**毕业设计（论文）**

|  |  |
| --- | --- |
| 课题名称 | 基于Spark的河北省旅游景区推荐 |
| 系统的设计与实现 |
| 学 院 |  |
| 专 业 |  |
| 学生姓名 |  |
| 学 号 |  |
| 指导教师 |  |
| 职 称 |  |

2023年 5 月 29 日

**学生毕业设计（论文）原创性声明**

本人以信誉声明：所呈交的毕业设计（论文）是在导师的指导下进行的设计（研究）工作及取得的成果，设计（论文）中引用他（她）人的文献、数据、图件、资料均已明确标注出，论文中的结论和结果为本人独立完成，不包含他人成果及为获得重庆工程学院或其它教育机构的学位或证书而使用其材料。与我一同工作的同志对本设计（研究）所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

毕业设计（论文）作者（签字）：

2023年 5 月 29 日

摘 要

随着国家不断发展和进步，人们物质水平也在不断提高。与此同时，人们也更加趋于追逐精神生活的富足。满足自身精神层面富足的方式有很多，而大多数人会选择旅游的方式来进行精神层面的放松。 旅游可以适当满足自身的情感需求，放松身心，使精神层面解除紧绷的状态。但是，在如今快节奏的时代里，人们的假期大都是有限并短暂的，并不足以让人们充分了解景区信息，制定自身旅游计划，充分满足自身需求。

同时，河北省作为一个拥有丰富自然和人文景观资源的地区，也吸引了大量的游客。然而，由于游客数量庞大和景区众多，游客如何选择合适的景区成为一个难题。因此，设计并实现一套基于Spark的河北省旅游景区推荐系统，将能够提供游客个性化的推荐服务，帮助游客更好地规划旅游行程，提升旅游的体验。

Spark技术具有快速处理大规模数据、支持多种语言、具有灵活性和拓展性、并且能应用于实时数据流场景的特点。因此通过Spark技术，可以实现对旅游数据的实时监测。基于此，本设计旨在通过数据爬取、数据存储和Spark技术实现一个基于Spark的便捷的、高效的河北热门旅游景区推荐系统。

关键词：Spark；旅游退货；ALS；协同过滤

**ABSTRACT**

With the continuous development and progress of the country, people's material level is also constantly improving. At the same time, people are also more inclined to pursue spiritual prosperity. There are many ways to satisfy one's spiritual abundance, and most people choose to travel for spiritual relaxation. Tourism can appropriately satisfy one's emotional needs, relax the body and mind, and relieve mental tension. However, in today's fast-paced era, people's vacations are mostly limited and brief, not enough to fully understand the information of scenic spots, develop their own travel plans, and fully meet their own needs.

Meanwhile, as a region with abundant natural and cultural landscape resources, Hebei Province has also attracted a large number of tourists. However, due to the large number of tourists and numerous scenic spots, it has become a challenge for tourists to choose suitable scenic spots. Therefore, designing and implementing a Spark based recommendation system for tourist attractions in Hebei Province will provide personalized recommendation services for tourists, help them better plan their travel itinerary, and enhance their tourism experience.

Spark technology has the characteristics of fast processing of large-scale data, support for multiple languages, flexibility and scalability, and can be applied to real-time data flow scenarios. Therefore, through Spark technology, real-time monitoring of tourism data can be achieved. Based on this, this design aims to achieve a convenient and efficient recommendation system for popular tourist attractions in Hebei Province through data crawling, data storage, and Spark technology.

**Keywords:** Spark; Travel returns; ALS; Collaborative filtering

目 录

[1 绪 论 1](#_Toc16877)

[1.1 研究背景及意义 1](#_Toc30951)

[1.1.1 研究背景 1](#_Toc10908)

[1.1.2 研究意义 1](#_Toc29313)

[1.2 国内外研究现状 1](#_Toc8434)

[1.2.1 国外研究现状 1](#_Toc27570)

[1.2.2 国内研究现状 1](#_Toc4076)

[1.3 本文主要任务 1](#_Toc16831)

[1.4 论文组织结构 2](#_Toc3315)

[1.5 本章小结 2](#_Toc5314)

[2 相关理论及主要技术 3](#_Toc24258)

[2.1 大数据计算平台 3](#_Toc17260)

[2.1.1 Hadoop 3](#_Toc22690)

[2.1.2 Spark 3](#_Toc28113)

[2.2 系统前端技术 3](#_Toc5249)

[2.2.1 Vue 3](#_Toc32685)

[2.2.2 Element-plus 3](#_Toc10914)

[2.3 系统后端技术 3](#_Toc1478)

[2.3.1 Django 3](#_Toc15360)

[2.5 推荐算法简介 4](#_Toc28496)

[2.5.1 APH层次分析法 4](#_Toc3225)

[2.5.2 ALS算法 4](#_Toc9087)

[2.5.3 大致流程 4](#_Toc5435)

[2.6 系统环境 5](#_Toc21680)

[2.7 本章小结 5](#_Toc28807)

[3 需求分析 6](#_Toc1486)

[3.1 系统流程分析 6](#_Toc25928)

[3.2 系统角色 6](#_Toc7129)

[3.2.1 系统管理员 6](#_Toc8278)

[3.1.2 用户 6](#_Toc25793)

[3.1.3 游客 7](#_Toc653)

[3.3 系统用例 7](#_Toc1517)

[3.3.1 系统管理员 7](#_Toc19378)

[3.3.2 用户 7](#_Toc6183)

[3.3.3 游客 8](#_Toc28977)

[3.4 系统非功能需求 9](#_Toc9159)

[3.5 本章小结 9](#_Toc10266)

[4 系统设计 10](#_Toc1871)

[4.1 系统架构 10](#_Toc9168)

[4.1.1 系统架构设计原则 10](#_Toc25565)

[4.1.2 系统架构设计 10](#_Toc22327)

[4.2 系统功能设计 11](#_Toc17939)

[4.2.1 游客 12](#_Toc13427)

[4.2.2 用户 12](#_Toc14187)

[4.2.3 管理员 12](#_Toc1901)

[4.3 数据库设计 12](#_Toc789)

[4.3.1 概念结构设计 13](#_Toc10687)

[4.3.2 物理结构设计 13](#_Toc4814)

[4.4 数仓设计 17](#_Toc689)

[4.5 本章小结 17](#_Toc11081)

[5 系统实现 19](#_Toc7276)

[5.1 数据采集与存储 19](#_Toc19757)

[5.1.1 实现步骤 19](#_Toc8090)

[5.2 数仓建模 21](#_Toc26292)

[5.2.1 ODS层 21](#_Toc20729)

[5.2.2 DWD层 21](#_Toc9055)

[5.2.3 ADS层 22](#_Toc18618)

[5.3 推荐算法实现 22](#_Toc21107)

[5.3.1 APH层次分析法求权重 22](#_Toc6558)

[5.3.2 带权重的余弦相似度算法 22](#_Toc25097)

[5.3.3 模型训练 22](#_Toc3630)

[5.4 系统功能实现 23](#_Toc7166)

[5.3.1 用户管理模块 24](#_Toc5251)

[5.3.2 系统首页模块 25](#_Toc31495)

[5.3.3 数据检索模块 26](#_Toc30068)

[5.3.4 推荐模块 26](#_Toc25006)

[5.3.5 其他功能模块 28](#_Toc6703)

[5.3.6 后台管理模块 30](#_Toc2574)

[5.4 本章小结 31](#_Toc31613)

[6 系统测试 32](#_Toc31869)

[6.1 功能测试 32](#_Toc28739)

[6.2 性能测试 34](#_Toc3215)

[6.3 本章小结 34](#_Toc32003)

[7 总结与展望 35](#_Toc5902)

[7.1 全文总结 35](#_Toc5224)

[7.2 展望 35](#_Toc20702)

[参考文献 37](#_Toc31005)

[致 谢 38](#_Toc13805)

1 绪 论

1.1 研究背景及意义

1.1.1 研究背景

随着社会发展，人们对精神生活的追求日益增加[1]，旅游成为满足这种需求的重要方式。然而，快节奏的生活使得游客的假期有限且短暂，难以有效规划旅游行程，尤其对于河北省这样拥有丰富景观资源的地区更为突出[2]。因此，基于Spark技术的旅游景区推荐系统的设计和实现具有重要意义，能够为游客提供个性化的推荐服务，提升旅游体验[3]。

1.1.2 研究意义

设计并实现一套基于Spark的河北省旅游景区推荐系统，将能够提供游客个性化的推荐服务，帮助游客更好地规划旅游行程，提升旅游的体验；并且还能在有效的时间内，弥补自己对旅游景区的不了解的不足[4]，制定令人满意的旅游计划，迎接愉快的度假。

1.2 国内外研究现状

1.2.1 国外研究现状

在国外，旅游推荐系统已经得到了广泛的应用。一些主流的旅游网站，如TripAdvisor、Expedia和Booking.com等[5]，已经在大力推广和运用旅游推荐系统。这些推荐系统主要基于协同过滤、内容过滤和混合过滤等技术，通过对用户的历史浏览记录、评价及喜好偏好进行分析，从而给出符合用户需求的推荐产品[6]。

1.2.2 国内研究现状

随着互联网的发展和旅游行业的兴盛，国内的旅游推荐系统也逐渐得到了关注和发展。目前，携程、去哪儿等大型在线旅游平台及一些小型旅游网站[7]，也主要通过协同过滤、内容过滤和混合过滤等技术实现推荐系统的应用。然而，目前国内旅游系统也有一些瑕疵。例如对于景区选择性推荐还不够完善。由于景区特色的不同，其适应的旅游季节也会有所不同[8]。因此，对于景区选择性推荐问题还需不断完善。

综上所述，旅游景区推荐系统在国内外都得到了广泛的关注和研究。通过spark和协同过滤的技术，这些系统能够提供个性化的旅游推荐服务，提升用户的旅游体验。而基于spark的河北省旅游景区推荐系统也借鉴了国内外所使用的良好算法，例如协同过滤算法，也可以弥补景区选择性推荐不够完善的问题[9]。

1.3 本文主要任务

本文的主要任务是通过以下几个步骤构建一个完整的就业推荐系统：

python爬虫获取河北省旅游景区信息，对所获取的数据进行清洗等，并将其存储到数据库中；

通过系统问卷的方式获取用户的年龄、喜欢的主题、旅游作息、旅游月份以及其他相关信息；

使用spark技术、django+vue框架、协同过滤算法等进行系统的搭建和实现；

1.4 论文组织结构

本文的组织结构如下：

第一部分是绪论。这一部分主要阐述了本文的研究背景和意义，描述国内和国外研究的现状，最后构建一个基于Spark的就业推荐系统。

第二部分是相关理论及关键技术。这一部分主要介绍了实现本文系统所需要采用的系统开发技术、系统开发工具以及相关的算法理论。

第三部分是对系统的需求分析。这一部分介绍该研究对象的业务背景，以及基于该业务背景所提出的需求，建立出与其相关的业务流程、系统角色分析、用例图、用例描述，并对其非功能性要求进行说明。

第四部分是系统的分析与设计。这一部分在系统的需求分析基础上，对系统架构进行了描述，详细介绍了系统的基本功能设计，包括系统层次结构图、功能模块设计、数据库设计，并设计了原型系统的数据库模式和系统的UI设计。

第五部分是系统的具体实现。这一部分介绍了数据的采集及处理，阐述大数据计算平台的搭建，说明系统主要功能模块的实现细节。

第六部分是系统测试。在这一部分中，分别使用接口测试和功能测试技术来测试系统的主要接口和功能。

第七部分是总结与展望。在这一部分中，回顾整个项目，发现其中的不足，总结了其中的经验教训，并考虑未来的改进和优化。

1.5 本章小结

本章旨在介绍研究的背景和意义，涵盖了全球就业市场的激烈竞争、年轻人和毕业生的就业困难，以及大数据和人工智能技术对人才匹配的需求。特别强调了就业推荐平台作为解决传统就业市场问题的必要选择，其为求职者和企业提供便捷、高效匹配服务的优势。研究意义方面，突出了就业推荐平台提高求职成功率、优化招聘流程以及提供市场情报和数据分析的重要性。此外，对国内外研究现状进行了综述，指出了国外就业推荐系统向个性化和智能化发展的趋势，以及国内互联网招聘市场的优势和不足。随后，明确了本文的主要任务，包括数据爬取与处理、数仓建模和数据同步、推荐算法实现、后端接口和前端界面开发等。最后，给出了本文的组织结构，概述了后续章节将对相关理论和关键技术进行详细介绍和阐述。

2 相关理论及主要技术

2.1 大数据计算平台

2.1.1 Hadoop

Hadoop是一个开源的大数据处理平台，由Apache基金会开发和维护。其核心技术包括Hadoop Distributed File System (HDFS) 和 Yet Another Resource Negotiator (YARN)。HDFS是一种分布式文件系统，设计用于在商用硬件上运行，能够处理海量数据并实现高容错性和高可用性。而YARN是Hadoop的资源管理和作业调度系统，负责管理集群中的计算资源，并根据应用程序的需求进行分配。

2.1.2 Spark

Apache Spark是一个快速、通用、可扩展的分布式计算框架，特别适合大规模数据处理和实时分析。Spark的核心技术包括Resilient Distributed Datasets (RDDs)、In-Memory Computing以及Spark Core和一系列库。RDDs是Spark的基本数据结构，提供容错机制；In-Memory Computing利用内存进行数据处理，提高速度；Spark Core是基础，包含任务调度、内存管理等功能；而Spark SQL、MLlib、GraphX和Spark Streaming等库为各种数据处理和分析任务提供支持。

