

基于 Web 的网盘系统 课程设计报告

组长：彭小野 1502040228

组员： 刘刻 1509020236

周懂贤 1506010233

刘云浩 1506010234

吴仕权 1506010235

二〇一七年十二月

第一章 目录

第二章 概述	6
2.1. 项目背景	6
2.2. 项目介绍	6
第三章 系统需求分析	8
3.1. 任务概述	8
3.2. 数据描述	8
3.2.1. E-R 图	8
3.2.2. 数据库介绍	10
3.2.3. 数据词典	10
3.2.4. 存储要求	10
3.3. 流程图	10
3.3.1. 业务流程图	10
3.3.2. 数据流程图与过程流程图	14
3.4. 功能需求	16
3.4.1. 功能划分	16
3.4.2. 功能描述	16
3.5. 性能需求	17
3.5.1. 数据精确度	17
3.5.2. 时间特性	17
3.5.3. 适应性	18
3.6. 运行需求	18
3.6.1. 用户界面	18
3.6.2. 硬件接口	19
3.6.3. 软件接口	19
3.6.4. 故障处理	19
3.7. 其它需求	20
3.7.1. 通信接口需求	20
3.7.2. 安全保密需求	20
3.7.3. 可使用性需求	21
3.7.4. 可维护性需求	21
3.7.5. 可移植性需求	21
3.7.6. 灵活性需求	21
3.7.7. 其他专门要求	21
第四章 系统概要设计	23
4.1. 总体设计	23
4.1.1. 需求规定	23
4.1.2. 运行环境	23
4.1.3. 基本设计概念和处理流程	23
4.1.4. 结构	24
4.1.5. 人工处理过程	25
4.1.6. 尚未解决的问题	25

4.2.	接口设计	26
4.2.1.	用户接口	26
4.2.2.	外部接口	26
4.2.3.	内部接口	26
4.3.	运行设计	27
4.3.1.	运行模块组合	27
4.3.2.	运行控制	27
4.3.3.	运行时间	28
4.4.	系统数据结构设计	28
4.4.1.	逻辑结构设计要点	28
4.4.2.	物理结构设计要点	28
4.4.3.	数据结构与程序的关系	28
4.5.	系统出错处理设计	29
4.5.1.	出错信息	29
4.5.2.	补救措施	29
4.5.3.	系统维护设计	29
第五章 系统详细设计		31
5.1.	程序系统的结构	31
5.2.	登陆验证模块设计说明	32
5.2.1.	程序描述	32
5.2.2.	功能	33
5.2.3.	算法	33
5.2.4.	流程逻辑	34
5.2.5.	接口	35
5.2.6.	限制条件	35
5.2.7.	测试计划	35
5.3.	用户注册模块设计说明	35
5.3.1.	程序描述	35
5.3.2.	功能	35
5.3.3.	算法	36
5.3.4.	流程逻辑	37
5.3.5.	接口	38
5.3.6.	限制条件	38
5.3.7.	测试计划	38
5.4.	用户信息管理模块设计说明	38
5.4.1.	程序描述	38
5.4.2.	功能	38
5.4.3.	流程逻辑	39
5.4.4.	限制条件	39
5.5.	文件管理模块设计说明	39
5.5.1.	程序描述	39
5.5.2.	功能	40
5.5.3.	算法	40
5.5.4.	流程逻辑	41

5.6.	回收站模块设计说明	44
5.6.1.	程序描述	44
5.6.2.	功能	44
5.6.3.	算法	45
5.7.	图床模块设计说明	45
5.7.1.	程序描述	45
5.7.2.	功能	45
5.7.3.	流程逻辑	46
5.7.4.	接口	46
5.7.5.	限制条件	46
第六章	数据库设计	47
6.1.	外部设计	47
6.1.1.	标识符和状态	47
6.1.2.	使用它的程序	47
6.1.3.	约定	47
6.2.	结构设计	47
6.2.1.	概念结构设计	47
6.2.2.	逻辑结构设计	49
6.2.3.	物理结构设计	52
6.3.	运用设计	52
6.3.1.	数据字典设计	52
6.3.2.	安全保密设计	54
6.3.3.	优化	54
第七章	系统测试	55
7.1.	测试概要	55
7.2.	测试环境	55
7.3.	测试结果	55
7.3.1.	文件上传	55
7.3.2.	容量信息显示	57
7.3.3.	缩略图显示	57
7.3.4.	保险箱	58
7.3.5.	回收站测试	59
7.3.6.	下载测试	59
7.4.	测试结论	60
第八章	关键实现技术	61
第九章	小结	62
9.1.	个人完成的程序模块和任务清单	62
9.2.	个人总结	62
第十章	致谢	66
	参考文献	67

