

郑州轻工业学院

本科毕业设计（论文）

题    目 基于 RDF 的 OS 课程知识库系统开发

学生姓名 谢 先 斌

专业班级 计算机科学与技术 10-1 班

学    号 541007010144

院（系） 计算机与通信工程学院

指导教师(职称) 王岩（讲师）

完成时间 2014 年 5 月 25 日

郑州轻工业学院

## 毕业设计（论文）任务书

**题目** 基于 RDF 的 OS 课程知识库系统开发

**专业** 计算机科学与技术 **学号** 541007010144 **姓名** 谢先斌

### 主要内容、基本要求、主要参考资料等：

**主要内容：**系统是以 RDF 和 SPARQL 为核心技术所做的课程知识库系统，系统采用 SSH（Struts、Framework、Hibernate）开发，并且集成 CAS 中央认证系统。用户可以通过查询本地 RDF 文件或调用 DBpedia 接口两种途径实现对所需词条快速、准确的检索，管理人员可以对所有词条管理。

**基本要求：**1. 测评系统各基本功能都能实现。

2. 设计严谨、功能完备。

3. 界面美观大方, 使用方便。

### 主要参考资料：

[1] 吴鸿汉, 瞿裕忠. 基于 RDF 句子的语义网文档搜索[J]. 东南大学计算机科学与工程学院

[2] 章志龙. 基于语义网的博客搜索系统研究[J]. 武汉理工大学

[3] 黎明. 基于语义网的信息检索技术的研究[J]. 著南京理工大学

**完 成 期 限：** 2014 年 5 月 1 日

**指导教师签名：** \_\_\_\_\_

**专业负责人签名：** \_\_\_\_\_

年      月      日

# 目录

摘 要.....	I
ABSTRACT.....	II
1. 绪论.....	1
1.1 项目背景.....	1
1.2 项目意义.....	1
1.3 现状和发展趋势.....	2
1.4 工作难度.....	2
1.5 实现目标.....	2
2. 相关技术.....	4
2.1 语义网.....	4
2.2 RDF 简介.....	4
2.3 SparQL 技术.....	5
2.4 B/S 结构.....	5
2.5 SSH 框架技术.....	6
2.6 CAS 单点登录技术.....	7
2.7 DES 算法加密技术.....	7
2.8 WEB 服务器.....	8
2.9 MySQL.....	9
3. 系统分析.....	10
3.1 可行性分析.....	10
3.1.1 技术可行性.....	10
3.1.2 经济可行性.....	10
3.1.3 操作可行性.....	10
3.2 需求分析.....	11
3.2.1 功能需求分析.....	11
3.2.2 系统性能需求分析.....	11
3.2.3 系统运行需求和开发工具.....	12
3.3 业务流程分析.....	12

3.3.1 前台业务流程分析.....	12
3.3.2 后台业务流程分析.....	13
3.4 数据流程分析.....	14
4. 系统总体结构设计 .....	16
5. 系统数据存储设计 .....	18
5.1 MySQL 数据字典 .....	18
5.1.1 数据字典.....	18
5.1.2 数据库设计.....	21
5.2 RDF 文件设计.....	24
5.2.1 RDF 文件结构设计 .....	24
5.2.2 RDF 文件操作 .....	25
6. 系统详细设计和实现 .....	31
6.1 功能模块设计.....	31
6.1.1 前台的功能模块.....	31
6.1.2 后台的功能模块.....	32
6.1.3 认证中心的功能.....	33
6.2 系统实现设计.....	33
6.2.1 前台的功能实现.....	33
6.2.2 后台和 CAS 功能实现.....	35
7. 系统测试.....	39
结束语.....	40
致谢.....	41
参考文献.....	42

## 基于 RDF 的 OS 课程知识库系统开发

### 摘 要

课程知识库系统使用 Java SSH<sup>[1]</sup>框架开发，以 RDF<sup>[2]</sup>（资源描述框架）和 SparQL<sup>[3]</sup>为核心，实现对知识词条快速准确的查询。

随着互联网快速的发展，网络上海量的信息资源充斥着人们的视野，这给人们准确、快速搜索所需的信息带来了极大的不便。课程知识库系统使用先进的 RDF 存储技术和 SparQL 检索技术，能够快速准确的查询到所需的知识词条。

课程知识库系统采用 B/S 结构，有三个项目组成，均采用 Java SSH 框架技术开发。前台项目以词条检索和用户注册为开发核心，其中词条检索包括本地 RDF 文件词条检索和调用 Dbpedia 接口<sup>[4]</sup>检索公网上的 RDF 文件；后台项目以词条管理、用户管理和统计系统检索信息为开发核心；中央认证系统项目以耶鲁大学开源项目 CAS<sup>[5]</sup>为基础做二次开发，并且集成 DES 加密算法<sup>[6]</sup>作为系统重要信息存储的安全保障。本文结合基于 RDF 的 OS 课程知识库系统的开发，介绍了系统的开发背景和意义、相关技术、系统分析、系统设计以及系统测试等，并对系统各个模块的功能和实现方案做了详细介绍。

课程知识库系统在开发阶段后期做了多次测试和修改，最终各项指标均能达到系统开发的目的和意义。

**关键词** RDF; SparQL; CAS; SSH;

## **The RDF-based development of OS curriculum knowledge**

### **ABSTRACT**

The development of OS curriculum knowledge system used Java SSH framework, which base on RDF (Resource Description Framework) and SparQL as the core, to realize fast and exact query terms for knowledge.

With the rapid development of Internet, network information resources is filled with people's field of vision, the accurate and quick search the information for people bring the great inconvenience. The OS curriculum knowledge system users advanced RDF storage technology and SparQL retrieval technology, which can fast and exact query to the required knowledge entry.

The OS curriculum knowledge system adopts B/S structure, consists of three projects which are adopt Java SSH framework technology development. The development of fronts project at the to Retrieval entry and user registration for the core, the backends project to entry retrieves information management, user management and statistical system for the development of the core, the central system seriously project on the basis of the Yale university open source projects CAS do secondary development, and integration of DES encryption algorithm, the security as an important information storage system. In this paper, based on the research and development of OS course based on RDF knowledge base, this paper introduces the system development background and significance, related technology, system analysis, system design and system test, etc., and a function of each module in the system and implementation scheme in detail.

The OS curriculum knowledge system did the test and modify many times in the late development stage, finally all the indexes can reach the purpose and significance of system development.

**KEY WORDS** RDF; SparQL; CAS; SSH;

## 1. 绪论

### 1.1 项目背景

目前因特网上充斥着海量的信息资源, 这些信息虽然极大的丰富了人们的知识面, 但巨大的信息量给人们搜索所需的资源带来的极大的不便。通常我们因在因特网上获取的信息中包含大量的无用信息, 在实际生产应用中剔除这些信息将花费大量的人力和财力, 不利于生产效率的提高。因此, 研究如何准确、快捷的在因特网中找到自己所需资源, 成为了人们迫在眉睫的问题。

目前因特网在信息表达和检索方面存在严重的缺陷, 例如, 对“UEFI”这个概念进行检索, 传统的搜索引擎仅能搜索到含有 UEFI 关键字的信息, 当并不一定是具体的概念或介绍。虽然这些信息用户可以直接阅读处理, 但计算机不能读取具体的语义信息, 因此限制了计算机智能处理信息的能力, 不适合对信息进行准确的定位<sup>[7]</sup>。而实现信息资源的语义检索, 是提高数字化信息资源的重要突破方向, 也是信息搜索领域面临的重要挑战。

### 1.2 项目意义

2000 年 12 月, 万维网创始人 Tim Berners-Lee 在 XML2000 会议上正式提出了语义网 (Semantic Web)<sup>[8]</sup>, 它是下一代互联网的概念, 目的在于赋予万维网上所有资源一个唯一标识, 并且在资源之间建立一种机器可处理的各类语义联系, 从而使资源之间的关系更加明确, 对资源的搜索更加精确。语义网是对当代万维网的延伸和扩展, 它使用一种明确的、形式化的方式来表示资源信息, 能有效的提高异构系统之间的有效检索和访问。因此, 研究语义网是实现资源精确检索的重要方向。

本论文是项目基于 RDF 的课程知识库系统 (以下简称: 知识库系统), 通过使用 SPARQL 和 RDF (资源描述框架) 技术能实现对课程资源的精确定位, 方便学生、教师等的检索学习, 从而提高学习和生产效率。

### 1.3 现状和发展趋势

国内外关于 RDF 的研究十分火热，基于 RDF 数字检索的研究主要分为以下几种：

DBpedia 是从维基百科中提取结构化信息，并使这些信息在互联网上可共享使用的网站。

基于 RDF 的科技论文搜索引擎的设计与实现，提出一种基于 RDF 的科技论文搜索引擎的设计与实现方案。

HealthCyberMap 是一个医学语义 Web 项目，基于修饰词的都柏林数据集描述 RDF 元数据。它使用 Protégé-2000 来进行数据建模，并使用相关技术连接美国国家医学图书馆的知识资源数据库。

### 1.4 工作难度

课程知识库系统设计与实现采用 B/S 的软件体系架构，共分为 3 个项目：

- 1) 检索前台，提供用户检索使用；
- 2) 系统后台，提供管理员对用户和 RDF 文件系统进行管理使用；
- 3) CAS 中央认证系统，是前后台的认证中心，为安全可靠的认证提供保障。

因此，本系统存在难度如下：

1) 整个系统各个项目之间的协调关系。如前后台之间的统一认证问题，地址跳转问题，用户权限问题等；

2) 系统的安全问题。采用 CAS 的高级定制功能，实现数据库检索匹配用户信息；对于用户的密码存储采用 DES 加密技术，保证即使数据库泄露，也不能直接登录系统；采用邮箱验证功能，为用户使用提供更高保证。

