学术会议推送系统的研究与建设

### 计算机科学与技术 14-2 罗希指导教师 田萱

摘要

随着国内外举行的学术交流越来越多，各种学术会议也日益频繁，科研人员对这些学术会议的 信息以及相关论文的征集和收录情况有明确的需求，同时希望及时得到会议的新动态和新通知。

本文研究建设的学术会议推送系统旨在为用户推送其感兴趣的学术会议信息。该系统集会议检 索、会议收藏、会议个性化推荐等功能为一体，方便用户跟踪会议最新信息。该系统运用python爬 虫实现了学术会议信息自动下载和自动更新；基于用户协同过滤推荐算法设计实现了会议推荐功能； 并为为学术会议信息和用户信息的存储和管理提供了完备的数据库体系支撑。该系统为科研人员等 用户提供更为方便、快捷的会议查询途径和及时的会议新闻通知，针对用户兴趣偏好进行个性化会 议推荐，具有较强的实际意义。

该系统以 IntelliJ idea 作为开发平台, 使用 Java EE 相关技术研发的 B/S 系统，以 MVC 架构作为开发框架，以 Hibernate 框架的数据持久层和 MySQL 进行数据的管理和存储。功能测试结果表明：系统整体具有实时更新、个性化推荐、界面简洁、友好等特点，有较强的实用性和推广应用前 景。

关键词： **B/S** 系统， **Python** 爬虫， 个性化推荐， **Hibernate**

**Research and Implement of Academic Conference Recommendation System**

### Computer Science and Technology 14-2 Luo Xi

Research Supervisor Tian Xuan

**Abstract**

As for the academic exchanges at home and abroad more and more as well as all kinds of academic conference also increasingly frequent. Therefore, the scientific researchers have clear requirements for the information and the collection of the relevant papers of these academic conferences. At the same time, they hope to get the conferences’ news and notifications in time.

The academic conference recommendation system developed in this paper aims to recommend the academic conference information that users are interested in. This system integrates the functions of meeting retrieval, meeting collection and meeting personalized recommendation, so as to facilitate users to track the latest meeting information. The system uses python crawler to download and update academic conference information automatically. Based on the design of user collaborative filtering recommendation algorithm, conference recommendation function is realized. It also provides a complete database system for the storage and management of academic conference information and user information. The system provides researchers and other users to provide more convenient and fast query way and timely meeting news announcements, user interest preference for personalized recommendation meeting, has strong practical significance.

With IntelliJ idea for the system as a development platform, using Java EE, related technology research and development of the B/S system to MVC architecture as a development framework, with Hibernate framework of data persistence layer and MySQL for data management and storage. The functional test results show that the system as a whole has the features of real-time update, personalized recommendation, simple interface, friendly and so on.

**Keywords: B/S system, Python crawler, personalized recommendation, Hibernate.**

目录

1. 绪论 3
   1. [研究背景和目的 3](#_TOC_250022)
   2. [国内外学术会议系统研究现状 3](#_TOC_250021)
      1. 国内外相似产品分析 4
   3. [系统设计目标 5](#_TOC_250020)
   4. [论文组织结构 5](#_TOC_250019)
2. [系统开发工具及关键技术研究 **7**](#_TOC_250018)
   1. B/S 模式体系结构 7
      1. 三层 B/S 结构 7
   2. Java EE 平台简介 8
   3. [Bootstrap 框架 8](#_TOC_250017)
   4. [Hibernate 框架 9](#_TOC_250016)
   5. Pyhton 网络爬虫 10
   6. [个性化推荐技术简介 10](#_TOC_250015)
      1. 推荐技术的分类和介绍 10
3. [系统需求分析和概要设计 **13**](#_TOC_250014)
   1. [系统需求分析 13](#_TOC_250013)
      1. 系统的功能性需求分析与建模 13
      2. 系统的非功能性需求 14
   2. [数据库设计 15](#_TOC_250012)
      1. 概念结构设计 15
      2. 数据表结构及作用 15
   3. [系统架构设计 18](#_TOC_250011)
   4. [系统功能模块设计 21](#_TOC_250010)
4. [详细设计与关键技术 **23**](#_TOC_250009)
   1. [会议信息的组织与浏览 23](#_TOC_250008)
      1. 会议信息的浏览 23
      2. 会议信息的分类 24
      3. 会议信息的搜索 25
   2. [会议推荐模块 26](#_TOC_250007)
      1. 用户兴趣偏好的获取 26
      2. 基于用户的协同过滤推荐算法（User CF） 27
   3. [会议更新推送模块 32](#_TOC_250006)
      1. 网络爬虫技术模块 33
      2. 收藏会议新闻更新推送 35
5. 系统使用和运行 **38**
   1. [系统使用说明 38](#_TOC_250005)
      1. 运行环境 38
      2. 系统安装 38
   2. [系统运行结果 38](#_TOC_250004)
6. 总结与展望 45

[6.1 总结 45](#_TOC_250003)

[6.2 展望 45](#_TOC_250002)

[致谢 46](#_TOC_250001)

[参考文献 47](#_TOC_250000)

**1** 绪 论

本章主要介绍了学术会议推送系统的研究背景和目的以及分析了国内外学术会议系统的研究现 状和国内外相似产品，阐述了系统设计目标和论文组织结构。

### 研究背景和目的

在互联网的普及的今天以及“信息爆炸”的背景下，人们对信息的需求越来越广泛，信息的时效 性也越来越强，然而如何快速准确地从互联网上获取自己所需的信息却越来越困难。在科研领域， 目前国内外举行的学术交流越来越多，各种学术会议也日益频繁起来。因此科研人员对这些学术会 议的动态以及相关论文的征集和收录情况有强烈的需求，也希望得到及时的会新闻信息。然而科研 人员查询最新的会议信息或往届论文收录，往往通过互联网查询搜索各会议的历年或历届的官方网 站，网络中及其庞大的信息量导致了搜索引擎查询效率的大幅降低，要查询哪些会议符合自己的需 求困难如同大海捞针，同时对于会议的新动态和新通知也难以得到及时通知。

为了节省科研人员的时间和精力，本文结合当今日益成熟的网络技术，研究建设一种学术会议 推送系统，能够从互联网上自动搜集学术会议信息推送给用户，同时为用户提供更为便捷的会议查 询途径。学术会议推送系统根据用户的兴趣领域和兴趣会议等信息，主动将用户偏好与会议内容相 匹配，再推送给用户，实现以用户为中心，适时适度精准地为用户进行个性化推荐服务。并且能将 会议的新消息、通知等及时向用户汇报，在会议结束后该系统还能及时将会议论文网址信息等向用 户及时告知，以方便用户跟踪会议主题。

### 国内外学术会议系统研究现状

当前世界上举办的学术会议日益增多，然而无论是国外学术会议还是国内学术会议，在互联网 上都有各自的官方宣传途径，具体到某一年某一届会议的官网地址都不相同。若仅仅通过传统的搜 索引擎策略，科研人员查找会议信息会十分受限，因此市面上出现了许多学术会议系统，收集了大 量不同领域的学术会议信息，例如，国际上会议信息比较齐全的学术会议系统有 Allconferences[1]、ourGlocal[2]，目前国内比较有代表性的学术会议系统有 cnki 中国学术会议网 [3]、学术会议搜索 [4]等。这些系统的出现节省了大量的人力物力以及时间成本，也更加高效便利。实际上目前许多学会、企 业甚至学院高校都创建了各自的相关会议系统，但这些系统的主要业务只是收集和汇聚大量会议信 息，对会议的后续消息没有进一步跟进，将重心放在发布会议信息上，面向不同的用户群体提供同 样的死板的信息空间, 缺乏灵活性，没有考虑到用户个体的兴趣的差异。这种服务模式使得用户必

须主动到服务器上去寻找信息，信息和服务器反而变成了被动应答一方[5]，然而在网络信息爆炸的 环境下，这种模式已经无法满足用户日趋多元化和个性化的信息需求。

* + 1. 国内外相似产品分析

调查研究发现， 众多大型学术会议系统均没有进一步开发针对用户信息的个性化会议推荐模块， 仅有少数国外会议系统对会议后续内容的跟进和新通知做了及时反馈。通过搜集查找，本文对国内外相关会议系统现状做了调查分析。主要产品有以下几个代表：

1. All Conferences 会议系统，如图 1.1 所示。该系统收集了大量英文学术会议，学科分类十分详细，信息量庞大；会议网站页面设计简洁，重点突出明了；对会议地点引入了地图定位 API；没有会议后续论文收录情况；会议新闻没有动态更新。

图 **1.1 All Conferences** 会议系统界面

**Fig.1.1 All Conferences system page**

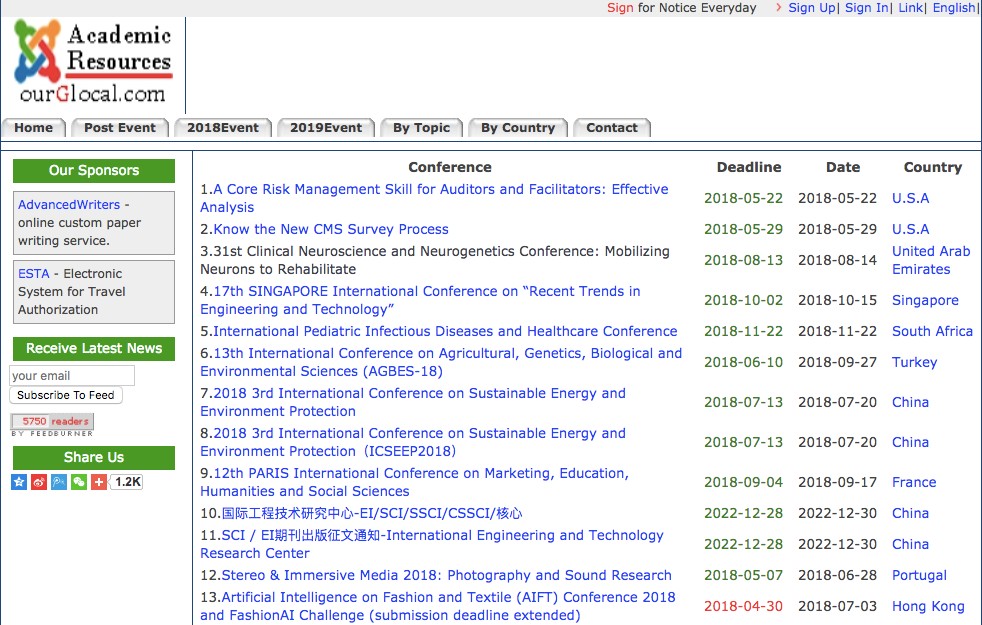
1. ourGlocal 会议系统，如图 1.2 所示。该系统对会议新消息有及时反馈和更新，为本文学术会议推荐系统的开发提供了良好的借鉴。

图 **1.2 ourGlocal** 会议系统界面

**Fig.1.2 ourGlocal system page**

1. cnki 中国学术会议网，如图 1.3 所示。该系统是国内主要的学术会议信息平台之一，支持中英文双语；收录国内会议较多，国外会议具体信息大多不详（如会议议程、主题报告等）；无后续动态更新。



图 **1.3** 中国学术会议网界面

**Fig.1.3 Chine academic conferences web page**

### 系统设计目标

本文研究建设的学术会议推送系统，结合实际需求，通过对一些主流业务系统开发技术的分析 与比较，出于对系统未来的可重用性、可维护性和可扩展性方面的考虑，系统采用三层 B/S 结构体系，使用 MySQL 数据库对信息数据进行管理和存储。选择 Java EE 平台，采用 MVC 分层设计模式，从表示层、业务逻辑层、持久层进行系统架构设计，实现系统低耦合、可维护和可扩展，前端 采用 Bootstrap 技术设计 Jsp 动态网页。通过调查分析国内外有借鉴意义的相似会议平台，从用户需求出发，实现以下功能:

1. 可浏览所有会议列表，查看会议详细信息和会议简单查询功能；
2. 在线注册，登录后可进行会议收藏功能、会议推荐功能、会议复杂查询功能；
3. 已收藏的会议有新动态，用户登录时，弹出提示框通知用户，同时收藏列表该更新会议有红字 提示。

