基于Web的网盘系统

概要设计说明书

组长：彭小野 1502040228

组员： 刘刻 1509020236

周懂贤 1506010233

刘云浩 1506010234

吴仕权 1506010235

二0 一七年十二月

目录

[1引言 3](#_Toc502266396)

[1.1编写目的 3](#_Toc502266397)

[1.2背景 3](#_Toc502266398)

[1.3定义 3](#_Toc502266399)

[1.4参考资料 3](#_Toc502266400)

[2总体设计 3](#_Toc502266401)

[2.1需求规定 3](#_Toc502266402)

[2.2运行环境 3](#_Toc502266403)

[2.3基本设计概念和处理流程 4](#_Toc502266404)

[2.4结构 4](#_Toc502266405)

[类包层次结构图 4](#_Toc502266406)

[2.6人工处理过程 6](#_Toc502266407)

[2.7尚未解决的问题 6](#_Toc502266408)

[3接口设计 6](#_Toc502266409)

[3.1用户接口 6](#_Toc502266410)

[3.2外部接口 6](#_Toc502266411)

[3.3内部接口 7](#_Toc502266412)

[4运行设计 8](#_Toc502266413)

[4.1运行模块组合 8](#_Toc502266414)

[4.2运行控制 8](#_Toc502266415)

[4.3运行时间 8](#_Toc502266416)

[5系统数据结构设计 9](#_Toc502266417)

[5.1逻辑结构设计要点 9](#_Toc502266418)

[5.2物理结构设计要点 9](#_Toc502266419)

[5.3数据结构与程序的关系 9](#_Toc502266420)

[6系统出错处理设计 9](#_Toc502266421)

[6.1出错信息 9](#_Toc502266422)

[6.2补救措施 10](#_Toc502266423)

[6.3系统维护设计 10](#_Toc502266424)

**概要设计说明书**

# 1引言

## 1.1编写目的

编写这份设计说明书是为了给出软件的设计方案，为编码、测试提供依据，并作为日后软件维护的参考。

## 1.2背景

软件系统名称：基于Web的网盘系统

任务提出者：软件工程实验任务组

开发者：彭小野、刘刻、吴仕权、刘云浩、周懂贤

用户：个人用户、企业用户

## 1.3定义

REST（Representational State Transfer，表述性状态传递）：一种软件架构风格，通过 URI 对资源进行唯一的标识，通过 HTTP 标准方法操作资源。

JSON（JavaScript Object Notation）：一种轻量级的数据交换格式，它基于 JavaScript 的语法，但又是完全独立于语言的文本格式。

## 1.4参考资料

《需求分析说明书》。

# 2总体设计

## 2.1需求规定

详见《需求分析说明书》。

## 2.2运行环境

硬件环境：

要求操作系统为 Linux（Ubuntu），CPU 主频 2.5GHz，内容 1GB，带宽 1M。

系统运行开发所使用的服务器如下表4-1所示：

表4-1 最低硬件配置表

|  |  |
| --- | --- |
| **设备名称** | **详细要求** |
| 处理器 | Celeron D 2.53G HZ以上 |
| 内存容量 | 至少1GB |
| 外存容量 | 至少30G |
| 联机/脱机 | 需要数据库服务器 |

软件环境：

Windows 2000以上版本系统，Windows 7为最佳

MySQL 5.7以上版本

JAVA虚拟机：JDK1.8或更高版本

Apache Tomcat 8 及以上

安装有 MongoDB，Redis，RabbitMQ等应用。

## 2.3基本设计概念和处理流程

本系统采用三层架构进行设计，层与层之间采用Java接口进行衔接，降低了模块耦合度，并在每一层功能完成后进行相应层的单元测试，对开发过程中避免了很多不必要的麻烦，节约了开发时间。

本系统在开发过程中实现了国际化框架，可以识别不同的浏览器语言相应的显示对应语言的内容，由于开发时间限制，语言包只提供了缺省的中文语言包。

表示层：该系统对每个表单设置表单号和验证码，可以有效防止用户的表单重复提交。本系统参考了Hibernate Validate框架的表单Bean检验方式，对用户输入的表单信息进行了相应的约束，并使用正则表达式进行校验，对于不符合约束的信息，及时在网页上回显给用户，只对符合要求的表单信息提交给下层Service业务处理层，保证了业务处理层接收到的表单信息的纯净性。

