本文基于微信公众号开发了新高考志愿填报系统,并将其应用于河北、山东和辽宁三省的一些试点高中,使用该系统帮助高考生进行填报。”

1.3 本文的主要工作

本文的主要工作是介绍实现系统的主要功能和模块，并在此基础上介绍笔者负责的爬虫子系统的设计与实现。本系统分为用户模块、院校信息模块、专业信息模块和院校专业推荐模块共四部分。其中用户模块主要实现了用户职业兴趣测试、人格测试和登录注册。院校信息模块和专业信息模块主要实现了不同院校、专业的信息模糊检索、精确匹配、分类筛选和信息对比。推荐模块主要实现了院校推荐和专业推荐。

第二部分本文介绍了该系统的数据库设计，包括数据库运行管理和数据库存储格式设计，并对爬虫子系统的具体实现进行说明，包括网络请求分析、spiders 模块、items 模块、pipelines 模块、settings 模块，实现了对院校信息、专业信息和录取信息的获取、处理以及存储。最后，文章对爬虫的实际运行，如脚本启动、多线程运行、模拟用户以及服务器部署做出一些探索工作。

1.4 论文的组织结构

本文一共包含七章，组织结构如下：

第一章介绍了该课题所处的背景及项目意义、国内外对于高考志愿推荐的研究现状、本文的主要工作和论文结构，对文章做出了总体概括。

第二章对所用的技术框架进行简要介绍，对爬虫子系统的 Scrapy 框架、前端的 Vue 框架、后端的 SpringBoot 框架和 MongoDB 进行了介绍，该部分对后文系

统设计及系统特性做出铺垫。

第三章详细展开系统需求分析，对系统开发的目标、受众群体、系统功能需求、用例和非功能需求进行展示，印证系统的功能作用及意义。

第四章对系统设计进行简要介绍，对系统总体设计方案和系统模块划分进行了说明。

第五章介绍数据库设计。对数据库的运行管理、数据存储格式设计和爬虫子系统的详细设计进行展示。

第六章介绍前文爬虫子系统的详细实现。数据获取模块中详细展示了关键步骤、过程困难及应对方法。

第七章：总结与展望。对本文的工作进行回顾和总结，以及提出未来展望的方向。

第二章 相关技术概论

2.1 爬虫技术

网络爬虫是一个自动下载网页的计算机程序或自动化脚本，通常从一个初始的网页链接集合构成的队列开始，根据一定的顺序对每个页面进行下载、解析内容、存储结果并根据需要获取新的迭代网页链接加入链接队列。根据分类，爬虫可以大致分为通用网络爬虫、聚焦网络爬虫、增量式网络爬虫和深层网络爬虫四类。

1) 通用网络爬虫通常是对全网信息进行无差别大规模爬取，对系统运行的可靠性、并发性、运行速度和存储空间要求较高，通常应用于搜索引擎。
2) 聚焦网络爬虫又称主题网络爬虫，是根据既定目标选择性的抓取相关信息的计算机程序或脚本，可用于各类主题系统的设计和实现。
3) 增量式网络爬虫指对特性范围的页面中对发生变化或新增加的页面进行爬虫，通常对增量爬取的设计和去重有一定要求。
4) 深层网络爬虫指爬取隐藏在网页表单、按钮和信息验证后的潜在页面，通过静态链接无法获取的信息和页面。

2.2 Scrapy

Scrapy 是一个用于抓取网站和提取结构化数据的应用程序框架，可用于广泛的有用应用程序，如数据挖掘、信息处理或历史存档。此外，它也可以用于使用 API 提取数据或作为通用网络爬虫。Scrapy 异步调度和处理请求，即 Scrapy 在等待请求完成和处理时会发送另一个请求或同时做其他事情，且请求失败或在处理请求时发生错误，其他请求也可以继续进行。

此外，控制爬行范围也是该项目选用 Scrapy 的原因之一。Scrapy 可以通过设置下载延迟，限制每个域或每个 IP 的并发请求数量，以及使用自动节流扩展来自动实现都可以实现对爬虫更加精确的控制。

Scrapy 包含 Scrapy Engine、Scheduler、Downloader、Spider、Item Pipeline 和 Spider Middlewares 几部分。引擎用于部件间通讯；调度器整理排列请求用于异步高效获取数据；下载器用于实际下载内容；爬虫则写入我们的具体逻辑和控制信息，并将所获得的信息交还管道；管道用于爬虫开始前或结束后的处理；中间件可以自定义拓展。其间的合作与配合间下文图解。

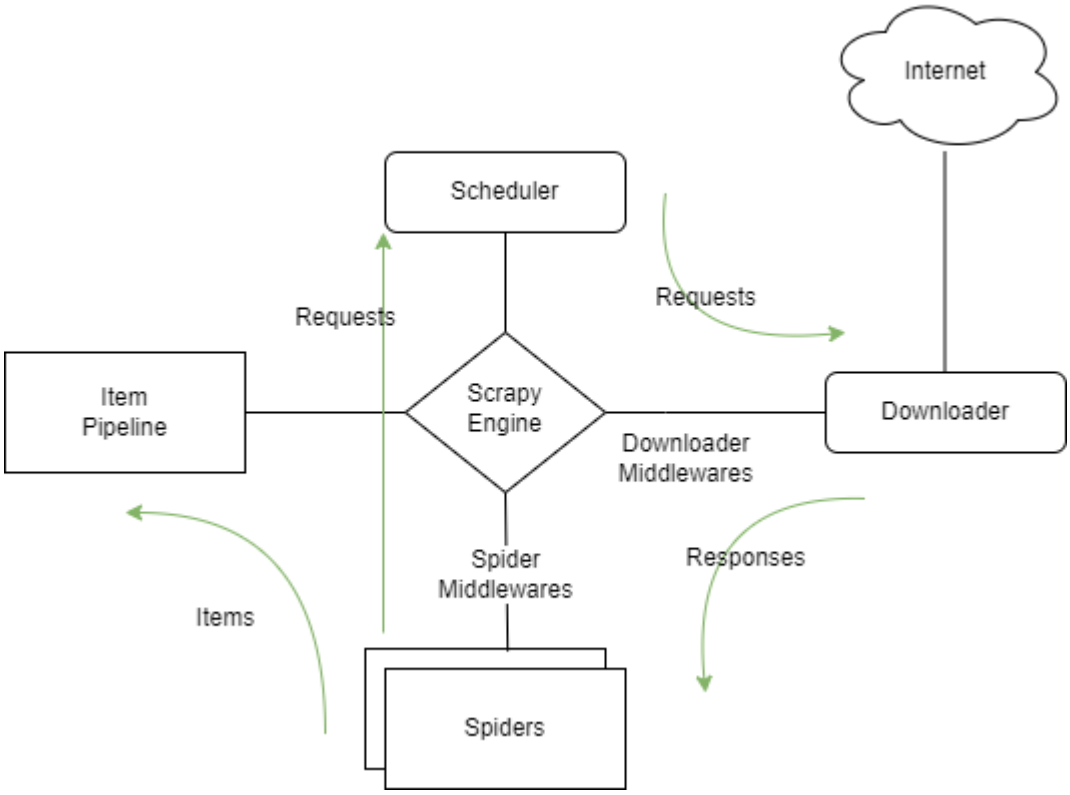

图 2-1scrapy 架构

2.2 MongoDB

MongoDB 是一个由 C++语言编写、基于分布式文件存储的数据库，旨在为 WEB 应用提供可扩展的高性能数据存储解决方案。MongoDB 中的记录是一个文档，它是由字段和值对组成的数据结构。MongoDB 文档类似于 JSON 对象。字段的值可以包括其他文档，数组和文档数组。该特性允许其将文档（即对象）对应于许多编程语言中的本机数据类型，同时嵌入式文档和数组减少了对昂贵连接的

需求；并且可以动态模式支持流畅的多态性。与此同时，MongoDB 能够对集合和视图按需实例化。此外，MongoDB 还具有高可用性、水平可伸缩性、支持多种存储引擎和提供可插拔的存储引擎 API 等多种特性。

2.3 SpringBoot

SpringBoot 派生于轻量级 Java 框架 Spring，是一个全新的自动化配置解决方案，其设计目的即是简化 Spring 应用的搭建和开发过程，并快速创建基于 Spring 生产级的独立应用程序。SpringBoot 默认配置了很多框架的使用方式，使用 SpringBoot 可以快速创建基于 Spring 生产级的独立应用程序，并提供了自动配置的“starter”项目对象模型（POMS）以简化 Maven 配置，最重要的是，使用时无需过多甚至不需要关注 XML 配置，使用起来较为便捷。SpringBoot 的主要的核心技术为 IOC（控制反转）和 AOP（面向切面）。控制反转是面向对象的一种设计原则，能够进行业务对象的控制管理，以及对业务对象多的依赖绑定，最大的作用就是降低代码的耦合度，提高代码的可用性。