建立学术会议推送系统，将为科研人员搜索合适的学术会议节省大量时间精力，提供更大的便

利。

### 论文组织结构

本文组织结构如下所示：

1. 阐述论文的研究意义，分析国内外学术会议系统发展与现状及相似系统的特点；
2. 简要介绍系统使用的开发工具及关键技术；
3. 系统需求分析和概要设计，总结了学术会议推送的功能性和非功能性需求，介绍了系统的整体 框架、功能模块以及数据库的设计；
4. 系统详细设计，介绍系统的主要功能模块的详细流程设计以及核心代码的实现；
5. 介绍系统使用说明，全面展示了系统最终的页面效果和运行结果；
6. 总结与展望，对论文总结，并对将来需要改进的方面进行了展望。

# **2** 系统开发工具及关键技术研究

本章介绍了系统开发工具和关键技术研究，包括 B/S 模式体系结构、Java EE 平台、Bootstrap 框架、Hibernate 框架、Python 网络爬虫和个性化推荐技术。

* 1. **B/S** 模式体系结构

B/S 结构（Browser/Server），即浏览器/服务器模式结构，是 Internet 技术和 WEB 兴起后形成的一种网络结构模式，是对 C/S (Client/Server)结构进一步的改进，客户端的主要应用软件是 WEB 浏览器[6]。在 B/S 结构下，服务器端实现所有软件应用的业务逻辑，Web 服务器实现表示层，而浏览器则是客户端进行业务请求和处理的唯一工具。B/S 结构的主要特点是分布性强、开发简单、维护便利、且共享性强、总体成本低。

这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维 护和使用。客户机上只要安装一个浏览器，如 Internet Explorer，服务器安装 SQL Server、Oracle、MYSQL 等数据库。浏览器通过 Web Server 同数据库进行数据交互[7]。B/S 是一种全新的软件系统构造技术，因此 B/S 结构更成为当今应用软件的首选体系结构。

* + 1. 三层 **B/S** 结构

三层 B/S 结构就是把二层 C/S 结构的事务处理逻辑模块从客户端的任务中分离出来，由单独组成的一层来负担其任务，于是客户机的压力便大大减轻了，从而把负荷均衡地分配给了 Web 服务器，于是由原来的两层的 C/S 结构转变成一种由表示层、功能层、数据库服务层构成的三层分布式结构[8]，B/S 结构是以访问 WEB 数据库 为中心，HTTP 为传输协议，客户端通过浏览器来访问WEB 服务器以及与其相连的后台数据库， B/S 三层体系结构如下图 2.1 所示：

Web浏览器 Web服务器 数据库服务器



事务处理逻辑

接受请求

数

据

库

数据处理逻辑

提出请求

显示逻辑

图 **2.1 B/S** 三层体系结构图

**Fig.2.1 B/S three-layer architecture diagram**

1. Web 浏览器

即客户端，主要完成与系统后台的交互及最终结果的输出功能。在客户端向指定的 Web 服务器提出请求，Web 服务器验证用户身份成功后，通过 HTTP 协议把所需文件资料传给用户，客户端接受并显示在浏览器上[9]。

1. Web 服务器

即功能层，客户端的请求被 Web 服务器接收后，第一步是执行相应的扩展应用程序与数据库进行连接，然后运用 SQL 语句等方式申请向数据库服务器进行数据处理，数据处理的结果由后台数据库服务器提交给 Web 服务器，最后传递回客户端[10]。

1. 数据库服务器

数据库服务器是接受 Web 服务器的客户请求对数据库独立地进行增加、查询、修改、更新等功能的操纵处理。

* 1. **Java EE** 平台简介

Java EE（Java Platform，Enterprise Edition）是 Sun 公司在 2006 年推出的企业级应用程序标准， 并公布了 EJB 3.0 和 JSF 1.2 主要规范。Java EE 大幅提升了软件的开放程度，得到了行业开发者的广泛支持，但其暴露出的应用程序冗杂、开发和维护成本高、难以测试等问题带来了比其商业价值更 大的复杂性，急需一种新型轻量级的 Java EE 方案来补充。

现在，轻量级 Java EE 平台已经成为企业开发的标准平台之一，由于其性能稳定、开放程度较高、跨平台性好、安全性高等优点，受到了企业级开发领域的喜爱与认同。对于企业而言，选择 Java EE 平台搭建应用特别是信息化平台等大型应用，就等于选择了开放、稳定、安全以及高效的开发道路[11]。据统计，在信息化需求较高的行业（如电信，证券，电子商务和银行等行业），选择 Java EE作为其开发平台的比例超过半数。

## Bootstrap 框架

Bootstrap 是 Twitter 公司开发的一款基于 HTML、CSS、Javascript 的前端框架，它提供了一套前端工具包 ，适用于快速开发 Web 应用程序和网站。Bootstrap 是基于 JQuery 框架开发的，它在JQuery 框架的基础上进行了更为个性化和人性化的完善，形成一套自己独有的网站风格，并兼容大部分 JQuery 插件。它为开发人员创建接口提供了一个简洁统一的解决方案，提供了一个带有网格系统、链接样式、背景的基本结构，自带全局的 CSS 设置，定义基本的 HTML 元素样式、可扩展的class，以及一个先进的网格系统[12]。包含了多个可重用的组件，用于创建图像、下拉菜单、导航、 警告框、弹出框等等，功能强大，还提供了基于 Web 的定制。同时，Bootstrap 也是一个开源的产

品。

## Hibernate 框架

Hibernate 是开源的一个 ORM（Object-Relational Mapping）框架，即对象关系映射框架，它的主要工作是在对象和关系型数据库之间进行了一次完整的映射，从对象（Object）映射到关系（Relation），再从关系映射到对象[13]。它对 JDBC 访问数据库进行了代码的封装，为数据层的频繁访问简化了大量重复性代码。

Hibernate 除了能在 Java 类与数据库表之间建立映射关系，还能自动生成 SQL 语句，提供查询获取数据的方法，自动执行，可以大幅度缩短使用 JDBC 处理数据持久化的时间。它是企业级开发中的主流框架，映射的灵活性很出色，支持很多关系型数据库。

Hibernate 的核心结构如图 2.2 所示：

应用程序

配置文件



Configuration

Hiibernate.cfg.xmll

映射文件

SessionFactory

xxx.hbm.xmll

查询



异常

正常

Transaction

Session

提交数据库

回滚

Query



图 **2.2Hibernate** 核心结构图

**Fig.2.2 Hibernate core structure diagram**

从上图中展现了 Hibernate 的六个核心接口和两个主要配置文件：

1. Configuration 接口：负责配置并启动 Hibernate；
2. SessionFactory 接口：负责 Hibernate 的初始化；
3. Session 接口：负责持久化对象的 CRUD 操作；
4. Transaction 接口：负责事务；
5. Query 接口和 Criteria 接口：负责执行各种数据库查询操作。
   1. **Pyhton** 网络爬虫

网络爬虫（Web Spider），又被称为网页蜘蛛、网络机器人。爬虫程序或脚本的基本原理是按照一定的规则，自动地从网站上检索和抓取相应的文本信息[14]，这种规则也称为正则表达式。相关模 块及框架有 Requests ，BeautifulSoup 等。Beautiful Soup 是 Python 的一个库，主要的功能是通过解析文档从网页抓取数据，它能提供一些简单的函数处理导航、搜索、修改分析树等功能。Requests 是基于 urllib3 的一个 Python 库，也是采用 Apache2 Licensed 开源协议的 HTTP 库。它比 urllib2 更加方便，可以节约大量的工作，完全满足 HTTP 测试需求。

爬虫具体流程如下：先由 urllib 的 request 打开 url 得到网页 html 文档；浏览器打开网页源代码分析元素节点；通过 Beautiful Soup 或则正则表达式提取想要的数据；存储数据到本地或数据库（抓取、分析、存储）。

### 个性化推荐技术简介

随着互联网的迅速发展，信息爆炸增长的同时也导致了信息获取效率大幅度降低。在信息过载 的背景下，个性化推荐技术出现了，面对海量的信息数据，通过筛选、过滤、计算，将用户最感兴 趣、关注度最高的信息展现给用户，节约了用户主动筛选信息所耗费的时间，极大地提高了信息获 取效率。

过去的 20 年间，个性化推荐技术得到充分的发展和进步。对个性化推荐技术的研究最早可追溯 到上世纪 90 年代——个性化导航系统 WebWatcher[15]，个性化推荐系统 LIRA[16]，个性化智能导航体 Letizia[17]，这三个系统可认为是个性化推荐技术发展初期最为经典的系统，标志着个性化推荐技术研究的开始。国外推出了多个实现初步个性化服务的原型系统，如 SavvySearch[18]，WebACE[19]， WebMate[20]等。自从 20 世纪 90 年代中期第一篇介绍协同过滤推荐算法的论文问世，来自不同的领域研究者们，提出了各种模型和解决方案，力求对个性化推荐算法进行优化。据粗略统计，所涉及 到的学科包括人工智能、机器学习、认知科学、信息抽取、数据挖掘、预测理论、近似理论，甚至是管理科学、市场营销和心理学。所使用的算法除了传统的协同过滤，还包括图模型（Graph Model）、 矩阵分解（Matrix Factorization）、奇异值分解（SVD，Singular Value Decomposition）、链接分析（Link Analysis）、回归分析（Regression Analysis），以及机器学习领域各种分类和学习算法[21]。

* + 1. 推荐技术的分类和介绍

目前，传统的推荐技术方法分为三大类，分别是基于内容的推荐、基于协同过滤的推荐和混合 推荐：

1. 基于内容的推荐（Content-based Recommendation）

基于内容的推荐是推荐系统中较为常用的一种技术，主要是向用户推荐那些与用户兴趣偏好相 似的物品。 主要内容是基于信息检索与信息过滤技术，以对文本信息和数据信息进行整理和建模为基础，针对用户的不同兴趣爱好进行推荐。

基于内容的推荐算法基本流程如下：首先，为每一个用户建立一个户兴趣偏好信息模型，用于 描述其喜好和兴趣属性，再为每一个物品建立物品属性特征模型，然后分析和进行匹配，根据物品 属性模型与用户信息模型的匹配程度或者直接根据比较物品属性模型之间的属性相似度，最后决定 是否要向用户推荐该物品。它只针对物品本身的特征进行建模，而不依赖于用户对物品的评价。

1. 基于协同过滤的推荐（Collaborative Filtering Recommendation）

协同过滤推荐技术是推荐系统中应用最为成功、广泛的技术之一，第一篇推荐技术相关的论文 就是介绍协同过滤推荐技术。协同过滤算法不需要提前得到用户的兴趣偏好信息或物品的特征属性等，它只依赖于用户过去的行为（如对物品的浏览或购买等），从而收集用户对物品的反馈，并以评分的形式表示，然后计算用户与用户之间的相似程度，然后分析与目标用户相似程度较高的邻居集 合对其他物品的评价，来计算目标用户对其没有进行反馈的其他物品的可能感兴趣程度，最后系统 根据这一兴趣程度来对目标用户进行物品推荐。协同过滤推荐的重点在于对用户群体的评分信息进 行整理和建模，而与物品的本身属性特征无关[22]，这恰好与基于内容的推荐相反。因此，协同过滤 最大优点是对推荐物品没有特殊的要求，能够简单处理非结构化的复杂物品对象。

基于用户的协同过滤(User-based Collaborative Filtering)，又被称为 KNN (K-Nearest Neighbor)， K 最近邻算法。其核心思想是根据相似度计算出目标用户的邻居用户集合，对物品的评分预测由邻居用户评分的加权组合计算得出，最后为目标用户推荐得分高的物品。这种算法可以总结成三步： 首先对用户的过去行为抽象成评分的形式来计算用户与用户之间的相似度；然后选择与目标用户相 似度最高的 K 个用户为邻居用户集合，来为目标用户作推荐；最后通过对邻居用户的评分的加权平均，来对物品进行评分预测，把评分最高的 N 个项推荐给目标用户[23]。

1. 混合推荐（Hybrid Recommendation）

因为各种推荐技术都有各自的优缺点，并没有十全十美的一项，所以在实际应用中，从各自需 求的差异以及不同用户对象群体和特定物品的特征属性的角度出发，混合推荐技术往往最常被采用， 多种推荐技术相混合，弥补各自推荐技术的不足和缺陷，从而设计出最符合用户需求的推荐技术。 基于内容的推荐和基于协同过滤的推荐组合是现阶段研究和应用领域最流行、运用面最广的混合推 荐技术。最简单的做法就是分别用基于内容的方法和协同过滤推荐方法去产生一个推荐预测结果， 然后用某方法组合其结果，例如：加权、变换、混合、层叠、特征组合、特征扩征等组合方式。虽