第二章 概述

2.1. 项目背景

我们经常有这样的需求：通过网络交换公共数据文件以实现资源共享，同时保护私有数据不被非法访问，并使用简单、直观的方式操作。我们常用的文件传输方式如 FTP、Email 邮件、网上邻居等都能实现文件的传送。其中，“FTP”功能最为强大，但使用起来却稍显复杂，一大堆设置足以让许多人望而止步，尤其用户数量不可预见时，针对特殊需求用户的设置将更加繁琐；“Email”是大家所熟悉的了，但它的传送不仅需要连入 Internet，而且安全性也是个问题，在企业内部，财务、劳资等文件资料需通过这种方式交流不是好办法；“网上邻居”是又一种传送文件的方法，通过在本机上指定共享的 Web 文件夹并放置资源文件，一定范围内的用户可以访问到这些文件，然而这种方式使用的范围相当有限，通常在同一个 DNS 段地址内的用户才能顺利访问，其它尤其是 Internet 上的用户很难使用，此外同 Email 类似，它的使用也不直观，很多时候不得不在许多列表计算机上一层层展开搜索才有可能获取想要的资源。

由此看来，以上三种方式虽然在一定程度上实现了文件数据的交流，但它们都主要面向“点对点”的传送（你只能被动地等对方放置数据而不能主动“可视化”地索取），无法实现“一块空间，资源互见”的应用需求。

这种“点对多”的共享方式需要寻求另外的传输途径，网络硬盘就是一种很好的解决方式。

所谓“网络硬盘”，是将服务器上的磁盘空间提供给合法用户使用，这样用户就可以在互联网上直接实现文件的备份与转移。当然用户也可以通过客户端软件将这个磁盘空间虚拟成一个本地的目录，并完全集成到 Windows 的资源管理器中，在操作方法上就像使用本地硬盘一样方便。通过网络硬盘，不仅可以备份电脑系统中的文件，而且可以轻松实现移动化办公。用户可以随时随地地进行个人文件的上传、下载、删除以及文件目录的新建和修改，甚至还可以实现网络中文件的共享等等，也就是说，只要用户愿意，就可以与网络上的所有人共享存放在网络硬盘上的文件。不受时空的限制，随时随地存储自己的个人文件。

2.2. 项目介绍

根据人们的实际需求，我们提出了基于计算机网络的文件共享系统的设计方案，并根据该设计方案部分地实现了文件的存储、访问、备份、共享及检索系统的基本功能，完成一个功能较完善的基于 B/S 模式的网盘系统。

对我们开发者而言，这次项目旨在培养我们的动手能力和团队合作精神，希望通过这次项目实习，综合运用以前所学知识（算法与数据结构、数据库、计算机网络、JAVA WEB 开发等相关知识）的能力，学会软件开发工程文档的书写，提高动手的能力，而不只是停留在理论层面上。

本系统实现了以下功能：

（1）用户信息管理模块。用户可以通过注册免费获取一个自己的网络硬盘，

然后进行登录就可以进行相应的在给定权限内的操作，如修改密码、修改个人资料、安全退出等。还包括会员充值和管理，会员用户有更多的特权和福利。

(2) 文件管理模块。用户在网盘内可以分类浏览各种格式的文件，可以预览编辑图片、视频、音乐、PDF、Office 等类型的文件，可以进行文件/文件夹的创建、删除、重命名、移动，可以批量上传、下载文件，并且支持断点续传等高级功能。

