软件设计说明书

**知识库管理系统**

组 号：0504

组员：冯凯 罗妍

指导教师：田刚

目录

[第一章 引言 3](#_Toc25992)

[1.1设计背景 3](#_Toc26169)

[1.2编写目的 3](#_Toc23165)

[1.3功能说明 3](#_Toc31385)

[第二章 相关技术和工具综述 5](#_Toc17334)

[第三章 系统的功能性需求分析 6](#_Toc19324)

[3.1系统功能描述 6](#_Toc11396)

[3.2数据需求分析 7](#_Toc31566)

[第四章 系统的非功能性需求 7](#_Toc26747)

[4.1软硬件环境需求 7](#_Toc18119)

[4.2系统质量需求 8](#_Toc25370)

[4.3资料存储 8](#_Toc11879)

[4.4文本处理 9](#_Toc23006)

[第五章 系统分析 9](#_Toc9421)

[5.1功能分析 9](#_Toc15284)

[5.1.1系统功能分析 9](#_Toc12502)

[5.1.2系统用例图分析 10](#_Toc287)

[5.1.3主要用例的用例规约 11](#_Toc10662)

[第六章 系统总体设计 13](#_Toc10909)

[6.1 系统总体架构设计 13](#_Toc9224)

[6.1.1 软件体系结构设计 13](#_Toc16131)

[6.1.2软件总体架构设计 14](#_Toc9166)

[6.2系统静态结构设计 14](#_Toc12779)

[6.3 数据库设计 14](#_Toc30989)

[6.3.1数据库逻辑结构设计 14](#_Toc6764)

[6.3.2数据库物理结构设计 15](#_Toc12399)

[6.3.3数据访问设计 17](#_Toc9393)

[6.4 类的详细设计 18](#_Toc30456)

[第七章 总结 18](#_Toc15660)

# 第一章 引言

## 1.1设计背景

在日常的生活和工作中，每个人在各自的范围内不断累计，形成不同层次的过程资源财富，涵盖了文本、传真、图形、视频、音频等各个类型的知识财富载体。在知识财富的不断累计过程中，由于知识财富的来源不同、用途不同、载体不同等，往往导致知识财富的时间链条错乱、关联知识无法有效检索、各类知识无法综合运用等问题的出现，伴随知识财富的持续积累，这种问题愈发突出。

## 1.2编写目的

编写软件设计说明书是软件开发过程必不可少的部分，其目的是为了使开发人员在完成设计说明书的基础上完成各项模块的具体实现的设计工作。

作为一个参赛项目，软件设计说明书同时也是为了更好的让评委老师了解项目开发的信息。

## 1.3功能说明

1、用户登录：

用户输入所使用的用户名和密码，判断用户是管理员还是普通用户，判断用户的合法性，如果是合法用户，则允许登录知识库管理系统显示首页；否则显示错误页面，点击跳转登录页面。