然在理论上混合推荐可以由很多种组合方式，但是针对某一具体问题并不一定完全适用，所以混合

推荐最主要的原则之一就是，通过混合后能够实现对各自推荐技术的缺点的补救和避免 [24]。

# **3** 系统需求分析和概要设计

本章主要阐述了系统需求分析和概要设计，对系统的功能性需求进行了用例图建模分析，详细 介绍了数据库设计、系统架构设计和系统功能模块设计。

### 系统需求分析

系统需求分析包括系统的功能性需求和非功能性需求的分析。

* + 1. 系统的功能性需求分析与建模

用户包括非注册会员用户和注册会员用户，两者的根本区别就是有无注册，两者使用系统的功 能是不同的，注册会员用户登录后能享受更丰富的系统功能。从用户角度考虑，学术会议推送系统 需要具备以下功能：

1. 针对非注册会员用户

非注册会员用户使用本系统能够完成注册功能、会议浏览功能（包括会议列表浏览和会议详情 浏览）以及会议简单搜索功能，该角色的用例图如图 3.1 所示。



注册

<<extends>>

会议列表浏览

会议浏览

<<extends>>

会议详情浏览

<<include>>

非注册会员

会议简单搜索

图 **3.1** 非注册用户用例图

**Fig.3.1 Non-registered user case diagram**

1. 针对注册会员用户

注册过的用户登录学术会议推送系统后，可对自己的个人信息进行查看和修改；除了浏览会议 信息、查看会议详情、会议的简单搜索功能外，还能进行会议的复杂搜索功能、根据用户兴趣偏好 进行会议推荐的功能；在浏览会议的过程中，还有把感兴趣的会议加入收藏的会议收藏功能，此外 通过查看我的收藏可以查看自己收藏过的会议，不满意可以删除，还有会议更新信息通知。该角色

的用例图如图 3.2 所示。



<<extends>>

会议列表浏览

会议浏览 <<extends>>

会议详情浏览

<<include>>

<<extends>>

会议简单搜索 会议复杂搜索

<<extends>>

查看

<<include>>

个人信息管理

<<extends>>

<<include>>

登录

注册会员

修改

<<include>>

<<include>>

会议推荐

<<extends>>

增加

会议收藏管理

<<extends>>

<<extends>>

<<extends>>

查看

会议更新通知

删除

图 **3.2** 注册用户用例图

**Fig.3.2 Registered user case diagram**

* + 1. 系统的非功能性需求

通过分析要求系统可以安全稳定的运行，具体系统的性能需求总结如下:

1. 健壮性

学术会议推送系统的网站页面需要布局稳定，使用不同浏览器的用户显示的界面应当一致。当 用户进行非法请求或输入字符非法时，系统会自动检测并提示错误，并且能够处理运行时出现的各 种异常状况。

1. 系统稳定性

学术会议推送系统要保证系统稳定，长时间运行不会出现错误，能顺利完成用户的各项操作请 求。即使用户进行了错误的操作，也能触发错误提醒，提示用户重新选择操作，同时不影响系统正 常运行。

1. 良好的用户体验

学术会议推送系统还需要做到用户在使用系统时，能够让用户很容易地进行良好的页面交互， 网页布局、图片位置、文字大学、按钮、下拉框等组件位置，这些细节都要做到设计合理，符合大

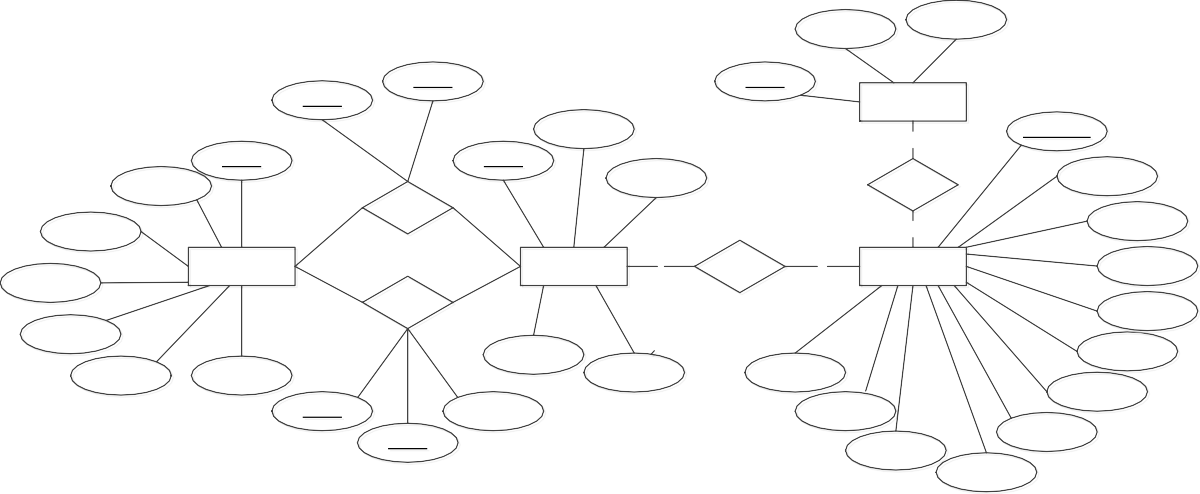
多数用户习惯。

### 数据库设计

系统将用户信息、会议基本信息、会议召开信息、会议新闻信息、用户收藏会议的信息以及用 户推荐会议信息进行了数字化，系统共设计并创建了 6 张数据表。

* + 1. 概念结构设计

图 3.3 展示了用户、会议和会议新闻的整体 E-R 模型，图中主要反应了各个实体的属性以及实体之间的联系。用户表和会议表是多对多的关系，两者之间采用外键的方式进行关联。会议表和会 议新闻表是一对多的关系，两者之间采用外键的方式进行关联。



内容

召开会议iid

会议iid

新闻iid

用户iid 会议新闻表

会议全称

N

召开会议iid

用户iid 会议iid

名称

会议介绍

对应

会议iid

收藏

密码

N

N

1

会议简称

用户表

会议基本表 1

对应

N 会议召开表

会议信息

国家

N

N

推荐

官网

兴趣领域

会议刊物

地点

兴趣会议

最新新闻iid

会议领域

新闻网址

开始时间

用户iid

推荐排序

会议进程

结束时间

会议iid

征稿信息

组织者

图 **3.3 E-R** 图

**Fig.3.3 E-R diagram**

* + 1. 数据表结构及作用

以下介绍用户信息表、会议基本信息表、会议召开信息表、会议新闻表、用户收藏会议表以及 用户推荐会议表 6 张表，包括表的名称，表中记录的字段名称、字段类型和长度限制、字段约束条件以及字段备注（表示该字段代表的含义）。

1. 用户信息表 USER

用于保存所有用户个人的信息。其主键约束为 USERID，外键约束为 PRE，如表 3.1 所示。

表 **3.1** 用户信息表

**Table 3.1 Table of user information**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 约束 | 备注 |
| USERID | VARCHAR(45) | 主键 | 用户 ID |
| NAME | VARCHAR(45) | 无 | 名称 |
| PASSWOED | VARCHAR(45) | 无 | 密码 |
| COUNTRY | VARCHAR(45) | 无 | 国家 |
| INTERESTFIELD | VARCHAR(45) | 无 | 兴趣领域 |
| INTERESTMEET | TEXT | 无 | 兴趣会议 |
| PRE | INT(11) | 无 | 该用户上次登录时，会议  新闻表最新新闻 id |

1. 会议基本信息表 MEETBASIC

用于保存会议的基本信息。其主键约束为 ID，如表 3.2 所示。

表 **3.2** 会议基本信息表

**Table 3.2 Table of meeting basic information**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 约束 | 备注 |
| ID | INT(11) | 主键 | 会议 ID |
| FULLNAME | VARCHAR(100) | 无 | 会议全称 |
| INFORMATION | TEXT | 无 | 会议介绍 |
| FIELD | VARCHAR(45) | 无 | 会议领域 |
| PUBLICATION | VARCHAR(45) | 无 | 会议刊物 |

1. 会议召开信息表 MEET

用于保存所有具体会议的召开信息。会议召开信息表的主键约束为 MEETID，外键约束为 ID， 如表 3.3 所示。

表 **3.3** 会议召开信息表

**Table 3.3 Table of the held meeting information**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 约束 | 备注 |
| MEETID | INT(11) | 主键 | 召开会议 ID |
| ID | INT(11) | 外键 | 会议 ID |
| NAME | VARCHAR(45) | 无 | 会议简称 |
| INFORMATION | TEXT | 无 | 会议信息 |
| WEBSITE | VARCHAR(45) | 无 | 官网 |
| ADDRESS | VARCHAR(45) | 无 | 地点 |
| STARTTIME | DATE | 无 | 开始时间 |
| ENDTIME | DATE | 无 | 结束时间 |
| ORGANIZATION | VARCHAR(45) | 无 | 组织者 |
| CALLFOR | VARCHAR(45) | 无 | 征稿信息 |
| PROGRAM | VARCHAR(45) | 无 | 会议进程 |
| NEWS | VARCHAR(45) | 无 | 新闻网址 |

1. 会议新闻表 NEWS

用于保存会议的新闻信息。该表的主键约束为ID，外键约束为MEETID，如表 3.4 所示。

表 **3.4** 会议新闻表

**Table 3.4 Table of meeting news**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 约束 | 备注 |
| ID | INT(11) | 主键 | 新闻 ID |
| MEETID | INT(11) | 外键 | 召开会议 ID |
| CONTEXT | TEXT | 无 | 新闻内容 |

1. 用户收藏会议表 COLLECT

用于保存用户收藏会议的信息。主键约束为 USERID，MEETID，如表 3.5 所示。

表 **3.5** 用户收藏会议表

**Table 3.5 Table of user collect meeting**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 约束 | 备注 |
| USERID | VARCHAR(45) | 主键 | 用户 ID |
| MEETID | INT(11) | 主键 | 召开会议 ID |

1. 用户推荐会议表 RECOMMEND

用于保存用户收藏会议的信息。主键约束为 USERID，MEETID，如表 3.6 所示。

表 **3.6** 用户推荐会议表

**Table 3.6 Table of user recommend meeting**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 约束 | 备注 |
| USERID | VARCHAR(45) | 主键 | 用户 ID |
| MEETID | INT(11) | 主键 | 召开会议 ID |
| NO | INT(11) | 非空 | 推荐会议排序 |

### 系统架构设计

本系统选择 Servlet 和 Hibernate 持久层框架来搭建。Servlet 实现了 MVC 模式，清晰地区分了模型层、视图层和控制层。既满足了系统当前的需求，也能够保证系统稳定运行，还可以对系统 进行进一步的扩展以及以后维护。系统的架构设计说明如图 3.4 所示：

Request

配置文件

Hibernate

映射文件



获取数据

Servlet (Controller)

JavaBeans (Model)



Java对象(PO)

Response



数据库



更新模型

JSP

(View)



图 **3.4** 系统架构设计图

**Fig.3.4 System structure design diagram**

1. 模型层（Model）

完成业务逻辑、数据持久化工作，通过 Hibernate 的 PO 对象来实现数据库的操作和相关计算， 存放 PO 对象（Entity 类）和.hbm.xml 文件配置。

①.hbm.xml 文件配置

.hbm.xml 文件，即 Hibernate 映射文件，是 Hibernate 与数据库之间进行持久化的桥梁。由节点定义映射内容并指定所对应的 JavaBean 的位置。最后要将该文件配置到 hibernate.cfg.xml 中。部分代码如下：

<hibernate-mapping>

// MeetEntity 类对应数据表 meet

<class name="com.rose.model.MeetEntity" table="meet" schema="meetingsys">

//主键映射，name 是类的属性名

<id name="meetid">

<column name="meetid" sql-type="int(11)"/>

</id>

//普通字段映射

<property name="name">

<column name="name" sql-type="varchar(45)" length="45"/>

</property>

<property name="information">

<column name="information" sql-type="text" not-null="true"/>

</property>

……

</hibernate-mapping>

②hibernate.cfg.xml 文件配置

包含了 Hibernate 与数据库的基本连接信息和 Hibernate 的基本映射信息，即系统中每一个 Java 实体类与其对应的数据库表之间的关联信息。在 Hibernate 工作的初始阶段，这些信息被先后加载到Configuration 和 SessionFactory 实例中。详细代码如下：

<hibernate-configuration>

//下列是配置 session-factory 的过程，SessionFactory 类主要负责保存 Hibernate 配置信息，以及对

Session 的操作

<session-factory>

//设置数据库的连接 url:jdbc:mysql://localhost:3306/meetingsys，meetingsys 是数据库名

<property name="connection.url">jdbc:mysql://localhost:3306/meetingsys</property>