3) RDF 技术和 SPARQL 检索。课程知识点如何存储在 RDF 文件中和如何使用 SPARQL 检索是本系统实现的主要难点和技术要点。

### 1.5 实现目标

本论文通过 RDF 技术、SSH 框架技术、CAS 点单登录等，编写课程知识库系统，该系统通过信息资源的语义检索，实现对具体知识概念的精确化定位。该



系统以大学课程中的名词概念为检索对象，构建语义词典本体。实现对课知识点的快速准确的检索。

## 2. 相关技术

基于 RDF 的课程知识库系统采用 B/S 系统架构，使用 JAVA 的 SSH 框架技术实现。系统采用 RDF 存储信息，采用 SPARQL 检索技术。关于系统安全，采用耶鲁大学开源项目 CAS 中央认证系统的高级定制功能，并添加 DES 加密算法，保证用户信息安全。

### 2.1 语义网

随着网络数据的爆炸式增长，网络的体验方式发生了重大的转变，因网络不能满足日益增长的网络数据处理需求，并且小有处理成本大大增加，语义网应运而生。语义网能建立对信息的有效标识，定制统一的标准，最终实现计算机可以对信息进行高效的自动化处理。

语义网<sup>[10]</sup>的核心是，通过给因特网上的文档（如：HTML）添加能够被计算机理解的语义元数据（Meta data），从而使整个因特网成为一个通用的信息媒介。语义网与传统的互联网有所不同，它更加重视计算机的“理解与处理”能力，并且使计算机具有一定的推理判断能力。

语义网处理的核心是语言，语言是所有知识的载体<sup>[9]</sup>。在具体的信息检索过程中，所有的查询的结果都将使用语言表达。

### 2.2 RDF 简介

资源描述框架（Resource Description Framework,RDF）是一个用来描述资源信息的框架，是 W3C 组织极力推荐使用的用来描述资源及其之间的关系的语言规范，具有简单、易扩展、易交换和开放性等特点，是语义网应用的重要方式。RDF 描述的资源可以是任何形式的，包括文档、人物、物体和其他抽象的概念，在本论文的课程知识库系统中，资源指的就是具体的名词概念。

RDF 使用的目的是方便 WEB 应用程序处理信息，而不是显示给普通个人。RDF 提供了一个通用信息资源的描述框架，所以资源可以在应用之间相互传递而不改变意义。因为 RDF 是一个常用框架，所以程序员可以利用常用的 RDF 解析器和处理工具处理 RDF 文件。由于资源信息可以在不同的应用之间传递，这

就使得资源在非创建者之外使用成为了可能。

在本论文，所有的检索信息都存在于 RDF 文件中，这样可以让计算机自动的处理检索任务，并且达到高效、准确的目的。

## 2.3 SparQL 技术

SparQL，代码“Simple Protocol and RDF Query Language”（SparQL 协议与 RDF 查询语言），是为 RDF 开发的一种查询语言和数据获取协议，它是 W3C 所开发的基于 RDF 数据模型的定义，并且它可以应用于任何使用 RDF 表示的信息资源，被认为是语义网科技的一个关键技术<sup>[11]</sup>。

SparQL 于 2008 年 1 月 15 日正式成为一项 W3C 推荐标准。SparQL 构建在以前的 RDF 查询语言（例如 rdfDB、RDQL 和 SeRQL）之上，拥有一些有价值的新特性。而且，SparQL 将 Web2.0 和 Semantic web 两种新的 web 技术联系起来了，很有可能成为将来的主流网络数据库的查询语言和数据获取标准。

## 2.4 B/S 结构

B/S 结构<sup>[12]</sup>（Browser/Server，浏览器/服务器模式）是随着 WEB 技术兴起后的一种重要的网络结构模式，WEB 浏览器是客户端使用的最主要的应用软件。该模式统一了客户端软件，并将系统功能实现的核心部分集中在服务器上，从而达到简化系统开发、维护和使用的目的。B/S 结构是对 C/S 结构的改进的结构，是互联网上重要的网络结构模式之一。

B/S 结构具有以下特点：

1) 系统开发、维护、升级方式简单

当服务端应用程序需要升级时，只要在服务器上升级应用程序即可，而用户计算机端的浏览器并不需要升级，该结构使得客户机端越来越“瘦”，而服务器端越来越“胖”，是将来信息发展的主流方向。

2) B/S 结构成本降低，客户端选择多样

在 B/S 结构下，用户通过浏览器进行访问，而不需要特定的客户端，并且当今浏览器已成为电脑的标准配置，客户端可以从不同的浏览器访问系统。

3) B/S 结构的结构易于扩展

由于 Web 平台具有无关性,因此 B/S 结构可以任意的扩展,可以从包含一台服务器和几个用户的小型系统快速的扩展成为拥有成千上万个用户的大型系统。

#### 4) 应用服务器运行数据负荷较重

由于 B/S 架构管理软件只安装在服务器端 (Server) 上,因此网络管理员仅需要管理服务器端即可,用户界面主要事务逻辑在服务器端 (Server) 全部通过 WEB 浏览器实现,极少部分事务逻辑在浏览器端 (Browser) 实现,所有的客户端只有浏览器,网络管理人员只需要对服务器端 (Server) 做硬件维护。但是,应用服务器运行数据负荷较重,服务器端存在系统崩溃、数据丢失等危险。

## 2.5 SSH 框架技术

SSH 为 Struts、Spring、Hibernate 的缩写,他们是 Java 的一个集成框架,是目前较流行的一种 Web 应用程序开源框架。本论文知识库系统采用 SSH 框架进行编写,下面介绍关于 SSH 的相关知识。

SSH 框架的系统可以分为四层: 1) 表示层 2) 业务逻辑层 3) 数据持久层 4) 域模块层,使用 SSH 的目的是在短期内搭建结构清晰、可复用性好、维护方便的 Web 网站应用。其中 Struts 扮演系统的整体基础架构的角色,利用 Struts 框架的模型部分控制业务的跳转,利用 Hibernate 框架对持久层提供支持,利用 Spring 来管理 struts 和 hibernate。实现原理是: 应用面向对象的分析方法,根据需求提出一些基础模型,并将上述模型实现为 Java 对象,然后编写 DAO 接口,并给出 Hibernate 的 DAO 实现,采用 Hibernate 架构实现的 DAO 类来实现 Java 类与数据库之间的转换和访问,最后利用 Spring 的管理功能,来管理 struts 和 hibernate。

系统的基本业务流程是: 在表示层中,首先通过 JSP 页面实现交互界面,负责接收请求(Request)和传送响应(Response),然后 Struts 根据配置文件(struts.xml)将 ActionServlet 接收到的 Request 委派给相应的 Action 处理。在业务层中,管理服务组件的 Spring IoC 容器负责向 Action 提供业务模型(Model)组件和该组件的协作对象数据处理(DAO)组件完成业务逻辑,并提供事务处理、缓冲池等容器组件以提升系统性能和保证数据的完整性。而在持久层中,则依赖于 Hibernate 的对象化映射和数据库交互,处理 DAO 组件请求的数据,并返回处理结果。

采用上述开发模型，不仅实现了视图、控制器与模型的彻底分离，而且还实现了业务逻辑层与持久层的分离。这样无论前端如何变化，模型层只需很少的改动，并且数据库的变化也不会对前端有所影响，大大提高了系统的可复用性，大大提高了开发效率。

## 2.6 CAS 单点登录技术

SSO(Single Sign-on)，即单点登录，它是在多个应用系统中，用户只需要在中央认证系统（CAS）中登录一次就可以访问所有的相互信任的应用系统。本论文知识库系统采用企业级架构，分为前后台两个项目，因此采用了 Yale 大学发起的开源项目 CAS（Central Authentication Service，即中央认证服务）作为认证中心，从而保障系统的安全。

CAS 的目的是为 Web 应用系统提供一种可靠的单点登录方法，由于本论文系统在存储用户密码等信息时采用 DES 加密储存，故在采用 CAS 认证时加入了其高级定制功能，即在认证时对用户的密码信息进行解密操作（加密、解密算法将在下节中介绍）。

CAS 的结构包括 CAS Server 和 CAS Client，其中 CAS Server 是独立部署的，主要负责对用户进行认证工作，CAS Client 负责处理对客户端受保护的访问请求，但需要登陆时，重定向到服务器端进行认证，并重定向到原的页面。CAS 的工作原理如下：当用户第一次访问应用后台应用系统（对应 rdf\_backend 项目）的时候，由于未登录，会被重定向到认证系统（对应 CAS 项目）中进行认证登录，cas 根据用户提供的登录信息，认证系统进行身份校验（从数据库中获取信息），如果通过校验，cas 将返回给用户一个认证的 ticket 作为凭证，并把用户的相关信息返回（通过后台项目的过滤器获取后存入 session）。当用户再访问别的应用的时候将携带该 ticket，作为自己认证的凭据，应用系统接受到请求之后会把 ticket 送到 CAS 进行校验，检查 ticket 是否合法。如果通过校验，用户将不再登录就可以访问前台应用系统（对应 rdf\_fronts 项目）。

## 2.7 DES 算法加密技术

数据存储安全一直是人们关系的问题，特别是有关密码等敏感信息，一旦泄

露后果将不堪设想。例如，2014 年 5 月份小米漏洞导致 800 万“米粉”的信息泄露<sup>[13]</sup>，给小米和用户都带来了极大的损失。本论文系统在考虑该问题时对用户的密码信息，采用了自己编写的 DES 算法进行加密存储。下面是关于 DES 算法的介绍。

DES 算法是密码体制中的对称密码体制，是 1972 年由美国 IBM 公司研制的对称密码体制加密算法。DES 算法入口参数有三个 key、data、mode。key 为加密解密使用的密钥，本论文采用自定义密钥；data 为加密解密的数据，即要加密的密码，mode 为其工作模式，标识是加密还是解密。该算法对明文按 64 位进行分组，密钥长 64 位，密钥事实上是 56 位参与 DES 运算（第 8、16、24、32、40、48、56、64 位是校验位，使得每个密钥都有奇数个 1）分组后的明文组和 56 位的密钥按位替代或交换的方法形成密文组的加密方法。

DES 算法具有极高安全性，在现有的破解方法中，除了运用穷举搜索法对 DES 算法进行攻击外，尚未发现更有效的办法。

## 2.8 WEB 服务器

课程知识库系统采用的 WEB 服务器前段是一个 Nginx 和后端是 3 个 Tomcat。其中 Nginx 反向代理<sup>[14]</sup>使用，Tomcat 发布 Jsp 程序<sup>[15]</sup>。下面分别介绍 Nginx 和 Tomcat 以及系统发布架构。