2、用户上传文档

用户可以根据系统所提供的分类按文件类型上传相关文件，并给文件添加相应的标签，且支持多文档同一分类同时上传。

3、用户查找、预览、下载文档

用户可以对整个系统当中的文档以及用户自己上传过的文档进行搜索，显示匹配内容并支持所有文件下载。部分文件支持在线预览。

4、管理员

管理员登录后页面跳转至管理员界面，管理员可以对系统的普通用户进行修改密码以及删除用户操作，并对不合适的文件进行屏蔽删除以保证系统的合法纯洁性。

5、形成知识

用户搜索文件之后可以选择保存此次搜索结果形成相关知识，方便下次查找使用；用户上传文件时，同一分类下的文档自动关联形成知识。

6、推荐系统

文档链接：用户在浏览文档时可发送文档至常用目录方便整个系统所有用户查看，也可发送文档至常用文档以便自己查看。

7、规则使用

系统支持为文件设定规则，指定动作（组合）、条件和操作，当动作触发符合设定的条件，系统则自动执行规则的操作。

# 第二章 相关技术和工具综述

1、Java

Java是一种可以撰写跨平台应用程序的面向对象的程序设计语言。

2、Struts2

Struts2是流行和成熟的基于MVC设计模式的Web应用程序框架，本系统基于struts作为控制层。

3、Dbutils （数据库访问层）

DbUtils是一个为简化JDBC操作的小类库，本系统用dbutils进行对数据的访问操作。

4、Openoffice（wps之类的）

对office系类的东西就行转换，方便用户进行预览

5、Mencoder（视频转换）

将绝大多数视频转换为swf文件，方便用户预览

6、Ffmpeg

是一套可以用来记录、转换数字音频、视频，并能将其转化为流的开源计算机程序，方便用户预览

7、Poi

对office系类进行文件的抽取，方便形成用户的知识

8、Pdfbox

对pdf文件进行抽取，方便形成用户的知识

9、elasticSearch

搜索引擎，对全文检索的性能进行优化，减少用户查询的时间

# 第三章 系统的功能性需求分析

## 3.1系统功能描述

1、文件集中存储上传

支持文件的批量上传及压缩包导入，支持Office文档、PDF、图像、音视频和图纸等各类型文件。

2、文档链接

支持将文档发送至常用文档，用户无需每次在打开各个目录中查找； 支持将文档发送至目录，方便其他用户查阅；

3、关联文档

文档上传后，系统会自动为文件关联。支持通过文件唯一编号，实现文档与相关文档、图纸、图片及其它格式附件进行关联，点击链接即可一键快速查看。

3、规则使用

系统支持为文件设定规则，指定动作（组合）、条件和操作，当动作触发符合设定的条件，系统则自动执行规则的操作。

4、智能检索

快速从海量资料中检索出所需文件，可根据文件类型、分类、标题等进行相关检索，也可根据某一次搜索后用户保存形成的知识进行检索。

5、预览和下载

可以直接在系统中预览各类格式的图片、播放视频、音频以及浏览各类Office文件，包括Word、Excel、PowerPoint、WPS等格式。支持所有格式文件下载。

## 3.2数据需求分析

用户数据库：个人信息，常用文档，浏览历史

文件数据库：文件的分类、类型、所有信息，常用目录

规则数据库：用户自定义的文件规则，系统自动为文件生成的规则

标签数据库：标签使用次数，用户使用过的标签

# 第四章 系统的非功能性需求

## 4.1软硬件环境需求

|  |  |
| --- | --- |
| **需求名称** | **详细要求** |
| 硬件 | 开发环境：服务器、PC机；  部署环境：服务器。 |
| 软件 | 开发环境：  开发终端操作系统： Windows XP SP3或Win7；  开发工具：Eclipse 3.6以上、JDK 1.6以上；  服务器操作系统：Windows 2003 server 64位。  部署环境：  服务器操作系统：Windows 2003 server 64位；  运行环境：JDK 1.6。  运行支撑环境：  数据库：Mysql或oracle10g。 |

## 4.2系统质量需求

|  |  |
| --- | --- |
| **主要质量属性** | **详细要求** |
| 正确性 | 运行无误。 |
| 健壮性 | 容错。 |
| 可靠性 | 数据完整性约束。 |
| 性能，效率 | 达到用户要求，秒级响应。 |
| 易用性 | 用户可以按系统提示操作。 |
| 清晰性 | 无二义。 |
| 安全性 | 严格用户权限。 |
| 可扩展性 | 可以根据需要，添加功能模块和基本表。 |
| 可移植性 | 兼顾Unix、Windows操作系统 |