第三章 系统需求分析

3.1 总体描述

3.1.1 系统目标

本系统作为一款 Browser/Server 应用，前端使用 Vue 框架进行开发，后端则采用 SpringBoot 框架与 MongoDB 数据库。系统主要分为用户模块、院校信息模块、专业信息模块和推荐模块。该应用具有院校信息检索、专业信息检索、获取用户特征和推荐专业院校等功能，能够方便、快捷、直观的展示各个院校的信息，同时根据考生的自身情况推荐分数和兴趣契合考生的院校专业。

该系统的爬虫子系统主要使用 scrapy 搭建，主要分为网络蜘蛛、管道、元素和爬虫控制四部分。该子系统能够实现特定主题的自动化抓取、数据更新和数据存储。该子系统能够提供全面、完整的数据信息，并对数据进行筛选和逻辑处理，从而为整个系统的查询、推荐功能提供坚实的数据支撑。同时该系统应能够支持并发、多线程，以保证高效的获取数据。对于需要连续运行的爬虫程序，该子系统还应能够通过脚本可以部署于服务器以便不间断运行。

3.1.2 用户特征

本系统的目标用户是高考考生，考生的需求是通过使用本系统来了解院校和专业信息，并参考系统推荐的志愿填报方案来选择志愿。考虑到我国高中生的学习情况，大部分考生在高考前都对高等院校和各个专业没有详细的了解，并且有不少考生由于之前的生活重心都放在了学习上，对电脑网页的操作并不熟悉，因此本系统应尽量让页面简单明了，容易上手操作，同时为考生提供获取院校和专业信息的渠道。

3.2 功能性需求分析

本系统的主要功能有：

1. 个人信息管理。用户可以填写并保存个人基本信息，如高考分数，排名，生源地，选科。此外，用户还可以通过填写问卷来完成性格和职业测试，测试结果也会被保存在用户个人信息中。

2. 院校信息检索。用户可以通过名称检索院校，同时还可以对检索结果进行排序或筛选来更容易找到要找的院校。在检索到目标院校后，院校的相关数据如院校基本信息，院校图片，专业历年分数线等会以可视化的形式呈现。

3. 专业信息检索。用户可以通过名称或者专业代码检索专业，在检索到目标专业后，可以查看目标专业的介绍，就业前景和知名院校等信息。

4. 志愿推荐。用户在填写了个人基本信息后即可进行基本的志愿推荐，填写系统提供的测试问卷则可以使志愿推荐的结果更加合理，志愿推荐系统会根据用户情况分别列出推荐的院校和专业列表供用户参考。

系统用例图如图 3-1 所示：

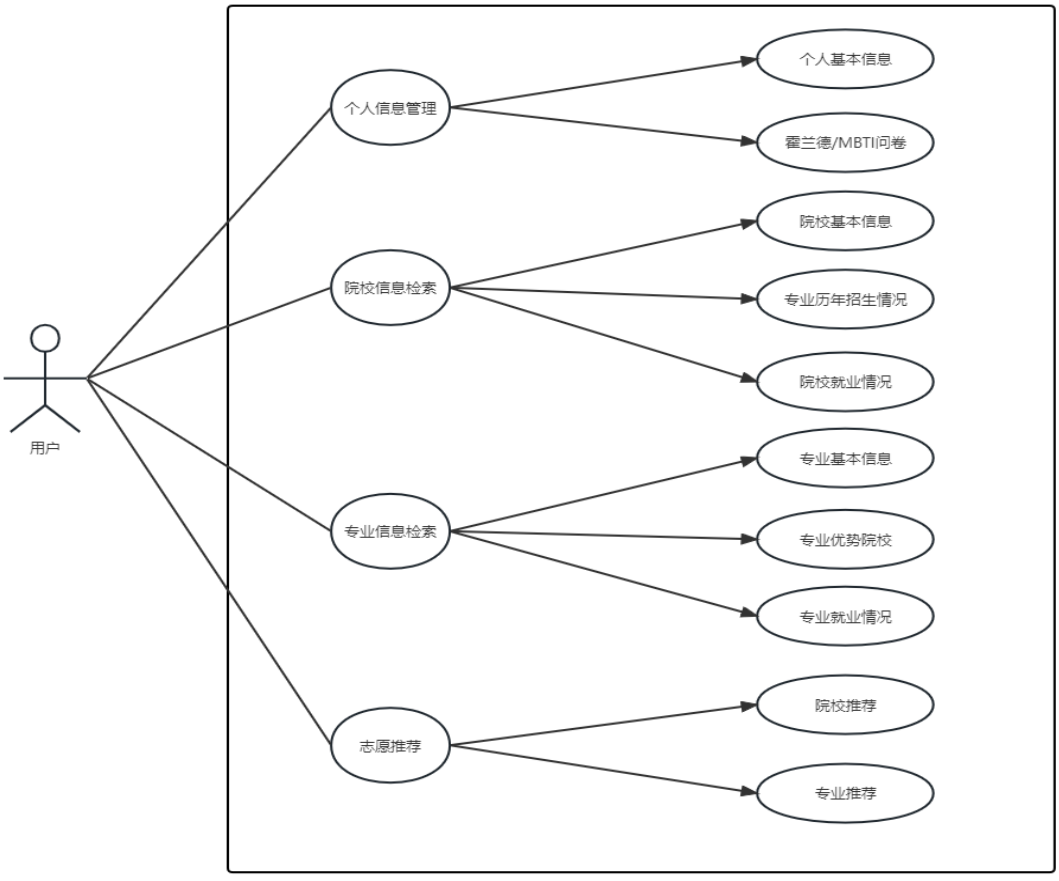

图 3-1 系统用例图

3.3 用例描述

这一节包含 4 个高考志愿推荐系统的功能用例和两个爬虫子系统的爬虫用例。系统的功能用例描述了用户和系统整体的不同交互过程。爬虫子系统的用例从爬虫子系统和数据库、爬虫子系统内部不同组件模块的交互过程。

3.3.1 高考志愿推荐系统的用例描述

1. 用户进行个人信息管理的用例描述如表 3.1 所示。此用例的参与者为高考考生，描述了考生完善个人信息并且进行修改的流程。

表 3-1 用户个人信息管理用例

名称	**内容描述**
ID	01
用例名称	个人信息管理
参与者	考生用户
描述	用户填写个人基本信息和问卷，如有需要可以进行修改
触发条件	用户打开个人基本信息或问卷页面
前置条件	用户处于登录状态
后置条件	如果用户修改个人信息或文件内容，系统需要将其保存
正常流程	1. 用户打开个人基本信息或问卷页面，系统显示信息 2. 用户修改信息，系统返回修改后的数据
拓展流程	无
特殊需求	无

2. 用户进行高校信息检索的用例描述如表 3.2 所示。此用例的参与者为高

考考生，描述了考生查找各大高校的详细资讯的流程。

表 3-2 院校信息检索的用例描述

名称	内容描述
ID	02
用例名称	院校信息检索
参与者	考生用户
描述	用户输入院校名称关键词，系统检索名称包含关键词的院校并展示，用户选择系统展示的某个院校，系统展示院校的详细信息
触发条件	用户输入院校名称关键词并检索
前置条件	用户处于登录状态
后置条件	无
正常流程	1. 用户输入院校名称关键词并检索 2. 系统以列表的形式展示名称包含关键词的院校 3. 用户选择系统列出的某个院校 4. 系统展示用户选择的院校的详细信息
拓展流程	无
特殊需求	无

3. 用户进行专业信息检索的用例描述如表 3.3 所示。此用例的参与者为高考生，描述了考生查找各专业基本资料和详细信息的流程。

表 3-3 专业信息检索的用例描述

名称	内容描述
ID	03
用例名称	专业信息检索

参与者	考生用户
描述	用户输入专业名称关键词或者专业代码，系统检索相关专业并展示，用户选择系统展示的某个专业，系统展示专业的详细信息
触发条件	用户输入专业名称关键词或者专业代码并检索
前置条件	用户处于登录状态
后置条件	无
正常流程	1. 用户输入专业名称关键词或者专业代码并检索 2. 系统以列表的形式展示相关专业 3. 用户选择系统列出的某个专业 4. 系统展示用户选择的专业的详细信息
拓展流程	无
特殊需求	无

4. 志愿推荐的用例描述如如表 3.4 所示。此用例的参与者是高考考生，描述了考生输入高考分数、省内排名、文理分科和生源地的信息后，系统根据推荐算法给出推荐的高校和专业名单的过程。

表 3-4 志愿推荐的用例描述

名称	**内容描述**
ID	04
用例名称	志愿推荐
参与者	考生用户
描述	用户在填写了个人信息后，选择志愿推荐，系统展示推荐的院校和专业

触发条件	用户点击志愿推荐按钮
前置条件	用户处于登录状态
后置条件	如果用户修改个人信息或文件内容，系统需要将其保存
正常流程	1. 用户点击志愿推荐按钮 2. 系统显示推荐的院校和专业列表
拓展流程	无
特殊需求	无