(3) 文件分享模块。用户可以自由分享选定的文件或文件夹到公共空间、好友或群组，并且支持访问密码设置和取消分享功能。

(4) 回收站模块。用户删除文件到回收站后，可以进行文件的恢复与清空。

(5) 云笔记模块。用户可以在平台上创建笔记，并能在其他设备上同步浏览、编辑笔记，方便办公场景。

(6) 图床模块。为博客用户提供图床平台，用户可上传多种格式的图片，生成 URL 链接后方便引用。

第三章 系统需求分析

3.1. 任务概述

3.1.1. 运行环境

服务器操作系统：Ubuntu 16.04

数据库：MySQL 5.7

开发环境：JDK 1.8

Web 应用服务器：Tomcat 8

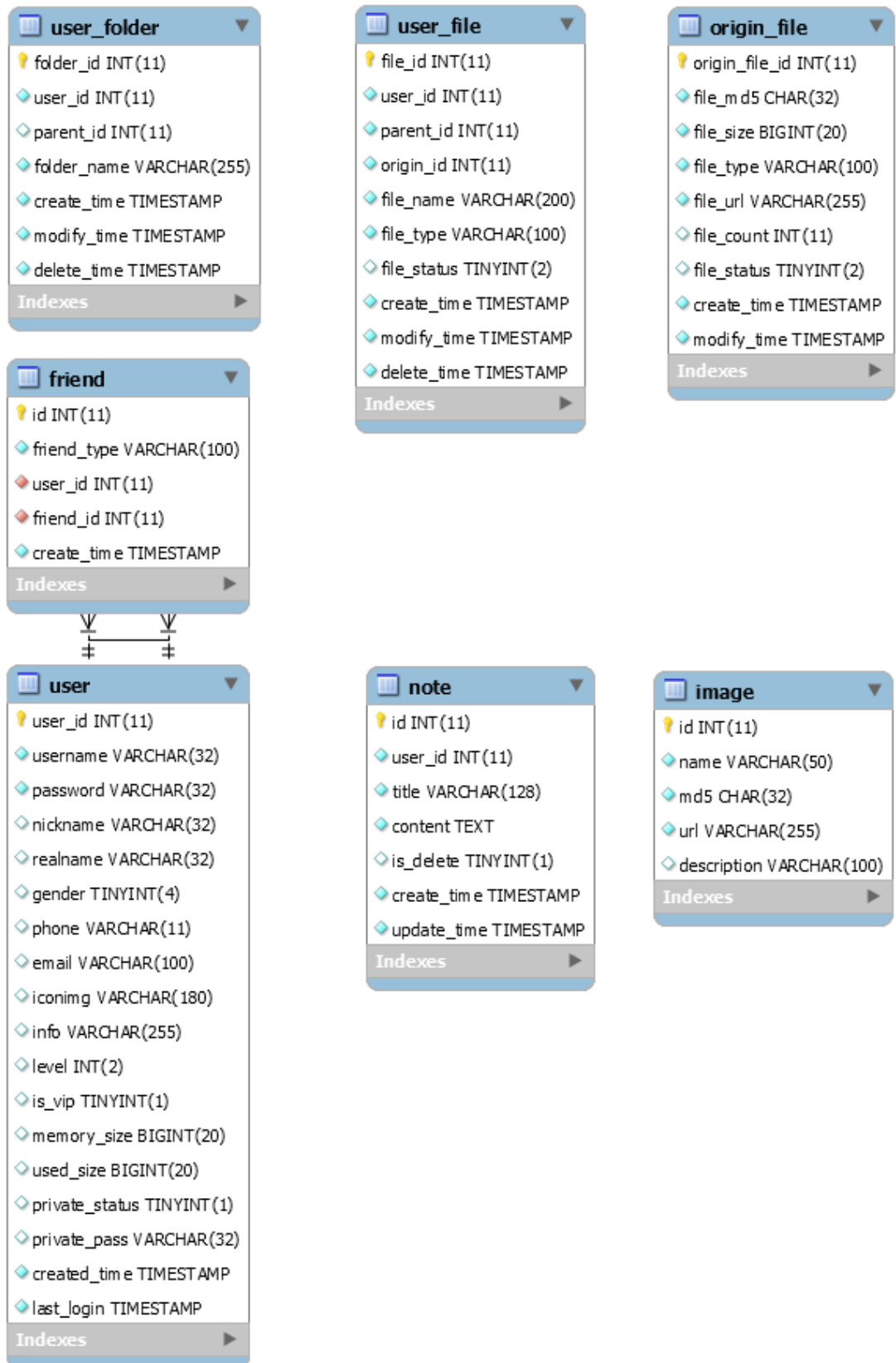
相关软件：Microsoft Visio 2016

3.2. 数据描述

3.2.1. E-R 图

数据对系统分析得到的数据字典中的数据存储进行分析，分析各数据存储之间的关系，可采用 E-R 图的方法进行数据结构分析。

主要实体及其属性如图所示。



3.2.2. 数据库介绍

见《数据库设计》部分。

3.2.3. 数据词典

见《数据库设计》部分。

3.2.4. 存储要求

本应用采用 MySQL 数据库进行数据存储, 存储引擎采用 InnoDB 引擎, 对于应对复杂的查询具有绝对优势, 但是负载能力一般, 为应对高负载的需求, 本系统采用以下策略进行数据存储:

- 数据库表格的设计上, 采用列维度和行纬度的拆分, 以此提高数据的存储效率;
- 采用读写分离策略提高性能, 读频率远大于写频率的数据单独存储;
- 对于系统中高并发、访问频繁的数据, 采用缓存数据库对 MySQL 进行隔离, 防止 MySQL 崩溃。
- 数据的备份采用单服务器集中备份的方法, 保证数据的高安全性和高可靠性。