## 4.3资料存储

1、管理的资料素材数目：≥1百万；

2、资料综合分析：通联关系分析时间：≥10万条记录下，<3分钟；

## 4.4文本处理

1、全文检索，当用户输入的查询词小于10个时，检索的响应速度小于5秒；

2、命名实体识别功能，能够支持中文，人名、组织机构名、时间、地名、目标类型、目标名称等；

3、文本抽取功能，能够对pdf, office 2003/2007, html, email，chm，zip，rar抽取文本内容。

# 第五章 系统分析

## 5.1功能分析

5.1.1系统功能分析

知识库管理系统的功能分解图如图所示，主要功能如下：

（1）登录：管理员登录，普通用户登录。

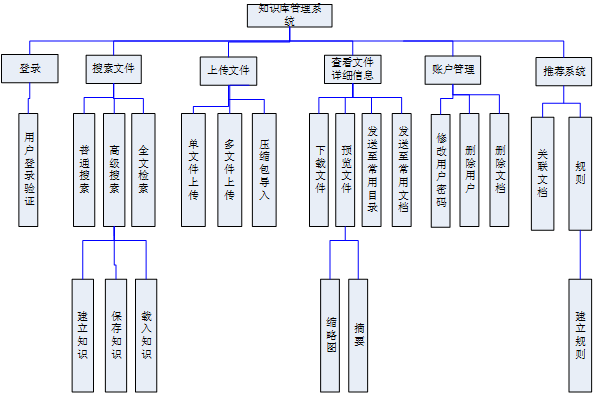
（2）文件上传：支持单文件上传、多文件上传、压缩包导入以及文档新建

（3）文件查找：普通查找，通过关系条件查找，全文检索。

（4）关联文档：通过唯一编号实现文档之间的关联

（5）查看文件：文件下载、文件预览、发送至常用目录、发送至常用文档。

（6）规则使用：系统为文件设定规则。



5.1.2系统用例图分析

根据在需求调研与需求分析结果，使用Rational Rose[8]建模工具建立了知识库管理系统用例。系统的主要业务角色分为管理员、普通用户，系统主要功能模块包括删除用户、删除文件、搜索文件、上传文件、下载文件、查看文件详细信息、预览文件、发送至常用文档、发送至常用目录等。系统主要用例图分析如图所示：

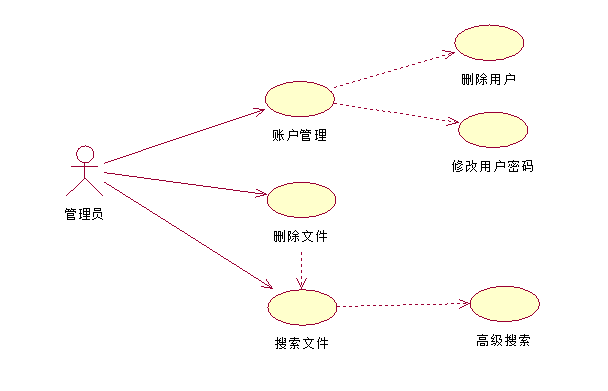


系统主要用例图

### 5.1.3主要用例的用例规约

系统主要功能模块包括删除用户、删除文件、搜索文件、上传文件、下载文件、查看文件详细信息、预览文件、发送至常用文档、发送至常用目录等。主要用例分析及用例规约如下：

系统管理员用例如图



前置条件：用户已登录，并且角色是管理员

基本事件流：

1. 账户管理模块：

①管理员进入删除用户界面，进行删除用户操作

②管理员进入用户信息页面，进行更改用户密码操作

1. 删除文件模块：

管理员进入指定文件页面，进行删除文件操作

1. 搜索文件模块：

管理员进入搜索文件页面，进行搜索文件操作

普通用户用例图



前置条件：用户已登录，并且角色是普通用户

基本事件流：

1. 搜索文件模块：

①用户进入搜索页面，选择全文检索

②用户进入搜索页面，选择高级搜索

1. 上传文件模块：

①用户进入压缩包导入页面，进行上传压缩包操作

②用户进入单文件上传页面，进行单个文件上传操作

③用户进入多文件上传页面，进行多个同一分类文档上传 操作

1. 查看文件详细信息模块：

①用户进入文件详细信息页面，点击下载进行下载文件操 作

②用户进入文件详细信息页面，点击预览进行预览文件操 作

③用户进入文件详细信息页面，点击发送至常用目录以便 其他用户查看

④用户进入文件详细信息页面，点击发送至常用文档以便 自己查看

# 第六章 系统总体设计

## 6.1 系统总体架构设计

6.1.1 软件体系结构设计

系统采取基于B/S结构的软件设计，在这种结构下，用户工作界面是通过WWW浏览器来实现，极少部分事务逻辑在前端实现，但是主要事务逻辑在服务器端实现。大大简化了客户端电脑载荷，减轻了系统维护与升级的成本和工作量，降低了用户的总体成本。