业务处理层：对本系统需要提供的功能进行封装和分类，分成若干个子系统。

持久层：持久层对应业务处理层，也分成了若干个子系统。持久层对MySQL的数据库操作进行了实现，数据库连接信息以配置文件的形式存放在网站目录中，增强了系统的灵活性，对于更改数据库只需更改相应的配置文件即可，并对Dao接口对应相应数据库进行实现即可。

## 2.4结构

### 类包层次结构图

**类包层次结构图**

主包名：com.bitswild.clouddisk

子包：

aop: AOP 层；

converter: 数据类型转换层；

dao：持久层接口（数据访问层），与底层MySQL、Oracle、Hbase等进行数据交互。其中有三个接口，分别是学生子系统，教师子系统，题库子系统的对应持久层接口；

dao.impl：持久层接口实现类，分别实现持久层接口中的三个接口；

dto: 数据传输对象层，Service或Manager向外传输的对象；

entity：实体类层，与数据库表结构一一对应，通过DAO层向上传输数据源对象。整个系统的所有实体JavaBean；

exception：自定义异常；

factory：工厂模式工厂包，工厂遵循单例模式，实现根据不同的需要加载对接口的不同实现类，通过Java类加载器实现，从配置文件factory.properties读取相应的类加载信息；

filter: 过滤器层，用于用户鉴权等；

service：业务处理层抽象接口，分为学生子系统，教师子系统，题库子系统三个业务处理接口；

service.impl：业务处理层接口对应的实现类，分别实现业务处理层的三个接口；

util：整个项目的所有工具类，包括每个层的对应工具类；

manager: 通用业务处理层，它有如下特征：

1. 对第三方平台封装的层，预处理返回结果及转化异常信息；

2. 对Service层通用能力的下沉，如缓存方案、中间件通用处理；

3. 与DAO层交互，对多个DAO的组合复用

web.controller：表示层控制型Servlet，实现对用户业务处理请求的初步处理和对业务处理层的请求；

web.form：用户提交表单JavaBean，实现对表单校验和相应的用户反馈信息生成；

web.ui：表示层显示型Servlet，实现对用户的请求预处理和过滤，并转发到JSP页面；

test：测试包

db.properties：JDBC连接数据库的相关配置信息

factory.properties：工厂模式加载相应实现类的映射文件

log4j.properties：log4j日志库配置信息

redis.properties：redis配置信息

**网站根目录层次结构图**

temp：网站缓存文件夹

images：网站图片文件夹

css：层叠样式表文件夹

js：JavaScript文件夹

jsp：网站JSP文件夹

lib：网站引用jar包文件夹

web.xml：网站配置文件

login.html：网站登录页面

home.html：网站首页

## 2.6人工处理过程

暂无人工处理过程。

## 2.7尚未解决的问题

需求中的问题在本设计中已经基本实现，但是基于一般的考虑，可能会在开发过程或者后期版本中增加的功能包括：

1. 图片、视频、音乐、PDF、Office等格式文件的预览；
2. BT种子下载；
3. 离线下载；
4. 文件分享排行榜，实时更新高热度文件分享，增强社交体验‘’
5. 敏感信息过滤、不良内容检测；
6. 会员充值、高级功能；
7. 签到系统，根据用户签到天数可发放相应奖励；
8. 加入企业网盘功能，在企业网盘内部加入用户权限分配、网盘空间分配等功能；
9. 加入消息队列中间件，耗时任务如发送邮件通过消息队列完成，增强系统地异步性；
10. 数据导出为 Excel 表格供下载；
11. 用户操作行为跟踪——数据收集和数据挖掘。