//配置 mysql 数据库驱动程序

<property name="connection.driver\_class">com.mysql.jdbc.Driver</property>

//连接数据库，用户名 rose，密码 11100

<property name="hibernate.connection.username">rose</property>

<property name="hibernate.connection.password">111000</property>

//指定映射文件，扩展名为.hbm.xml，可映射多个映射文件

<mapping resource="com/rose/model/CollectEntity.hbm.xml"/>

<mapping resource="com/rose/model/MeetEntity.hbm.xml"/>

……

<mapping resource="com/rose/model/NewsEntity.hbm.xml"/>

1. 视图层（View）

负责展示用户界面，是整个MVC架构中唯一显示给用户看，同时接收用户的输入，显示处理结 果，主要由 JSP 文件来显示视图，选取了目前国内使用较多的 Bootstrap 框架作为系统的前端框架， css、js、images目录下存放了各种资源文件。

1. 控制层（Control）

主要负责视图层和模型层之间的连接，根据用户的输入来控制用户界面数据的显示。它由各类

Servlet 响应文件实现控制，从视图层接收用户请求，传递给模型层，调用相应 PO 对象来访问数据库，然后 Servlet 把处理结果传递到相应的 JSP 页面，协同完成请求。本系统的 Servlet 类功能如表 3.6 所示。

①Servlet 配置

表 **3.6 Servlet** 功能表

**Table 3.6 Table of Servlet function**

|  |  |
| --- | --- |
| Servlet 类 | 功能 |
| controller.Login | 用户登录、  用户收藏会议更新 |
| controller.Register | 用户注册 |
| controller.Save | 用户信息修改 |
| controller.View | 会议详情浏览 |
| controller.Search | 会议分类浏览 |
| controller.SimpleSearch | 会议简单搜索 |
| controller.ComplexSerch | 会议复杂搜索 |
| controller.Collect | 会议收藏 |
| controller.ReCollect | 会议取消收藏 |
| controller.Recommend | 会议推荐 |

Servlet 需要配置在 web 应用中才能响应用户请求，通过修改 web.xml 文件，配置<servlet>和

<servlet-mapping>完成。其中<servlet>中的<servlet-name>取为类的名称，例如类名为 Collect，这里命名为 Collect， <servlet-class> 是类的具体路径；<servlet-Mapping> 是配置映射信息， 此次的

<servlet-name>对应的是 Servlet 名，和<servlet>中的<servlet-name>相同，这里是 Collect，<url-pattern>

是映射路径，也就是访问 Servlet 的名称。详细代码如下：

<servlet>

<servlet-name> Collect </servlet-name> //Servlet 名称

<servlet-class>com.rose.controller. Collect </servlet-class> //具体路径

</servlet>

<servlet-mapping>

<servlet-name> Collect </servlet-name> //Servlet 名称

<url-pattern>/ Collect </url-pattern> //映射路径

</servlet-mapping>

### 系统功能模块设计

系统功能模块设计如图 3.5 所示。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | | 学术会议推送系统 | | | | | | | | |  | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | |  | | | | | | | | | | | | | | | | | |
|  | | | |  | | | | | | | | | | |  | | | | | | | | | | | |  | | | | | | | | |  | | | |
| 会议浏览模块 | | | | | | | |  | | | 会议搜索模块 | | | | | | | |  | | | | 会议收藏模块 | | | | | | | |  | 用户信息管理模块 | | | | | | | |
|  | | | |  | | | | | | | | | | |  | | | | | | | | | | | |  | | | | | | | | |  | | | |
|  |  | | |  | | |  | | |  | | |  | | | |  | | |  | | | |  | | |  | | |  | | |  | | |  | | |  |
| 所有会议 | |  | 会议分类 | |  | 会议推荐 | |  | 关键词搜索 | |  | 类别搜索 | |  | | 时间搜索 | |  | 地点搜索 | |  | | 添加收藏会议 | |  | 删除收藏会议 | |  | 更  新会议新通  知 | |  | 用户注册 | |  | 会员登录 | |  | 资料修改 | |

图 **3.5** 系统功能模块图

**Fig.3.5 System function module diagram**

1. 会议浏览模块

用户登录系统后，会议按照所有会议、我的收藏、推荐会议列表展示。用户也可以在会议分类 列表下浏览会议信息，会议分为数据库类、数据挖掘类与内容检索类，可以点击查看会议详情。系 统根据用户的研究兴趣推送相关会议。

1. 会议搜索模块

用户根据会议的基本信息进行查询，可以输入会议关键词、会议领域、会议开始时间、结束时 间、是否征稿、会议地点，在所有会议中搜索。

1. 会议收藏模块

用户可以对感兴趣的会议进行收藏，会议被收集到收藏列表中，用户也可以随时删除收藏会议。 系统会实时更新会议动态，若用户收藏的会议有新通知，会在用户登录时弹出提示消息。会议结束 后，系统还能连接 DBLP 计算机科学文献库查询会议论文收录情况。

1. 用户信息管理模块

提供新用户注册、会员登陆、会员资料修改功能。用户注册时，对用户的兴趣信息进行收集。

# **4** 详细设计与关键技术

本章介绍了系统的详细设计和关键技术的实现，主要包括会议信息的组织与浏览、会议推荐模 块和基于用户的协同过滤推荐算法，以及会议更新推送模块和网络爬虫技术。

### 会议信息的组织与浏览

会议信息的组织与浏览的详细设计包括会议信息的浏览、会议信息的分类和会议信息的搜索。

* + 1. 会议信息的浏览

用户登录学术会议推送系统后的首页，即所有会议列表浏览的界面。列表只显示了会议的基本信 息，如会议名称、会议领域、开始时间、结束时间、地点、用户收藏情况。如果用户对某个会议感 兴趣，想要查看该会议详细信息，可以点击“查看”按钮，系统会跳转到该会议的详细介绍界面。会 议浏览顺序图如图 4.1 所示。



用户

选择要查看的会议

传递会议id

连接数据库查询该会议记录

query(meetid):m

传递查询结果的会议对象

session.setAttribute(m)

展示会议m的详细信息

会议详细界面single.jsp

页面控制器controller.View

会议列表界面all.jsp

图 **4.1** 会议浏览顺序图

**Fig.4.1 Meeting view sequence diagram**

会 议 的 浏 览 功 能 由 表 3.6 中 的 Servlet 类 controller.View 的 doPost 方 法 实 现 ， request.getParameter("meetid")获取客户端的请求的会议 id，从数据库的 meet 查询会议 id 相同的记录，找到符合条件的记录 m,通过 session.setAttribute("m",m) 传递会议对象 m 到会议详情界面single.jsp。详细代码如下：

int meetid = Integer.parseInt(request.getParameter("meetid"));//获取客户端请求的会议 id Configuration cfg = new Configuration().configure();//Hibernate 连接数据库SessionFactory sf = cfg.buildSessionFactory();

Session hsession = sf.openSession(); Transaction ts = hsession.beginTransaction();

Query hquery = hsession.createQuery("from MeetEntity m where m.meetid=?");//查询会议 id

hquery.setInteger(0,meetid);

List<MeetEntity> m = hquery.list();//查询到的会议对象session.setAttribute("m",m.get(0)); //传递会议对象response.sendRedirect("single.jsp");//跳转到会议详情页面

* + 1. 会议信息的分类

为了便于用户查找会议，在本系统把会议按照会议领域和会议刊物名称进行分类，用户通过点 击相应类别名称的多选框，来浏览此类别的会议，缩小用户查找范围，节约用户时间。所有会议列 表浏览页面的 CATEGORIES 一栏展示了本系统的所有会议领域，包括：数据库类、数据挖掘类、内容检索类；会议刊物名称包括：SIGMOD、SIGKDD、SIGIR、VLDB、ICDE、CIKM、DASFAA、ECML-PKDD、CIDR、WWW、ESWC 等。勾选会议领域，点击底部“提交”按钮，系统会搜索所有会议返回包含该类别的会议列表。会议分类顺序图如图 4.2 所示。



用户

会议列表界面all.jsp

页面控制器controller.View

会议结果界面result.jsp

选勾要查看的会议类

别

传递会议类别

连接数据库查询该类别会议

query(field,

publication):smList

传递查询结果的会议对象集合

session.setAttribute(smList)

展示会议集合smList的列表信息

图 **4.2** 会议分类顺序图

**Fig.4.2 Meeting classification sequence diagram**

会 议 的 分 类 功 能 由 表 3.6 中 的 Servlet 类 controller.Search 的 doPost 方 法 实 现 ， request.getParameterValues("checkbox")获取客户端的请求表单，用数组s[]存储会议领域和刊物名称， 然后在从数据库的 meet 表的 field 和 publication 属性中通过是否包含关键字的方法来匹配会议，找到符合条件的记录，结果集为 smList。最后通过 session.setAttribute("smList",smList) 传递结果集到搜索结果展示界面 result.jsp。

* + 1. 会议信息的搜索

会议信息的搜索功能包括会议简单搜索和会议复杂搜索，且只有登录后的用户能够使用会议复 杂搜索功能。会议简单搜索功能通过 SQL 语句的模糊查询会议名称来实现，用户只需要系统首页顶端的会议搜索栏中输入关键字，系统会返回会议名称中包含有该关键字的所有会议集合。用户登录 后，点击会议搜索栏的“高级搜索”或系统导航栏的“会议搜索”，可以触发系统的会议复杂搜索功能， 用户可以输入会议名称的关键字，选择会议领域，是否征稿，开始时间、结束时间，会议地点，系 统会返回所有符合条件的会议集合。会议搜索顺序图如图 4.3 所示。



用户

首页index.jsp 简单搜索控制器controller.SimpleSearch

搜索页面search.jsp 复杂搜索控制器controller.ComplexSearch 结果界面result.jsp

输入关键字key

传递关键字key

连接数据库查询该类别会议

query(key):smList

传递查询结果的会议对象集合

session.setAttribute(smList)

展示会议集合smList的列表信息

登录（账号、密码）

登录验证成功，选择查询条件condition

传递查询条件condition

连接数据库查询该类别会议

query(condition):smList

传递查询结果的会议对象集合

session.setAttribute(smList)

展示会议集合smList的列表信息

图 **4.3** 会议搜索顺序图

**Fig.4.3 Meeting search sequence diagram**

实现方法与 4.1.2 章节会议信息的分类功能类似，主要由表 3.6 中的 Servlet 类 controller. ComplexSearch 的 doPost 方法实现，从数据库中查找匹配的会议，最终系统会遍历所有会议信息并返回结果集，通过 session.setAttribute 方法传递到搜索结果展示界面 result.jsp。

### 会议推荐模块

会议推荐模块主要功能是，通过统计用户个人信息和用户的行为，获取用户兴趣偏好，和各个 会议进行相似度计算，为用户过滤相似度较低的会议，从而推荐用户可能会感兴趣的会议。

用户登录系统后，点击会议推荐列表，系统自动推荐会议给用户。根据用户是否进行实时推荐， 以及有无收藏会议，系统推荐会议的活动图如图 4.4 所示。



用户

学术会议推送系统

登录

查看推荐会议

读取数据库用户推荐会议表中该用户的推

荐会议信息

得到推荐会议列表

[否]

是否进行实时推荐

[是]

获取用户信息

[否]

用户是否有收藏会议

[是]

根据用户个人信息（兴趣 运用基于用户的协同过滤

领域、兴趣会议、国 推荐算法，计算用户的相 家），初始化用户的兴趣 似度，得到用户相似集合

偏好标签矩阵

与数据库召开会议表中的 计算用户对集合中其他用

各个会议进行关联度计算 户收藏过，而自身没有收

藏的会议的兴趣程度

对各会议的得分进行倒序排序

得到更新后的推荐会议列表

更新数据库用户推荐会议表中该用户的推荐会议信

息

图 **4.4** 推荐会议活动图

**Fig.4.4 Meeting recommend activity diagram**

* + 1. 用户兴趣偏好的获取

用户的兴趣偏好通常以标签的形式来表示，这里我们采用会议实体域的分类标签。例如：会议 领域分为数据库、数据挖掘、内容检索，这 3 个标签。

获取用户兴趣偏好的方法分为显式和隐式：显式获取用户兴趣偏好需要用户主动提供清晰明确

的兴趣偏好信息或对物品的评分信息，本系统主要通过用户注册时提供的个人信息（包括兴趣领域、 兴趣会议、地点）来获取用户兴趣偏好信息；隐式获取用户兴趣偏好，是指获取用户没有明确描述 的兴趣偏好，则不需要用户提供具体的兴趣偏好信息和评分信息，而是通过协同过滤、内容过滤、 点击流分析等思想或技术手段进行分析计算，从而获得用户兴趣偏好[25]。本系统则是根据用户的会 议收藏行为，运用基于用户的协同过滤的思想来计算用户的兴趣偏好。