B/S最大的优点就是可以在任何地方进行操作而不用安装任何专门的软件，只要有一台能上网的电脑就能使用，客户端零安装、零维护。B/S架构的软件只需要管理服务器就行了，所有的客户端只是浏览器，根本不需要做任何的维护。无论用户的规模有多大，有多少分支机构都不会增加任何维护升级的工作量，所有的操作只需要针对服务器进行。

6.1.2软件总体架构设计

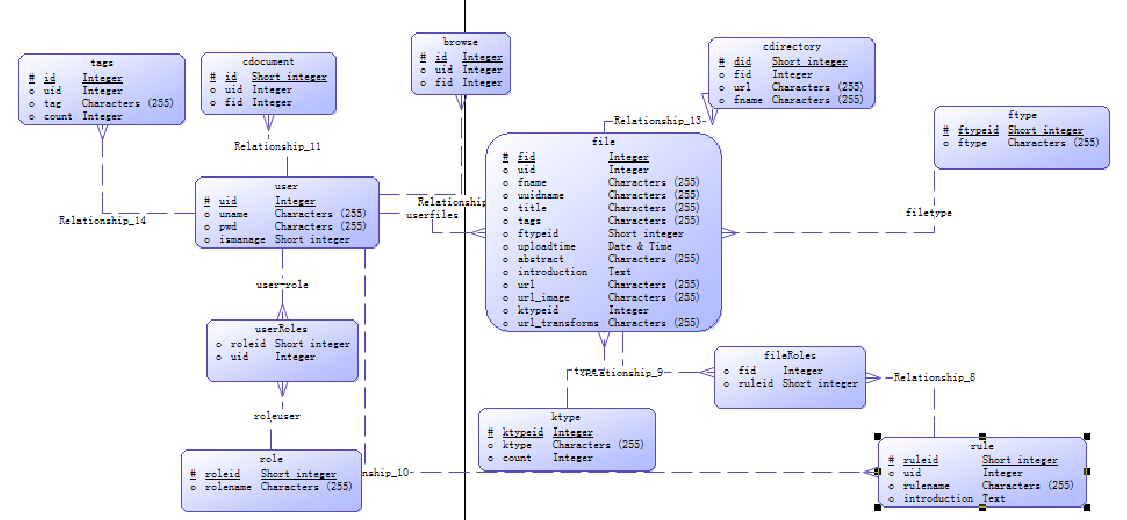
Web项目经常采用的架构形式是MVC和分层架构，本系统采用MVC架构。MVC模式（Model-View-Controller）是软件工程中的一种软件架构模式，把软件系统分为三个基本部分：模型（Model）、视图（View）和控制器（Controller）。MVC目的实现Web系统的职能分工。其中Model层实现系统中的业务逻辑，用JavaBean来实现； View层用于与用户的交互，用JSP来实现； Controller层是Model与View之间沟通的桥梁，它可以分派用户的请求并选择恰当的视图以用于显示，同时它也可以解释用户的输入并将它们映射为模型层可执行的操作。系统整合Struts2框架，该系统用JavaBean 作Model层，View层用jsp实现，由Struts负责，将Action充当Controller层，由Struts2负责，数据交互层用DAO实现，由Dbutils负责。

## 6.2系统静态结构设计

在用例模型的基础上，抽象出了项目管理系统的静态模型，即类模型。系统的类模型如下图

## 6.3 数据库设计

6.3.1数据库逻辑结构设计



### 6.3.2数据库物理结构设计

标签存储数据库

|  |  |
| --- | --- |
| 名称 | 注释 |
| id | 唯一ID |
| uid | 标签对应的用户 |
| tag | 标签的内容 |
| count | 标签出现的次数 |