# 3接口设计

## 3.1用户接口

用户开启浏览器访问：http:// 118.89.142.148:8080，使用用户名和密码登录，访问相应的网页，进行查看和编辑。

服务端不向用户提供任何命令，服务端维护人员可以通过修改配置文件\*.properties文件来实现对服务端启动方式的修改。

## 3.2外部接口

服务端需要在运行有Linux系统主机的shell下启动，无交互模式。服务端还需要JVM的支持。

## 3.3内部接口

控制器接口：处理用户请求的接口，分别有用户信息管理控制器，网盘文件操作控制器，主页控制器，文件分享控制器，会员管理控制器，图床控制器。

模型接口：提供数据库访问模型，分别有用户信息模型，原始文件信息模型，用户网盘文件信息模型，用户网盘文件夹信息模型，好友信息模型，云笔记信息模型，图床信息模型。

视图接口：提供视图渲染、数据展示的接口，每个控制器对应一到多个视图接口。

内部接口规定了服务器和客户端之间的交互协议。服务器和客户端之间的接口均为 HTTP 协议，遵循 REST 设计原则。

服务器接收 HTTP 请求对情报信息、指挥命令、文档列表、用户信息进行操作。如无特殊说明，下面接口中 request 和 response 的格式都为 JSON，request 和 response 的 header 中 Content-Type 应为 application/json。

如果 response 是表示一个错误（4xx 和 5xx 返回码），返回一个 Error 对象，Error 对象的属性有：

status：状态码

message：错误信息

常见的返回状态码如下：

200 OK - [GET]：服务器成功返回用户请求的数据

201 Created - [POST/PUT/PATCH]：用户新建或修改数据成功

204 No Content - [DELETE]：用户删除数据成功

400 Invalid Request：用户发出的请求有语法错误，如 JSON 的格式错误

401 Unauthorized：用户没有权限，token 错误

403 Forbidden：用户的访问被禁止，如普通用户试图删除其他用户发布的情报

404 Not Found：用户试图获取不存在的资源，如查看命令详情，但命令 ID 不存在

406 Not Acceptable：请求数据的格式错误，如要求 application/json 的格式，但请求的格式是 text/plain

409 Conflict：被请求的资源和资源当前状态存在冲突，如用户试图新建一个命令，但该 ID 的命令已经存在

422 Unprocessable Entity：请求格式正确，但是由于含有语义错误，无法响应。如用户发送 PUT 请求修改每个情报的内容，但发送的 JSON 对象不完整

500 Internal Server Error：服务器内部错误，如数据库访问出错

根据 RESTful 接口的语义，GET、PUT、DELETE 方法都是幂等的，即多次发送 GET、PUT、DELETE 方法的请求，效果应该和发送一次的效果相同。POST 方法不是幂等的。下面对于 GET、PUT、DELETE 方法不再做特殊说明，认为是幂等的。

# 4运行设计

## 4.1运行模块组合

用户登录，查看网盘全部文件，编辑文件，退出；

用户登录，点击个人头像，修改个人信息，退出；

用户登录，点击“添加文件”，上传文件，退出；

用户登录，查看网盘某种类型的文件（文档、图片、视频、音乐），编辑日程，退出；

用户登录，点击“回收站”，恢复、清空回收站文件，退出；

用户登录，查看云笔记，编辑、预览云笔记，退出；

用户登录，查看图床，上传、下载图床图片，退出；

## 4.2运行控制

用户输入用户名和密码登录成功后，点击“全部”，查看相关信息，点击相关按钮进行操作，点击 Logout 退出登录；

用户输入用户名和密码登录成功后，点击用户头像，进入个人信息管理页面，对个人信息进入编辑；

用户输入用户名和密码登录成功后，点击“文档”“图片”“视频”“音乐”，查看分类过滤后的网盘文件相关信息，点击相关按钮进行操作；

用户输入用户名和密码登录成功后，点击“添加文件”，从客户端电脑中勾选出要上传的文件，点击“确定”，文件进入上传队列，等待文件上传完成；

用户输入用户名和密码登录成功后，点击“回收站”，查看放入回收站的文件和文件夹，右击文件选择“恢复”“删除”等选项完成对应功能，点击页面中的全部清空将回收站中所有文件全部删除；