Nginx ("engine x") 是一个高性能轻量级的 HTTP 和反向代理服务器，同时也是一个 IMAP/POP3/SMTP 代理服务器。Nginx 在知识库系统中做反向代理服务器使用，通过 80 端口代理到各项目的发布端口。

Tomcat 是由 Apache、Sun 及其他一些公司共同开发而成，是 Apache 软件基金会（Apache Software Foundation）的 Jakarta 项目中的核心项目之一。它是一种运行 servlet 和 JSP Web 应用的基于 Java 的容器，可以用来发布 JSP 应用程序。

课程知识库系统的发布架构如图 2-1 所示。

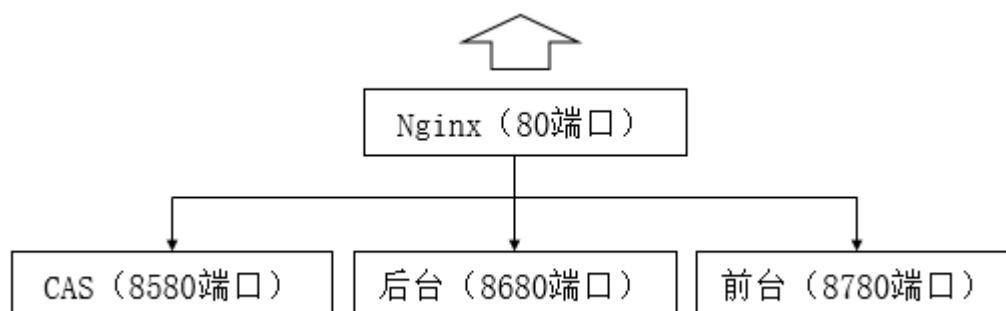


图 2-1 知识库系统发布架构

## 2.9 MySQL

课程知识库系统选用 MySQL<sup>[16]</sup>作为后台数据库,用来存储用户的基本信息,搜索记录,收藏记录和用户登录记录。数据库为 openrdf。下面对 MySQL 做简单介绍。

MySQL 是一个源码开放的小型关联式数据库管理系统,瑞典 MySQL AB 公司开发,目前隶属于 Oracle 公司。MySQL 如今被广泛地应用在互联网的中小型网站中。由于 MySQL 体积小、速度快、总体拥有成本低,并且开放源码等特点,成为了许多中小型网站为了降低网站总体运行成本的最佳选择。MySQL 的 SQL 语句是访问数据库最常用标准化语言。由于 MySQL 开发源码,为系统的开发提供了强有力的支持,并为 MySQL 以后的扩展提供了坚实的基础。

### 3. 系统分析

知识库系统的系统分析包括可行性分析、需求分析、业务流程分析和数据流程分析四个方面，下面分别介绍分析。

#### 3.1 可行性分析

下面对知识库系统从技术可行性、经济可行性和操作可行性三个方面进行分析。

##### 3.1.1 技术可行性

RDF 和 SparQL 是语义网检索的基础，课程知识库系统利用惠普实验室的开放源代码 Jena Semantic Web Framework 在 Java 应用程序中搭建 RDF 数据模型，完成系统对 RDF 文件的操作需求，加以老师的指导和帮助可以解决该技术问题。

利用 SSH 框架技术可以快速搭建基于 B/S 架构的先进、安全和跨平台的 JAVA WEB 系统，利用 CAS 可以保障系统各项目之间的安全，应用也不是问题。

DES 算法是常见的加密算法，该应用主要难点是加密模型的建立，网上有很多该应用实例，使用起来也不是问题。

综上技术可行性分析，课程知识库系统不存在技术性问题。

##### 3.1.2 经济可行性

由于课程知识库系统的主要应用背景是毕业课程设计，其实并不注重直接的经济效益，其实现目的是针对毕业生自身水平和能力的提高，并且程序发布的要求不高，只要有一台可以运行 JDK、Tomcat 和 Nginx 的服务器即可，对系统环境和配置并没有刚性的需求，所以不用考虑该系统的经济问题。

##### 3.1.3 操作可行性

课程知识库系统主要面向的使用对象是学生等对课程知识点进行精确、快速的查询，只要用户熟悉常用的搜索引擎如百度、Google 等就可以使用该系统，并且由于系统采用 B/S 架构，客户端只需要一个浏览器即可访问系统。因此，用



户只要会一些电脑操作，就可以进行查询所需的知识点，而不需要用户有较高的计算机专业知识。该系统对于网站的维护也十分方便，系统管理员只需登录系统后台就可以对系统信息进行维护。

## 3.2 需求分析

信息检索技术随着科技的发展而发生着巨大的变化，基于 RDF 的 OS 课程知识库系统将会比传统的检索更加准确和快速。基于语义网的 RDF 技术必将成为下一代互联网的标识。

### 3.2.1 功能需求分析

课程知识库系统是 B/S 架构的基于 RDF 的知识检索系统，该系统有三个项目组成，其中两个是自己开发的，分别是前台和后台；另一个是 CAS 中央认证系统，采用耶鲁大学开源项目，并集成 DES 加密算法和数据库等。

#### (1) 前台相关功能

- a) 从知识库系统检索所需的知识点
- b). 注册用户
- c). 认证邮箱
- d). 收藏知识点

#### (2) 后台相关功能

- a). 系统管理，包括对知识点的增、删、改、查。
- b). 搜索管理，包括搜索统计、搜索查询和收藏查询。
- c). 用户管理，包括用户列表、添加用户、删除用户。
- d). 个人中心，包括对个人信息的操作。

#### (3) CAS 功能

- a). 登录系统。
- b). 退出系统。

### 3.2.2 系统性能需求分析

实用性高：为平时学习具体知识提供方便，能有效的对知识点进行管理。

**操作简单：**课程知识库系统同常见的搜索系统如百度、Google 类似，系统对使用者水平没有要求，同时系统操作简单明了。

**技术先进：**系统设计和开发能紧跟着整个计算机发展潮流。其中，RDF 是语义网发展的重要方向，CAS 系统和 DES 加密算法是常见的企业级应用。利用先进的设计思想，较新的开发技术和工具，使整个系统无论是功能设计上，还在技术上，都能处于前沿地位。

**安装使用简便：**系统采用将服务器和客户机分开，服务器端仅需要简单的配置就能运行。客户机端只需用浏览器就可直接访问使用。

**代码可读性好：**系统开发过程中涉及的大量代码均有注释，方便阅读和维护。

### 3.2.3 系统运行需求和开发工具

**操作系统：**Ubuntu 12.04

**必要环境：**JDK 1.6

**硬件环境：**CPU 英特尔 Xeon L5410

**数据库：**MySQL5.10

**Web 应用服务器：**Tomcat 7.0、Nginx1.6.0

**开发工具：**MyEclipse10.6

## 3.3 业务流程分析

本节主要介绍课程知识库系统的业务流程，分为前台业务流程和后台业务流程分别介绍。

### 3.3.1 前台业务流程分析

前台的业务流程如下图所示，主要描述了用户在前台页面进行的功能操作，其中用户业务包括查询知识点和收藏知识点，用户管理业务包括搜索词条、注册用户和用户登录（需要跳转到 CAS 认证），系统业务流程图如图 3-1~3-3 所示。

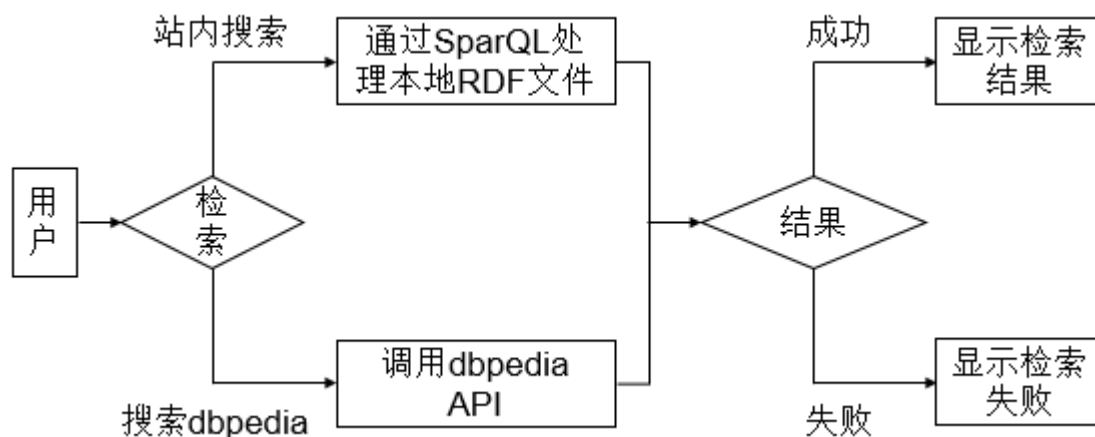


图 3-1 搜索词条流程图

本系统词条数增加的重要途径是通过用户添加的，因此，注册用户是该系统必不可少的操作。当新用户注册时，首先调转到注册页面，用户输入用户名、密码等必要的信息后点击注册，但注册成功后会收到注册邮件，点击即可验证账户。

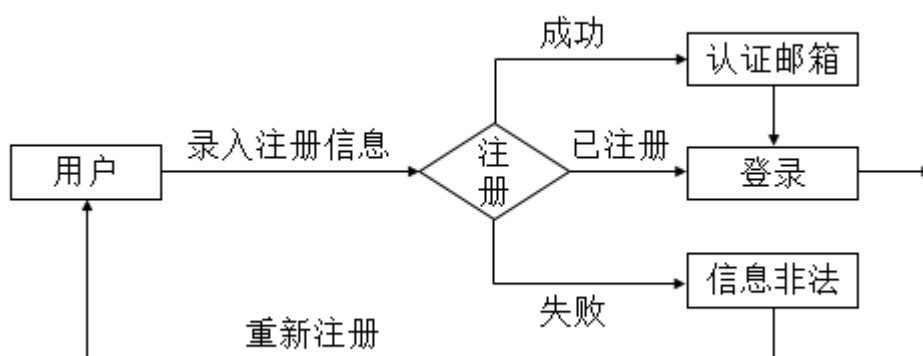


图 3-2 注册用户流程图

### 3.3.2 后台业务流程分析

课程知识库系统的后台业务流程如下所示，主要描述管理员在后台进行的管理操作，系统发送请求和相应的流程。如图 3-3 所示，知识库系统后台功能可分为 4 个模块，分别是系统管理模块、搜索管理模块、用户管理模块和个人中心。系统模块是对系统的管理模块，主要功能是对 RDF 文件的管理，搜索管理模块主要做系统的搜索记录和相关数据的统计工作，用户管理模块是对系统的用户的管理，包括增、删、改、查等，个人中心展示当前登录用户的信息，其中包括修