3.3.2 爬虫子系统的用例描述

爬虫子系统对于网络上的各类院校专业信息以及高考相关资讯进行搜集、下载、处理和存储。主要模块有调度器、网络蜘蛛和管道。

1. 表 3-5 描述了爬虫调度用例。爬虫调度器对页面链接集合构建队列，并根据当前爬虫执行情况进行调度。

表 3-5 爬虫调度器的用例描述

名称	**内容描述**
ID	05
用例名称	爬虫调度
参与者	爬虫引擎
描述	爬虫引擎检查页面链接的有效性，并对网页链接设置优先级，按照优先级进行调度
触发条件	爬虫启动
前置条件	无
后置条件	将调度后的网页链接分配给爬虫进行后续的网页爬取
正常流程	1.获取爬虫进行的网页链接

	2.检查链接的合法性 3.设置网页链接优先级 4.按照优先级进行排序
拓展流程	无
特殊需求	无

2. 表 3-6 描述了爬虫爬取网页链接的用例。爬虫引擎启动爬虫程序并选取网页链接队列中的第一个链接，加载页面，对下载的文档树或 json 文件进行解析，并提取需要的数据返回至管道。

表 3-6 爬虫爬取网页链接的用例描述

名称	**内容描述**
ID	06
用例名称	爬取网页链接
参与者	爬虫引擎
描述	爬虫程序选取网页链接队列中的第一个链接，加载页面，对下载的文档树或 json 文件进行解析，并提取需要的数据返回至管道。
触发条件	爬虫引擎运行爬虫程序
前置条件	爬虫正常启动，调度器完成网页链接的调度
后置条件	将网页中提取的数据送至管道
正常流程	1.获取目标网页内容 2.解析网页 3.提取网页中的有效信息
拓展流程	若网页中包含其他链接，需将链接添加至调度器的链接队列
特殊需求	若因网络或服务器原因下载失败，需提示错

误信息

3.4 非功能性需求描述

易用性：系统应符合日常使用习惯，操作方便，用词易于理解，用户无需参考手册即可直接使用系统，同时，系统应对用户的操作给出显式的反馈，并对可能出现的误操作提供一定的容错处理。

时间性能：用户的操作应在 500ms 内得到反馈，对于耗时更久的大数据量操作等，应出现加载提示以减少用户焦虑。

安全性：系统应只允许已经登录的用户访问内容，此外，系统需要保证用户的个人信息不会泄露。

可扩展性：系统应采用模块化组件化开发，尽可能降低耦合提高复用，便于日后的功能扩展。

可移植性：系统应正常在绝大部分浏览器上运行，需要适配不同分辨率的不同种类浏览器。

3.5 本章小结

该系统是一款前后端分离、基于 Vue+SpringBoot 框架，面向高考考生的志愿推荐系统，包含用户信息模块、院校信息模块、专业信息模块和志愿推荐模块共四部分。系统主要包含用户信息、院校检索、专业检索和志愿推荐功能，具有良好的易用性、安全性、可拓展性和可移植性。

第四章 系统设计

4.1 总体设计

本节将使用“4+1”视图模型来进一步描述系统的架构设计，将展示系统的总体架构设计图，逻辑视图，物理视图，处理视图和开发视图。

系统的总体架构分为交互层，执行层和数据层。其中交互层负责和用户的交互，并向执行层传输交互数据，执行层负责处理逻辑，数据层则负责向执行层提供需要的数据。在交互层，系统通过 UI 组件展示窗口页面和用户进行交互，并将用户所执行操作和请求参数返回给执行层。在执行层中，系统模块收到相应的请求和参数，进行逻辑处理，在数据层获取相应的数据后返回结果。数据层则包含各类信息的存储、维护、检索和更新，除了返回相应的数据，在数据的可视化、视图的构建和数据的动态更新方面也有相应的职责。