2.2 系统前端技术

2.2.1 Vue

Vue.js是一个流行的前端JavaScript框架，用于构建用户界面和单页应用。Vue的核心技术特点包括Reactive Components、Virtual DOM以及Templating and Directives。Vue通过组件化的方式组织UI，每个组件都是独立且反应式的；使用虚拟DOM优化渲染性能，仅更新实际发生变化的部分DOM；提供易于理解的模板语法和一系列指令，方便声明式地定义组件的结构和行为。

2.2.2 Element-plus

Element Plus 是一个基于 Vue.js 的开源 UI 组件库，是对 Element UI 的升级和扩展。它提供了一套美观、易用且高性能的组件，旨在帮助开发者快速构建现代化的 Web 应用程序。它继承了 Element UI 的优点，包括丰富的组件库、灵活的布局系统和完善的文档。同时，Element Plus 在功能和设计上进行了进一步的改进。它提供了全新的组件样式风格和更加现代化的外观，同时增加了一些新的组件和功能。支持自定义主题，开发者可以根据自己的需求来定制组件的样式和主题色，从而使应用程序更符合自身的品牌和设计风格。除此之外，Element Plus 还提供了丰富的指令、插件和工具集，帮助开发者更快速、高效地构建和维护 Vue.js 应用程序。

2.3 系统后端技术

2.3.1 Django

Django是一个高级的[10]、Python编写的Web框架，遵循MVC（Model-View-Controller）架构模式。Django的核心技术特性包括MTV Architecture、ORM (Object-Relational Mapping) 以及“Batteries Included”理念。Django采用Model-Template-View（MTV）架构，其中Model处理数据，Template负责呈现，View处理业务逻辑；内置强大的ORM系统，允许以Python对象的方式操作数据库；秉承“包含电池”的理念，提供了丰富的内置功能，如用户认证、权限管理、站点地图、RSS/Atom feeds生成等，简化Web应用的开发过程[11]。

2.5 推荐算法简介

本文介绍了一种结合了APH层次分析法和ALS算法的推荐系统算法，以解决多因素推荐问题。该算法流程包括定义权重列表、特征处理、标准化、权重计算、相似度计算、模型训练、性能评估和生成推荐结果等步骤。

在这一算法中，APH层次分析法用于计算不同特征的权重，以进行加权余弦相似度计算。同时，ALS算法用于学习用户和物品的特征向量，进一步计算相似度并进行个性化推荐。通过对用户画像和招聘信息的特征进行处理和标准化，并结合权重计算，该算法能够更精确地评估用户与招聘信息的相似度，并给出针对用户的推荐结果。

2.5.1 APH层次分析法

APH（Analytic Hierarchy Process）层次分析法是一种多准则决策分析方法，用于处理复杂的决策问题。它将问题分解为一系列层次结构，通过两两比较不同元素的重要性来建立判断矩阵，然后通过数学计算得出最终的决策结果。在推荐系统中，APH层次分析法可以用于处理多因素推荐问题，例如在考虑用户偏好、物品特征等因素时进行权重分配和决策。

2.5.2 ALS算法

ALS（Alternating Least Squares）算法是一种协同过滤推荐算法，常用于处理用户-物品矩阵进行推荐。ALS算法通过学习用户和物品的特征向量来计算相似度，并通过推荐最相似的物品给用户。在推荐系统中，ALS算法可以根据用户的画像和物品的画像来计算它们之间的相似度，从而进行个性化推荐。用户画像可以包括求职者简历中的期望薪资、期望城市等信息，物品画像可以包括物品的薪资、城市、类别等信息。

2.5.3 大致流程

本文提出的推荐算法大致流程如下：

1. 将用户画像数据和招聘信息数据进行特征处理生成用户画像和招聘信息的特征向量。
2. 对特征向量进行标准化。
3. 使用APH层次分析法计算出各个特征对应的权重。
4. 对用户画像向量和招聘信息特征向量进行笛卡尔积操作，并使用带权重的cos相似度算法得到所有用户和招聘信息对的相似度计算结果。
5. 将数据集拆分为训练集和测试集为8:2
6. 使用ALS模型，并使用训练集作为评分进行训练。
7. 使用测试集评估模型性能。
8. 为所有用户生成推荐结果并写入MySQL数据库。

2.6 系统环境

本系统数据采集、开发阶段在 Windows11 系统开发，数据库使用MySQL（5.7.26），编程语言选择Python （3.8），编程工具Pycharm。系统配置如表 2.1：

表2.1 系统配置表

|  |  |
| --- | --- |
| 项目 | 信息 |
| 操作系统 | Windows 11 |
| CPU | 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz |
| GPU | NVIDIA GEFORCE RTX 1060 |
| 内存 | 16 GB |
| 硬盘 | 1T |

2.7 本章小结

本章详细介绍了大数据计算平台、数据采集框架、系统前端技术、系统后端技术以及推荐算法等关键内容。在大数据计算平台方面，深入探讨了Hadoop和Spark两个平台的核心架构和生态系统。数据采集框架方面，介绍了Scrapy、Scrapyd和Gerapy等爬虫管理工具。系统前端技术部分涵盖了Vue、Vite和Element Plus等重要技术。后端技术采用了Django框架，并进行了全面介绍。推荐算法部分介绍了APH层次分析法、ALS算法和带权重的余弦相似度算法，以及算法的大致流程。最后，还介绍了系统的开发环境和配置。第三章将着重介绍系统的需求分析。

3 需求分析

3.1 系统流程分析

基于Spark的旅游景区推荐系统的工作流程可分为五个主要环节：数据采集、数据处理、模型训练、推荐生成和推荐结果展示。

在数据采集部分，首先确定需求并选择合适的采集对象，然后进行数据采集工作。

数据处理阶段，包括ETL(Extract, Transform and Load)过程，即抽取、清洗和加载数据，确保数据质量和一致性。

模型训练环节，其中使用APA层次分析法获取用户偏好的权重值，并利用此信息构建ALS推荐模型，用于个性化推荐。

推荐结果展示，通过后端(Django)与前端(Vue)之间的协作实现推荐结果的呈现，提供给用户参考和选择。

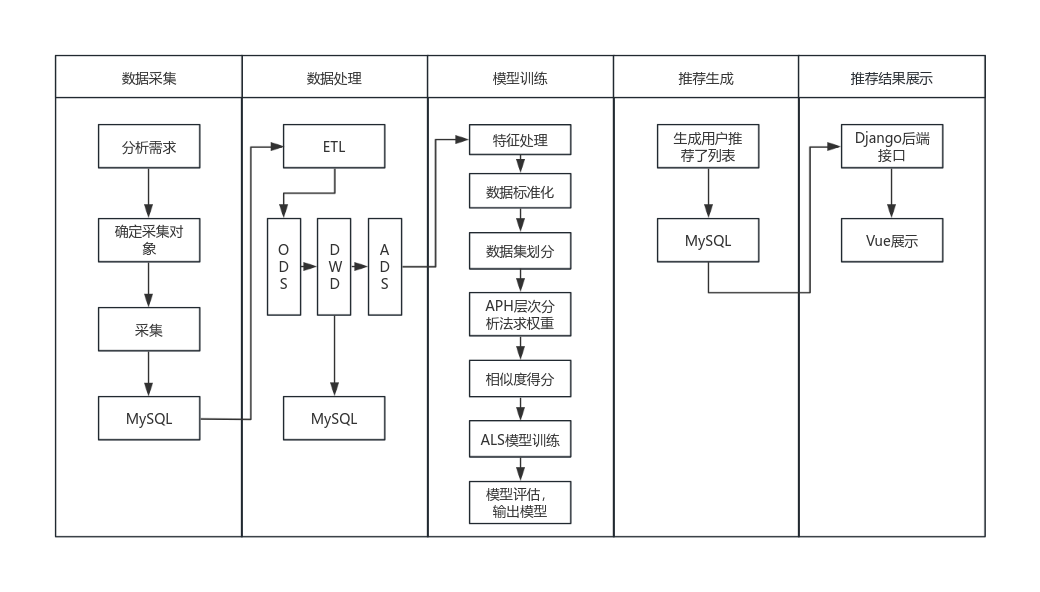


图3.1 系统业务流程图

3.2 系统角色

根据系统的业务流程，系统角色可分为两类：管理员、用户和游客。

3.2.1 系统管理员

系统管理员可以对系统模块进行管理，包括系统数据的增添、删除、查询、修改等操作，能够控制所有模块的相关权限，对所有系统功能进行操作。对普通用户有控制权，控制着普通用户的系统功能的使用。

3.1.2 用户

用户代表的是系统的一般使用人员，其可操作空间受管理员限制，某些系统功能的使用需要得到管理员的授权。在本系统中，用户可进行企业、职位等数据的浏览、查询、收藏、评论等操作。

3.1.3 游客

游客代表的是系统的体验用户，没有注册账号，只能浏览系统的极少部分页面。

3.3 系统用例

3.3.1 系统管理员

系统管理员在系统中的权限要大于普通用户的权限，系统管理员输入账号和密码登录到系统后台，拥有系统的所有权限，可直接对系统数据进行管理。

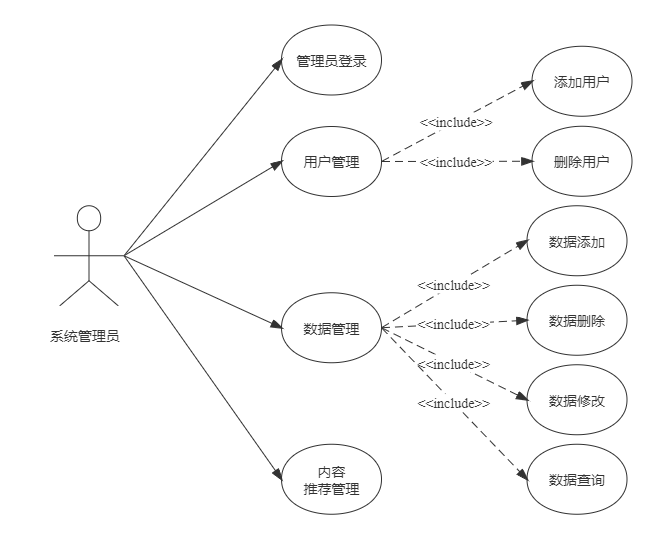


图3.2 系统管理员用例图

系统管理员可以对用户进行管理，包括添加用户和删除用户。对企业、职位等数据管理，包括数据的添加、删除、修改和查询。同时，还可以对推荐内容进行管理和更新。

3.3.2 用户

用户是系统的普通使用用户，权限远不及系统管理员，只能在系统进行已授权的操作，可通过输入正确的账号和密码登录到前端页面。

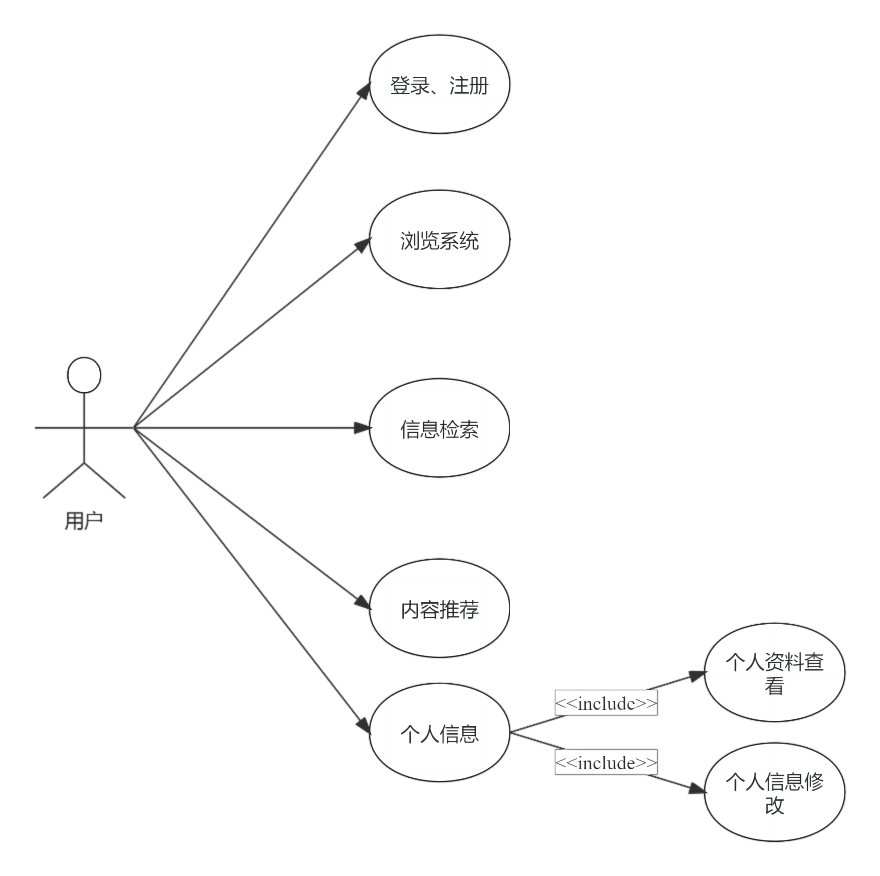


图3.3 用户用例图

如图3.3用户用例图所示，用户登录系统后，可以直接跳转到首页，包括热门企业、热门职位的推荐。用户可以通过系统的导航栏，可浏览推荐页面、企业查询页面、职位查询页面、我的信息页面。