若用户没有收藏会议，首先假设相似度的衡量标准为权值 W，相似度的计算需要在会议标签中寻找与用户兴趣偏好标签是否有一致项，如果有一致的项目则累加到权值 W 中。权值与用户对会议的感兴趣程度呈正相关，权值越大，即用户对会议的感兴趣程度越大。最后将各个会议的权值计算 结果进行倒序排序。具体计算流程图如图 4.5 所示。

开始

初始化用户的兴趣偏好标签

根据用户兴趣标签与每个会议进行相似度计算

结束

对权值进行排序，获得会议推荐列表

图 **4.5** 未收藏会议用户的推荐会议计算流程图

**Fig.4.5 Flow chart of recommended meetings for uncollected conference users**

若用户有收藏会议，运用基于用户的协同过滤算法，推荐与该用户兴趣偏好相似的其他用户收藏过的会议。首先根据用户的行为数据（收藏会议），来找出该用户的相似用户，然后将相似用户收藏过的而该用户没有收藏的会议进行推荐度计算，推荐度的计算结合了相似用户与该用户的相似度 累加。

* + 1. 基于用户的协同过滤推荐算法（ **User CF**）

User CF 算法流程图如图 4.5 所示。

开始

根据用户行为数据（收藏会议），来找出指定用户的相似用户集合

建立会议-用户倒查表和用户相似度矩阵

W，运用余弦相似度公式计算用户相似度

搜索相似用户收藏过而指定用户没有收藏的会议，计算指定用户对该会议的兴趣程度

结束

会议兴趣程度倒序排序

图 **4.5User CF** 算法流程图

**Fig.4.5 Flow chart of User CF algorithm**

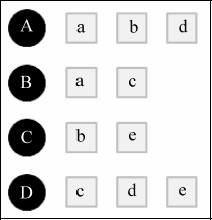
1. 找到和目标用户兴趣相似的用户集合

首先计算两两用户相似度。协同过滤算法主要根据用户行为的相似度计算兴趣的相似度。给定两个用户 u、v，N(u)表示用户 u 有收藏行为的会议集合，N(v)表示用户 v 有收藏行为的会议集合。通

过余弦相似度公式计算：

𝑊!"

= |𝑁(𝑢)⋂𝑁(𝑣)|

举例：下图 4.6 表示用户 A 对会议{a,b,d}有过收藏行为，用户 B 对会议{a,c}有过收藏行为。



|𝑁(𝑢)||𝑁(𝑣)|

图 **4.6** 用户行为记录图

**Fig.4.6 User behavior record diagram**

利用余弦相似度计算可得，用户 A 和用户 B 的兴趣相似度为：

𝑊!"

= |{𝑎, 𝑏, 𝑑}⋂{𝑎, 𝑐}| = 1

这种方法需要计算每一对用户之间的相似度，时间复杂度是 O(U^2)，然而实际上很多两两用户之间并没有对相同会议有过收藏行为，因此当分子为 0 时无需再计算分母，所以此处进行优化：即



|{𝑎, 𝑏, 𝑑}||{𝑎, 𝑐}|

6

首先计算出|N(u)∪N(v)| != 0 的用户对(u,v)，然后计算分母以得到两个用户的相似度。针对此优化， 需要 2 步：

① 建立会议到用户的倒查表 T，一个会议可能有多个用户与其产生过行为，对于每个会议都建立对该会议产生行为用户的列表；

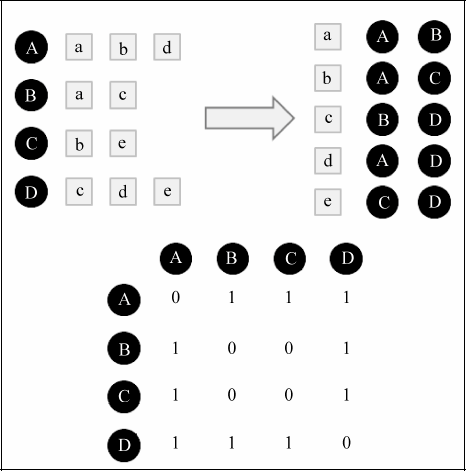
② 根据倒查表 T，建立用户相似度矩阵 W：在 T 中，对于每一个会议 i，设其对应的用户为 j、k ， 在 W 中，更新相应的元素值，w[j][k]=w[j][k]+1，w[k][j]=w[k][j]+1，以此类推，扫描完倒查表 T 中的所有会议后，就可以得到最终的用户相似度矩阵 W，这里的 W 是余弦相似度中的分子部分，然后将 W 除以分母可以得到最终的用户兴趣相似度。

图 **4.7** 会议**-**用户倒查表

**Fig.4.7 Meet - user inversion list**

1. 找到这个集合中的用户收藏过的，且目标用户没有收藏的会议推荐给目标用户。 如下公式计算用户 u 对会议 i 的感兴趣程度：

𝑝 𝑢, 𝑖 =  𝑊!"𝑟!"

!∈!(!,!)∩!(!)

其中， p( u , i ) 表示用户 u 对会议 i 的感兴趣程度，S( u , K )表示和用户 u 最为相似的 k 个用户，N( i )表示对会议 i 有过收藏行为的用户集合，Wuv 表示用户 u 和用户 v 的兴趣相似度，rvi 表示用户 v 对会议 i 的感兴趣程度（这里默认是用户单一行为的隐反馈数据，故所有 rvi =1）[26]。

继续图 4.6 中的例子，根据 UserCF 算法可以算出，用户 A 对会议 c、e 的兴趣为：

p A, c = 𝑊!" + 𝑊!" = 0.7416

p A, e = 𝑊!" + 𝑊!" = 0.7416

1. 本系统的会议的推荐功能由表 3.6 中的 Servlet 类 controller.Recommend 的 UserCF 方法实现，详细代码如下：

public static List<Grade> UserCF(String userid,Session hsession){

String recommendUser = userid;

Transaction ts = hsession.beginTransaction();

Query hq = hsession.createQuery("from UserEntity "); List<UserEntity> all = hq.list();

int N = all.size();

List<Grade> m = new ArrayList<Grade>();

int[][] sparseMatrix = new int[N][N];//建立用户稀疏矩阵，用于用户相似度计算【相似度矩阵】Map<String, Integer> userItemLength = new HashMap<>();//存储每一个用户对应的不同会议总数Map<String, Set<String>> itemUserCollection = new HashMap<>();//建立会议到用户的倒排表Set<String> items = new HashSet<>();//辅助存储会议集合

Map<String, Integer> userID = new HashMap<>();//辅助存储每一个用户的用户 ID 映射Map<Integer, String> idUser = new HashMap<>();//辅助存储每一个 ID 对应的用户映射for(int i = 0; i < N ; i++) {//依次处理 N 个用户

Query hq2 = hsession.createQuery("from CollectEntity c where c.userid=?"); hq2.setString(0,all.get(i).getUserid());

List<CollectEntity> c = hq2.list();

String[] user\_item = new String[c.size()+1]; user\_item[0] = all.get(i).getUserid();

for(int j = 0; j< c.size();j++){user\_item[j+1] = Integer.toString(c.get(j).getMeetid());} int length = user\_item.length;

userItemLength.put(user\_item[0], length-1);

userID.put(user\_item[0], i);//用户 ID 与稀疏矩阵建立对应关系

idUser.put(i, user\_item[0]);

//建立物品--用户倒排表

for(int j = 1; j < length; j ++){ //如果已经包含对应的会议--用户映射，直接添加对应的用户if(items.contains(user\_item[j])){

itemUserCollection.get(user\_item[j]).add(user\_item[0]);

}else{//否则创建对应会议--用户集合映射items.add(user\_item[j]);

//创建会议--用户倒排关系

itemUserCollection.put(user\_item[j], new HashSet<String>()); itemUserCollection.get(user\_item[j]).add(user\_item[0]); }

}

}

//计算相似度矩阵

Set<Map.Entry<String, Set<String>>> entrySet = itemUserCollection.entrySet(); Iterator<Map.Entry<String, Set<String>>> iterator = entrySet.iterator(); while(iterator.hasNext()){

Set<String> commonUsers = iterator.next().getValue(); for (String user\_u : commonUsers) {

for (String user\_v : commonUsers) { if(user\_u.equals(user\_v)){ continue; }

//计算用户 u 与用户 v 都有正反馈的会议总数

sparseMatrix[userID.get(user\_u)][userID.get(user\_v)] += 1; }

}

}

//余弦相似性计算用户之间的相似度

int recommendUserId = userID.get(recommendUser); for (int j = 0;j < sparseMatrix.length; j++) {

if(j != recommendUserId){

double ss = sparseMatrix[recommendUserId][j]/Math.sqrt( userItemLength.get(idUser.get(recommendUserId))\*userItemLength.get(idUser.get(j))); }

}

//计算指定用户 recommendUser 的会议推荐度

for(String item: items){//遍历每一个会议

Set<String> users = itemUserCollection.get(item); //收藏当前会议的所有用户集合if(!users.contains(recommendUser)){ //如果目标用户没有收藏当前会议

double itemRecommendDegree = 0.0; //进行推荐度计算for(String user: users){

itemRecommendDegree += sparseMatrix[userID.get(recommendUser)][userID

.get(user)]/Math.sqrt(userItemLength.get(recommendUser)\*userItemLength.get(user)); } if(itemRecommendDegree!=0.0){

Grade t = new Grade(Integer.parseInt(item),itemRecommendDegree); m.add(t); }

}

}

return m; }

### 会议更新推送模块

运用 python 爬虫，每天定时运行，从各个会议官方网站爬取会议新闻信息以 txt 文档格式存储在本地，当用户登录系统时，再在本地读取会议新闻 txt 文档，降低对网络的依赖，减少系统运行时所耗费时间。数据库内的收藏会议新闻与本地新闻进行比较和判断，若本地新闻数量大于数据库内 新闻数量，则会议新闻有更新，将会议新闻的新信息插入数据库新闻表中，系统首页会弹出提示框“您 收藏的会议有更新”，用户进入“我的收藏”页面，有更新的会议旁有红字提示更新；若会议新闻无更 新，无提示。具体活动图如图 4.8 所示。



用户

学术会议推送系统

登录

爬虫实时爬取会议网页新闻动态并保存在本地

获取用户收藏会议信息

在数据库新闻表中查找收

藏会议的新闻信息

在数据库新闻表中查找

收藏会议的新闻信息

[是]

本地新闻数量>数据库内新闻数量

[否]

收到会议更新通知

会议有更新，插入数据库

图 **4.8** 会议新闻更新活动图

**Fig.4.8 Meeting news update activity diagram**

* + 1. 网络爬虫技术模块

会议新闻网站的实时信息爬取由 python 的网络爬虫 News.py 实现，会议新闻文件以 txt 文本形式保存在本地。本系统设计的爬虫主要由三个模块组成，包括网页抓取模块、网页分析模块和数据 存储模块，模块之间相互协作，共同完成网页数据的抓取[27]。

（1）网页抓取模块

打开某学术会议官网，该网站 URL 是爬虫主要的处理对象，对服务器发出请求，打开会议网页，并分析网页源代码结构，会议网页源码如图 4.9 所示。

图 **4.9** 会议网页源码图

**Fig.4.9 Meeting website source code diagram**

核心代码及注释如下:

import requests import re

def getsource(self, url):

#获取网页 html 文本

html = requests.get(url) return html.text

def getpage(self, source):

#使用正则表达式，提取关键内容

page = re.findall('<a class="external-link" href="http://(.\*?)">Conference Website</a>', source,

re.S)

#循环获取多个子网页 url for i in range(0, len(page)):

page[i]='http://'+page[i] return page

（2）网页分析模块

确定爬虫方法，本爬虫程序使用 BeautifulSoup 的 find 方法进行网页分析抓取指定标签的信息， 因此需要导入 BeautifulSoup 库，部分代码及注释如下:

#导入程序中所用到的库from bs4 import BeautifulSoup

#通过 BeautifulSoup 的 find 方法解析页面，查找会议名称、地点、组织者、日程安排等信息

soup = BeautifulSoup(html, 'html.parser')

name = soup.find('h1', attrs={'align': True}).string address= soup.find('h1', attrs={'title': 'Address'}).string

organization = soup.find('a', attrs={'title': 'Conference Officers'}).get('href') news = soup.find('a', attrs={'title': 'News and Site Updates'}).get('href')

……

（3）数据存储模块

该爬虫对会议信息的爬取结果，保存到本地，存储格式为.txt 文件。部分代码如下: def save(self, classinfo):

# 打开要存储的文件

f = open('meet.txt', 'a')

# 将获取的会议信息写入文件中

for each in classinfo: f.writelines(each['name'] + '||') f.writelines(each['information'] + '||') f.writelines(each[address] + '||') f.writelines(each[website] + '||') f.writelines(each[starttime] + '||') f.writelines(each[endtime] + '||')

……

f.writelines(each['news'] + '@@@') f.writelines('\n')

f.close()

* + 1. 收藏会议新闻更新推送

收藏会议新闻更新推送的顺序图如图 4.10 所示：



用户

登录（账号，密码）

登录验证成功

连接数据库查询收藏会议新闻和本地会议新闻对比，判

断是否有会议更新

传递更新的会议id集合

session.setAttribute(newmid)

弹出提示框，提醒用户浏览会

议收藏页面，查看会议更新

更新的会议旁有红字提示更新

会议收藏界面collect.jsp

会议界面all.jsp

登录控制器controller.Login

系统首页index.jsp

图 **4.10** 会议新闻更新顺序图

**Fig.4.10 Meeting news update sequence diagram**

会议的更新功能由表 3.6 中的 Servlet 类 controller.Login 的 doPost 方法实现，在数据库 collet 表和 meet 表中查询到用收藏的会议记录 List collect，根据会议名称查找会议本地新闻文件，找到后存储会议 id 和会议新闻文件本地路径为 List filepath，根据路径读取该文件内信息并存储到 List news 中，如果本地该会议新闻数量大于数据库内现存的该会议新闻数量，即会议有更新，将会议新闻插 入数据库。查询数据库用户表中该用户的 pre 属性，即该用户上一次登录时会议新闻表中最后一条记录的 id，再查询当前更新后的会议新闻表中所有 id 大于 pre 的新闻记录，如果存在该用户收藏过的会议，则将该会议 id 存储到 List newmid 中，最后通过 session.setAttribute("newmid", newmid)传递有更新的收藏会议 id 集合到客户端界面 all.jsp 和 collect.jsp。核心代码如下：

int pre = uu.getPre();//用户上次登录时，会议新闻表最后一条记录 id

Query h = hsession.createQuery("select m from MeetEntity m,CollectEntity c where c.userid=? and c.meetid=m.meetid");

h.setString(0,id);

List<MeetEntity> collect = h.list();//用户收藏的会议

String path = "/Users/Rose/PycharmProjects/untitled4";

List<String> filepath = getFile(path,0);//读取文件夹下所有会议本地新闻文件for(int i = 0; i<filepath.size();i++){

if(filepath.get(i).contains("txt")){

String[] na = filepath.get(i).split("/");

Query qqh = hsession.createQuery("select ne from MeetEntity m,NewsEntity ne where m.name like '%" + na[0] + "%' and m.name like '%" + na[1] + "%' and m.meetid=ne.meetid");

List<NewsEntity> nList = qqh.list();//数据库内现存的该会议新闻数量

List <String>news=readTxtFile(filepath.get(i));//读取会议新闻文件

int m = news.size()-1-nList.size();//本地该会议新闻数量-数据库当前该会议新闻数量Query hq = hsession.createQuery("from NewsEntity");

List <NewsEntity> all = hq.list();

int q = all.get(all.size()-1).getId();//数据库新闻表末尾 id int no = q+m;//数据库新闻表将要插入的会议新闻 id if(m>0){//会议新闻有更新

for(int j = 1; j<=m;j++) {

NewsEntity n = new NewsEntity(); n.setId(no--); n.setMeetid(nList.get(0).getMeetid()); n.setContext(news.get(j));

hsession.merge(n);//插入数据库 }

}

}

}

//判断更新的会议新闻表中是否有用户收藏的会议Query qq = hsession.createQuery("from NewsEntity n"); List<NewsEntity> nList = qq.list();//用户收藏的会议

Query qq2 = hsession.createQuery("FROM NewsEntity n WHERE n.id>"+pre);

List<NewsEntity> newList = qq2.list();//会议新闻表中，大于用户上次登录时最后一条记录的所有新闻记录

List <Integer> newmid = new ArrayList<Integer>();//用户收藏会议，更新会议 id for (int i = 0;i<newList.size();i++){

for (int j = 0; j<collect.size();j++){ if(newList.get(i).getMeetid()==collect.get(j).getMeetid()){

newmid.add(collect.get(j).getMeetid()); }

}

}

int now = nList.get(nList.size()-1).getId();//当前会议新闻表最后一条记录 id uu.setPre(now);

hsession.merge(uu);//更新用户表session.setAttribute("newmid", newmid);//传递会议更新信息

**5** 系统使用和运行

本章介绍了系统使用说明和系统运行结果，展示了大量系统界面图。

### 系统使用说明

系统使用说明介绍了系统的运行环境和系统安装。

* + 1. 运行环境

1. Web服务器：apache tomcat 6.0/7.0

Tomcat 是一个轻量级应用服务器，是 Apache 软件基金会（Apache Software Foundation）的 Jakarta项目中的一个核心项目，由 Apache、Sun 和其他一些公司及个人共同开发而成。Tomcat 性能稳定、技术先进，运行时占用的系统资源小，而且免费并在不断改进完善中，深受部分软件开发商和广大 程序员的喜爱，成为目前比较流行的 Web 应用服务器。

1. 数据库服务器：MySQL

MySQL 是一种关系型数据库管理系统，关联数据库在不同的表中存储数据，而不是放在一个大仓库中，可提高数据操作的速度和灵活性。MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。由于其体积小、速度快、总体成本低，尤其是开放源码这一特点，一般中小型网站的 开发都选择 MySQL 作为网站数据库。

* + 1. 系统安装

服务器端，将数据库部署到 MySQL 数据库服务器上。用 IntelliJ idea 软件导入 meetingsys.war 项目，运行。

### 系统运行结果

在系统的运行过程中，通过对系统的整体页面布局和按钮、图片的位置大小进行了细微调整， 使系统界面更加友好、美观；在系统退出和界面跳转间添加了提示信息，使系统的提示功能得到进 一步完善，增强了系统的用户导向性。系统运行结果表明：本系统操作流畅，页面设计简洁美观， 体系结构完整，功能模块合理。系统的部分界面截图如图 5.1 至图 5.13 所示。

系统主页面，可以进行简单的会议搜索，如会议名称搜索和会议刊物类别搜索，以及进行注册、 登录操作，如图 5.1 所示。

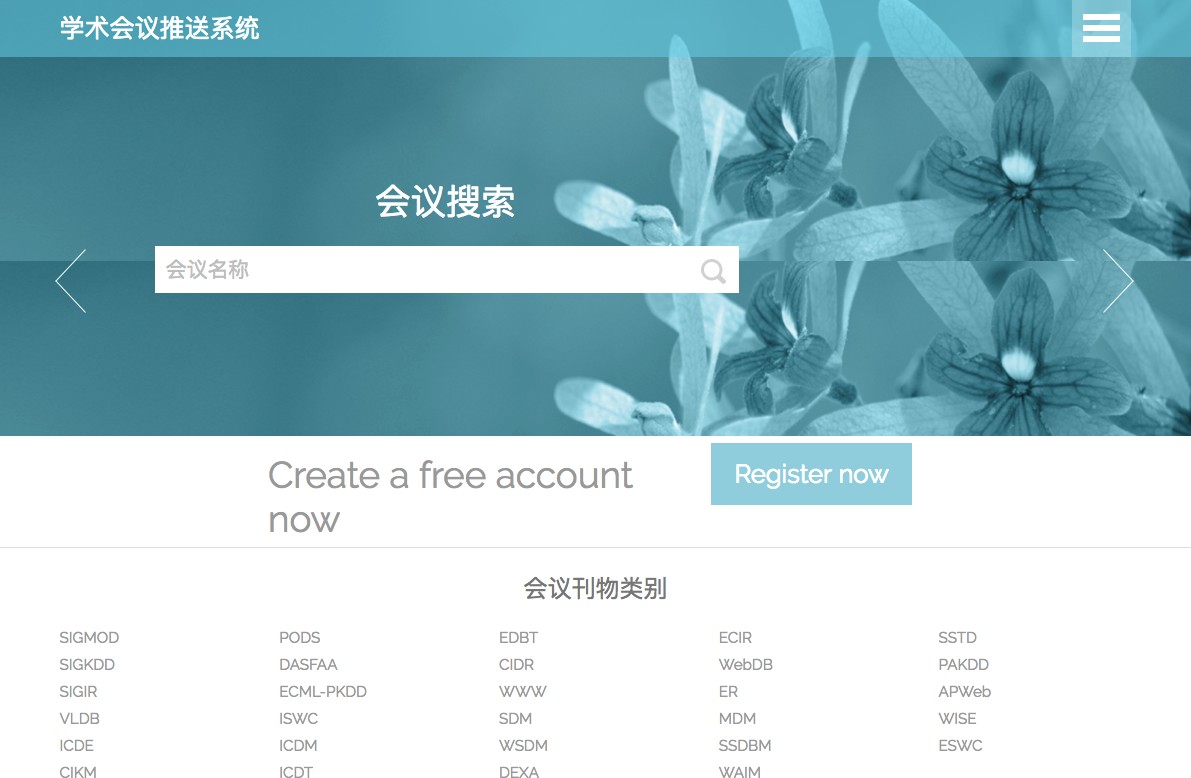


图 **5.1** 系统主界面

**Fig.5.1 Main page**

用户注册时，必填项为 email、名称、密码，可以选填国家、兴趣领域、兴趣会议，如图 5.2 所

示。

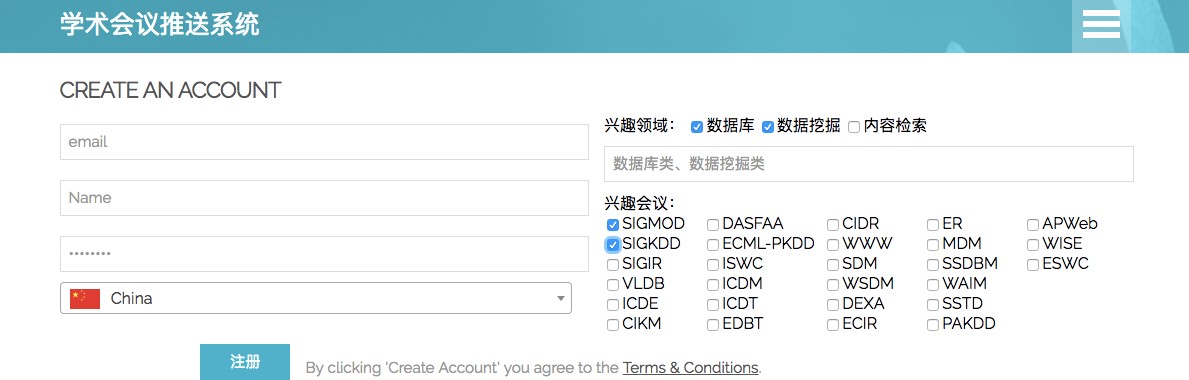


图 **5.2** 用户注册界面

**Fig.5.2 User register page**

图 5.3 为所有会议浏览页面，也是用户登录后的首页面，显示了系统时间，收录会议总数，统计了未召开会议总数，可以进行简单的搜索，包括会议名称搜索、会议领域搜索和会议刊物搜索。 会议列表展示了会议的名称、类别、开始时间、结束时间、地点，点击查看按钮可查看具体会议详 情，点击收藏按钮可以收藏该会议，已经收藏过了的会议显示为已收藏。点击界面底端的有分页浏