系统的总体架构设计图如图 4-1 所示。

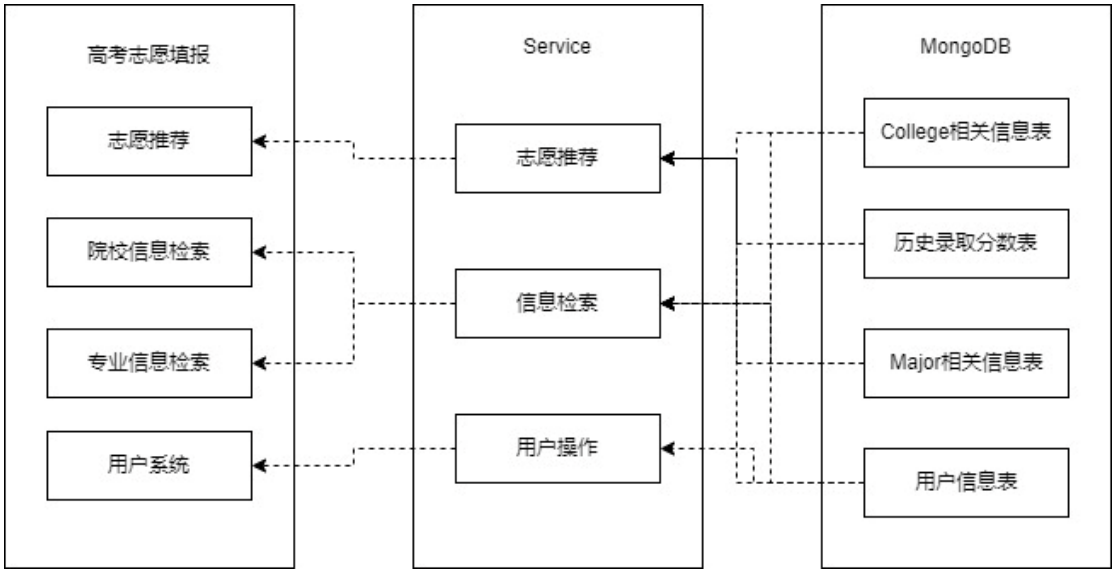

图 4-1 系统架构设计图

图 4-2 描述了系统的逻辑视图。交互层将交互数据传递给业务层，业务层再去从数据层获取数据来处理业务。

以登录行为为例，用户登录时的数据和行为会由 SignIn.vue 收集并向业务

层发起请求，业务层的 UserImpl 会处理该请求并对数据层中 User 表的相关数据进行更新处理。

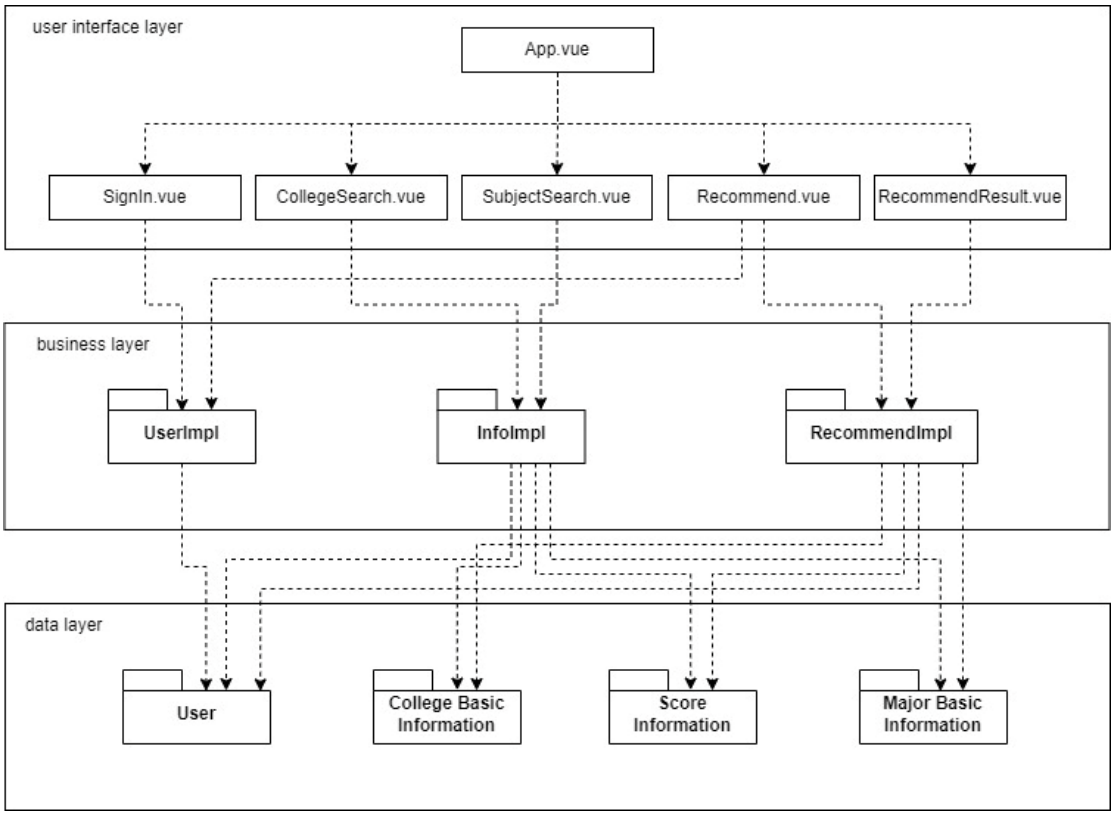

图 4-2 系统逻辑视图

物理视图描述了系统如何部署于硬件上。本系统的前端页面运行于浏览器中，而后端以及数据库则运行于服务器上，即前后端和数据库均通过 docker 部署于云服务器。

系统的物理视图如图 4-3 所示。

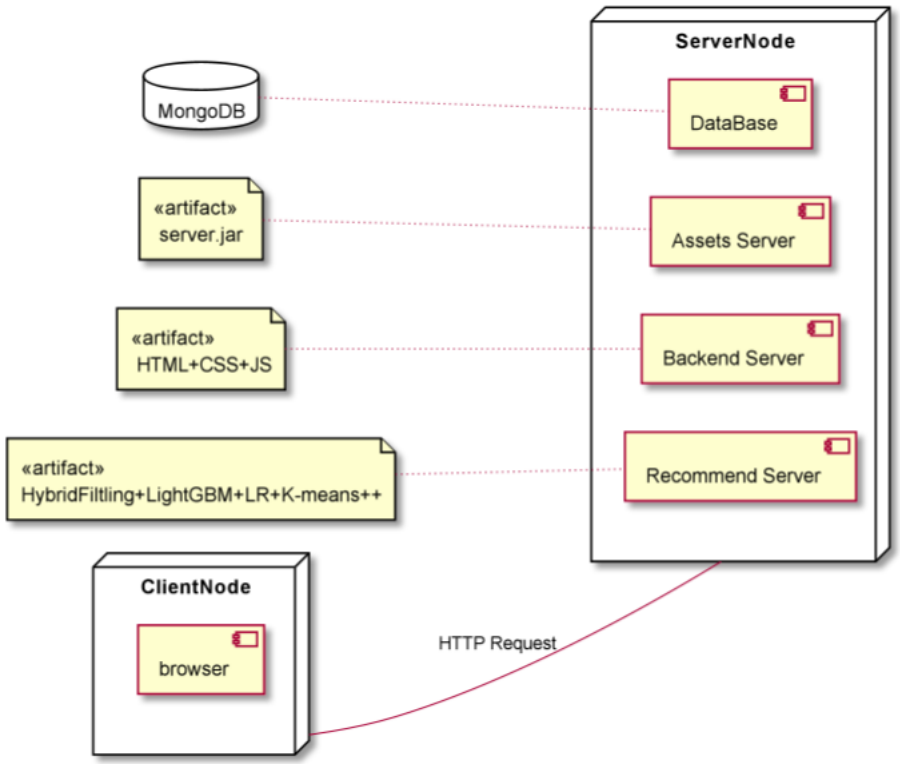

图 4-3 系统的物理视图

图 4-4 描述了处理视图。处理视图描绘的是系统软件组织之间的通讯时序，数据的输入和输出，在本处以时序图的形式来表示。

以用户对院校的模糊查询为例，已经注册的用户需要首先进行登陆，否则需要注册。登陆成功后在相应的界面输入查询需求，系统会携带用户的 token 发送请求。收到请求的模块系统会进行逻辑处理，并根据参数获取数据库数据。若过程没有异常则返回结果，否则提示报错。

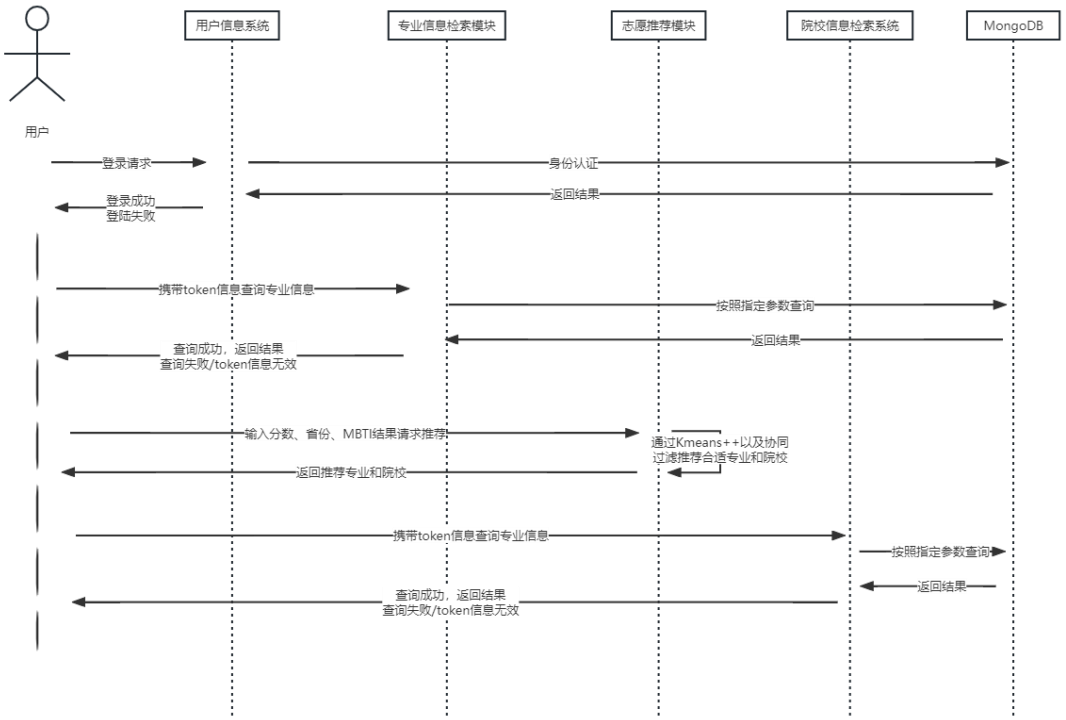

图 4-4 系统的处理视图

图 4-5 描述了开发视图。开发视图清晰地展现了系统的开发逻辑，包括前后端各个模块的分层架构和它们之间的交互情况。

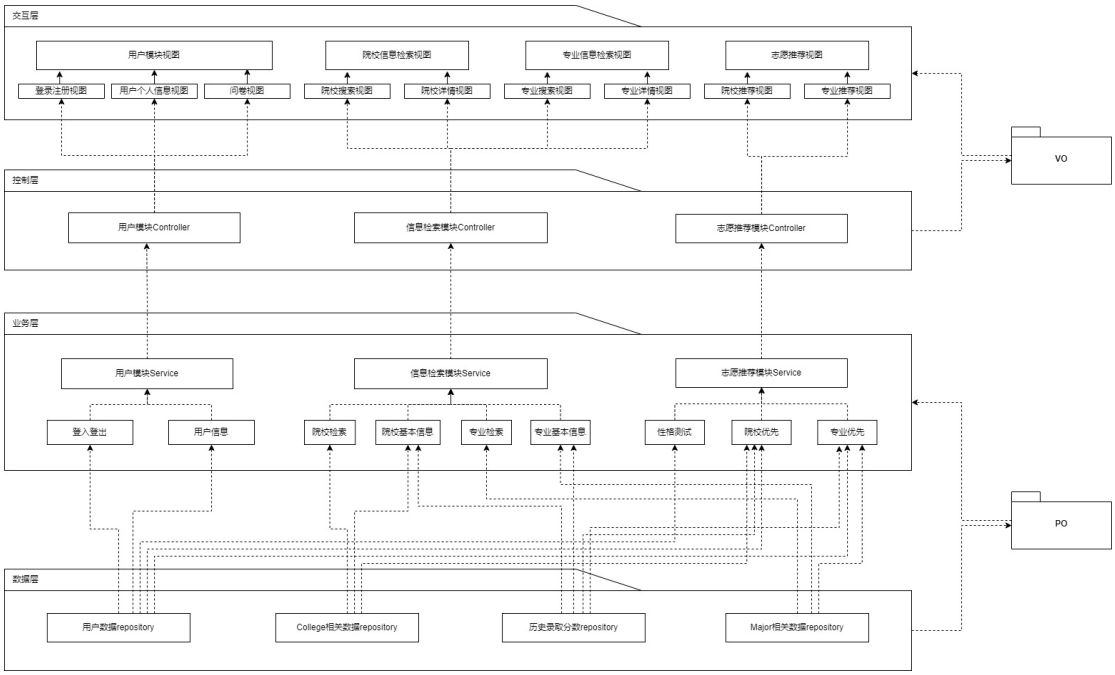

图 4-5 系统的开发视图

4.2 系统模块划分

系统分为四个主要功能模块，如图 4-6 所示，分别为用户信息系统，院校信息检索系统，专业信息检索系统和志愿推荐系统。这四个功能模块可以细分为十一个小模块：登录注册模块，用户基本信息模块，问卷模块，院校搜索模块，院校详情模块，专业搜索模块，专业详情模块，院校推荐模块，专业推荐模块，数据获取模块和数据处理模块。