改登录密码功能。

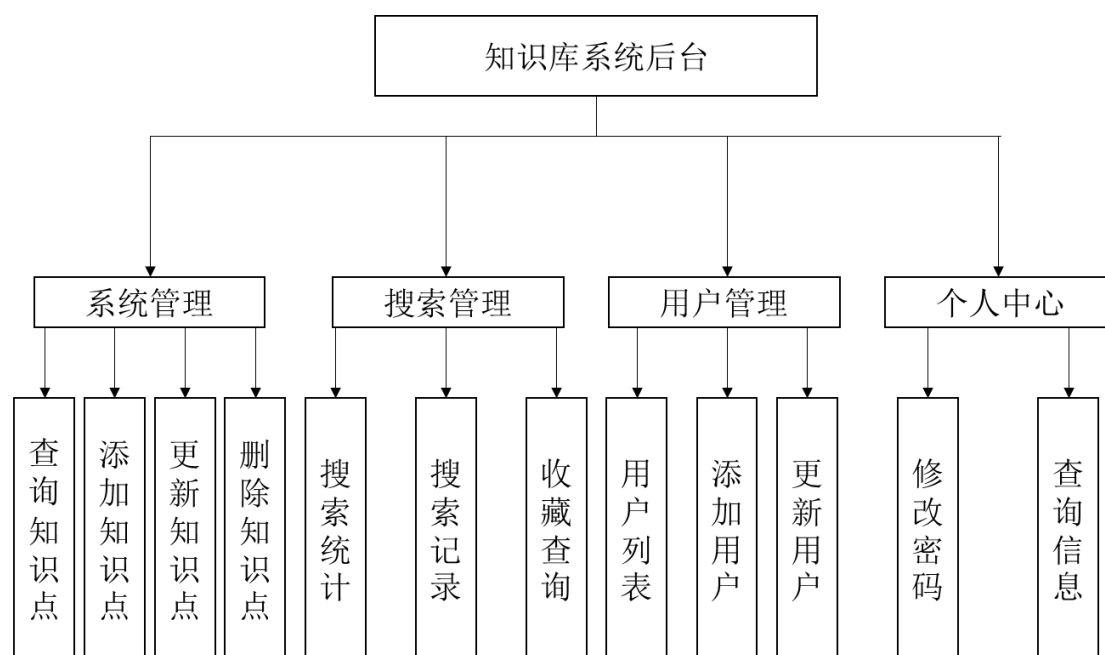


图 3-3 后台功能

### 3.4 数据流程分析

根据前台业务流程,可以将数据流分为两类。一类使用户检索信息的数据流,主要是对 RDF 文件进行查询操作;另一类是用户的查询记录和收藏记录,主要是对 MySQL 数据库中相应的数据库进行操作。同样,根据后台的业务流程,也可以将数据流分为两类,一类是对系统的管理时产生的数据,主要是对 RDF 文件的增、删、改、查等;另一类这是通过搜索管理、用户管理和个人中心操作产

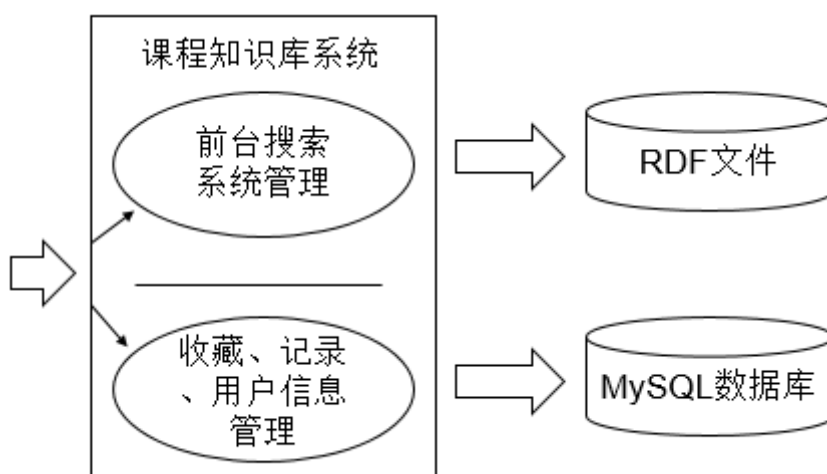


图 3-4 知识库系统数据流

生的数据，同样是对 MySQL 进行的操作。课程知识库系统数据流如图 3-4 所示。

## 4. 系统总体结构设计

对于课程知识库系统的设计采用功能前台、管理后台和认证中心相分离的设计系统，整个系统共分为 3 个项目，分别是功能前台项目、管理后台项目和认证中心项目。所有的项目采用统一的设计模板，保证系统整体风格统一。知识库系统的系统架构如 4-1 图所示。下面分别说明每个项目的作用和实现的功能。

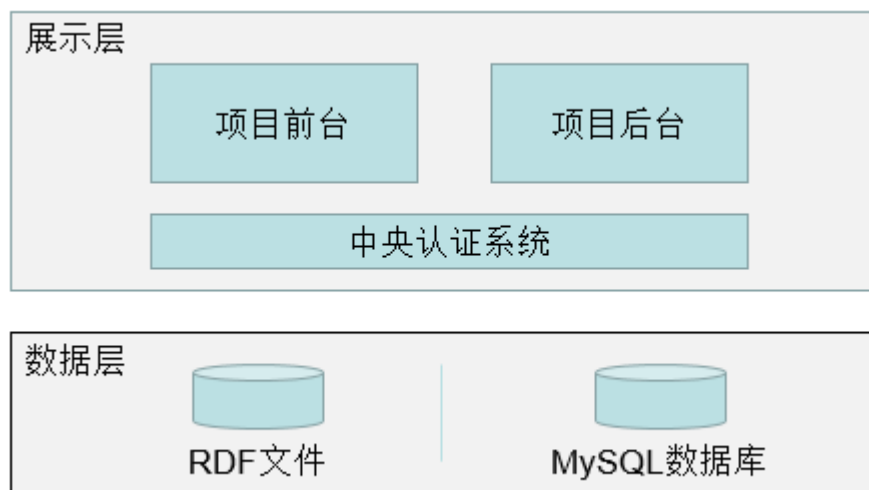


图 4-1 知识库系统系统架构图

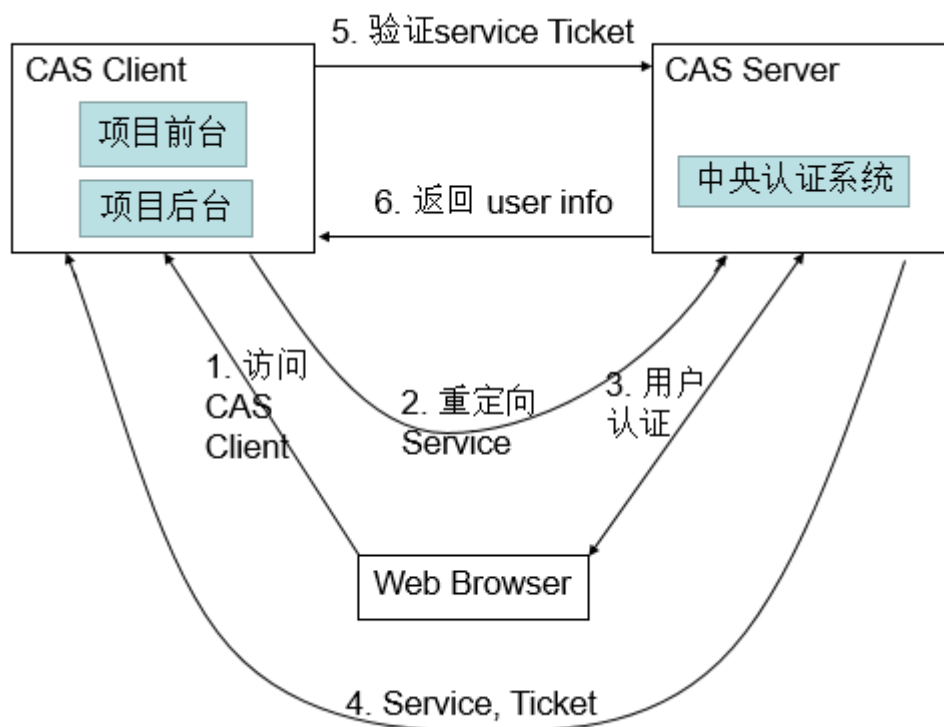


图 4-2 系统安全架构图

认证中心项目：主要为前台和后台项目提供统一的认证。主要包括登录、退

出和 ticket 的验证等功能，系统安全架构图如 4-2 所示。

功能前台项目：主要提供词条的检索功能，词条收藏功能。其中还包括用户注册功能。

管理后台项目：主要是对课程知识库系统提供管理使用，包括词条管理，用户管理和系统数据统计等功能。

本系统数据的操作核心是包括两个部分，一是 MySQL 数据库，另一个是 RDF 文件。所有项目同时对统一 MySQL 数据库和 RDF 文件进行操作。

## 5. 系统数据存储设计

课程知识库的数据存储包括两个部分：1) 用户信息、词条检索、词条收藏等信息，保存在 MySQL 数据库中 2) 知识库系统词条信息保存在 RDF 文件中，下面分别介绍两种存储的设计模式。