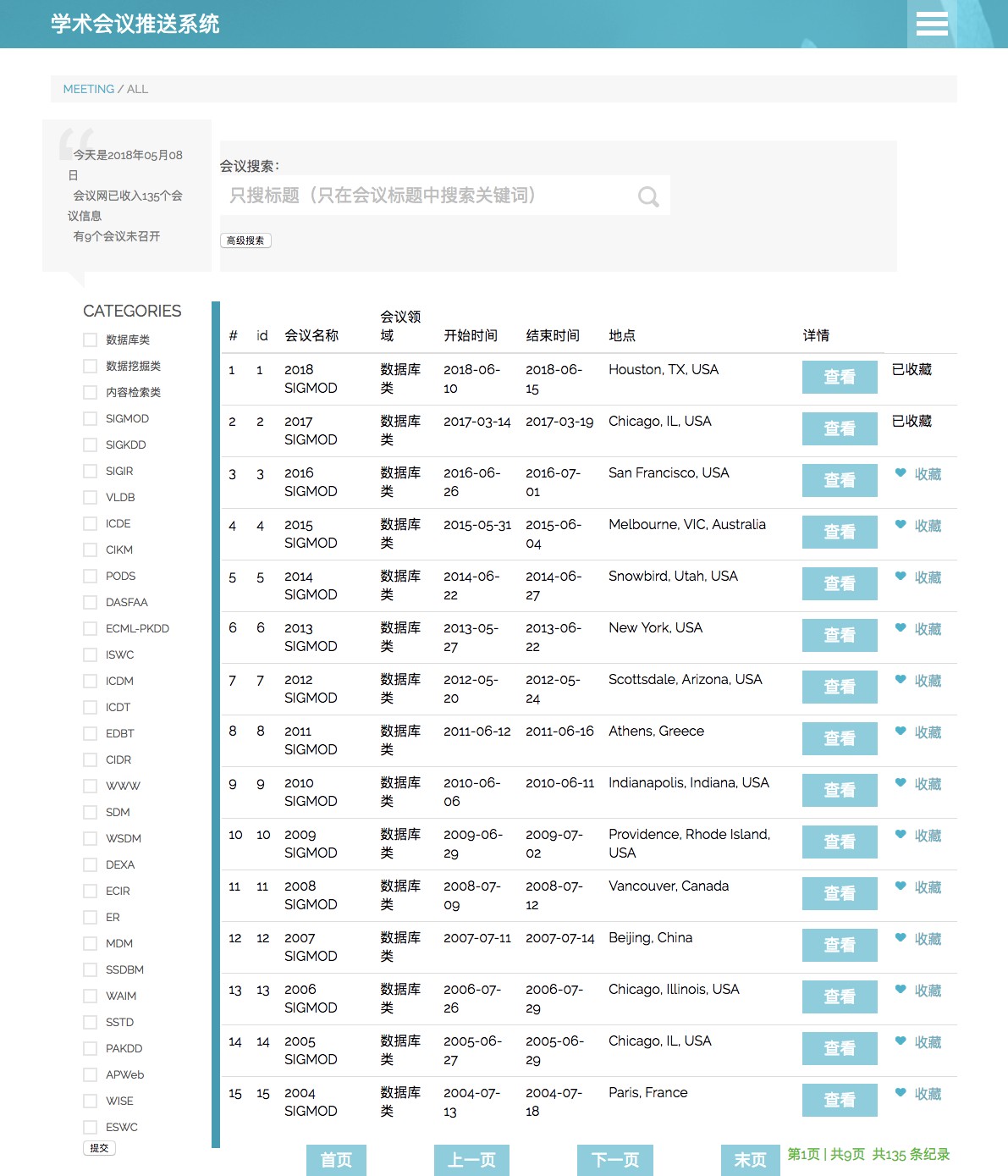
览功能，每页有 15 条会议记录。

图 **5.3** 会议浏览界面

**Fig.5.3 User register page**

图 5.4 为系统的导航栏，用户登陆方可后显示，分别是“我的收藏”、“推荐会议”、“所有会议”、

“会议搜索”、“个人信息”和“退出”。

图 **5.4** 导航栏界面

**Fig.5.4 Navigation bar page**

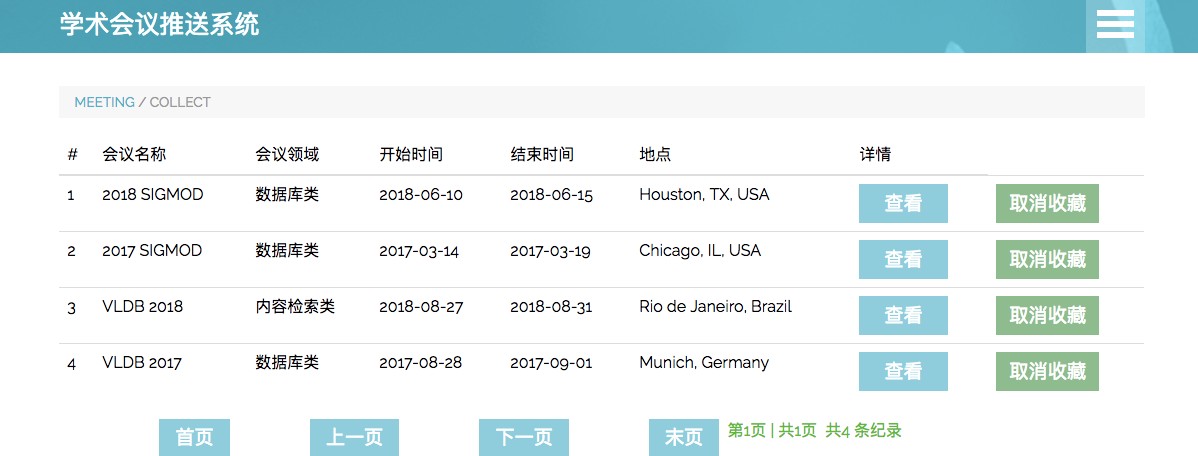
图 5.5 为收藏页面，用户登陆方可后显示，展示了所有该用户收藏的会议，会议列表展示了会议的名称、类别、开始时间、结束时间、地点，点击查看按钮可查看具体会议详情，点击取消收藏 按钮该会议会从会议列表中移除。

图 **5.5** 收藏界面

**Fig.5.5 Collect page**

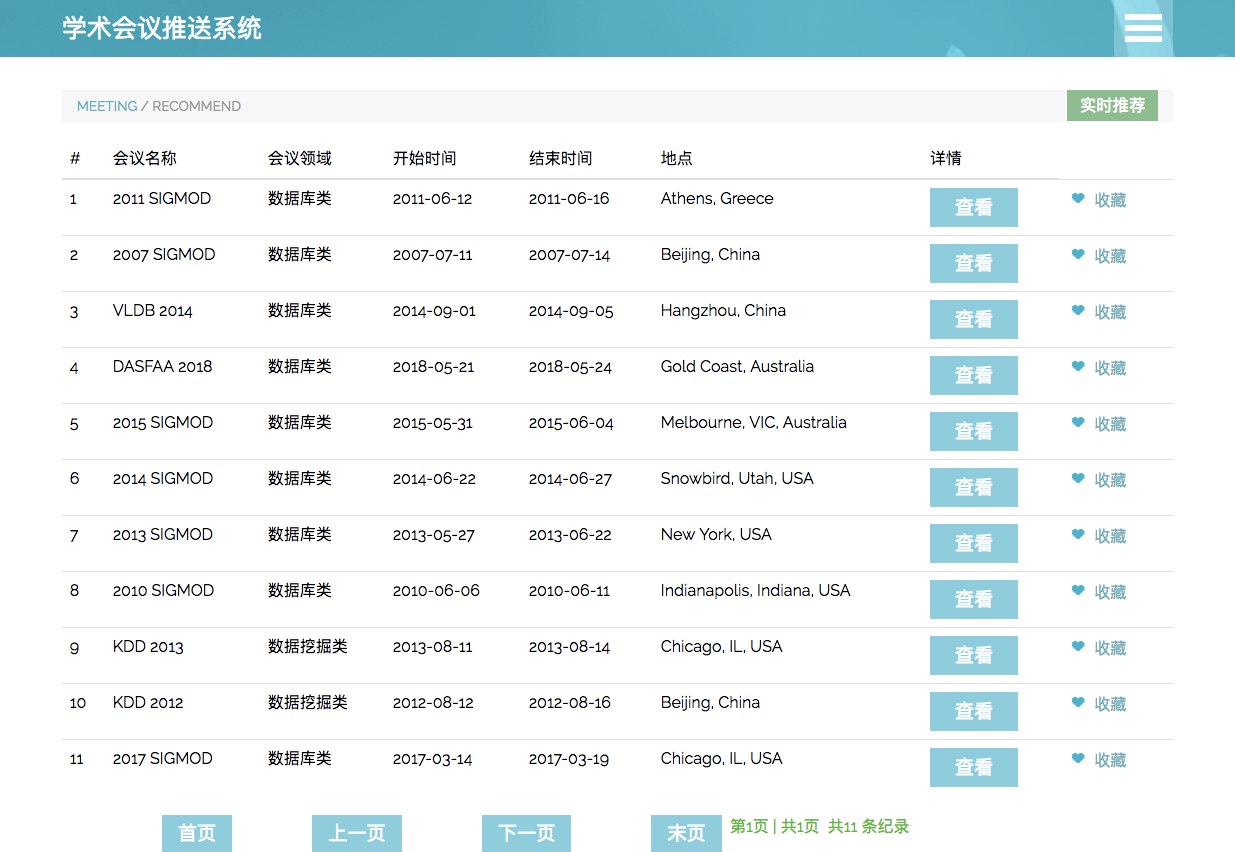
图 5.6 为推荐页面，用户登陆方可后显示，展示了所有系统推荐且该用户未收藏的会议，会议列表展示了会议的名称、类别、开始时间、结束时间、地点，点击右上角实时推荐按钮，可根据用 户实时信息进行会议推荐，点击查看按钮可查看具体会议详情，点击收藏按钮可以收藏该会议。

图 **5.6** 推荐会议界面

**Fig.5.6 Recommend meeting page**

图 5.7 为会议搜索页面，用户登陆方可后显示，属于会议高级搜索功能，包括会议关键字、会议领域搜索、是否征稿、开始日期、结束日期、会议地点搜索。

图 **5.7** 会议搜索界面

**Fig.5.7 Meeting search page**



图 **5.8** 用户信息界面

**Fig.5.8 User information page**



图 **5.9** 用户信息修改界面

**Fig.5.9 User information change page**

如果用户对某个会议感兴趣，想要查看该会议详细信息，在会议列表中点击“查看”按钮，系统 会跳转到该会议的详细介绍界面。图 5.10 为会议信息页面，该界面把会议的所有信息进行展示，包括：会议全称、会议信息、会议领域、刊物简称、官网、地点主办单位、征文信息、日程安排、论