3.3.3 游客

游客只能浏览系统的极少部分内容，其权限要少于用户的权限。

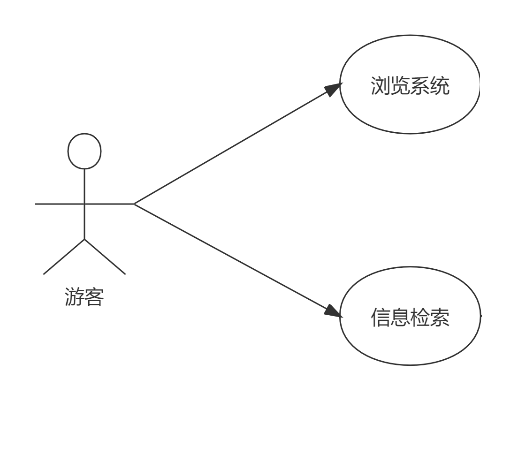


图3.4 游客用例图

如图3.4游客用例图，游客不用登陆系统，可体验系统的部分功能，如浏览、检索企业和职位等信息。

3.4 系统非功能需求

基于Spark的就业推荐系统包含的非功能性需求如下：

（1）系统的安全性要求

系统需要具备高安全性，确保用户数据和系统机密性得到保护。在用户注册账号时，要对用户的密码进行加密；在用户注册账号后，必须使用注册的账号和密码进行登录，系统还需在注册时判断账号是否已经被注册，防止注册重复。登录系统后，为了确保安全，防止账户密码被泄漏，不会使账户和密码出现在网页的浏览地址里面。用户需要登录和确认后才可以进入系统主页，反之则会出现错误提示。

（2）系统的可扩展性

系统需要具备高可扩展性，即能够轻松地进行扩展、升级以及介入更多的模块、插件或功能等。因此，在系统开发设计的过程中，需严格根据业务需求进行设计，增强系统的完整性。

（3）系统的可靠性

系统需要具备高可靠性，即能够持续稳定地运行，避免出现严重的系统错误或故障，否则一旦系统发生故障，会给用户带来严重的影响，导致业务中断、数据丢失、生产损失等问题。因此，系统需要使用高质量、可靠的硬件和设备，确保系统组件的稳定性和可靠性。

（4）系统的性能需求

系统需要具备高性能，即能够在高负载情况下保持良好的响应速度，用户可以快速地响应系统请求。因此，需要使用高效的编程语言和算法，提高系统运行效率，同时优化系统架构和系统设计，提高系统的并发处理能力和吞吐量。

3.5 本章小结

本章主要介绍系统的业务背景及流程，了解系统应该设计的方向和目标，阐述了系统角色和系统用例，并分析了系统的非功能性需求。第四章将要介绍系统设计。

4 系统设计

4.1 系统架构

4.1.1 系统架构设计原则

（1）简单性

系统的架构应该尽量简单，避免不必要的复杂性，造成用户体验不佳。这样不仅可以降低开发和维护的成本，减少出错的机会，也可以提高系统的稳定性和可靠性。

（2）可扩展性

本系统也将在后期逐步完善后续的需求功能，若在设计时考虑不周，会带来诸多麻烦。因此，一个良好的系统应该支持可持续的扩展，能够满足后续的业务需求，系统的各个模块应该独立可扩展，扩展模块应该有较低的风险和成本。

（3）安全性

安全性是一个系统需要重点考虑的方面，若被攻击造成数据泄露等问题，将会给用户带来不可挽回的损失。在系统设计中应该充分考虑安全方面的需求，实现可靠的安全控制系统。

4.1.2 系统架构设计

系统应该在满足用户需求的前提下，设计一个符合系统性能、可扩展、安全等要求。如下图所示：

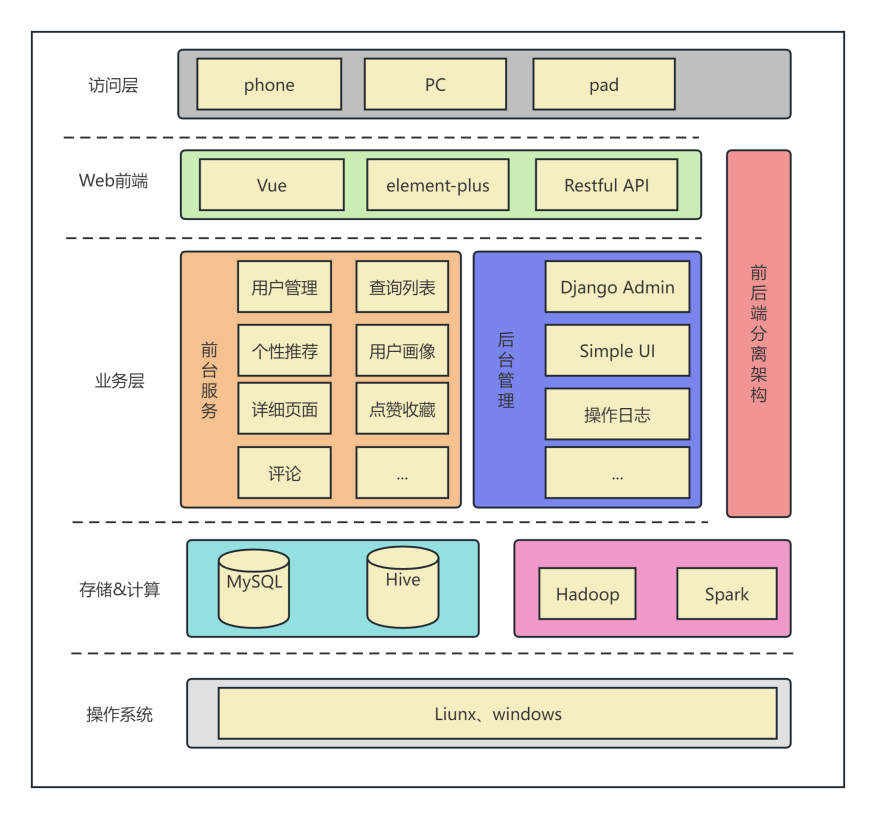


图4.1 系统架构设计图

本系统由四个主要层次组成：访问层、Web前端、业务服务以及存储与计算。

在最上一层是访问层，它包括Vue（一个流行的JavaScript框架），element-plus（一个UI组件库）和Restful API（一种网络应用接口）。这些工具提供了用户界面，并通过API与其他部分交互。

接下来的是Web前端，同样使用了Vue和element-plus作为开发工具。此外还引入了一个名为Django Admin的后台管理平台，这是Python Web框架Django的一部分，可以用来管理和维护网站内容。

再往下就是业务服务层，这部分包含了前后端分离的架构设计。在这个层级中，有一个个人化的推荐模块，允许对用户的兴趣爱好做出预测并提供个性化的建议；还有一个详细的页面展示功能，可能包含个人信息或者工作信息等详情；同时还有评论和点赞收藏等功能来增强用户体验。

最后，在底部是存储与计算层，这里采用了MySQL数据库管理系统和Hive数据仓库技术来进行数据处理和分析。而更深入的数据挖掘则依赖于Apache Spark的大规模分布式运算能力。

整个系统运行在一个Linux或Windows操作系统之上，为所有其他组成部分提供基础支持和服务。这种多层次的设计使得该系统能够有效地满足各种需求，从简单的查询到复杂的个性化推荐都能得到良好的实现。

4.2 系统功能设计

本系统的系统功能设计图如下图所示：

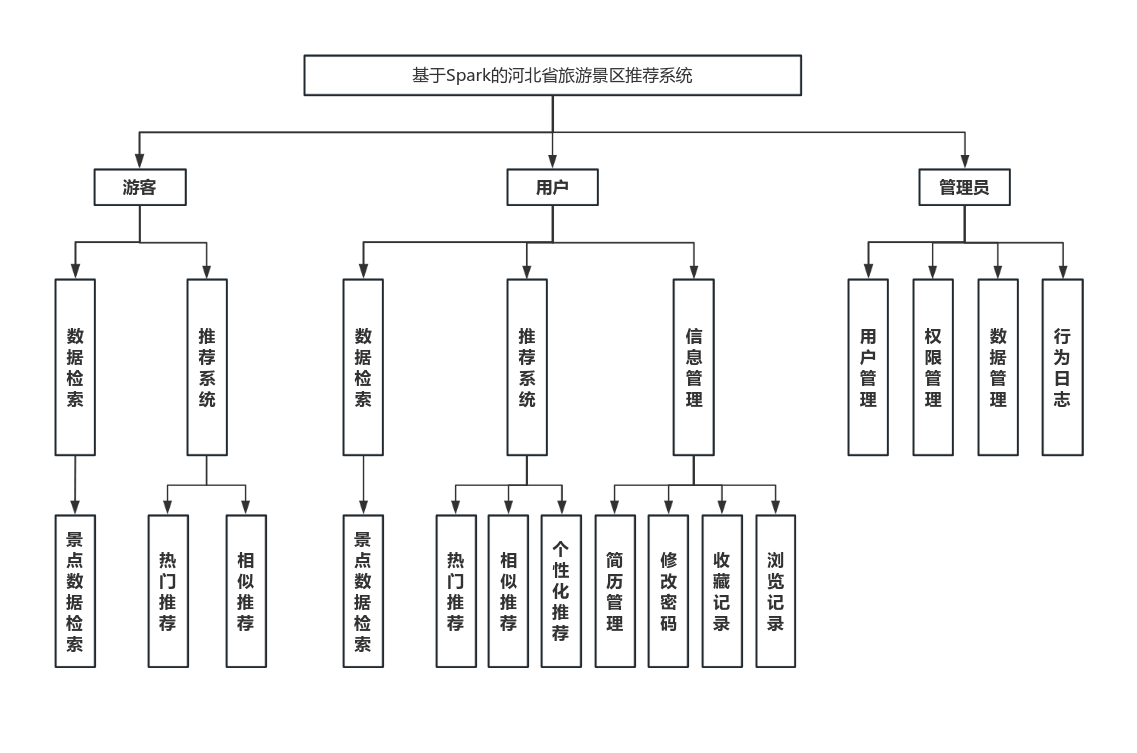


图4.2 系统功能设计图

系统的使用对用对象主要由用户和管理员组成，用户主要进行信息管理、内容浏览以及推荐浏览三个模块的功能，管理员主要进行信息管理、数据管理以及推荐管理三个模块的功能。各功能模块说明如下：

4.2.1 游客

①浏览系统首页包括轮播图、热门景点等等，方便用户快速了解当前最热门的旅游信息；

②浏览详细页，用户可以查看景点的详细信息，包括景点基本信息、景点描述、景点门票价格等等；

③景点详细页上，用户可以查看景点的基本信息、相似景点信息和评论区数据，以便更好地了解这个景点；

⑤注册功能，用户可以通过填写注册信息来创建自己的账号；

⑥登录功能，用户可以通过输入用户名和密码来登录自己的账号。

4.2.2 用户

①浏览系统首页：浏览系统首页包括轮播图、热门景点等等，方便用户快速了解当前最热门的旅游信息；

②浏览详细页：用户可以查看景点的详细信息，包括景点基本信息、景点描述、景点门票价格等等。

③景点详细页：用户可以查看景点的基本信息、相似景点信息和评论区数据，并可以进行评论和收藏操作，方便用户记录感兴趣的景点信息。

⑤推荐景点数据：系统通过实时推荐和匹配推荐的方式提供个性化的景点推荐，根据用户的兴趣和需求，推荐最适合的景点信息。

⑥景点喜好修改和查看：用户可以对个人景点喜好进行修改和完善。

⑦个人信息修改：用户可以随时修改个人信息。

4.2.3 管理员

①用户管理：管理员可以对系统中的用户进行管理，包括查看用户信息、新增用户、编辑用户信息、冻结或解冻用户账号等操作，以确保用户信息的准确性和安全性。

②数据管理：管理员可以对系统中的数据进行管理。

③权限管理：管理员可以设置用户的权限和角色，在系统中进行用户权限的分配和管理，以确保系统功能的合理使用和安全性。

④行为日志：管理员可以查看系统中的用户行为日志，包括用户的登录记录、操作记录等，以便进行系统安全监控和异常处理。

4.3 数据库设计

数据库设计是系统设计说明中不可缺少的一部分，需要结合业务需求进行综合、归纳，形成一个具体的数据库管理的模型。接下来将从概念结构设计、物理结构设计两个方面进行阐述。

4.3.1 概念结构设计

概念结构设计是一个在设计数据库、信息系统等复杂系统时，对其所涉及到的关键性概念进行梳理、抽象及组织的过程。能够将所得到的信息进行抽象和建模，制定出一个概念层面上的架构和框架，包括实体、关系、属性、约束等内容。为了更明确地理解系统业务中实体之间的关系及特性，本文通过E-R图将实体、实体的属性及实体间的关系表示出来。本系统的E-R图如下图所示：

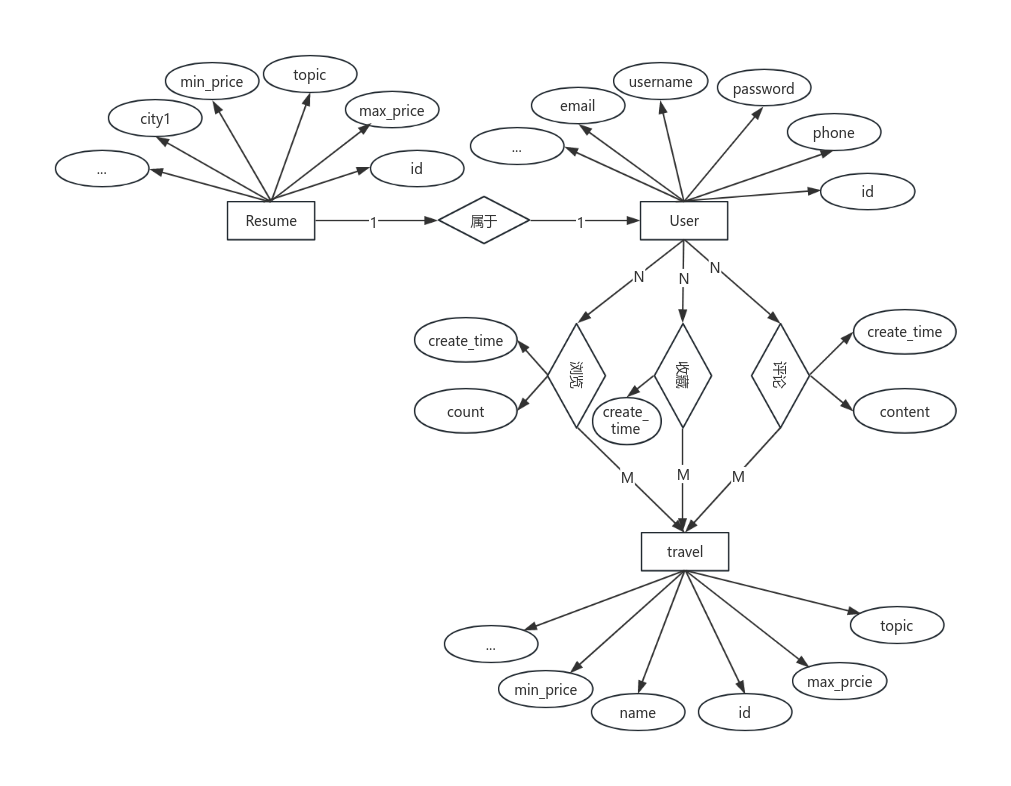


图4.3 系统E-R图

系统由用户、用户喜好、景点三个实体组成，用户浏览、收藏、评论景点，一个用户可以浏览、收藏、评论多个景点，一个景点也可以被多个用户浏览、收藏、评论。所以用户浏览、收藏、评论景点的三个行为都是多对多的关系。