### 5.1 MySQL 数据字典

数据字典是系统对数据的数据项、数据结构、数据流、数据存储、处理逻辑、外部实体等进行定义和描述。数据字典是数据库的中心，也是知识库系统需求分析的重要驱动之一。

#### 5.1.1 数据字典

本系统数据库共包含登录表、用户信息表，词条搜索记录表和词条收藏表。下表是数据库中各表的数据字典。

表4-1 登录表

字段名称	类型	长度	是否为空	是否主键	说明
userId	varchar	50	否	是	用户 ID (UUID)
userLoginEmail	varchar	30	否	否	用户登录邮箱
password	varchar	24	否	否	登录密码
userQQ	varchar	10	是	否	用户 QQ

登录表（表 4-1）是存储用户的登录信息的表，主要使用在用户登录的验证使用。登录表的主键 userId 是用户 ID，该 ID 采用 32 位的 UUID，保证所有的用户 ID 永不重复；userLoginEmail 是用户登录时使用的邮箱；password 是登录密码，该密码在系统存储时为考虑安全问题，采用 DES 算法加密；userQQ 使用用户的登录 QQ。

表4-2 用户信息表

字段名称	类型	长度	是否为空	是否主键	说明
userId	varchar	50	否	是	UUID
userName	varchar	20	否	否	用户名称
userStatus	varchar	24	是	否	用户状态



aliPayAccount	varchar	30	是	否	支付宝账户
accountCreateTime	float		是	否	创建时间
userBirthday	varchar	24	是	否	生日
userSex	varchar	10	是	否	性别
userCountry	varchar	20	是	否	国家
userProvince	varchar	20	是	否	省份
userCity	varchar	20	是	否	城市
userAddress	varchar	100	是	否	地址
userPostCode	varchar	10	是	否	邮编
userEmail	varchar	30	否	否	邮箱
emailAuthentication	int	1	否	否	是否认证
userPhone	varchar	15	是	否	手机
userQQ	varchar	10	是	否	QQ
userMSN	varchar	30	是	否	MSN
userBriefInfroduction	text		是	否	简介
userOthers	text		是	否	其他

用户信息表(表 4-2)中保存的是用户的基本信息,是对登录表的扩充。`userId` 是主键,同登录表的 `userID`; `userName` 是用户的姓名, `userStatus` 是用户状态,表示是否禁用; `accountCreateTime` 表示账户的创建时间, `userBirthday` 表示用户的出生日期, `userSex` 表示用户的性别, `userCountry`、`userProvince`、`userCity` 分别表示用户的国家、省份和所在的城市, `userAddress` 表示用户的地址, `userPostCode` 表示邮编, `userEmail` 同登录表中的登录邮箱, `emailAuthentication` 标识用户邮箱是否认真,其中 0 表示未认证,1 表示已认证。`userPhone` 表示用户的联系方式, `userQQ` 和 `userMSN` 分别表示用户的 QQ 和 MSN, `userBriefInfroduction` 表示用户的简介, `userOthers` 记录用户的其他信息。

词条搜索记录表中主要是在用户搜索时做数据统计。其中, `id` 是主键,记录系统的搜索次数, `keyword` 表示每次搜索的关键字, `searchTime` 表示搜索的时间, `whoSearch` 是记录谁搜索的,主键是用户表中的 `userId`, `other` 表示该词条是否在该系统中,其中,0 表示搜索失败,1 表示搜索成功。

表4-3 词条搜索记录表

字段名称	类型	长度	是否为空	是否主键	说明
id	int	11	否	是	编号
keyword	varchar	100	否	否	搜索关键字
searchTime	varchar	100	否	否	搜索时间
whoSearch	varchar	50	否	否	搜索人
other	text		否	否	是否查询到

表4-4 词条收藏表

字段名称	类型	长度	是否为空	是否主键	说明
id	int	11	否	是	编号
userId	varchar	50	否	否	用户 ID
keyword	varchar	100	否	否	收藏关键字
storeTime	varchar	100	否	否	收藏时间
other	text		是	否	其他

词条收藏表主要记录已登录用户收藏词条的记录。其中，id 为主键，userId 表示用户，主键是用户表中的 userId，keyword 是收藏的关键字，storeTime 表示收藏时间，other 记录其它一些信息。

以上是基本表的设计，在本系统中因添加了统计用户搜索词条的功能，因此建立了一个用于统计查询词条的视图。

视图是在基础表上建立的虚表，其内容是由查询语句定义的。本系统中因涉及对用户搜索记录的统计，故在词条搜索记录表的基础上创建了搜索统计视图，该视图的创建代码如下：

```
DELIMITER $$
USE `openrdf` $$
DROP VIEW IF EXISTS `search_statistics` $$
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL
SECURITY DEFINER VIEW `search_statistics` AS (
SELECT
  `search_history`.`keyword` AS `keyWord`,
  COUNT(0) AS `searchCount`,
  SUM(`search_history`.`other`) AS `successCount`,
```

```
MAX(`search_history`.`serachTime`) AS `lastSearchTime`
FROM `search_history`
GROUP BY `search_history`.`keyword`
ORDER BY COUNT(0)DESC,`search_history`.`keyword`)$$
DELIMITER ;
```

### 5.1.2 数据库设计

本系统数据库设计主要分为 E-R 图设计和数据库连接池两部分。

#### 1) E-R 图

E-R 图<sup>[17]</sup>也称实体-联系图(Entity Relationship Diagram)，是描述客观世界的概念模型图，也是课程知识库系统数据库设计的重要驱动。

知识库系统共涉及 2 个实体，分别是用户和词条。图 5-1 和图 5-2 分别是用户和词条的 E-R 图。

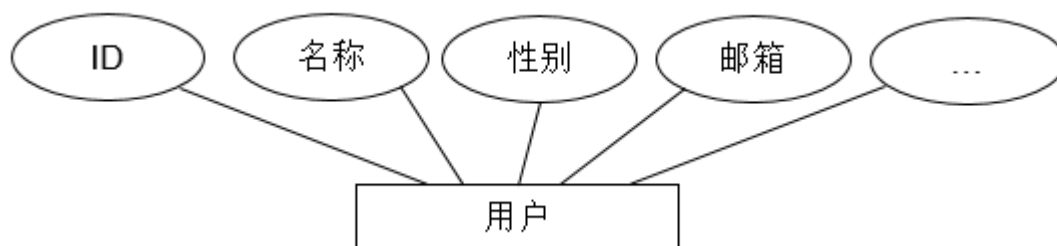


图 5-1 用户表 E-R 图

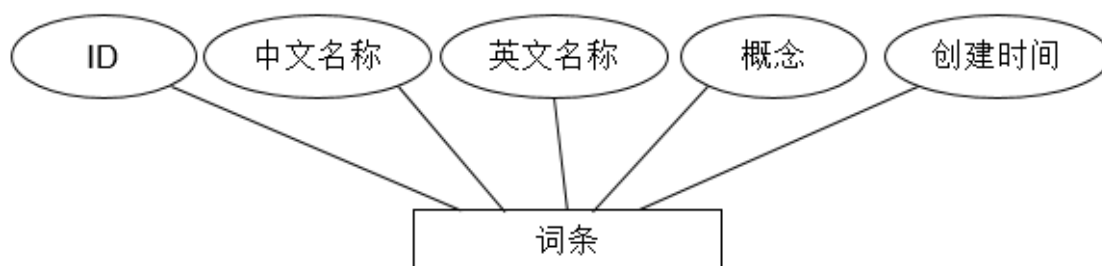


图 5-2 词条表 E-R 图

用户实体和词条实体在联系时会产生两个联系实体，如图 5-3 所示。在查询时，用户与词条是 p:k 的关系，故产生了词条搜索记录表；在收藏词条是，用户和词条是 m:n 的关系，故产生了词条收藏表。

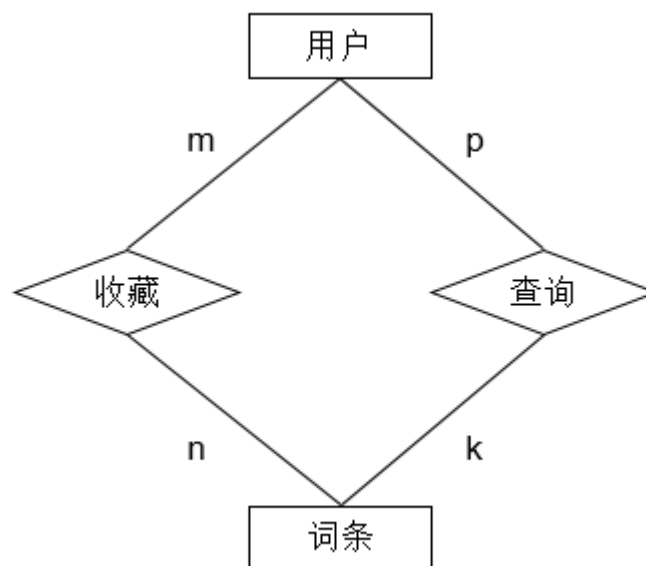


图 5-3 联系实体图

## 2) 数据库连接池

本系统采用 SSH 框架技术，采用 C0P3 数据库连接池。采用数据库能够极大的改善 JAVA 程序的性能，提高程序访问数据库的速度，简化开发模式，间接地给用户带来更好的用户体验。c0p3 是常见的数据库连接池，创建数据库连接池主要是由 hibernate 管理，c0p3 的配置主要分为 3 部分：1) 在 WEB-INF/lib 目录下添加 c0p3 的 jar 包 c3p0-0.9.1.2.jar。2) 配置 src 文件下的 applicationContext.xml 文件中添加如下配置：

```

<bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource"
destroy-method="close">
<!-- database driver -->
<property name="driverClass" value="${jdbc.driverClassName}"></property>
<!-- database URL -->
<property name="jdbcUrl" value="${jdbc.url}" />
<!-- MySql user -->
<property name="user" value="${jdbc.user}"></property>
<!-- MySql password -->
<property name="password" value="${jdbc.password}"></property>
<!-- initialize database connect pool -->
<property name="initialPoolSize" value="${jdbc.initialPoolSize}"></property>
<!-- minimum pool size -->