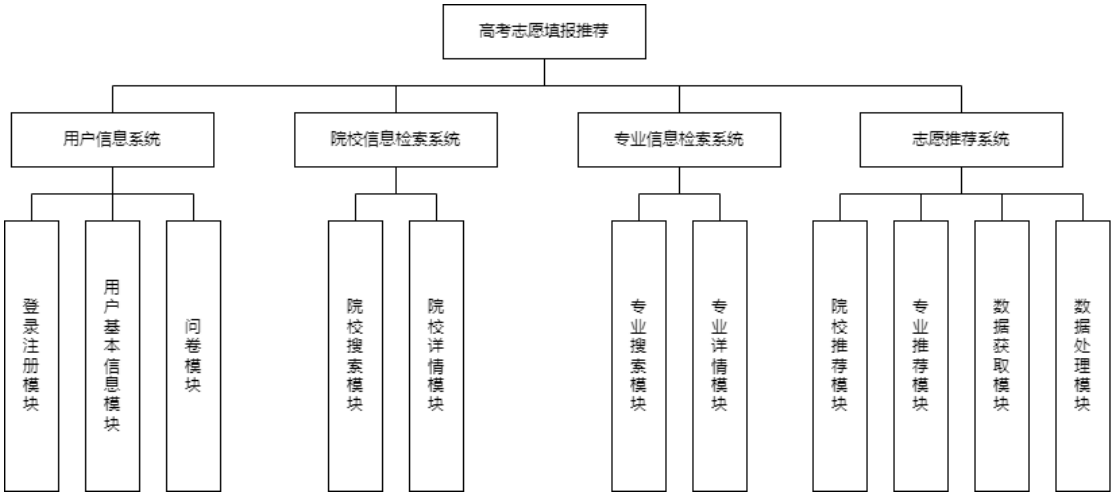

图 4-6 系统的功能模块

4.3 爬虫子系统的总体结构

图 4-7 描述了爬虫子系统的系统流程图。首先子系统根据爬取对象确定网页的链接队列，之后根据该网络请求是否控制并发访问决定运行方式。若限制网络访问则通过服务器部署，并设置间隔时间；若不限制则本地运行。然后对应的蜘蛛程序对请求进行爬取，并将获取的数据写入数据库。

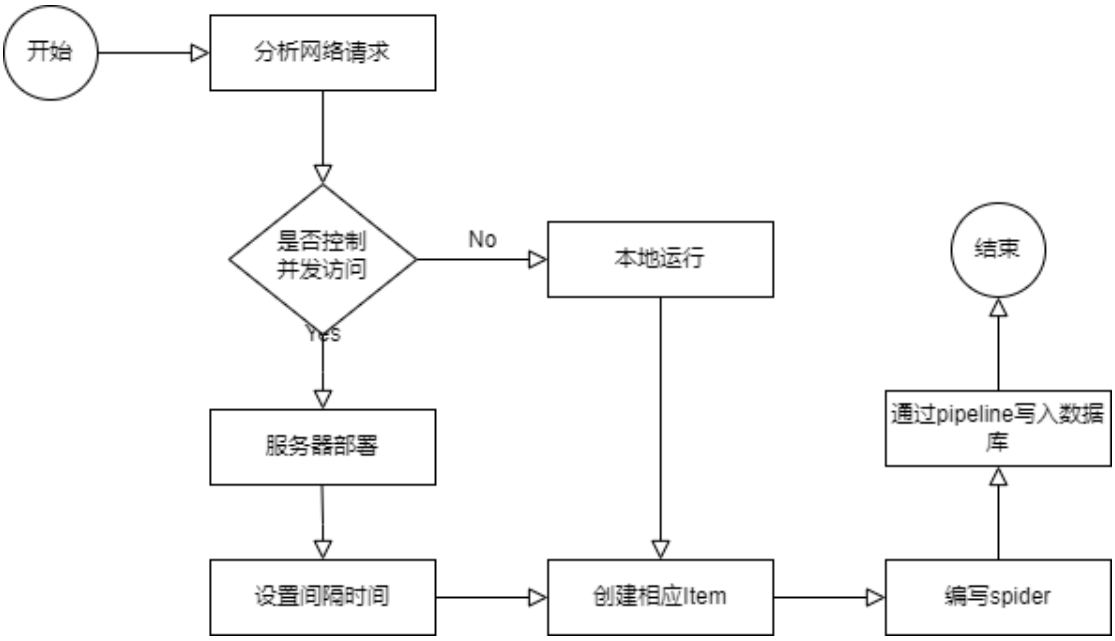

图 4-7 爬虫系统流程图

第五章 数据库设计

5.1 数据库运行管理

5.1.1 安全管理

在设置了密码的前提下，我们启动了验证权限。MongoDB 连接数据库的用户信息注册在数据库的 admin 表中，默认情况下 MongoDB 不会启用授权认证，无权限的用户也可访问数据库。当启用权限时，只有被 admin 授权的用户方可允许登录。部分黑客会直接扫描 Mongo 的默认 IP:27017 端口，检测到了无权限和密码的服务器就恶意删库勒索，这样可以有效防止这样的行为。

5.1.2 可视化管理

该项目使用 MongoDB Compass 工具作为 MongoDB 的可视化工具，从而对数据进行管理，如图 5.1 所示。

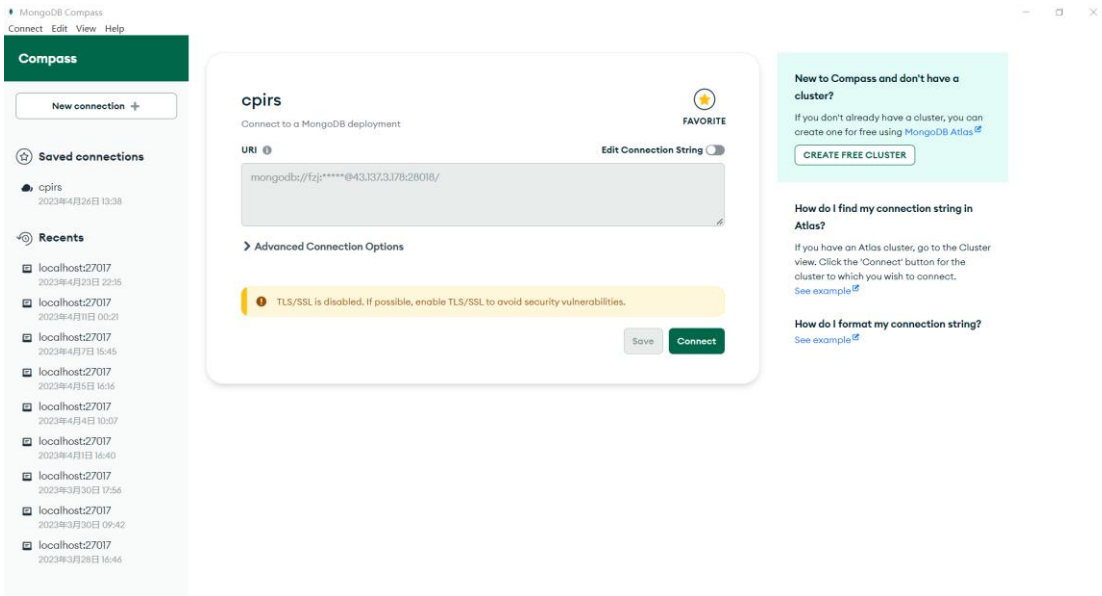

图 5-1 MongoDB 界面

5.1.3 查询管理

该项目使用 mongo 语句设计查询语句，并保存为查询视图。以对学校 id 的查询视图 school_id group 为例，通过构建查询语句并保存为视图获得全部学校的 id 列表，并基于此进行后续的数据获取和更新工作。