4.3.2 物理结构设计

物理结构设计是在进行数据库设计时确定数据库实际的存储结构和物理组织方式的过程，它基于概念结构设计，决定如何在计算机硬件上实现概念结构设计。根据微博数据的关系特性，本文选择关系型数据库MySQL存储数据，主要的数据表及表结构如下所示：

表4.1 景点信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型**  **（精度范围）** | **空/非空** | **约束条件** |
| id | int(11) | NO | PRI |
| name | longtext | NO |  |
| topic | int(11) | NO |  |
| topicTranslation | longtext | NO |  |
| low\_price | float | YES |  |
| high\_price | float | YES |  |
| address | longtext | YES |  |
| city | int(11) | YES |  |
| cityTranslation | longtext | YES |  |
| url | longtext | YES |  |
| img | longtext | YES |  |
| create\_time | timestamp | YES |  |

表4.2 用户简历信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型**  **（精度范围）** | **空/非空** | **约束条件** |
| id | bigint(20) | NO | PRI |
| city1 | int(11) | YES |  |
| city1Translation | varchar(20) | YES |  |
| city2 | int(11) | YES |  |
| city2Translation | varchar(20) | NO |  |
| city3 | int(11) | YES |  |
| city3Translation | varchar(20) | YES |  |
| topic | bigint(20) | YES |  |
| topicTranslation | varchar(20) | YES |  |
| low\_price | double | YES |  |
| high\_price | double | YES |  |
| created\_time | datetime(6) | YES |  |
| last\_update | datetime(6) | YES |  |
| user\_id | bigint(20) | YES |  |

表4.3 用户信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型**  **（精度范围）** | **空/非空** | **约束条件** |
| id | bigint(20) | NO | PRI |
| password | varchar(128) | NO |  |
| last\_login | datetime(6) | YES |  |
| is\_superuser | tinyint(1) | NO |  |
| username | varchar(150) | NO | UNI |
| first\_name | varchar(150) | NO |  |
| last\_name | varchar(150) | NO |  |
| email | varchar(254) | NO |  |
| is\_staff | tinyint(1) | NO |  |
| is\_active | tinyint(1) | NO |  |
| date\_joined | datetime(6) | NO |  |
| name | varchar(10) | YES |  |
| birth | date | YES |  |
| genderCode | int(11) | YES |  |
| genderTranslation | varchar(2) | YES |  |
| phone | varchar(11) | YES |  |
| photo | varchar(100) | YES |  |
| init | tinyint(1) | YES |  |
| last\_update | datetime(6) | YES |  |
| resume\_id | int(11) | YES |  |

表4.4 景点评论表

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型**  **（精度范围）** | **空/非空** | **约束条件** |
| cid | bigint(20) | NO | PRI |
| content | longtext | NO |  |
| create\_time | datetime(6) | NO |  |
| travel\_id | bigint(20) | NO | MUL |
| user\_id | bigint(20) | NO | MUL |

表4.5 景点收藏表

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型**  **（精度范围）** | **空/非空** | **约束条件** |
| sid | bigint(20) | NO | PRI |
| create\_time | datetime(6) | NO |  |
| travel\_id | bigint(20) | NO | MUL |
| user\_id | bigint(20) | NO | MUL |

表4.6 景点浏览表

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型**  **（精度范围）** | **空/非空** | **约束条件** |
| cid | bigint(20) | NO | PRI |
| count | int(11) | NO |  |
| create\_time | datetime(6) | NO |  |
| last\_update | datetime(6) | NO |  |
| travel\_id | bigint(20) | NO | MUL |
| user\_id | bigint(20) | NO | MUL |

表4.7 用户岗位推荐表

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **数据类型**  **（精度范围）** | **空/非空** | **约束条件** |
| user\_id | int(11) | YES |  |
| recommendations | longtext | YES |  |

4.4 数仓设计

在系统中，数据从多个来源收集，并通过一系列的数据处理步骤进行清洗、转换和整合。

在最底层是MySQL数据库，它是整个系统的数据存储中心。这个数据库包含了爬虫抓取到的各种旅游信息以及一些基础的系统数据。这些原始数据被导入到Hive表中进一步处理。在Hive中，我们可以对数据进行复杂的查询操作，例如计算每个景点的人流量或者游客满意度等指标。我们使用Spark SQL来执行更复杂的数据分析任务。比如，可以利用Spark ML库中的机器学习算法来进行用户行为预测或兴趣偏好挖掘；也可以用SQL语言直接对大数据集进行实时查询和更新。

经过上述过程得到的结果会被用于生成个性化的旅行建议。这部分工作通常由一个叫做ADS（Advertising Decision System）的应用完成，它可以根据用户的个人信息和历史记录为他们提供最适合他们的旅行方案。

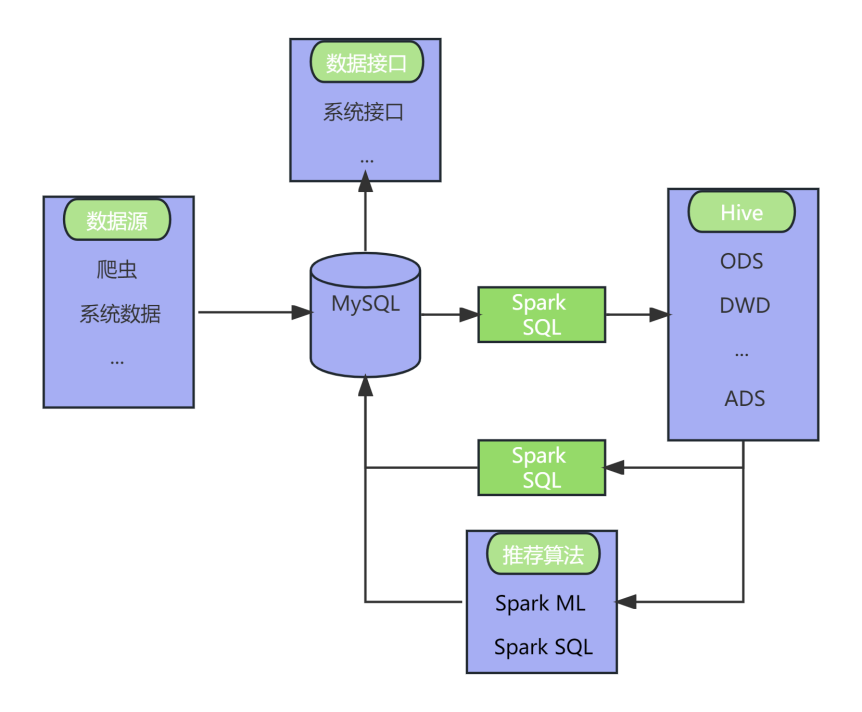


图4.4 数仓架构图

4.5 本章小结

本章以系统总体架构为基础，设计了一个满足用户需求、具备良好性能、可扩展性和安全性的系统。首先阐述了系统的层次结构设计，并对功能层次进行了详细说明。同时，进行了数据库设计，包括概念结构设计和物理结构设计两个阶段。系统主要表及表结构也被列举出来。第五章将深入介绍系统的具体实现细节。

5 系统实现

5.1 数据采集与存储

在数据采集模块，本文使用requests库对驴妈妈旅游网（https://www.lvmama.com）的数据接口进行请求，将格式化数据JSON解析存入数据库中

5.1.1 实现步骤

①筛选出河北省旅游景区



图 5.1 驴妈妈旅游网

②确定需要的字段

表 5.1 需要爬取的字段

|  |  |
| --- | --- |
| 字段名 | 字段类型 |
| id | int(11) |
| name | text |
| topic | text |
| low\_price | text |
| high\_price | double |
| address | text |
| city | text |
| url | text |
| img | text |
| create\_time | datetime |

③找出分页规则

http://s.lvmama.com/ticket/H9K310000P{page}PRO3?keyword=%E6%B2%B3%E5%8C%97&tabType=route350#list

该URL中的page就是分页参数



图 5.2 分页规则

④页面爬取与解析

核心解析代码如下



图 5.3 核心解析代码

⑤存入数据库

使用pymysql库将爬取的数据存入MySQL的spider数据库中

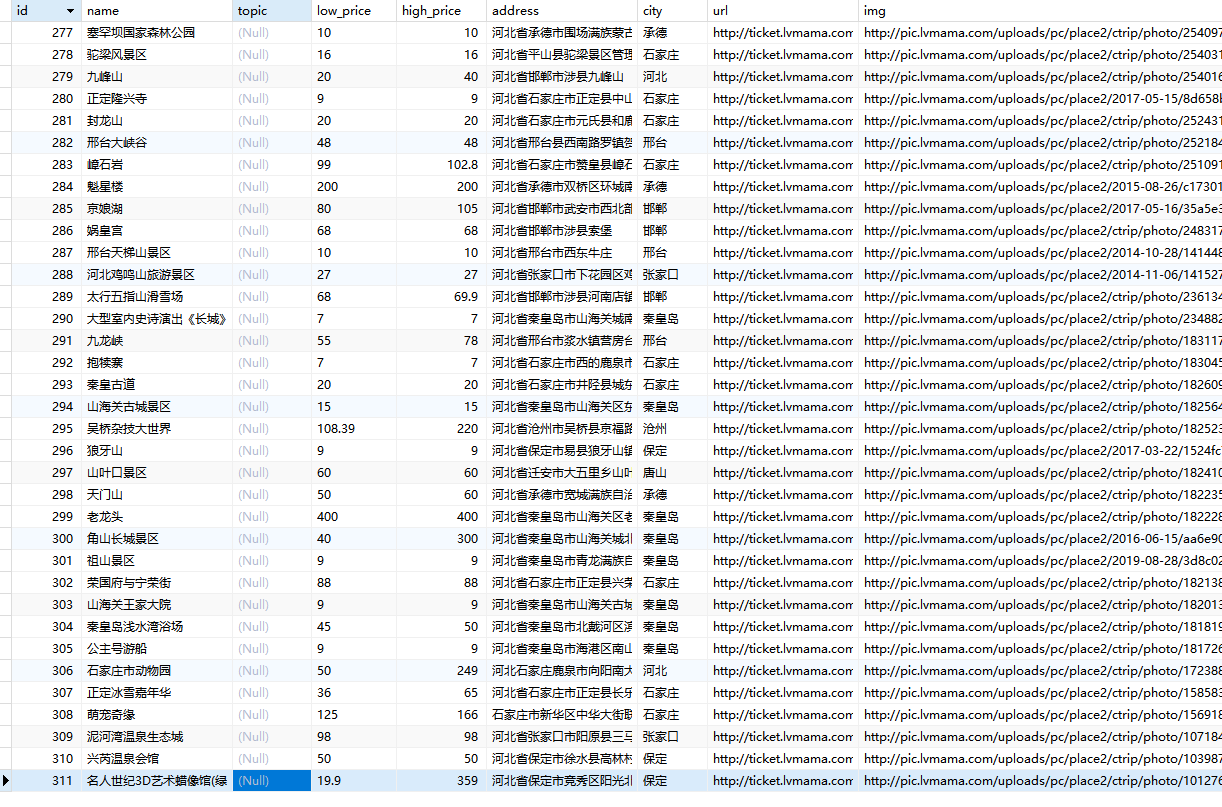


图5.4 存入数据库

5.2 数仓建模

数据仓库建模是指根据业务需求和数据分析目的，对企业数据进行组织、整合和抽象，以建立一个能够支持数据分析与决策的数据存储体系的过程。

利用Spark强大的并行计算和数据处理能力，实现了从数据提取到数据处理再到数据存储的全流程，为后续的数据分析和机器学习提供了高质量的基础数据，同时也提高了数据处理的效率和可靠性。

5.2.1 ODS层

将爬取的数据以及系统中的用户数据、用户行为数据和用户简历数据进行提取、转换和加载（ETL），导入到Hive的ODS层。在ODS层，进行进一步的数据清洗，包括格式化、字段映射、去重、空值处理等操作，以确保数据质量和一致性。同时，对JSON字段进行解析，保证数据结构化存储。

5.2.2 DWD层

在这一阶段，再次进行数据清洗和处理，例如数据合并、数据分割、数据过滤等，以满足业务需求和数据分析目的。根据业务需求，进行维度建模和事实建模，确保数据仓库设计符合系统应用的要求。整合并清洗好的招聘信息数据和企业数据将会被使用Spark导入到MySQL数据库中，以提供一个更加高效的数据存储和管理环境。在这一过程中也需要考虑数据同步的频率和数据一致性的问题，确保数据的及时性和准确性。