用户输入用户名和密码登录成功后，点击 “云笔记”，查看云笔记列表，点击某个笔记缩略图进行对应笔记的编辑和预览；

用户输入用户名和密码登录成功后，点击“图床”，用户可拖拽要上传的文件到框内进行上传，上传完成后界面显示生成的对应图片的链接，用户可点击此链接查看图片；

用户输入用户名和密码登录成功后，点击顶部的“上传队列”图标，查看正在上传和上传完成的文件列表；

用户输入用户名和密码登录成功后，在顶部的“文件搜索”框中搜索目标文件，界面将只展示符合搜索条件的文件和文件夹；

## 4.3运行时间

用户登录：约一分钟

查看相关模块：时间不定，超过30分钟不操作登录失效；

编辑相关数据：时间不定，超过30分钟不操作登录失效；

退出：立即。

文件上传、下载：时间不定，根据文件大小决定。

文件其他操作：立即。

即时聊天：文字消息立即接收，语音、视频消息根据大小决定。

# 5系统数据结构设计

## 5.1逻辑结构设计要点

数据结构的物理存储由 MySQL数据库进行控制，并进行了一些特定配置。

聊天信息部分的数据结构使用MongoDB进行存储，系统使用 MongoDB 的默认配置。

## 5.2物理结构设计要点

本系统内所使用的每个数据结构中的每个数据项均以数据库表的形式存储在数据库中，不考虑存取的物理关系。

访问权限：只有授权的用户才能进入系统进行数据操作，而且对一些重要数据，系统设置为只有更高权限的人员方可读取或是操作。所有操作都必须再登陆后操作，系统安全保密性较高。

## 5.3数据结构与程序的关系

1. 数据结构以数据库的表的形式存储在数据库服务器上；

2. 程序访问服务器上的数据库；

3. 用不同权限用户登录的系统只能访问对应的数据库。

# 6系统出错处理设计

## 6.1出错信息

error01-同步出错。

error02-文件上传失败。

error03-文件分享失败。

error04-上传文件被屏蔽。

error05-文件已被删除。

error06-用户未登录。

error07-上传图片格式不正确。

error08-用户名密码格式不正确。

error09-密码错误。

error10-用户搜索无结果。

## 6.2补救措施

出错信息：提示同步出错

补救：以管理员身份访问

出错信息：上传不了文件（过大）

补救：将浏览器切换单兼容模式、检查网络

出错信息：分享不了文件

补救：检查所传信息是否违反规定、是否有敏感词、检查网络

出错信息：上传文件被屏蔽

补救：压缩文件或改后缀名、检查内容是否合法

出错信息：查看文件已被删除

补救：系统提示用户文件不存在

出错信息：分享上传文件提示未登录

补救：系统转到登录注册页面

出错信息：上传图片格式不对

补救：系统直接返回上传界面

出错信息：注册时用户名密码格式不对

补救：系统提示用户重新输入

出错信息：登录时密码错误

补救：系统直接跳转登录界面，提示用户重新输入密码

出错信息：用户搜索无结果

补救：系统提示不存在符合条件的文件

## 6.3系统维护设计

1. 本系统采用日志框架记录程序运行情况和崩溃信息，开发人员可以第一时间知道程序发生错误并及时Debug。同时，如果用户在使用网站的过程中发现任何问题，可以以邮件方式通知开发人员。系统尽可能保证每一个月发布一次更新补丁。
2. 监测系统日志，通过运行系统日志程序，系统会记录下所有用户使用系统的情形，包括最近登录时间、使用的账号、进行的活动等。日志程序会定期生成报表，通过对报表进行分析，可以知道是否有异常现象。
3. 使用CVS进行版本控制，跟踪文件和文档的变更，同时可以防止覆盖其他设计者开发者在同一项目中的文件。
4. 考虑浏览器差异兼容问题，加入IE浏览器条件式、CSS Hack和其他解决方案。
5. 保障网站的服务器安全，保证服务器所在地的物理安全，防止意外事件和人物破坏物理设备。
6. 定期对数据库进行更新、维护，并对冗余数据进行处理。
7. 定期对数据库进行备份工作。为防止不能预料的系统故障或用户不小心的非法操作，必须对系统进行安全备份。除了对全系统进行每月一次的备份外，还应对修改过的数据进行每周一次的备份。同时，应该将修改过的重要系统文件存放在不同的服务器上，以便出现系统崩溃时(通常是硬盘出错)，可及时地将系统恢复到正常状态。