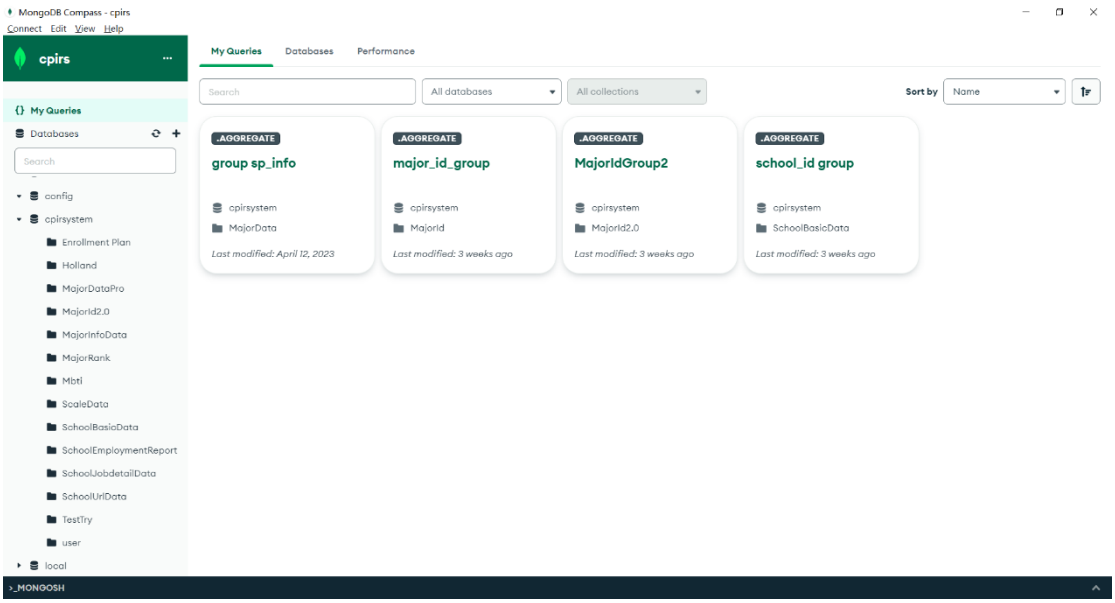

图 5-2 MongoDB 查询视图

5.2 存储格式设计

Mongo 涉及的数据为类 json 或 bson 的文档型数据，除院校名称及其编号和省份名及其编号外，各个集合间没有明显的数据并集，每个集合可近似看做独立。Mongo 的存储形式允许嵌套文档存在，故不适宜用表格进行描述，因此数据库字段均以“示例 + 注释”的 json 字段形式进行描述。

图 5-3 描述了数据库的总体设计。共包含 13 张表，分别对应用户信息、院校信息、专业信息和录取信息四个部分。其中用户信息主要包含 User、Holland 和 Mbti 三张表，User 存储用户的基本信息，Holland 是职业兴趣测试表单，Mbti 是人格测试表单，用户可以在系统界面填写表单并将结果存储在

user 表中，作为后续志愿推荐的算法参数。院校信息主要包含 SchoolBasicData、SchoolEmploymentReport、SchoolJobdetailData 和 SchoolUrlData 四张表，分别存储院校基本信息、院校就业报告、院校就业统计信息和院校图片。专业信息存储于 MajorID、MajorInfoData 和 MajorRank 三张表中，MajorID 负责各个专业的统计和编码，而在 MajorInfoData 中具体存储每个专业的详细数据，MajorRank 包含专业评级信息，主要为第四轮学科评估数据。录取信息主要包含 MajorDataPro 和 ScaleData 两张表，分别是某院校某专业在某省份某一年的具体录取信息，以及该省份当年的位次信息。

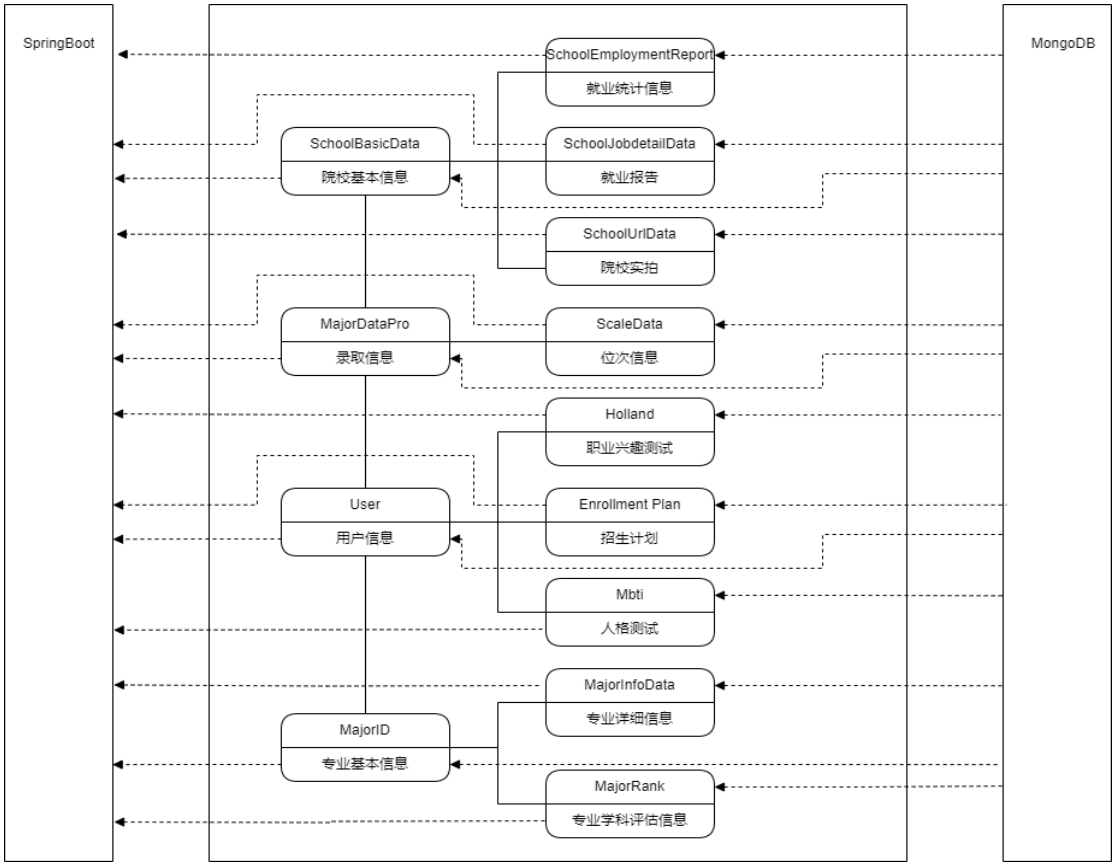

图 5-3 数据库设计

表 5-1 各个数据表的内容描述

数据表名称	描 述
SchoolBasicData	院校的详细信息

SchoolEmploymentReport	院校就业质量报告
SchoolJobdetailData	院校毕业生就业统计数据
SchoolUrlData	院校实景图
MajorID	专业编码和基本信息
MajorDataPro	专业录取信息（包括某院校某专业在某年某地区的录取分数和专业编码）
MajorInfoData	专业详细信息
MajorRank	学科评估参评院校及评审结果
ScaleData	各省历年一分一档表
User	用户信息
Holland	哈兰德职业兴趣测试表单
Mbti	16 型人格测试表单
ErollmentPlan	招生计划表

5.2.1 院校信息

SchoolBasicData 表存储院校基础信息，该数据由数个单独描述院校基础信息的集合组组成，集合间以院校编号和院校名称一一对应，共计 2826 条数据，主要包含院校名称、联系方式、院校地址、学校层次和院校类型。

SchoolEmploymentReport 存储院校就业质量报告，描述该院校的毕业生就业情况情况，共计 1048 份数据。每条按数据中包含院校 id 用以查询和匹配相应院校、就业质量报告列表，列表中每个元素包含年份、报告题目和报告链接三部分。

SchoolJobdetailData 存储院校就业信息，描述院校毕业生去向，共计 2676 份数据。每条数据包含该学校某一年份的综合就业率、升学率、出国率，以及地域统计数据、行业统计数据和企业统计数据。

SchoolUrlData 存储院校实拍，描述院校实况。每条数据包含院校 id、头像 url 和院校实拍列表，列表中每一个元素项包含名称如体育场和相应的 url 链接。共 2788 份数据。

5.2.2 录取信息

MajorDataPro 存储原始录取分数数据信息，每条数据包含院校名称、专业名称、专业 ID、录取要求、招生省份、招生年份、最低投档线、最低录取名词和招生人数。共 2590184（约 260 万）条数据。

5.2.3 一分一档表

ScaleData 表又称一分一档表，存储近三年的位次信息，每条数据包含分数、省份、选科类型、位次信息列表和同分数段的人数，共 159 条数据。

5.2.5 用户信息

User 表存储了用户信息，包括账号、加密后的密码、电话、省份、分数数据，以及 MBTI 人格测试和哈兰德职业兴趣测试的数据。

5.2.6 专业信息

MajorID 存储专业基本信息，包括细分专业名称、编号、其所在的 1 级学科目录、2 级学科、3 级学科等信息，共计 1524 条数据。

MajorInfoData 存储专业详细信息，同 MajorID 一样具有 1524 条数据。

MajorRank 存储专业排名数据。这里选取了第四次学科排名的信息。

5.3 数据爬取模块详细设计

本项目以 scrapy 框架为基础，对每一项所需数据在 spiders 中构建其对应的解析类，同时在 items 中创建其对应的数据类。Settings 中存放项目的各项配置。pipeline 中存放对 spider 获取数据的处理。外层使用 entrypoint 脚本进行快速运行。具体类文件见第六章详细设计。