5.2.3 ADS层

清洗好的DWD层数据存储到ADS（Application Data Store）层。这些数据作为根据DWD层数据清洗而成的训练集数据，在后续的数据分析和机器学习任务中发挥关键作用。数据经过ADS层的存储，为系统提供了可用于决策和业务应用的高质量数据。

5.3 推荐算法实现

5.3.1 APH层次分析法求权重

层次分析法（Analytic Hierarchy Process，AHP）是一种用于求解权重和进行多标准决策分析的数学方法。它最初由美国运筹学家托马斯·L·塞蒂尔（Thomas L. Saaty）在1970年代提出。该方法基于将问题分解成一系列层次结构，从而使得决策者可以在不同的层次上进行比较和评估。

在推荐算法的实现中，AHP可以用来确定不同因素（如用户兴趣、物品特征等）的权重。具体步骤如下：

①定义决策问题：确定需要比较和评估的因素集合。

②构造层次结构：按照问题的特性和因素之间的关系，构建一个递阶层次结构。

③建立判断矩阵：通过对每对因素进行比较，得到一个判断矩阵，其中包含了因素之间的相对重要性。

④计算权重：使用数学方法，将判断矩阵进行一致性检验，并计算出各个因素的权重。

⑤一致性检验：检查所得的判断矩阵是否具有一致性，以确保数据的可信度。

⑥综合评价：根据各个因素的权重，进行综合评价，并得出最终的推荐结果。

5.3.2 带权重的余弦相似度算法

带权重的余弦相似度算法是一种高级的向量相似度计算方法，它考虑了特征权重的影响，以更准确地反映不同特征在相似度计算中的重要性。该算法首先对特征向量中的每个特征应用权重，然后计算加权后的特征向量之间的余弦相似度，从而得到精确的相似度度量。在实际应用中，通过合理设置不同特征的权重，这种算法能够更准确地评估用户与物品之间的相关性，为推荐系统提供更符合用户偏好的推荐结果。

 (5.1)

5.3.3 模型训练

1. 加载和处理数据：

①使用 SparkSession.builder 创建一个 SparkSession 对象，用于操作 Spark 功能。

②使用 spark.sql 方法加载用户画像和招聘信息的数据集。这两个数据集可以使用 SQL 查询语句从数据库表中加载。

③使用 VectorAssembler 对象将用户画像和招聘信息的特征列合并为一个特征向量列。这一步骤使特征能够作为输入进行处理。

④使用 MinMaxScaler 进行特征向量的标准化处理，以确保特征值在相同的尺度范围内。

⑤对于用户画像和招聘信息数据集，使用 .transform 方法将转换后的特征向量列添加到数据集中。

1. 相似度计算：

①定义 weighted\_cos\_similarity 函数用于计算加权余弦相似度。该函数使用 Numpy 库进行向量计算。

②使用 udf 函数将 weighted\_cos\_similarity 函数转换为 Spark UDF（用户定义函数）。

③使用 cartesian\_df 创建的笛卡尔积数据集，对每个用户和招聘信息的特征向量应用 UDF 函数。将计算得到的相似度作为新的列添加到数据集中。

④使用 .orderBy 方法对数据集进行降序排序，根据相似度列的值。

1. ALS模型训练和评估：

①创建 ALS 对象用于实例化 ALS 模型。设置模型的参数，如用户列、物品列、评分列、迭代次数等。

②使用 training\_data 训练 ALS 模型，使用 .fit 方法。

③使用 model.transform 将测试集数据应用于训练好的模型，得到预测结果。

④使用 RegressionEvaluator 创建评估器对象，用于计算模型的 RMSE。根据需要可以选择不同的评估指标。

⑤使用评估器 .evaluate 方法，传入预测结果和实际评分列，计算得到模型的 RMSE。

1. 结果保存：

①使用 model.recommendForAllUsers 方法为所有用户生成推荐结果。可以根据实际需求指定推荐的数量。

②将生成的推荐结果保存在 DataFrame 中，并使用 .createOrReplaceTempView 将 DataFrame 转换为临时视图。

③使用 Spark SQL 语句将推荐结果写入数据表 ads\_jobfree.ads\_jobfree\_db\_jobfree\_t\_recommendforallusers。

④使用 df2.write.format('jdbc') 以 JDBC 方式将结果 DataFrame 保存到 MySQL 数据库中的指定表。

1. 模型训练代码：

模型训练的代码在项目路径下的，model/ALS.py中。

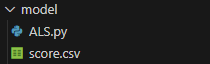


图5.1 模型训练代码路径

5.4 系统功能实现

基于Spark的河北省旅游景区推荐系统主要利用了Django和Vue两大框架进行实现。Django作为后端框架，负责处理用户请求、数据处理和交互逻辑。而Vue则作为前端框架，负责构建直观友好的用户界面，并与Django后端进行数据交互和展示。通过整合Spark、Django和Vue，系统能够高效地提供景点推荐服务，为用户提供个性化、精准的景点旅游推荐，从而提升旅游满意度和用户体验。

5.3.1 用户管理模块

1. 注册模块

用户打开前端页面，输入注册信息，包括用户名、密码和邮箱等个人信息，邮箱验证码等，前端页面通过 Vue 发起请求，将用户输入的注册信息发送到后端 Django 服务器，后端 Django 服务器接收到请求后，首先对用户提交的信息进行验证和处理，包括验证用户名是否已经存在、验证邮箱格式等，如果用户信息无误，则将用户数据存入数据库中。

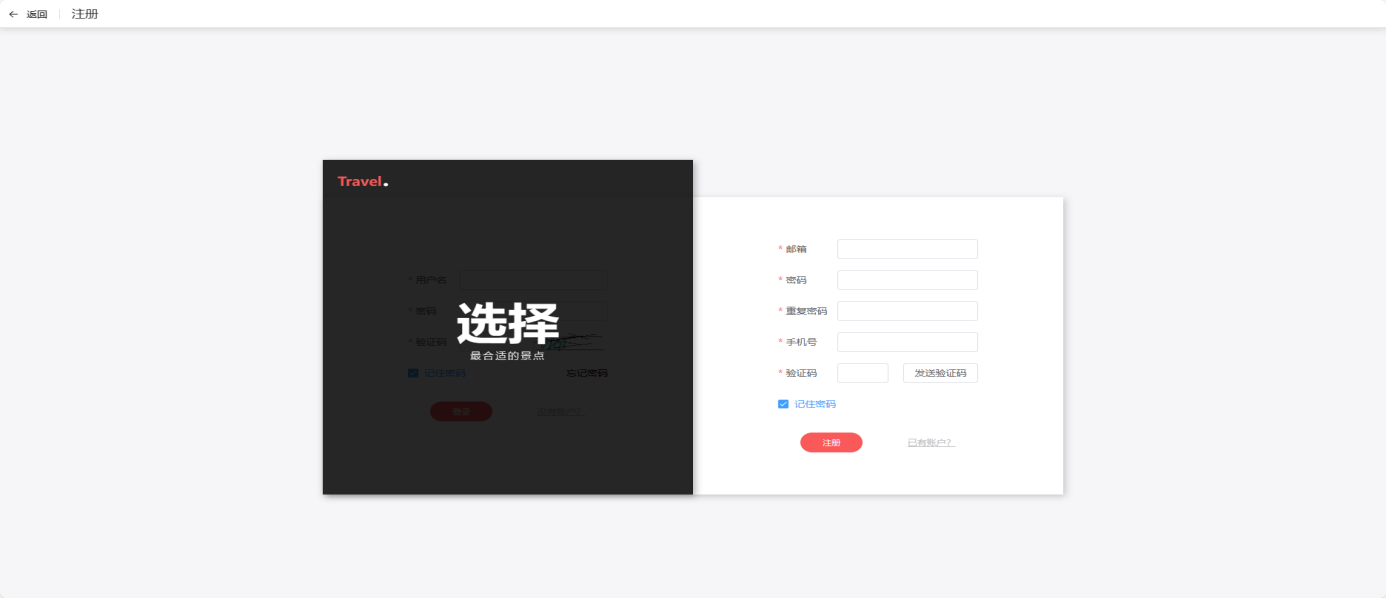


图5.2 用户注册页面

1. 登录模块

在登录模块中，用户打开前端登录页面，输入用户名和密码，后端接口生成图形验证码并将加密的验证码存入Cookie，用户输入图形验证码并传入提交表单后，后端解密在Cookie中的验证码并判断和用户传入的是否一致，如果传入数据都无误，则进入系统首页。

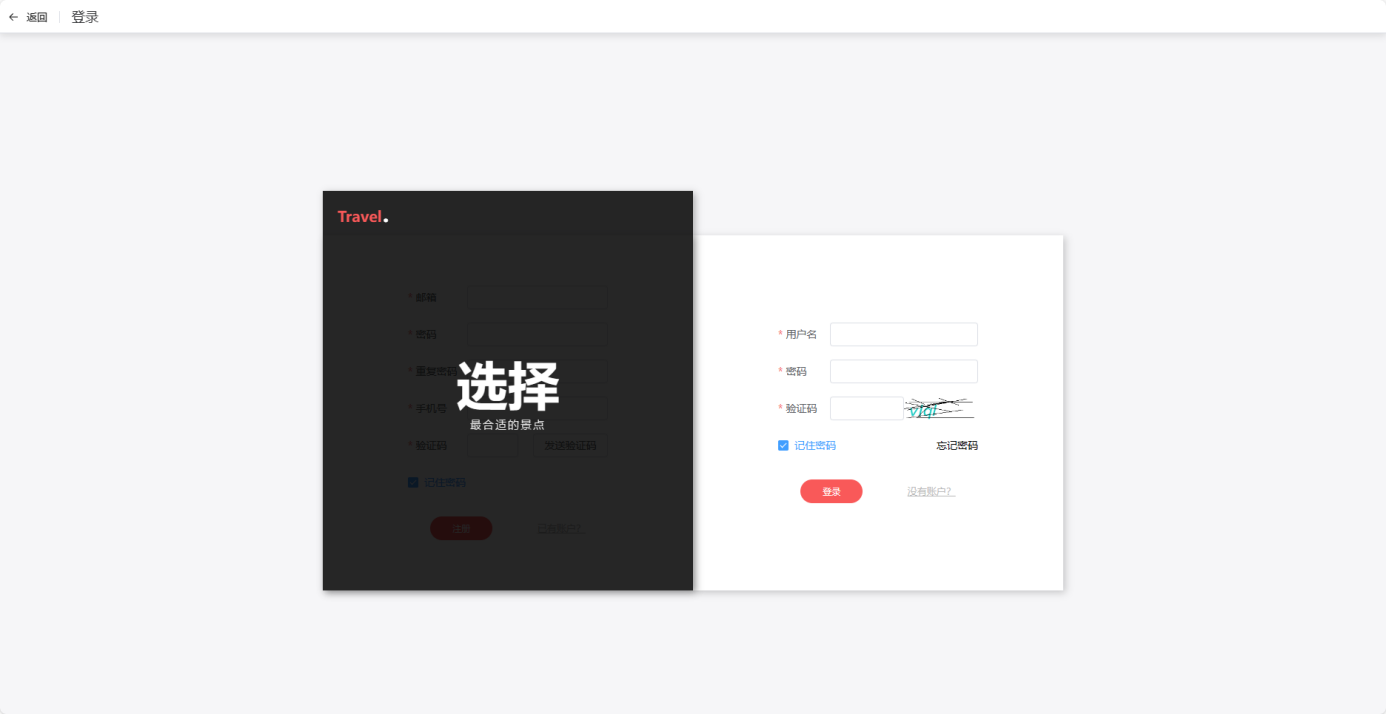


图5.3 用户登录页面

5.3.2 系统首页模块

系统首页模块主要由导航栏、搜索框、轮播图、轮播图左侧边导航栏、热门景点几个部分组成：

导航栏：提供网站的整体导航功能，包括各个主要页面的链接，首页、推荐景点、找景点、我的。

搜索框：用户可以在搜索框中输入关键词，便于用户快速找到所需信息。

轮播图：通过轮播图展示网站的重点信息或优质内容，以引导用户关注和点击进入相关页面。

轮播图左侧边导航栏：在轮播图的左侧提供导航栏，包括分类导航、特别推荐内容链接等，以便用户快速找到感兴趣的内容。



图5.4 首页页面

5.3.3 数据检索模块

筛选查询功能：用户可以根据关键词、城市、类型、门票预算等条件进行筛选查询。

分页功能：为了提供更好的用户体验和对大量数据的处理，找工作子模块支持将结果分页展示，可以通过页面上的翻页按钮或者页码输入实现翻页操作。

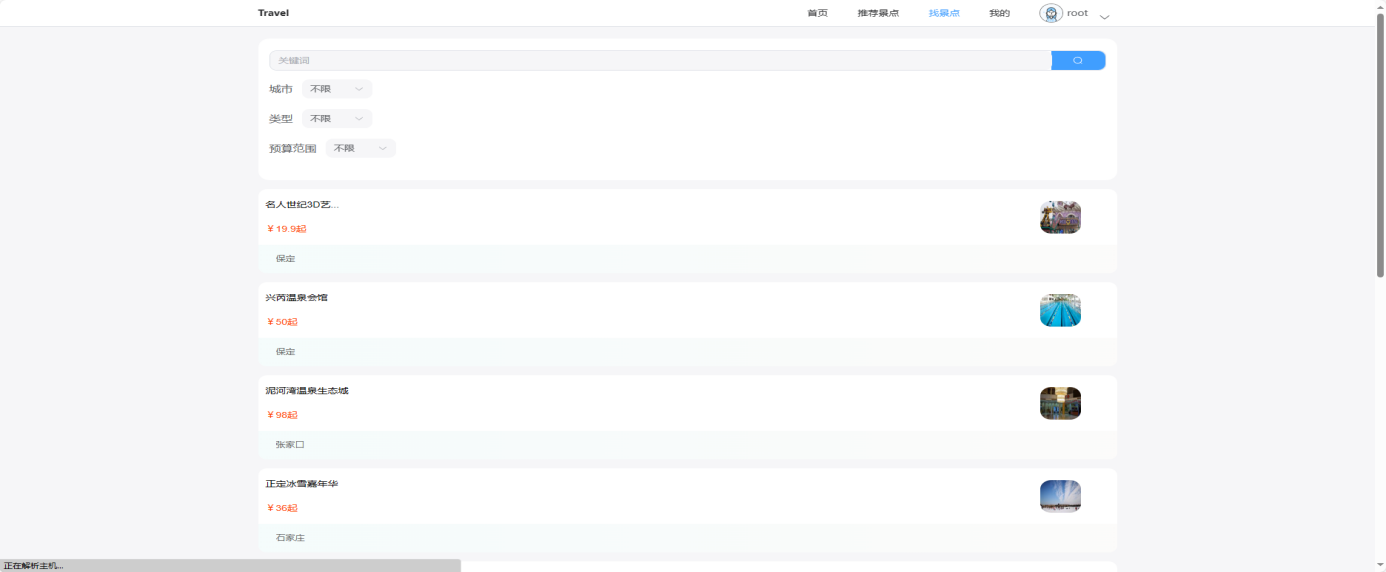


图5.5 找景点页面

5.3.4 推荐模块

1. 冷启动推荐

冷启动推荐旨在向用户展示热门信息，即当前流行的景点信息。它可以根据系统中的热门数据或热门搜索关键词来推荐最受欢迎的景点。这样的推荐可以帮助用户快速了解当前热门景点，提高用户的参考价值。