```

```
<property name="minPoolSize" value="\${jdbc.minPoolSize}"></property>
<!-- Maximum pool size -->
<property name="maxPoolSize" value="\${jdbc.maxPoolSize}" />
<!-- maximum idle time, default 0(s), never drop -->
<property name="maxIdleTime" value="\${jdbc.maxIdleTime}"></property>
<!-- acquireIncrement -->
<property name="acquireIncrement" value="\${jdbc.acquireIncrement}" />
<!-- acquireRetryAttempts -->
<property name="acquireRetryAttempts" value="\${jdbc.acquireRetryAttempts}" />
<!-- idleConnectionTestPeriod -->
<property name="idleConnectionTestPeriod"
value="\${jdbc.idleConnectionTestPeriod}" />
</bean>
```

为了能从配置文件中读取数据库连接的配置信息，在该文件中添加如下代码，用户从 jdbc.properties 中读取数据库配置信息。

```
<bean id="propertyConfigurer"
class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
    <property name="locations">
        <list>
            <value>WEB-INF/classes/jdbc.properties</value>
        </list>
    </property>
</bean>
```

其中，jdbc.properties 的配置文件示例如下：

```
jdbc.driverClassName=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://127.0.0.1:3306/openrdf?useUnicode=true&characterE
ncoding=utf-8
jdbc.user=root
jdbc.password=root
jdbc.initialPoolSize=10
jdbc.minPoolSize=3
jdbc.maxPoolSize=25
jdbc.maxIdleTime=60
```

```
jdbc.acquireIncrement=5
jdbc.acquireRetryAttempts=30
jdbc.idleConnectionTestPeriod=60
```

至此，数据库连接池的配置完成。

## 5.2 RDF 文件设计

在课程知识库系统中，系统操作的核心是 RDF 文件，RDF 文件中存储着所有的词条信息。下面分别介绍 RDF 文件的结构和主要操作。

### 5.2.1 RDF 文件结构设计

RDF 是本系统中词条的存储文件，也是本系统设计的关键部分。本系统中将一个词条抽象为一个本体，并主要存储词条的中文名称、英文名称、具体概念、创建时间和编号共 5 个属性，一个简单的 RDF 文件示例如下：

```
<?xml version='1.0' encoding='UTF-8'?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#"
  xmlns:openrdf="http://www.w3.org/2000/01/rdf-schema#" >
  <rdf:Description rdf:about=" http://www.166801.com/source#Memory">
    <vcard:SOURCE rdf:parseType="Literal">内存是计算机中重要的部件之一，
它是与 CPU 进行沟通的桥梁...</vcard:SOURCE>
    <vcard:NAME xml:lang="zh-cn">内存</vcard:NAME>
    <vcard:NAME xml:lang="en">Memory</vcard:NAME>
    <vcard:BDAY
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">1910-20-23</vcard:BD
AY>
    <vcard:N rdf:nodeID="A0"/>
  </rdf:Description>
  ...
</rdf:RDF>
```

该文件对词条“内存”进行定义，具体说明如下：第一行指明文件的类型是

XML 文件，并且使用 UTF-8 编码方式；所有的词条都包含在“rdf:RDF”节点内，该节点可以包含任意多个“rdf:Description”子节点，其中每个子节点就是一个词条；每个词条子节点都有一个“rdf:about”属性，该属性是本词条的一个唯一标示，每个词条节点下有包含 5 个子节点，表示该词条的属性。下图 4-4 是 RDF 文件的结构。

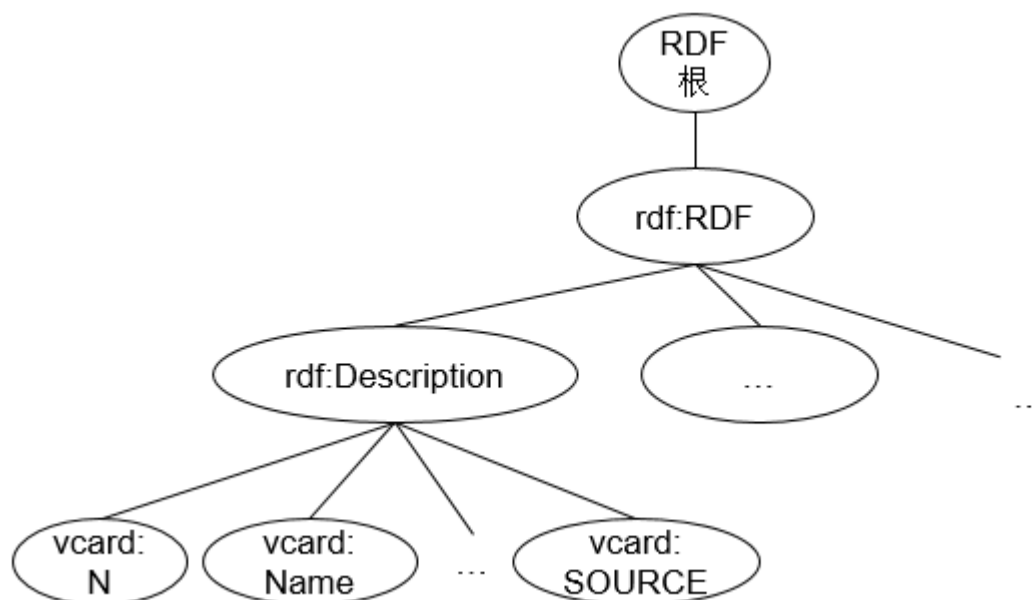


图 4-3 RDF 文件结构

### 5.2.2 RDF 文件操作

对 RDF 文件的操作主要可以概括为对子节点的增、删、改、查，映射到对词条就是对词条的增、删、改、查。对 RDF 文件的操作主要参照 Jena API。下面以增加词条和搜索词条分别介绍。

增加词条就是对 RDF 文件添加一个子节点，主要的实现代码如下：

```

public boolean AddConcept2RDF(Concept concept) {
    // get rdf file location
    String          filePath          =          URLPropertiesUtils
        .getPropertiesUrl(URLPropertiesUtils.RDF_FILE_LOCATION);

    // create an empty model
    Model model = ModelFactory.createDefaultModel();
  
```

```
// create inputStream
InputStream in = FileManager.get().open(filePath);
if (in == null) {
    throw new IllegalArgumentException("File: " + filePath
        + " not found.");
}

// read the RDF/XML file
model.read(in, "", "RDF/XML");

try {
    in.close();
} catch (IOException e) {
    e.printStackTrace();
}

model.write(System.out, "RDF/XML");

// build rdf_about
String rdf_about = "";
if (concept.getEnName() != null) {
    rdf_about = "http://www.166801.com/source#" + concept.getEnName();
} else {
    rdf_about = "http://www.166801.com/source#" + concept.getCnName();
}

// create the resource
Resource r = model.createResource(rdf_about);

// add the property
r.addLiteral(VCARD.BDAY, concept.getBday())
.addProperty(VCARD.N, model.createResource())
.addProperty(VCARD.NAME,
    model.createLiteral(concept.getEnName(), "en"))
```



```

        .addProperty(VCARD.NAME,
                     model.createLiteral(concept.getCnName(), "zh-cn"))
        .addProperty(VCARD.SOURCE,
                     model.createLiteral(concept.getSource(), true));

// write to RDF file
OutputStream outputStream = null;
try {
    outputStream = new FileOutputStream(filePath);
    // 写入编码
    outputStream.write("<?xml version='1.0' encoding='UTF-8'?>\r\n\r\n"
                       .getBytes());
    // model.write(new PrintWriter(outputStream), "RDF/XML");
    model.write(new PrintWriter(new OutputStreamWriter(outputStream,
                                                         "UTF-8"), true), "RDF/XML");
    outputStream.flush();
    return true;
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
} finally {
    try {
        // 关闭
        outputStream.close();
        model.close();
    } catch (IOException e) {
    }
}
return false;
}

```

对 RDF 文件的检索主要是通过 SparQL 实现的，SparQL 的主要代码如下：

```
DESCRIBE *
WHERE {
    ?x <http://www.w3.org/2001/vcard-rdf/3.0#NAME> ?y
    filter regex(str(?y), "keyword", "i")
};
```

使用 SparQL 搜索实现的核心代码如下：

```
public List<Concept> SearchByKeyword(String keyword) {

    // get rdf file location
    String filePath = URLPropertiesUtils
        .getPropertiesUrl(URLPropertiesUtils.RDF_FILE_LOCATION);

    // create an empty model
    Model model = ModelFactory.createDefaultModel();

    // create inputStream
    InputStream in = FileManager.get().open(filePath);
    if (in == null) {
        throw new IllegalArgumentException("File: " + filePath
            + " not found.");
    }

    // read the RDF/XML file
    model.read(in, "", "RDF/XML");

    try {
        in.close();
    } catch (IOException e) {
        e.printStackTrace();
    }

    String queryString = "DESCRIBE *"
```

```
+"WHERE {
+ " ?x <http://www.w3.org/2001/vcard-rdf/3.0#NAME> ?y "
+ "filter regex(str(?y), \"\" + keyword + "\", \"i\")"
+ " }";
```

```
Query query = QueryFactory.create(queryString);
```

```
// Execute the query and obtain results
```

```
QueryExecution qe = QueryExecutionFactory.create(query, model);
```

```
// ResultSet results = qe.execSelect();
```

```
Model resultModel = qe.execDescribe();
```

```
// resultModel.write(System.out, "N-TRIPLE");
```

```
// resultModel.write(System.out, "");
```

```
List<Concept> conceptList = new ArrayList<Concept>();
```

```
Concept concept = new Concept();
```

```
StmtIterator stmtIterator = resultModel.listStatements();
```

```
// 声明计数器
```

```
int i = 0;
```

```
while (stmtIterator.hasNext()) {
```

```
    Statement statement = stmtIterator.nextStatement();
```

```
    Property predicate = statement.getPredicate();
```

```
    if (predicate.toString().contains(
```

```
        "http://www.w3.org/2001/vcard-rdf/3.0#BDAY")) {
```

```
        String bday = statement.getAlt().toString().substring(1, 10);
```

```
        concept.setBday(bday);
```

```
    } else if (predicate.toString().contains(
```

```
        "http://www.w3.org/2001/vcard-rdf/3.0#NAME")) {
```

```
        String name = statement.getAlt().toString();
```

```
        if (name.contains("@en")) {
```

```
            String enName = name.substring(1, name.indexOf("@") - 1);
```

```

        concept.setEnName(enName);
    } else if (name.contains("@zh-cn")) {
        String cnName = name.substring(1, name.indexOf("@") - 1);
        concept.setCnName(cnName);
    }
} else if (predicate.toString().contains(
    "http://www.w3.org/2001/vcard-rdf/3.0#SOURCE")) {
    String source = statement.getAlt().toString();
    source = (String) source.subSequence(1,
        source.indexOf("^^") - 1);
    concept.setSource(source);
}
// 当获取前 5 个节点时，保存一次
if (++i % 5 == 0) {
    conceptList.add(concept);
    concept = new Concept();
}
}
qe.close();
resultModel.close();
return conceptList;
}