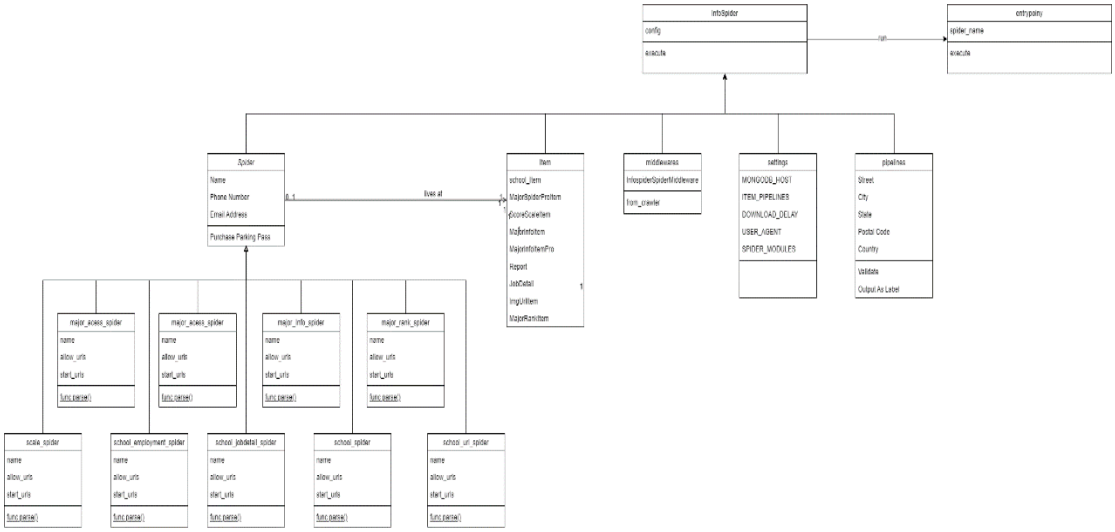

图 5-4 数据爬取模块类图

5.4 本章小结

本章主要对数据库的管理、字段设计进行了详细说明，并对涉及数据库的 Python 爬虫子系统进行了简单的概要设计赘述。

第六章 爬虫子系统的具体实现

该部分涉及项目数据系统的具体逻辑结构与实现。第一部分为请求分析，确定项目数据源；第二部分为 Scrapy 的具体实现，包括指定数据爬取的 Spider 模块、负责结果与数据库交互的 Item 模块、包含各个配置项的 settings 模块和负责 response 处理的 pipeline 模块；第三部分介绍项目实际运行所需的一些额外工作，如运行脚本、多线程、匿名请求、服务区部署等操作。

6.1 请求分析

本系统数据来源于掌上高考（https://www.gaokao.cn/）和中国教育在线（https://www.zhijiao.cn/）。截止文章完成，掌上高考网站收录了直至 2022 年的高考各类信息，包括院校信息、专业信息、招生信息等。中国教育在线提供了各院校各学科的学科评估数据。Chrome 浏览器的调式界面的 Network 版块展示了网站的所有网络请求。

经汇总分析，网路请求主要分为两类，分别为静态请求和动态请求，具体展示如下：

表 6.1 源网站接口分析区别

接口类型	**动态**	**静态**
前缀	api.eol.cn	static.*.cn
传参方式	param 传参	作为 url 一部分
传参类型	字典	数字编号
有无密文策略	部分有	无
是否限制 IP	被系统检测后会暂时限制访问约 2 小时	无，完全不受限制
是否限制 Header	有，未伪装的请求会被禁止	有，未伪装的请求会被禁止

6.2 该项目数据系统的各个模块

该项目的数据系统采用 Scrapy 框架获取数据，MongoDB 进行数据存储。其中 Scrapy 主要使用了 spiders 模块、items 模块、pipelines 模块和 settings 模块。

6.2.1 Spider 模块

Spider 模块是 Scrapy 最关键的部分，在这一部分中 scrapy 处理全部的网络请求，提取数据，获取 Item 字段需要的数据，并将需要跟进的 URL 提交给引擎，再次进入调度器。在 Spider 中指定了数据获取的范围、数据解析方式和数据返回结果。对于不同的数据，采用不同的 Spider 进行爬取和解析。本部分共 9 个类（即 9 个文件），分别为获取专业信息编码的 major_-id_spider，获取院校专业在不同省份不同年份录取信息的 major_acess_spid-er，获取该专业相关介绍的 major_info_spider，获取不同专业学科评估参评院校及结果的 major_rank_spider，获取某年份各省份一分一档表的 scale_sp-ider，获取学校基本信息的 school_spider，获取该院校近年来毕业生就业报告链接的 school_employment_spider，获取学校就业数据的 school_jobdetail_spider 和获取院校实拍的 school_url_spider。下文将依次展开各个 Spider 类。

表 6-2 spider 各个子类功能描述

类 型	描 述
school_spider	院校的详细信息
school_employment_spider	院校就业质量报告
school_jobdetail_spider	院校毕业生就业统计
school_url_spider	院校实景图
major_id_spider	专业编码和基本信息
major_acess_spider	专业录取信息（包括某院校某专业在某年某地区的录取分数和专业编码）
major_info_spider	专业详细信息
major_rank_spider	学科评估参评院校及评审结果

scale_spider	各省历年一分一档表

SchoolSpider 以 https://api.eol.cn/gkcx/api/?page=1&request_type=1&signsafe=&size=20&sort=view_total&top_school_id=&type=&uri=apidata/api/gk/school/lists 为起始请求，获得院校信息列表作为后续请求的 id 来源，同时供用户数据查询使用。其中 parse 函数定义了获取数据的递归方法，detail_parse 函数返回获得数据的具体结果。

school_employment_spider、school_jobdetail_spider 和 school_url_spider 三个类结构和内容较为一致，仅请求链接不同，故合并展示说明。这里以 SchoolEmploymentReportSpiderSpider 为例，在该类中首先取得前文获取的院校 id，然后根据列表中的 id 完善请求链接并逐个添加到爬虫队列以获取院校的就业情况、毕业生就业报告和院校实景图。

MajorIDSpider 中，专业分为专科专业和本科专业，每个层次对应不同的网络请求。这里分别将对应的请求添加到队列后，执行请求。Parse 请求不再赘述。

MajorAcessInfoSpider 表对应各院校各专业在各地区不同年份的录取状况，由于参数较多因而相对复杂，这里采用循环嵌套和递归结合的方式执行请求，嵌套在外层，内省执行递归。同时由于算法需要将选科信息转化为 01 序列便于对接。

MajorInfo 通过 major_id 获取专业编码，在根据编码获取专业的详细信息。其逻辑结构与协作关系类似 school_spider 和 school_jobdetail_spieder，这里不再赘述。

major_rank_spider 根据信息网站获取学科评估结果，根据其不同大类的专业编码获取请求链接，并根据链接执行 request 请求。

scale_spider 与 school_spider 类似，不再赘述。

6.2.2 items 模块

Item 模块负责明确需要抓取的目标，spider 根据 item 类创建对象并返回结果。通常每个 spider 对应一个 item，但 item 也可以复用或是一个 spider 获得多个 item。下文将以 MajorSpiderPro 为例进行展示，其他 item 类结构类似仅属性不同不在重复。

表 6-3 各个 item 对应的 spider

Spider	Item
school_spider	SchoolItem
school_employment_spider	ReportItem
school_jobdetail_spider	JobDetailItem
school_url_spider	ImgUrlItem
major_id_spider	MajorInfoItem
major_acess_spider	MajorSpiderProItem
major_info_spider	MajorInfoItemProItem
major_rank_spider	MajorRankItem
scale_spider	ScoreScaleItem

6.2.3 pipelines 模块

pipeline 模块负责设计管道存储爬取内容，包括处理 Spider 中获取到的 Item，并进行进行后期处理（详细分析、过滤、存储等）的地方。在该项目中，我们主要使用的是 open_spider 函数，process_item 函数和 close_spider

函数，其中 open_spider 用于爬虫开始之前进行参数配置、process_item 用于处理返回的 item，close_spider 用于 spider 运行结束后的处理。

在 open_spider 中首先取得 settings 中定义的数据库参数，并通过参数和 MongoDB 建立连接。中间增加了依据运行 spider 名称自动识别数据表的判断语句，避免手动修改配置项。若网页蜘蛛名称为 major_info_spider，则指定数据库表项为 MajorInfoData，否则检查下一项，直到查询到对应的 spider_name 或是检查结束，使用 test 数据库等待处理