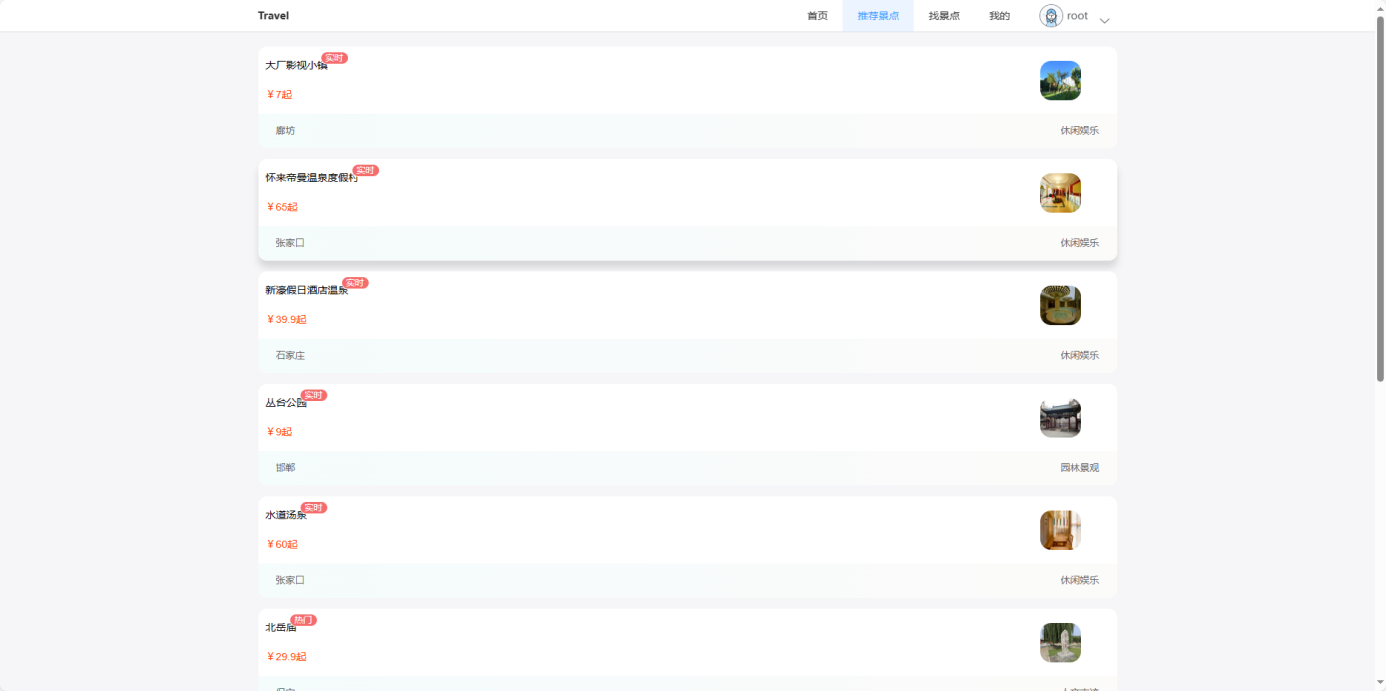


图5.6 热门推荐

1. 实时推荐

实时推荐是根据用户的收藏列表来生成个性化推荐结果。首先，系统会查询出与当前用户收藏的景点相匹配的其他用户列表（排除了自己），然后进一步查询出这些用户收藏的其他景点列表，再将这些列表整合起来作为推荐结果返回给当前用户。

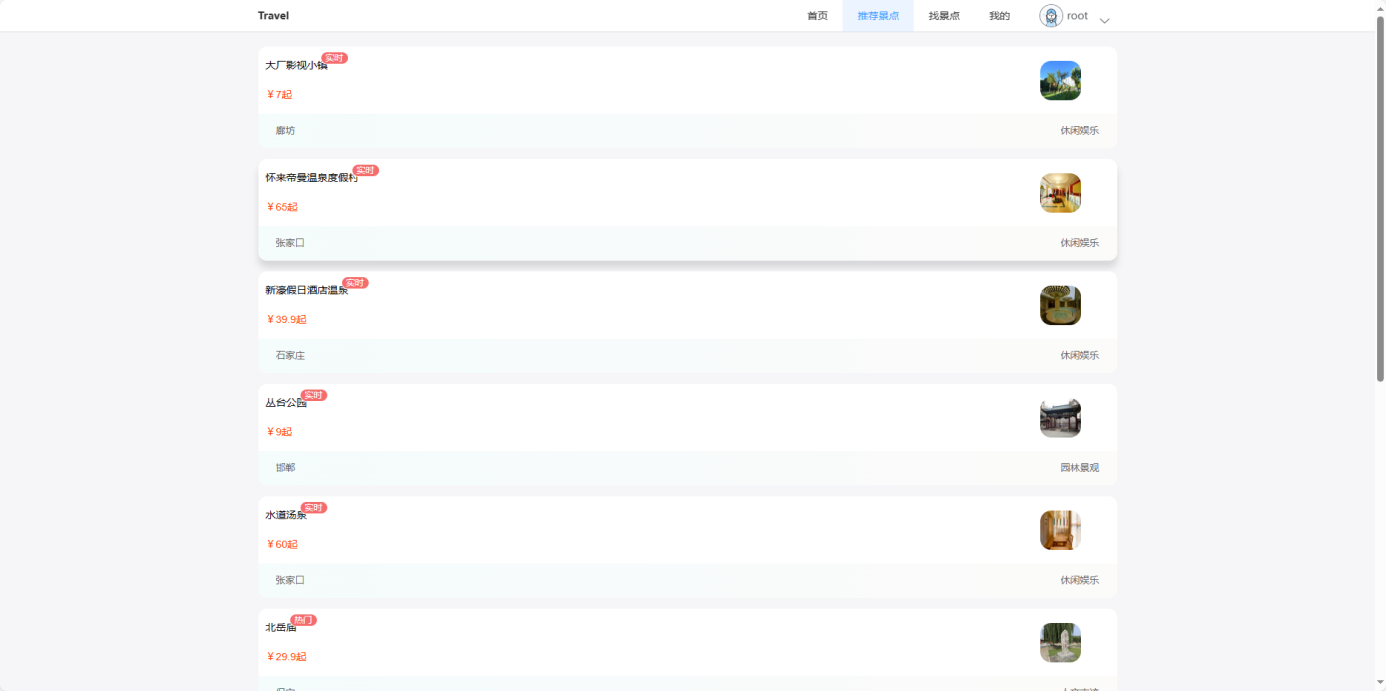


图5.7 实时推荐

1. 离线推荐

离线推荐是基于用户的喜好，以及景点信息，计算景点和用户喜好之间的向量相似度。这里使用ALS（Alternate Least Squares）算法进行训练，通过对用户历史行为数据的分析和建模，得到一个可用于推荐的离线模型。

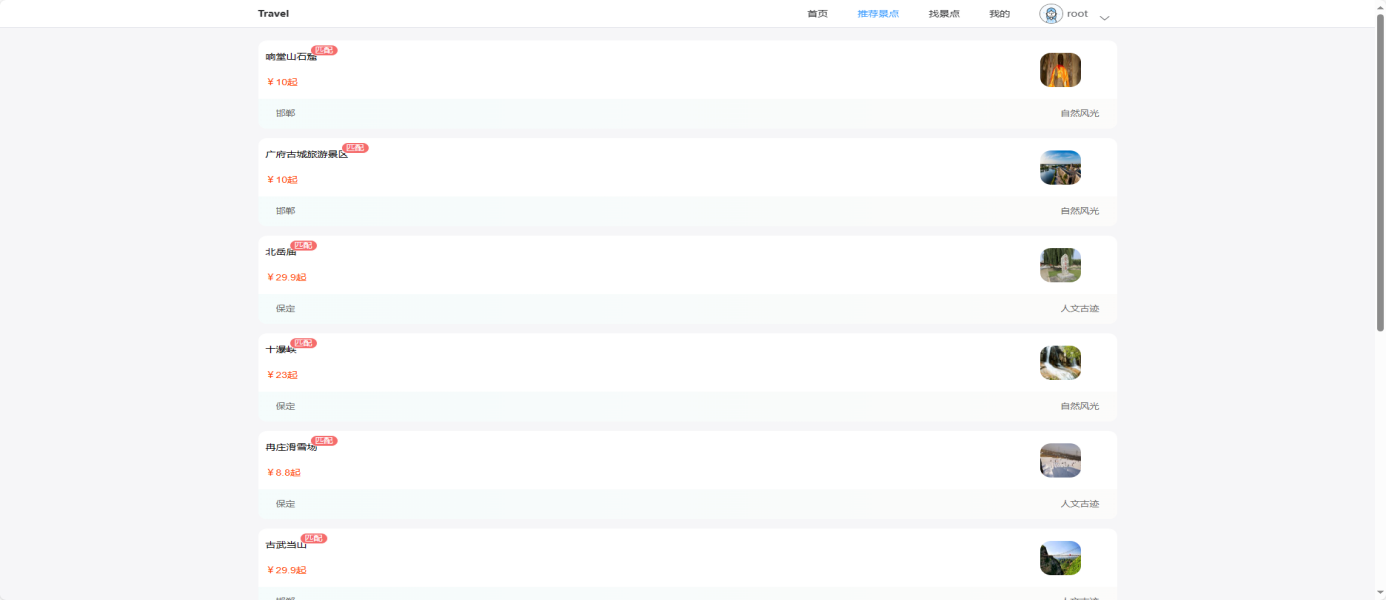


图5.8 匹配推荐

1. 相似推荐

相似推荐主要在景点详细页上，向用户推荐与当前景点相似的景点信息。

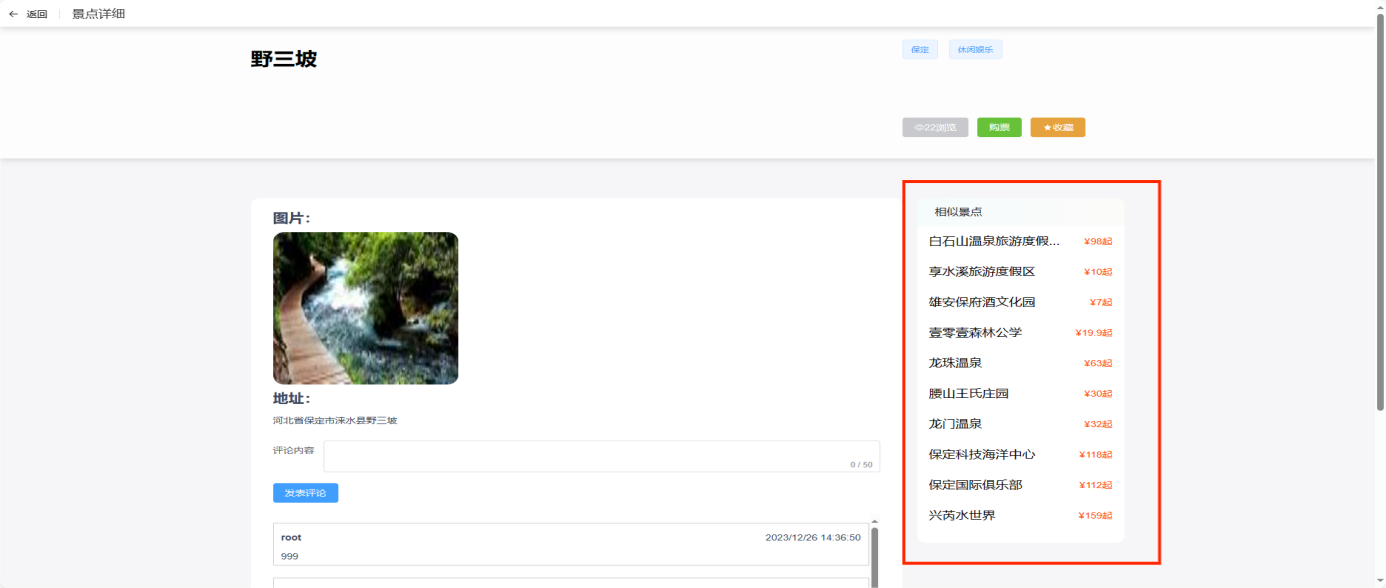


图5.9 相似推荐

5.3.5 其他功能模块

1. 收藏功能

收藏功能模块允许用户在景点详细页上收藏感兴趣的景点。用户可以通过点击收藏按钮将特定职位加入到自己的收藏列表中。在"我的"界面，用户可以方便地查看已收藏的景点列表，随时了解自己感兴趣的景点信息。