常用文档，历史记录数据库

|  |  |
| --- | --- |
| 名称 | 注释 |
| id | 唯一id |
| uid | 对应的用户 |
| fid | 对应的文件 |

用户数据库

|  |  |
| --- | --- |
| 名称 | 注释 |
| uid | 用户的ID |
| uname | 用户的名称 |
| pwd | 用户的密码 |
| isamage | 是否为管理员 |

文件信息数据库

|  |  |
| --- | --- |
| 名称 | 注释 |
| fid | 文件的ID |
| uid | 上传的用户 |
| fname | 文件的名称 |
| uuidname | 文件的随机ID |
| title | 文件的标题 |
| tags | 文件的标签 |
| ftypeid | 文件的类型ID |
| uploadtime | 文件的上传时间 |
| abstract | 文件的摘要 |
| introduction | 文件的描述 |
| url | 文件的存储为主 |
| url\_image | 文件的缩略图存储位置 |
| ktypeid | 文件的分类ID |
| url\_transforms | 文件的预览文件的存储位置 |

分类数据库

|  |  |
| --- | --- |
| 名称 | 注释 |
| Ktypeid | 文件的分类ID |
| ktype | 文件分类的名称 |

规则数据库

|  |  |
| --- | --- |
| 名称 | 注释 |
| ruleid | 规则的ID |
| uid | 规则对应的用户 |
| rulename | 规则的名称 |
| introduction | 规则的具体描述 |

类型数据库

|  |  |
| --- | --- |
| 名称 | 注释 |
| ftypeid | 文件类型的ID |
| ftype | 文件类型的名称 |

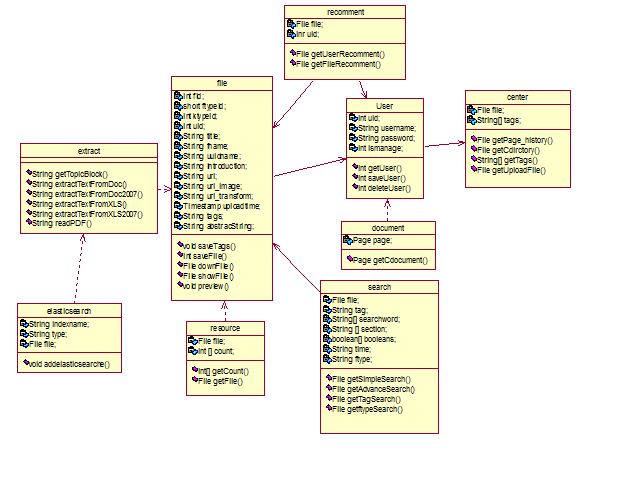
常用目录数据库

|  |  |
| --- | --- |
| 名称 | 注释 |
| did | 常用目录文件的唯一ID |
| fid | 对于的文件ID |
| url | 对应文件的存储位置 |
| fname | 文件的名称 |

### 6.3.3数据访问设计

数据访问设计采用面向对象的方法，将实体类映射封装成JavaBean，数据交互层用Dbutils实现，将数据数据封装成JavaBean作为参数传递给DAO实现类，进行增删改查等操作。QueryRunner查询的结果自动封装成JavaBean，方便系统业务逻辑层对数据进行操作。

## 6.4 类的详细设计



# 总结

本文采用面向对象的思想实现了基于J2EE的B/S架构的知识库管理系统，运用MicrosoftVisio、rationalrose辅助完成了系统的需求设计过程，运用myecplise开发工具进行系统开发，数据库采用mysql，系统发布运行服务器采用TOMCAT，在windows环境下开发。系统架构采用MVC架构模式，整合了struct2、dbutils、word分词、POI、logging、pdfbox、elasticsearch、mahout等开源框架，前台设计采用jQuery等开源js库，按照软件工程定义的软件开发过程来开发和维护系统。