表 6-4 各个 spider 对应的 colname

spider_name	colname
school_spider	SchoolBasicData
school_employment_spider	SchoolEmploymentReport
school_jobdetail_spider	school_jobdetail_spider
school_url_spider	school_url_spider
major_id_spider	MajorId2.0
major_acess_spider	MajorDataPro
major_info_spider	MajorInfoData
major_rank_spider	MajorRank
scale_spider	ScoreScale
其他	Test

process_item 对得到的数据项进行检查和处理，并将返回的 item 插入数据库。数据的处理主要包括清洗和位次波动统计。

数据清洗指通过对爬取到的信息进行检查、筛选、去重和格式化等操作，用以保证数据的质量和准确性。为了避免无效数据项影响，该爬虫系统对缺失信息标记为“-1”，便于后续的处理和运行报错。同时，对于录取信息中的选科信息，为了便于算法逻辑处理，该项目将不同选科信息转化为二进制字符串用于算法输入。

位次波动统计主要针对学校历年的录取位次，通过统计近年来该院校的录取位次波动情况并计算方差，对该院校的录取情况作出定性说明用以提醒填报考生。

close_spider 关闭数据库连接，避免资源占用。

6.2.4 settings 模块

Settings 模块用于配置项目的各项参数，包括项目名称、请求头、协议、请求间隔、pipeline 及其权重以及数据库连接参数。其中 BOT_NAME 指明了运行的项目名称，SPIDER_MODULES 指明运行的网络蜘蛛集合，USER_AGENT 中设置爬虫的网络请求头，DOWNLOAD_DELAY 用于设置请求时间间隔，通常用于限定访问的网站，减轻服务器负担。ITEM_PIPELINES 设置数据处理管道并指定优先级，这里我们把项目的通用处理管道 InfospiderPipeline 优先级设为 300，对图片处理管道 ImgPipeline 优先级设为 200。最后添加数据库的连接信息，便于 pipeline 中创建连接请求时切换和复用。

```
MONGODB_HOST = '43.137.3.***'
MONGODB_PORT = 28018
MONGDB_USERNAME = '***'
MONGODB_PASSWORD =    '******'
MONGODB_AUTHENTICATION = 'admin'
MONGODB_DBNAME = 'cpirsystem'
```

6.3 实际部署

6.3.1 运行脚本

在项目中添加自动运行脚本 entrypoint，便于调试和自动运行。其中 execute 是系统的执行函数，scrapy 指定命令行执行程序，crawl 代表通过 scrapy 引擎执行爬虫，最后 school_spider 是具体的爬虫名称。通过将多个爬虫执行语句放入脚本构建项目的快速运行脚本。

```
execute(['scrapy','crawl','school_spider']
```

6.3.2 多线程

对于爬虫中的多线程，通常采取切分 url 或是切分项目的方式。Scrapy 自身使用 Twisted 异步网络框架处理并发请求，并以此设置多线程。中途尝试使用 Python 的 thread 函数库自定义多线程方式。CONCURRENT_REQUESTS 设置了最大同时并发请求数，CONCURRENT_REQUESTS_PER_DOMAIN 设置了每个数据域允许的请求数，CONCURRENT_REQUESTS_PER_IP 设置了每个 IP 允许的请求数。

```
CONCURRENT_REQUESTS = 32
CONCURRENT_REQUESTS_PER_DOMAIN = 32
CONCURRENT_REQUESTS_PER_IP = 32
```

6.3.3 更改请求头

对于本项目所涉及的站点，需要模拟浏览器请求信息进行数据获取。本项目中使用 fake_useragent 内的 UserAgent 设置 headers 设置。

```
USER_AGENT = UserAgent().random
```

6.3.4 服务器部署

由于爬虫过程不可中断，对于较大数据需要将项目部署到服务器运行。将项目克隆到服务器之后，在项目目录下建立运行脚本，并用 nohup 执行程序，即可实现服务器上的不间断运行。

```
nohup python -u exe.py > .log 2>1& &
```

其中&代表重定向，即将前者的输出作为后者的输入。0 代表标准输入 stdin (standard input)，1 代表标准输出 stdout (standard output)，2 代表标准错误 stderr (standard error)；2>&1 代表将标准错误重定向到标准输出，标准输出再被重定向输入到 out.log 文件中。

第七章 总结和展望

7.1 总结

随着互联网时代院校信息的爆炸增加、人们对高考报考的重视提升，如何快速获得关注院校的信息、如何根据自身兴趣与分数快速匹配心仪的院校专业成为一个亟待解决的问题。针对该社会现象，我们团队设计并开发了高考志愿智能推荐系统，该系统能够便捷直观的展示、检索院校和专业的各类信息，并向考生推荐与考生兴趣、省份位次匹配的专业和院校。

本文基于 scrapy 框架、应用 MongoDB 进行存储，实现了高考志愿填报推荐系统的爬虫子系统，能够对各类相关信息进行获取、处理和存储。该子系统可以在不同的系统环境下运行，并可以通过脚本自动运行。

7.2 展望

目前为止，该系统已能够进行较为全面的信息展示、并根据考生进行院校推荐和专业推荐，但受限于时间、精力，该系统未能做到完美，有些方面仍存在改进的空间。

首先是网络爬虫的负载均衡模块，若果未来继续拓展系统功能、数据量进一步增加，目前通过本地/服务器异步运行的方式可能无法满足需求，需要尝试分布式运行，对多个网络链接进行去重、分配。

其次系统的算法部分，由于用户兴趣与专业匹配涉及协同过滤算法和相似推荐，难免产生冷启动的问题。具体表现为早起系统样本较少，产生的推荐较为刻板。随着样本数量增加和系统的实际应用，该问题有望得以解决。

此外，该项目爬虫子系统的健壮性有待进一步提升。目前项目各个模块间还存在一定依赖，一旦某一模块出现故障可能会影响其他部分的功能。未来可以进一步优化代码逻辑和实现。

参考文献

[1] 杨玉春.从新高考透视我国公共教育政策走向[J].北京师范大学学报(社会科学版),2021(04):74-81.

[2] 孙立伟,何国辉,吴礼发.网络爬虫技术的研究[J].电脑知识与技术,2010,6(15):4112-4115.

[3] 李玺作.爬虫逆向进阶实战[M].北京：机械工业出版社,2022

[4] 潘晓英，陈柳，余慧敏，赵逸喆，肖康泞.主题爬虫技术研究综述[J].计算机应用研究,2020,第 37 卷(4): 961-965，972

[5] 周立柱,林玲.聚焦爬虫技术研究综述[J].计算机应用,2005(09):1965-1969.

[6] 陈丛，周力臻.基于 Python 爬虫技术的虚假数据溯源与过滤[J].计算机仿真,2021,第 38 卷(3): 346-350

[7] 孙川钰，朱镕申，黎秀.基于 Python 语言的网络爬虫 KMR 研究[J].计算机仿真,2023,第 40 卷(3): 504-507

[8] 罗春.基于网络爬虫技术的大数据采集系统设计[J].现代电子技术,2021,第 44 卷(16): 115-119

[9] 郭一峰. 分布式在线图书爬虫系统的设计与实现[D].北京交通大学,2016.

[10] 胡博. 基于网络爬虫的内容资源评价研究[D].北京理工大学,2015.

[11] 赵鹏程. 分布式书籍网络爬虫系统的设计与实现[D].西南交通大学,2014.

[12] 王柏琦. 基于多特征权重的新高考志愿填报系统的设计与实现[D].中国科学院大学(中国科学院沈阳计算技术研究所),2022.DOI:10.27587/d.cnki.gksjs.2022.000035.

[13] 王泽卿,季圣鹏,李鑫,赵子轩,王鹏旭,韩霄松.基于分数线预测的多特征融合高考志愿推荐算法[J].计算机科学,2022,49(S2):254-260.

[14] 白俊杰. 基于混合推荐的高考志愿推荐系统的设计与实现[D].内蒙古大学,2022.DOI:10.27224/d.cnki.gnmdu.2022.001490.

[15] 王柏琦,付立军,周晓磊,高思达,张永宏.新高考志愿推荐算法研究[J].中国教育信息化,2023,29(04):112-120.

致谢

当深夜笼罩的我写下这段话的时候，我的本科生涯已接近尾声。踏入校园的光景仿佛仍在昨日，而我即将和这里的四年告别。人们称呼这里的学生为天之骄子，大家谈论的是“建设第一个南大”、“做最好的本科教育”，这里留下的是主席对青年学子的殷切期盼和谆谆教诲。在这样的环境中我受益匪浅、成长良多。在这大学四年的最后阶段，感谢葛季栋老师在整个毕业设计的过程中对项目的耐心指导和关注，并为项目提出诸多建设性的建议；感谢范子君、任毅、吴子玥三位队友，在完成项目的过程中共同进步、成长。感谢我的朋友们在项目完成过程中的帮助。感谢我的家人，为我提供坚强的后盾。最后，感谢学校给予我优秀的平台，让我不断进步与成长。

风雨四载，金陵一梦，时常庆幸能与诸位相逢。前路漫漫，愿大家前途璀璨，初心未泯，多年后依旧相逢如故。