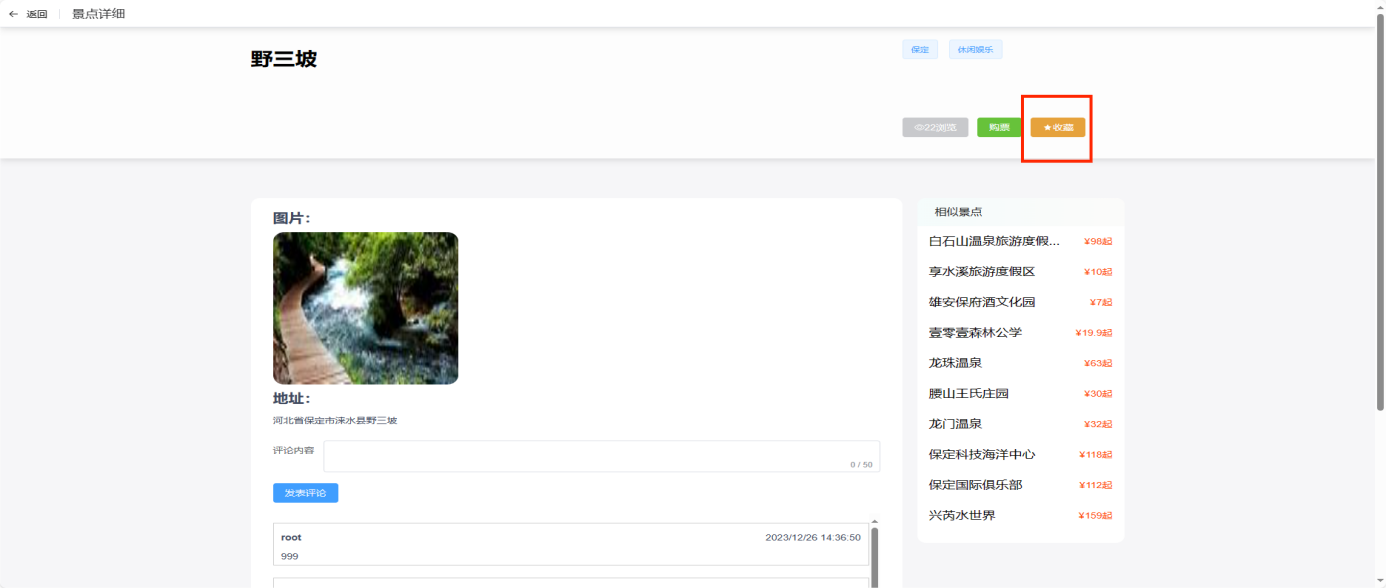


图5.10 收藏功能1



图5.11 收藏功能2

1. 评论功能

评论功能模块使用户可以在景点详细页对特定景点进行评论。用户可以分享自己对景点的看法。

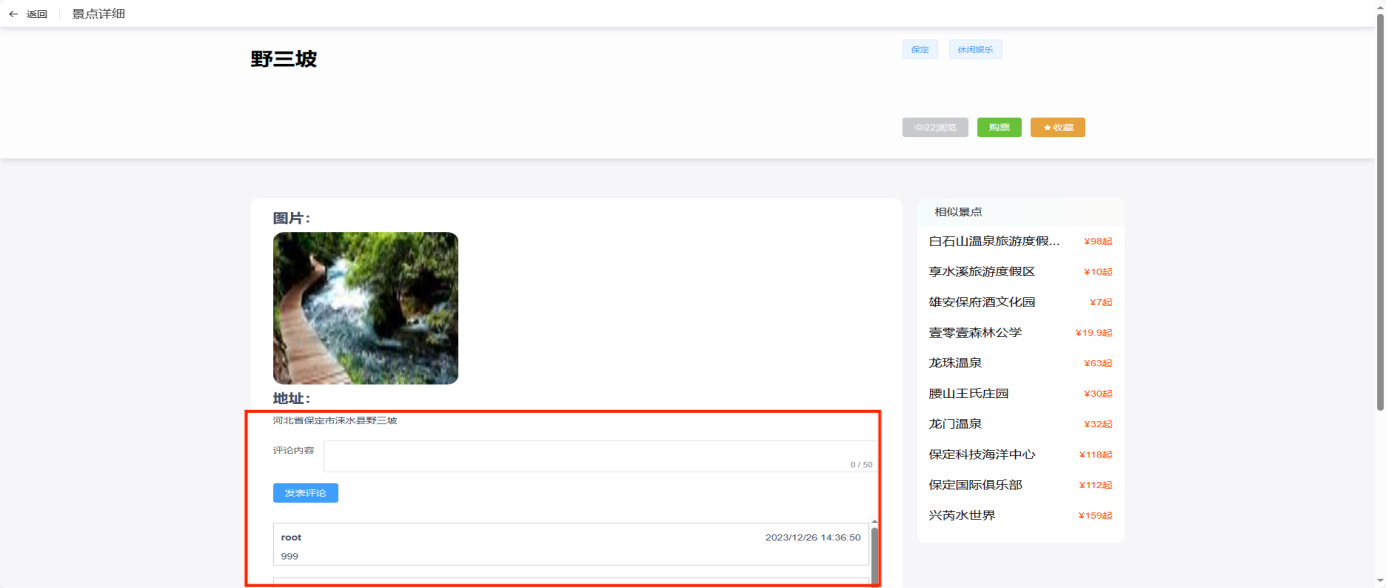


图5.12 评论功能

1. 用户喜好管理

用户喜好管理模块包括个人喜好的创建、修改和查看。

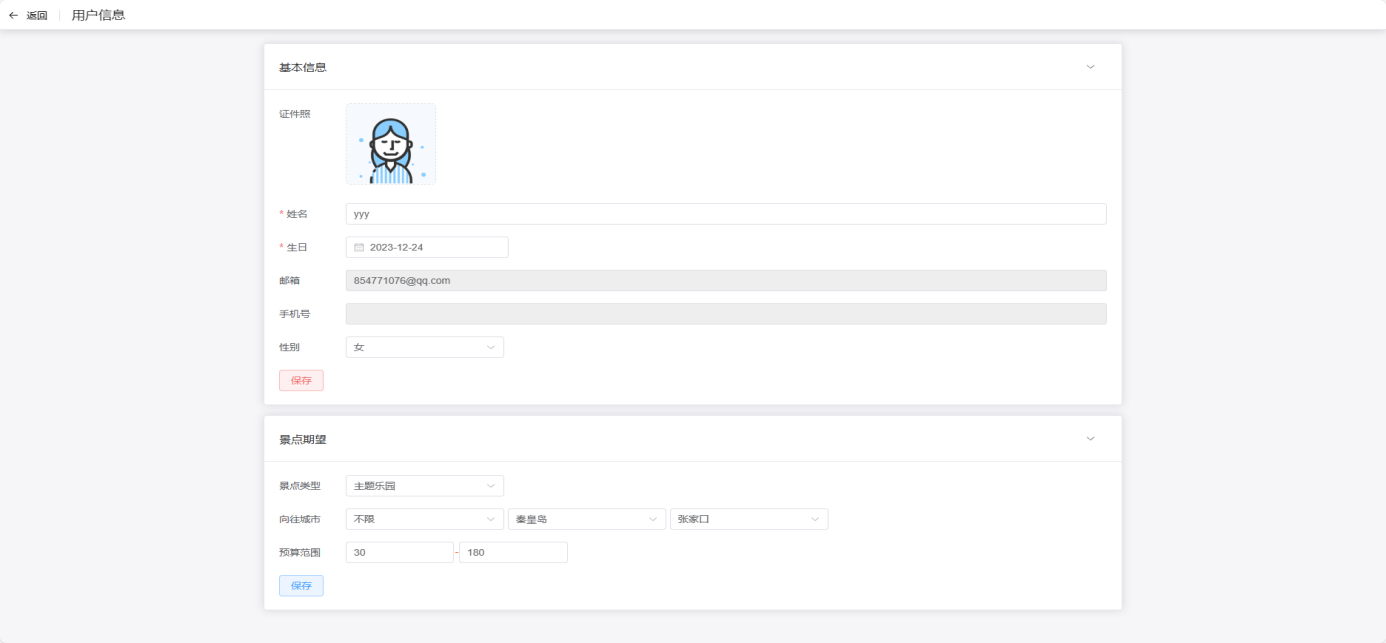


图5.13 用户喜好管理

5.3.6 后台管理模块

本文后端管理系统是基于Django内置的后端框架+SimpleUI组件，可使后台页面更加美观。

1. 登陆页面

登录页面模块是为了管理员登录后台管理系统而设计的。管理员需要输入正确的账号和密码才能成功登录。登录页面可以使用Django的内置模块进行实现，以降低开发成本并提供安全认证。

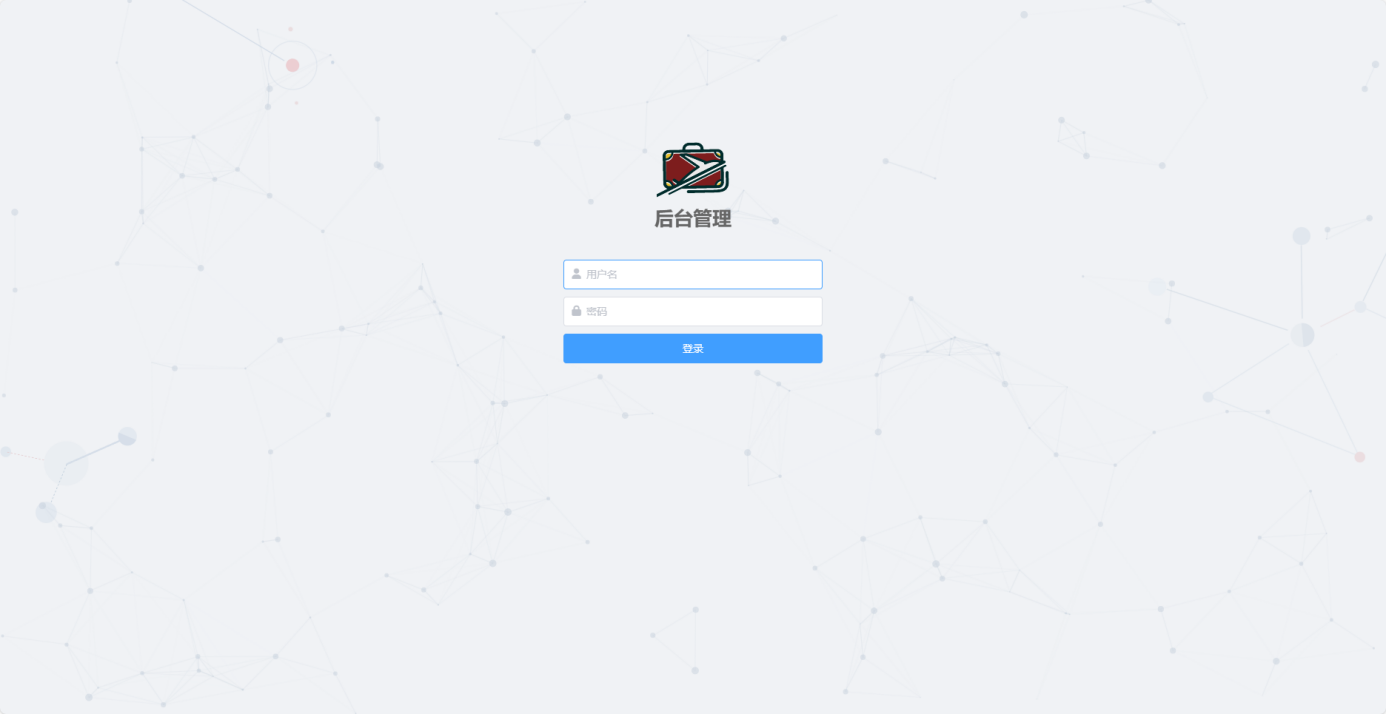


图5.14 后台登陆页面

1. 数据管理

数据管理模块是后台管理系统的核心功能之一，用于管理网站上的各种数据信息。管理员可以对数据库数据进行增删改查操作。这包括对数据库的读写操作、数据导入导出、数据备份恢复等操作，以及数据的搜索、排序和分页等功能。