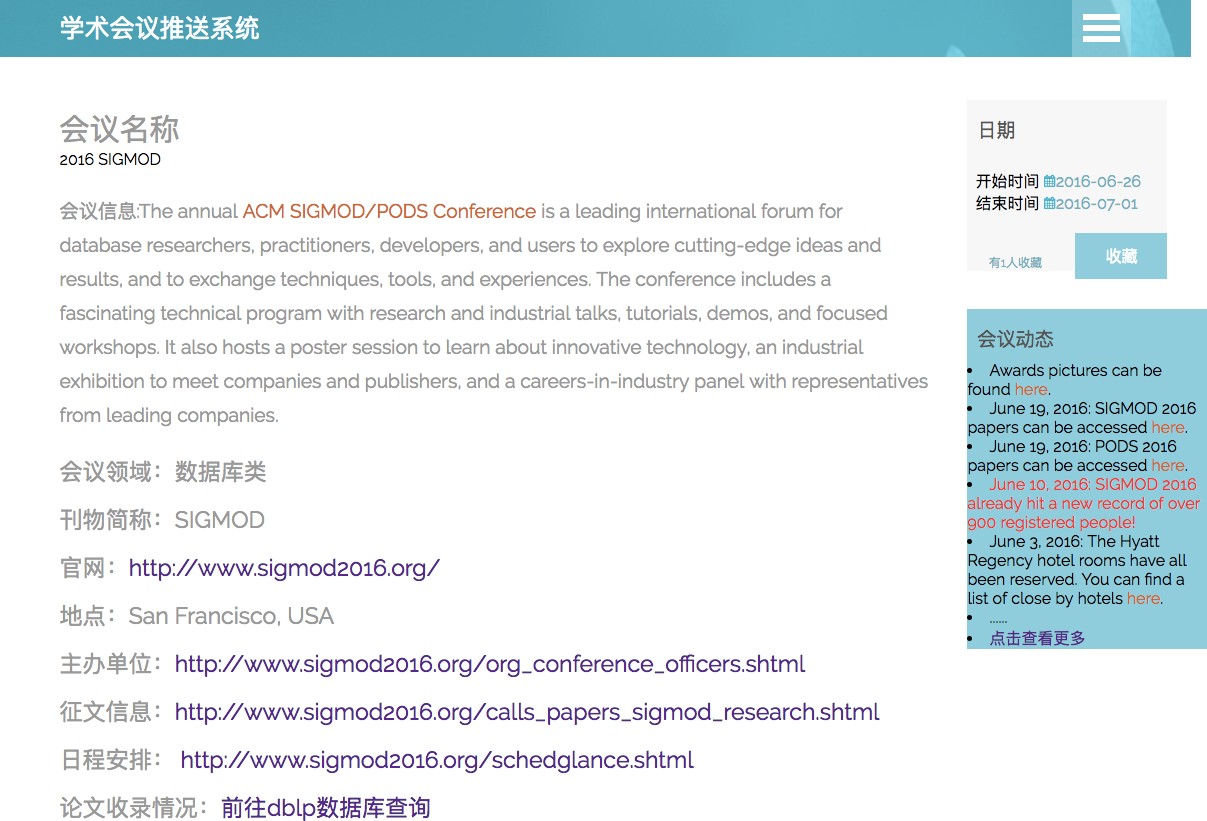
文收录情况、日期、该会议的收藏人数、会议动态信息。点击收藏按钮可以收藏该会议。

图 **5.10** 会议信息界面

**Fig.5.10 Meeting information page**

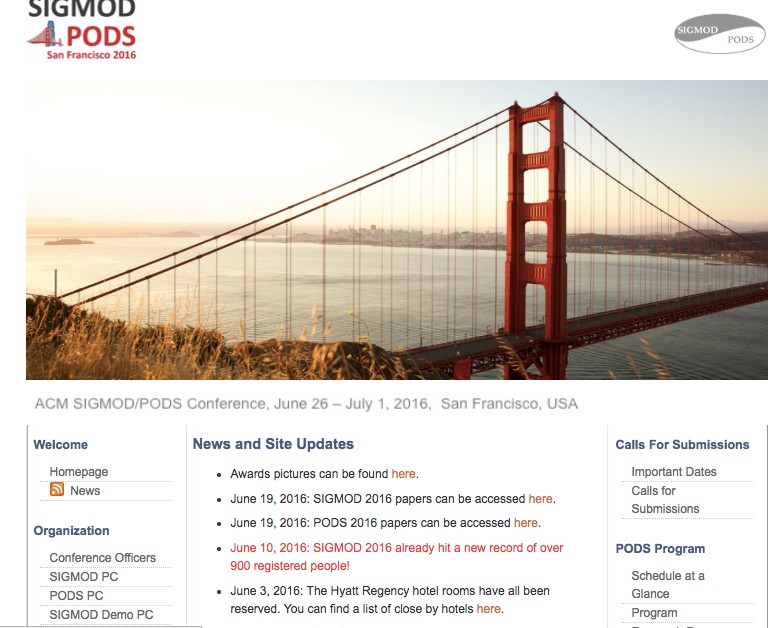
点击会议动态栏目的最末端的点击查看更多，可以跳转到该会议的新闻网页，如图 5.11 所示。

图 **5.11** 会议新闻网页界面

**Fig.5.11 Meeting news web page**

点击论文收录情况，跳转到 dblp 数据库，查询该会议名称，查看论文集，如图 5.12 所示。

图 **5.12** 会议论文界面

**Fig.5.12 Meeting paper page**

当用户收藏的会议有更新时，用户登录系统时会弹出提示框提示更新，如图 5.13 所示。

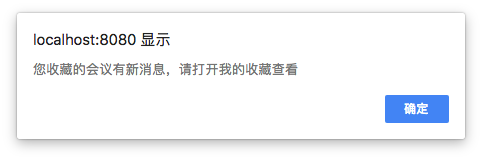


图 **5.13** 会议更新提示界面

**Fig.5.13 Meeting news inform page**

用户登录后，进入我的收藏页面，会议列表后有红色字体提示会议有更新，点击查看按钮，进 入会议页面，会议动态一栏有更新，最新通知在最顶端，如图 5.14 所示。

图 **5.14** 会议更新界面

**Fig.5.14 Meeting news page**

**6** 总结与展望

本章是根据本文研究内容做出的总结和对未来研究工作的展望。

### 总结

本文通过收集分析学术会议系统平台的需求和调查国内外文献，总结和归纳各类学术会议系统 平台设计特点和所使用的关键技术，进行了细致的系统需求分析、概要设计以及详细设计。通过 Java 开发语言， 设计并实现了一个 B/S 模式学术会议推送系统。使用 Tomcat 应用服务器，采用 Servlet 的 MVC 三层架构和 JSP 动态网页开发技术，实现了视图层、逻辑层、数据层的分离，使得系统具有很好的扩展性。服务器端使用 MySQL 数据库存储和管理会议信息、用户信息、收藏信息和会议新闻信息，保证数据安全性。通过 Bootstrap 框架较好地完成了界面设计，精心编写了各类 Servlet 以完成用户的输入和请求。

在针对目前学术会议系统在信息过载方面出现的瓶颈上，本文设计实现的学术会议推送系统引 入了个性化推荐的概念，根据用户的兴趣偏好信息，运用基于用户的协同过滤推荐算法，针对不同 用户进行个性化推荐。用户还能随时进行收藏会议或取消收藏，针对用户收藏的会议，本系统运用 了 python 爬虫技术，实时爬取会议网页的新闻动态信息并及时提醒用户更新消息，在会议后续反馈的跟进方面为用户提供了极大的便利。

本文设计实现的学术会议推送系统不仅能快速推荐符合用户兴趣的学术会议，还能自动更新会 议新闻通知用户，同时提供了方便、快捷的会议查询途径，学术会议推送系统的研究与建设具有较 强的现实意义和实用价值。

### 展望

本系统实现了各项预期目标，但在今后的版本中仍有值得改进的地方：

1. 扩大系统会议数量规模

收集更多会议信息，目前系统只收集了中国计算机学会推荐国际学术会议 (数据库、数据挖掘与内容检索) ，扩充更多类别的会议，以满足更多的用户的需求。

1. 移动端

移动终端越来越受追捧，希望未来学术会议推送系统也能通过移动端进行，方便而且能随时随 地的浏览和收藏会议。

# 致谢

时间匆匆，大学生活即将结束，在这四年里有许多难忘的事和值得感谢的人，在此，我要特别 感谢我的指导老师田萱老师。田老师兢兢业业，学识渊博，非常认真地教授我们知识和指导我们动 手实践开发。于我个人而言，田老师更是像慈母一般，对我的求学道路细心叮咛，不管是日常学习 上的问题，还是我申请出国读研时的困惑，她都给了我很多建议和帮助，让我不断成长。所以在最 后的毕业设计的导师选择时我选择田老师作为我的指导老师，同时也非常幸运的通过了田老师的筛 选。

毕业设计开题后，田老师多次与我讨论系统需求并指导我做需求分析和设计，让我有了清晰的 设计思路，督促和指导我控制在期限的时间内完成开发。这个完整过程让我深刻的体会到一个产品 是一个从需求到设计、根据设计来开发实现、到最后变成一个成品上线的过程。田老师还给我们介 绍前沿的集成开发环境和前端开发技术，带领我们探索和学习，给我带来很大的收获。田老师的为 人、态度、责任心以及她开阔的视野深深的影响我，让我清楚的知道该以什么样的心态、什么样的 态度来面对以后的工作和生活以及为人处世。

此外，我还要感谢大学里授予我知识的每一位任课老师，是你们的热情和付出使我学到了许多 人生重要的技能。还记得上课时听着老师讲述做笔记时的场景，在机房调试代码求助的老师的场景， 期末考试前紧张复习去老师办公室答疑的场景，这些画面是如此温馨，这一切都让我十分怀念。还 要感谢计算机 14-2 班的大家庭，作为其中的一员我感到无比的骄傲与自豪。能够认识每一位同学真的很高兴，我从大家身上学到了很多开发的知识和技能，给了我大学成长进步最肥沃的土壤，感谢 同学们四年来给予我的支持、鼓励和关怀。我还要感谢我亲爱的家人和朋友们，谢谢他们平时的鼓 励和陪伴，让我感到非常地开心快乐和收获了满满的正能量。

最后，我要感谢我的母校北京林业大学，让我拥有了数不清精彩回忆的四年，跌宕起伏迅速成 长的四年，时而惆怅时而惶恐的四年，收获了知识和朋友的四年。一身轻松的来，却要即将沉甸甸 的离开，无论是母校给予我的东西还是我自己的心情。希望我的母校越办越好，哺育越来越多的学 子，也希望越来越多的学子在母校的光芒下茁壮成长，成为国家栋梁之才，为母校增光添彩。

# 参考文献

1. Allconferences.[EB/OL[].http://www.allconferences.com,](http://www.allconferences.com/) 2018-05-22.
2. ourglocal.[EB/OL[].http://www.ourglocal.com,](http://www.ourglocal.com/) 2018-05-22.
3. cnki 中国学术会议网.[EB/OL[].http://c](http://conf.cnki.net/)o[nf.cnki.net,](http://conf.cnki.net/) 2018-05-22.
4. 学术会议搜索.[EB/OL[].http://w](http://www.searchconf.net/)w[w.searchconf.net,](http://www.searchconf.net/) 2018-05-22.
5. 章春梅. 网络教育环境下个性化信息定制服务系统的设计与开发[D].南京师范大学,2006.
6. 韩燕丽,杨慧炯.基于 Web 的毕业设计过程管理平台的研究与设计[J].软件工程,2016,19(10):40-43.
7. 顾苗.关联规则挖掘算法及应用研究[D].南京邮电大学,2015.
8. 杨玉, 李建军. 网络信息系统 Client/Server 到 Browser/Server[J]. 哈尔滨商业大学学报( 自然科学版),2002, 18(5):515-518.
9. 樊胜. C/S 与 B/S 的结构比较及 Web 数据库的访问方式[J].情报科学,2001(04):443-445.
10. 李欣.基于云平台的智能家居管理系统设计与实现[D].电子科技大学,2015.
11. 杨涛,王建桥,杨晓云.深入浅出 STRUTS2[M].北京:人民邮电出版社,2009.
12. 严春燕,戴仕明. 基于框架的 web 前端(Bootstrap 和 MUI)之比较[J]. 网络安全技术与应用, 2017(4):83-84.
13. 卢凯霞,程耕国.Struts+Spring+Hibernate 在 OA 系统开发中的应用[J].信息技术,2011,35(07):145-148.
14. 牛敏,米石云,张倩.网络数据资源自动获取技术研究与应用[J].信息技术,2013(12):23-26.
15. Robert Armstrong, Dayne Freitag, Thorsten Joachims, Tom Mitchell. Web Watcher: A Learing Appretice fir the World Wide Web[C].
16. Marko Balabanobie,Yoav Shoham.Learing Infirmation Retrieval Agents: Experiments with Automated Web Browsing[C].1995:13-18.
17. Henry Lieberman.Letizia:An Agent That Assists Web Browing[J].1995,95(3):97-102.
18. Daniel Dreilinger, Adele E. Hoew. Experiences with selecting search engine using metasearch[J].ACM Transactions on Infirmation Systems,1997,15(3):195-222.
19. Eui-Hong Han,Daniel Bolcy.et al.WebACE.1998:408-415.
20. Liren Chen, Katia Sycara.WebMate: A Personal Agent for Browing and Searching[C].New York:ACM Press,1998:132-139.
21. 袁伟. 科技论文社区网络结构划分及论文推荐算法的研究[D].燕山大学,2014.
22. 陈伟, 师秦龙. 一种改进协同过滤推荐算法[J]. 咸阳师范学院学报, 2014,29(6):47-49.
23. TBN\_MSnail.推荐算法浅析. [EB/OL] . https://blog.csdn.net/lza1989/article/details/40656199, 2018-05-22.
24. 百 度 百 科 . 推 荐 系 统 . [EB/OL]. https://baike.baidu.com/item/%E6%8E%A8%E8%8D%90%E7%B3%BB%E7%BB%9F/10267357,2018-05-22.
25. 何军, 刘业政. 基于社交关系和影响力的在线社交网络用户兴趣偏好获取方法研究[J]. 情报学报,2014, 33(7):730-739.

[26] 苏林宇,陈学斌.基于用户的协同过滤算法的改进研究[J].软件,2017,38(04):127-132.

[27] 郭丽蓉.基于 Python 的网络爬虫程序设计[J].电子技术与软件工程, 2017(23):248-249.