```

## 6. 系统详细设计和实现

课程知识库系统的详细设计和实现可以分为系统功能设计和系统实现两部分，下面分别介绍。

### 6.1 功能模块设计

由于课程知识库系统分为 3 个项目，下面分别介绍三个项目的功能设计。其中代码的托管地址分别是：

- 1) 前台项目：[https://github.com/OpenRDF/rdf\\_fronts](https://github.com/OpenRDF/rdf_fronts)
- 2) 后台项目：[https://github.com/OpenRDF/rdf\\_backend](https://github.com/OpenRDF/rdf_backend)
- 3) 认证中心项目：[https://github.com/OpenRDF/rdf\\_cas](https://github.com/OpenRDF/rdf_cas)

#### 6.1.1 前台的功能模块

前台的主要功能可以概括为搜索词条功能、收藏词条功能以及用户注册和验证的相关功能，如图 6-1 所示。

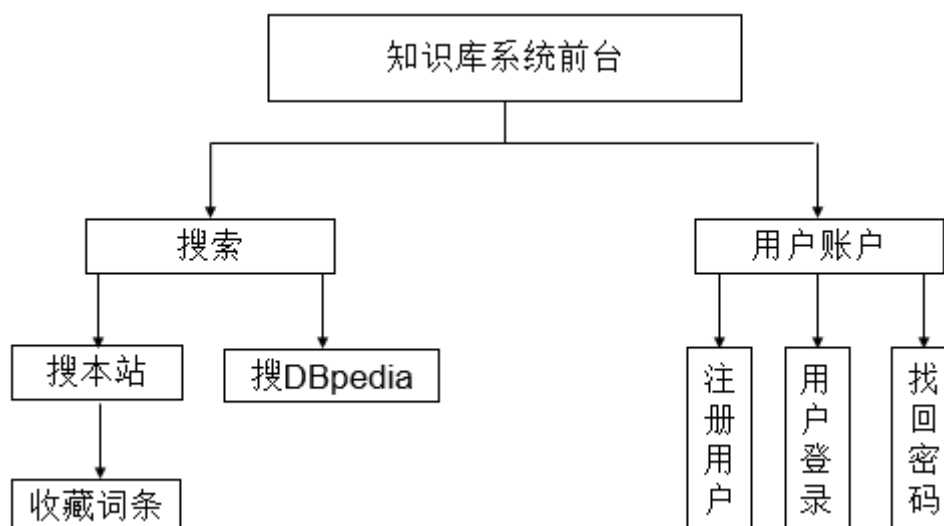


图 6-1 系统前台功能

前台的主要功能是实现对词条的检索功能，该功能是课程知识库系统实现的核心功能。搜索分为两类，一是本地 RDF 文件搜索，二是调用 dbpedia 接口实现搜索公网上的 RDF 文件。功能描述如下：当用户在搜索框中输入所需搜索的词

条后，点击搜索，系统将跳转到结果页面，若系统并未收录该词条，将提示“您搜索的关键字不存在”，若已收录该关键词，将显示该词条的相关信息。

对于用户，搜索词条后可以收藏该词条，这样用户可以在以后方便的查找到自己收藏的记录。

系统词条的增加由用户或管理员添加的，因此，用户注册是本系统不可或缺的功能。为了简化用户注册的流程，本系统用户注册仅需提供所需的登录邮箱和密码即可，当用户注册成功后，系统将会发送注册认证邮件到用户注册邮箱。当用户点击认真链接后将能完成相应的注册流程。

### 6.1.2 后台的功能模块

后台的主要功能主要是对课程知识库系统的管理功能，包括系统词条管理，搜索管理和用户管理等功能，如图 6-2 所示。

词条管理主要是对词条的增、查功能，增加词条是系统的重要功能，该功能允许用户将系统中并不存在的新词条录入系统。系统还实现将所有的词条列出的功能。

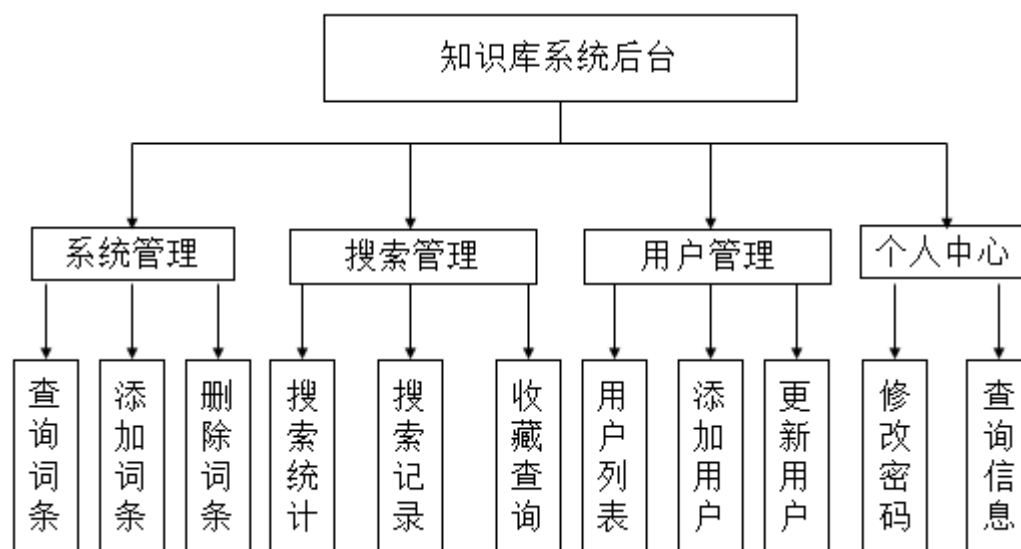


图 6-2 系统后台功能

搜索管理功能主要实现对用户搜索记录和统计，用户收藏词条等功能。在大数据时代，人们要实现对数据的挖掘，就必须收集大量的数据并且分析数据。本系统在实现对所有用户的搜索进行记录和统计的功能。

用户管理是系统的重要模块，本系统也实现了对用户的增、删、改、查等功

能。

### 6.1.3 认证中心的功能

认证中心采用耶鲁大学的开源项目 CAS 做的二次开发。该项目的主要功能是保证本系统认证的统一和安全，并且在该项目中加入了自己编写的 DES 说法对用户密码进行加密，进一步保证了系统的可靠性。

## 6.2 系统实现设计

系统的功能实现可以分为前台功能实现、后台功能实现和 CAS 功能实现。

### 6.2.1 前台的功能实现

前台实现的整体可以包括用户搜索和用户账户两个部分，用户搜索又包括两个部分，一部分是用户对课程知识库系统自身的 RDF 文件进行检索，该搜索是通过 SparQL 语句实现，首页如图 6-3 所示，检索结果如图 6-4 所示。



图 6-3 系统首页

图 6-3 页面下半部分展示了系统所使用相关技术的参考页面和部分功能链接。

当在图 6-3 的页面搜索框中输入“Memory”后回车，系统搜索结果跳转到如

图 6-4 所示的页面。该页面信息是系统搜索本地 RDF 文件的信息。

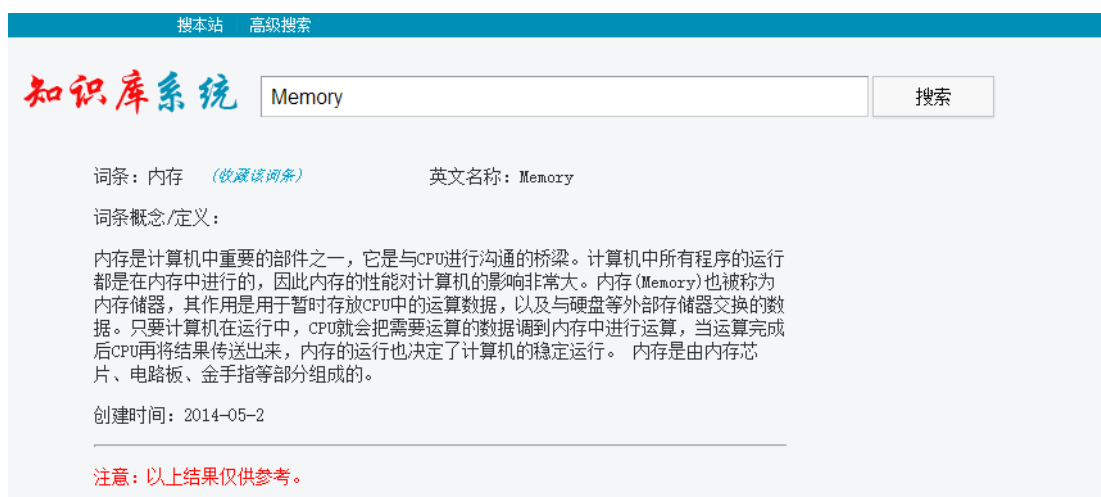


图 6-4 系统检索结果页面

另一部分作为本地 RDF 文件知识内容的补充和完善，通过调用 DBpedia 对外提供的 API 接口进行检索，首页如图 6-5 所示，检索结果如图 6-6 所示。



图 6-5 调用 DBpedia 接口搜索首页

为了填补系统 RDF 文件的局限性，系统提供了“高级搜索功能”。图 6-5 是系统调用 DBpedia 的页面，该功能后台实现是通过使用 REST 框架调用 DBpedia 对外提供的接口实现的。

下面介绍如何使用系统的“高级搜索功能”。在图 6-5 页面搜索框中输入“CPU”回车后，系统跳转到如图 6-6 所示的页面，该页面展示了 DBpedia 所提供的所有关于“CPU”的相关信息。