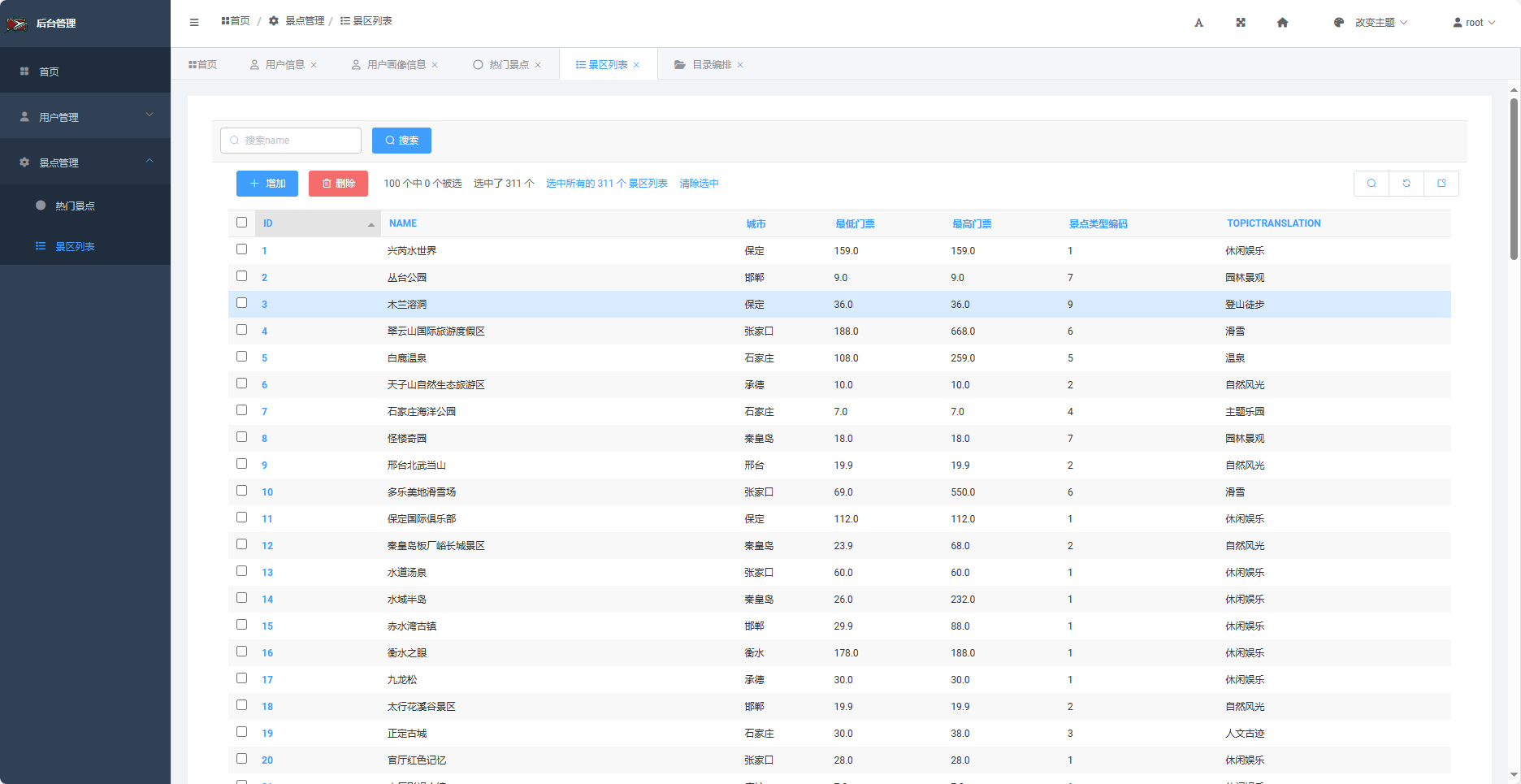


图5.15 后台数据管理

5.4 本章小结

本章主要介绍系统的实现，包括数据采集、数仓建模、推荐模型介绍以及系统功能的具体实现。第六章将介绍系统测试。

6 系统测试

系统测试是开发过程中的一项重要环节，用于验证系统是否满足用户需求，其目的是发现和纠正系统中的错误，确保系统的稳定性、安全性、可靠性和性能。基于Spark的就业推荐系统的使用对用主要是用户和管理员，故以两者为角度，采用黑盒测试，对系统的登录注册、信息管理、数据浏览、推荐管理等核心功能进行测试，分析业务流程是否符合需求。

6.1 功能测试

用户的核心功能有登陆注册、信息管理、数据浏览、推荐浏览等模块的核心功能，用户的功能模块的测试情况如下表：

表6.1 用户核心功能测试情况表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 功能  名称 | 测试流程 | 预期结果 | 实际结果 | 测试  结果 |
| TC1-1 | 用户  注册 | 进入注册页面，输入注册信息，点击提交，跳转登陆页面 | 注册成功，能够成功跳转到  登陆界面 | 注册信息能够保存在数据库中，注册成功 | 通过 |
| TC1-2 | 用户登陆 | 在登陆界面，输入账号和密码，点击登陆，进入系统 | 登陆成功，能够跳转到系统首页 | 登陆成功，能够进入系统首页 | 通过 |
| TC1-3 | 用户收藏、评论 | 用户通过点击信息列表，进入详细页页面中的收藏按钮、发表评论按钮 | 用户的操作，能够正常进行 | 页面短暂加载，功能实现正常 | 通过 |
| TC1-4 | 信息管理 | 用户点击个人信息按钮，浏览个人信息 | 页面正确显示个人信息数据 | 个人信息能够正确显示 | 通过 |
| TC1-5 | 数据浏览 | 用户点击数据浏览按钮，浏览数据信息，输入搜索条件，页面返回搜索结果 | 页面能够实现搜索，并正确返回结果 | 页面正确返回搜索条件要求的数据信息 | 通过 |
| TC1-6 | 推荐浏览 | 用户点击热门推荐按钮，浏览系统默认推荐内容，修改推荐条件，并返回新的推荐内容 | 页面初始显示默认推荐内容，其次显示修改推荐条件后的推荐内容 | 页面能够正确显示修改推荐条件前后的推荐内容 | 通过 |

管理员的核心功能有用户管理、权限管理、数据管理、推荐管理等模块的核心功能，管理员的功能模块的测试情况如下表：

表6.2 管理员核心功能测试情况表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 功能  名称 | 测试流程 | 预期结果 | 实际结果 | 测试  结果 |
| TC2-1 | 管理员  登录 | 在登陆界面，输入账号和密码，点击登陆，进入后台系统 | 登陆成功，能够跳转到系统首页 | 登陆成功，能够进入系统首页 | 通过 |
| TC2-2 | 用户管理 | 点击用户信息，页面显示用户注册信息以及添加、删除、修改、查询等功能按钮 | 页面正确显示用户信息并含添加、删除、修改、查询功能按钮 | 页面正确显示并存在添加、删除、修改、查询功能按钮 | 通过 |
| TC2-3 | 权限管理 | 点击权限按钮，修改用户权限，点击保存，页面返回修改成功 | 页面正确返回修改成功弹窗 | 页面返回修改成功的弹窗 | 通过 |
| TC2-4 | 数据管理 | 页面显示数据信息以及添加、删除、修改、查询等功能按钮 | 页面正确显示用户信息并含添加、删除、修改、查询功能按钮 | 页面正确显示并存在添加、删除、修改、查询功能按钮 | 通过 |

根据表6.1和表6.2两个功能测试情况表可知，系统整体运行表现良好，功能需求基本得到满足。

6.2 性能测试

分析系统测试的结果，有助于了解系统的缺点和不足，而性能测试便于系统的改进和扩展，保证系统运行的稳定性。本文选择多个性能指标，将用户和管理员的测试情况进行汇总，如下表：

表6.3 系统性能测试情况表

|  |  |  |  |
| --- | --- | --- | --- |
| 性能指标 | 预期结果 | 实际结果 | 测试结果 |
| 注册响应时间 | 低于3秒 | 1-2秒 | 通过 |
| 登陆响应时间 | 低于3秒 | 1秒内登陆成功 | 通过 |
| 数据加载时间 | 低于5秒 | 2-3秒 | 通过 |
| 查询数据返回时间 | 低于3秒 | 1-2秒 | 通过 |
| 平均业务处理时间 | 低于3秒 | 约1.5秒 | 通过 |

根据表6.3系统性能测试情况表，系统整体表现良好，各性能指标均达到预期结果。

6.3 本章小结

本章主要介绍系统的系统测试情况，阐述功能测试和性能测试，从测试结果上分析，系统整体表现良好，功能需求、性能指标等均达到预期结果。

7 总结与展望

7.1 全文总结

我选择开发基于Spark的河北省旅游景区推荐系统的主要原因源自对旅游信息推荐功能的极大兴趣。我经常浏览旅游网站，并对推荐系统的工作机制产生了浓厚兴趣，这促使我决定运用大数据技术深入研究其内部原理。通过挖掘用户的潜在需求，我期望为用户提供个性化、准确的景点推荐，以提升其旅游体验。

在系统准备阶段，我进行了大量文献和技术文档的深入阅读，为系统设计奠定了坚实基础。我选用Django作为后端框架，而前端页面则运用了Vue和Element-plus。为了数据治理，我采用了Spark SQL+Hive技术组合。

系统设计阶段，我以使用对象为切入点，细致考量了系统管理员、用户和游客三类用户的需求。透过研读设计文档和剖析微博页面呈现的数据，我完成了系统使用角色、用例和功能的详尽设计。

总体而言，基于Spark的河北省旅游景区推荐系统的开发旨在处理大量数据、实现实时推荐，以帮助用户及时获取最新的旅游景点信息，从而提升他们的旅游满意度和选择效率。通过深度分析用户行为和喜好，并结合机器学习算法，该系统能够为用户提供更符合个性化需求的景点推荐，进一步提升用户对旅游的满意度和匹配度。

7.2 展望

本文实现的Spark的河北省旅游景区推荐系统，尽管已取得一定成果，但在系统设计和功能实现方面仍存在一些值得改进的问题。

数据的单一来源： 目前系统主要采集驴妈妈旅游网的数据，导致数据来源单一，可能无法全面覆盖其他旅游网站的数据。为了提供更全面和多样化的推荐服务，需要考虑增加其他旅游网站的数据源，以丰富系统的数据，提高推荐的覆盖范围和准确性。

缺乏用户反馈机制： 系统并未涉及用户对推荐结果的反馈机制，这在优化推荐算法和提高个性化程度上显得不足。引入用户反馈机制，如用户评分、点击行为、满意度调查等，可以帮助系统更好地理解用户偏好，优化推荐算法，提高推荐的准确性和个性化程度。

伪实时推荐： 系统未实现实时数仓的构建，导致推荐结果更新可能存在一定延迟。实时推荐能够根据用户当前行为和上下文，快速生成相应的推荐结果，提升用户体验。引入流式处理技术，结合用户实时行为，实时更新推荐结果，让用户能够即时获取最新的旅游信息。

平台可扩展性考虑不足： 系统设计中未充分考虑到系统的可扩展性，可能面临性能瓶颈。针对大规模数据处理和用户量的增长，可以考虑引入分布式数据库等技术，以提高系统的扩展性和性能，应对未来数据规模和用户量的增长挑战。

综上所述，通过解决以上问题并不断优化系统设计和功能实现，将能够提升系统的覆盖范围、推荐准确性和用户体验，使系统更好地满足用户需求并适应未来的发展需求。

参考文献

1. Yonghong Tian,Bing Zheng,Yanfang Wang,Yue Zhang,Qi Wu.College Library Personalized Recommendation System Based on Hybrid Recommendation Algorithm[J].Procedia CIRP,2019,():490-494.
2. 李英楠. 基于Hive的购销数据仓库系统的设计与实现[D].重庆：西南大学,2020.
3. 朱本瑞.基于Spark的离线与实时的电影推荐系统设计与实现[D].南京：南京信息工程大学,2022.
4. 张婳,彭海英.基于混合深度神经网络的就业推荐方法[J].计算机工程与设计,2022,43(04):995-1001.
5. 梁文楷,凃红玲,陈佳欢.一种基于ElasticSearch全文检索技术的研究[J].中国科技信息,2021(18):82-84+87.
6. 丁健龙.一种基于画像关联及本体相容匹配的就业推荐方法[J].浙江工业大学学报,2022,50(03):270-275+298.
7. 张坤.基于Spark机器学习的电影推荐系统的设计与实现[D].南京：南京邮电大学,2022.
8. 金铄. 基于用户画像的高校毕业生就业推荐方法研究[D].大庆：东北石油大学,2023.
9. 李艳杰.MySQL数据库下存储过程的设计与应用[J].信息技术与信息化.2021
10. 胡阳.Django企业开发实战[M].人民邮电出版社
11. 张凡.基于Web的旅游服务平台的设计与实现.[D].软件工程.2018
12. 綦慧，徐晓慧.基于Web系统的大数据搜索技术的实现和优化.[A].信息学.2021
13. 白秀军.基于大数据的枣庄市智慧旅游服务平台构建研究.[D].旅游管理.2018

致 谢

时光如梭，转眼间我即将踏上人生新的征程，而在这段宝贵的大学岁月里，我经历了无数的学习与成长，甘苦与欢笑交织成了我毕业论文的篇章。

倾注在毕业论文上的心血，离不开许多人的关怀、支持与勉励。在此，我要向所有给予我帮助与指导的导师们表达最衷心的感谢。您们无私地传授知识，为我点亮前行的道路，塑造了我坚实的学术基础和独立思考的能力。您们的辛勤付出和悉心指导使我的论文得以深入探讨和广泛研究，为我奠定了坚实的研究基础。

特别要感谢我的指导老师，正是您在整个毕业论文的写作过程中给予我孜孜不倦的指导和启发。您对我毕业论文的每个细节都给予了独到的见解和建议，激发了我对知识的渴望和探索的热情。您的教诲不仅开拓了我的思路，还塑造了我扎实的学术风范和精益求精的态度。

同时，我衷心感谢每一位在大学中与我一同奋斗的同学们。在知识的海洋中，我们并肩前行，一起追求学术的高峰。你们的智慧和勇气让我们共同成长，相互鼓励，激发出了无限的创造力和思维的火花。你们的友谊和支持是我宝贵的财富，让我在学术的道路上始终不感孤寂和压力。

在此，我还要感谢我的家人和朋友们。感谢你们在我求学的道路上始终的陪伴与支持。你们的无私的付出与爱意是我学业成功的最大动力，你们的理解和鼓励让我坚持不懈，追求卓越。

再次对所有给予我帮助和支持的人们表示最衷心的感谢！没有你们的关怀和支持，我将无法走到论文的最后一刻，也无法圆满地完成我的学业。感谢大家的支持与鼓励，祝福我们共同迎接未来的挑战与机遇！

谢谢！