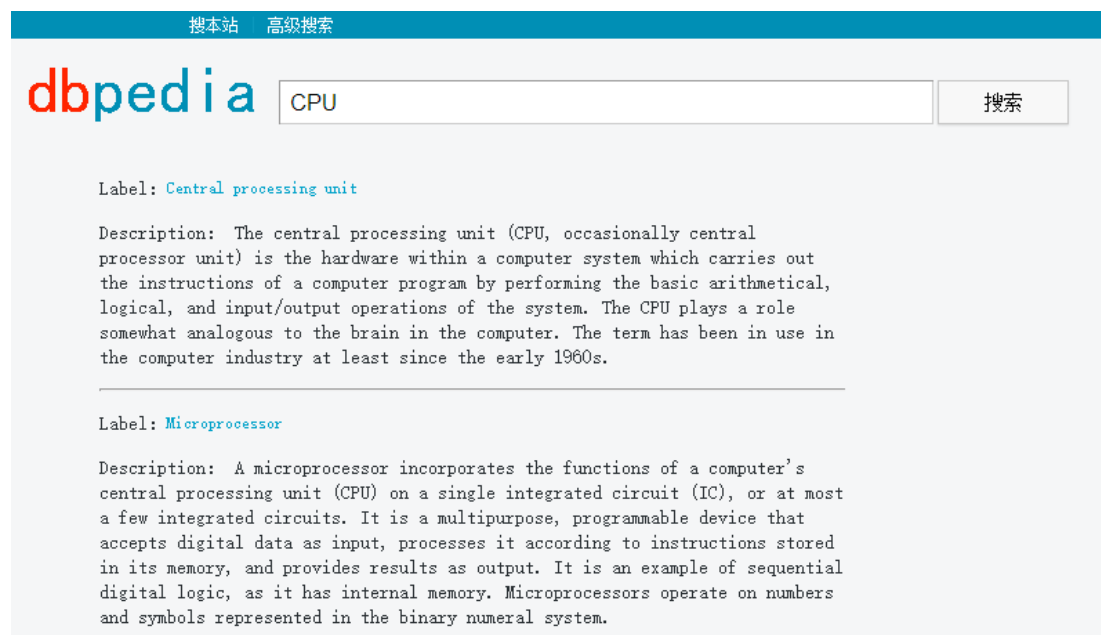


图 6-6 调用 DBpedia 接口搜索结果页面

用户账户管理包括用户注册和用户邮箱激活页面，其中用户注册如图 6-7 所示。



图 6-7 系统注册页面

## 6.2.2 后台和 CAS 功能实现

CAS 是前后台项目的认证中心，后台项目的所有功能都要经过 CAS 认证后才能操作，因此登录后台时系统会跳转到中央认证系统认证，登录如图 6-8 所示，当用户注销后，跳转到如图 6-9 的页面。

图 6-8 展示了系统的登录页面，当用户登录时，需要输入已经注册的邮箱和对应的密码回车即可登录。当用户密码错误时，提示“您提供的凭证有误”，当

用户信息正确时，跳转到系统后台欢迎页面。

图 6-8 系统登录页面

图 6-9 系统退出页面

登录后系统跳转到用户中心, 如图 6-10 所示。所有系统的管理菜单如图 6-11 所示。

图 6-10 系统管理后台欢迎页面



图 6-11 系统管理菜单

系统最为关键的词条的添加和管理页面，添加词条页面如图 6-11 所示，词条列表如图 6-12 所示。

系统管理
查询知识点 | 添加知识点 | 更新知识点 | 知识点列表

添加知识点

名词（中文）：

名词（英文）：

概念/定义：

确认添加

重置

图 6-12 添加词条页面

系统管理		查询知识点   添加知识点   更新知识点   知识点列表		
中文名称	英文名称	概念/定义	更新时间	操作
内存	Memory	内存是计算机...	2014-5-27	详细信息   更新   删除
寄存器	Register	寄存器是内存...	2014-5-27	详细信息   更新   删除
c语言	The C Programming Language	C语言是一种...	2014-5-27	详细信息   更新   删除
硬盘	HDD	硬盘（港台称...	2014-5-27	详细信息   更新   删除
鼠标	Mouse	鼠标：是计算...	2014-5-27	详细信息   更新   删除

图 6-13 词条列表页面

系统搜索统计如图 6-13 所示。其中排名第一的词条“Memory”搜索了 4 次，最后搜索时间为“2014-5-28 22:57:22”。

搜索管理		搜索统计   搜索记录   收藏查询		
次序	概念/定义	搜索次数	结果	最后搜索
1	Memory	4	成功	2014-5-28 22:57:22
2	C语言	3	成功	2014-5-27 13:51:06
3	bad	1	失败	2014-5-28 21:13:17
4	cpu	1	失败	2014-5-28 22:57:18
5	hdd	1	成功	2014-5-27 13:49:02
6	m	1	成功	2014-5-27 14:29:16
7	sd	1	失败	2014-5-28 22:49:05
8	百度	1	失败	2014-5-28 17:59:47

图 6-14 系统搜索统计

系统搜索查询如图 6-14 所示，其中“次序”表示词条检索次数多少的排名，“概念/定义”表示检索的词条，“搜索次数”表示搜索该词条的次数，“结果”中“成功”中表示系统已经搜索该词条，“失败”表示系统并未搜索该词条，搜索失败。

搜索管理		搜索统计   搜索记录   收藏查询	
编号	词条	用户	收藏时间
1	The C Programming Language	a. b	2014-5-27 13:51:08
2	Register	a. b	2014-5-29 14:37:52

图 6-15 词条收藏查询

词条收藏查询如图 6-15 所示，编号表示主键，“词条”表示收藏的词条内容，“用户”表示是收藏者，“收藏时间”是该词条被收藏的时间。

## 7. 系统测试

系统测试是系统开发的重要环节,因为软件的质量不仅仅体现在系统运行的正确性上,而且它和开始编码前所做的系统需求分析和架构设计有着密切的关系。生产环境证明,许多系统的错误未必是在软件开发环节造成的,在需求分析阶段系统就有可能存在一定的问题。因此,在系统开发的前期阶段,就必须对系统做完善的系统设计。

为了保证系统的质量,课程知识库系统着眼于整个系统的生存期,特别是在开发阶段的系统分析。因此本系统测试的概念和实施范围在整个开发各阶段做了多次的复查、评估和检测。

通过严格的单元测试和系统测试得出,课程知识库系统性能良好,安全可靠。

## 结束语

毕业设计是我在大学阶段的最后一个环节,是对我大学所学专业知识的综合应用,在这一过程中我的学习能力和综合素质得到了很大的提高。

通过本次毕业设计,我对 RDF 和语义网等知识有了更深入的认识,对 Java 的 SSH 框架和 CAS 等有了自己独特的理解。在课程设计的过程中,我对以前熟悉的框架技术得到了更加深入的认识,对那些不熟悉或未知的知识,如 RDF、CAS 等,有了一定的认识。

这次毕业设计给我的最大启示是任何困难的事情只要做就一定会有结果。在做毕业设计前我对 RDF 等知识并不了解,经过了这几个月的努力,现在对这些知识有了一定的了解并能完成简单的应用。事实再一次证明任何事情只要做,就一定会有结果,相信自己。

## 致谢

三个月的课程设计转眼即逝，在这成长的三个月中我学到了许多新知识，并且得到了很多人的帮助和鼓励。

首先要感谢我的指导教师王岩老师，在毕业设计的过程中，王老师每周两次的例会为我们汲取新知识的提供了平台。在此次开发过程中，遇到过许许多多的问题，王老师耐心的指导和严谨的工作态度让我受益匪浅。在此，向王岩老师表示最诚挚的敬意和最衷心的感谢。再次，是感谢那些在毕业设计中帮助我的同学们，感谢你们提供的宝贵素材和建议。

感谢在毕业设计期间给予我指导和建议的老师，在我遇到问题时，他们耐心的讲解使我能够很好的解决遇到的问题，积累宝贵的经验。

感谢在这次设计中给予我帮助的同学，他们让我体会到了团队精神的重要性。

## 参考文献

- [1] 郑晨. 基于 SSH 框架的图书馆管理系统分析与设计[O]. 云南大学 2012
- [2] 吴刚. RDF 图数据管理的关键技术研究[O]. 清华大学 2008
- [3] 高志强, 潘越, 马力等. 语义 Web 原理及应用[M]. 北京: 机械工业出版社, 2009
- [4] 朝乐门, 张勇, 邢春晓. DBpedia 及其典型应用[J]. 现代图书情报技术: 2011(03)
- [5] Yale University. Central Authentication Service project[EB/OL]. [2014-05-29] <http://www.jasig.org/cas>
- [6] 李联. 信息安全中的 DES 加密算法[J]. 现代电子技术:2005(09)
- [7] 吴鸿汉, 瞿裕忠. 基于语义网的数字图书馆信息检索模型研究[J]. 计算机研究与发展:2010(02)
- [8] 王咏刚. Web 开发技术史话[J]. 程序员:2004(04)
- [9] 袁颖. 基于语义网的数字图书馆信息检索模型研究[O]. 西安电子科技大学 2010
- [10] 李洁, 丁颖. 语义网关键技术概述[J]. 计算机工程与设计:2007(08)
- [11] 章志龙. 基于语义网的博客搜索系统研究[O]. 武汉理工大学 2009
- [12] 张新曼. 精通 JSP-Web 开发技术与典型应用[M]. 北京: 人民邮电出版社, 2007, 516-564.
- [13] 张倩怡. 800 万“米粉”数据泄露? [N]. 北京日报, 第 11 版, [2014-05-15]
- [14] 杨伟强. 安全高效的 Nginx 反向加速技术在站群中的应用[J]. 信息系统工程 2013(03)
- [15] 张新曼. 精通 JSP-Web 开发技术与典型应用[M]. 北京: 人民邮电出版社, 2007, 516-564.
- [16] Ben Forta. Mysql Crash Course Forta[J]. Ben Sams Publishing (2005-06)
- [17] 杨睿娜. 数据库设计过程中 E-R 图向关系模型的转换[J]. 硅谷:2009(11)