3.3. 流程图

3.3.1. 业务流程图

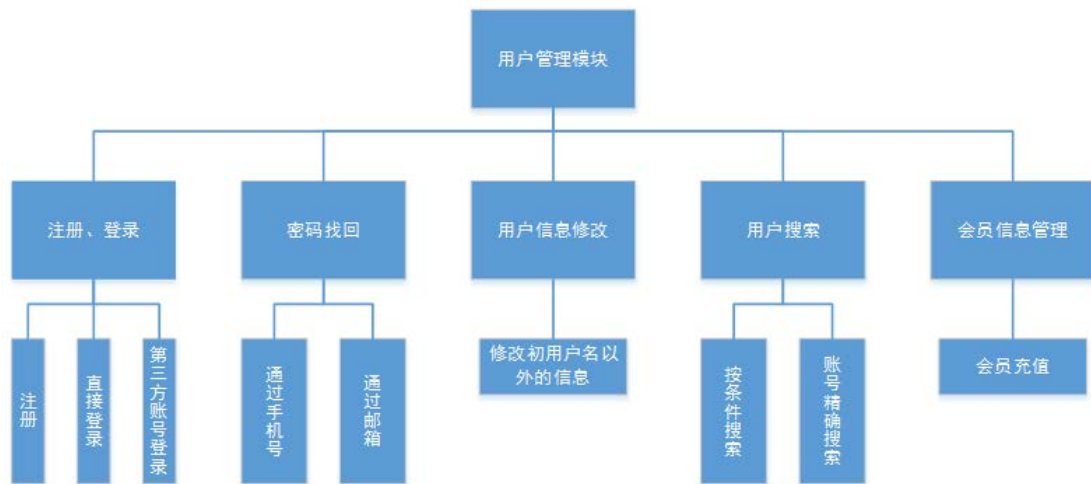
用户管理模块:

客户端用户信息模块包括用户的注册、登录及用户对个人信息的管理。

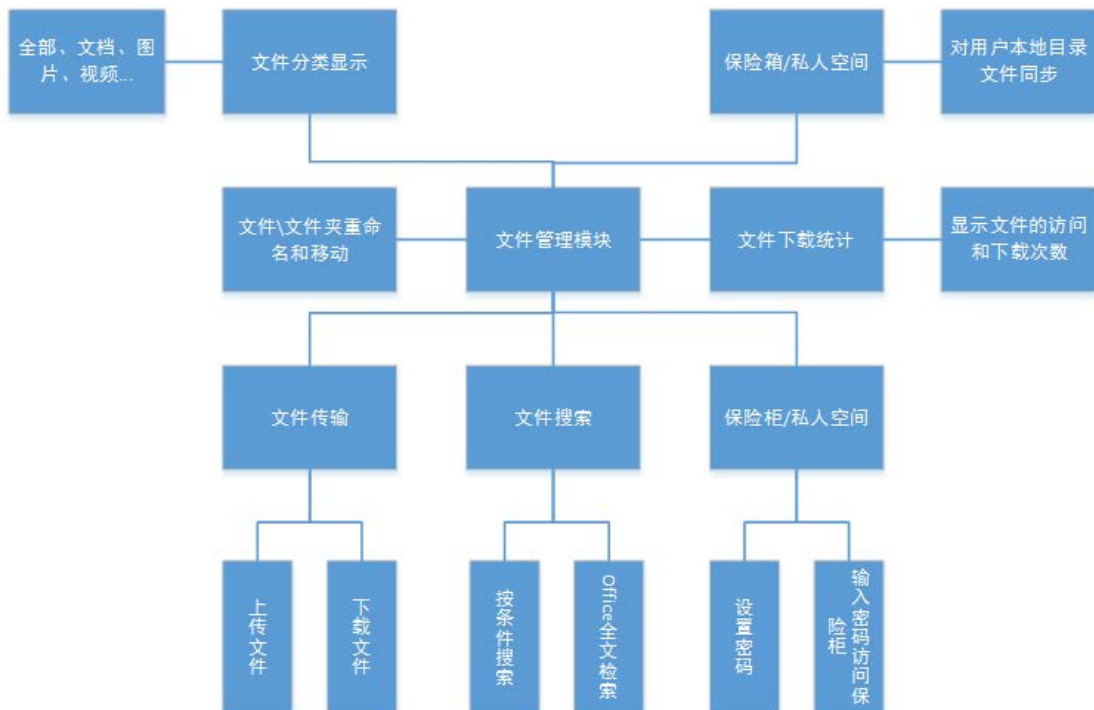
(1) 注册。通过客户端访问用户注册页面填写相关的个人用户信息获取网盘使用权限和个人网盘空间。

(2) 登录。通过登录对话框输入个人相关信息登录个人空间访问个人网盘资源。

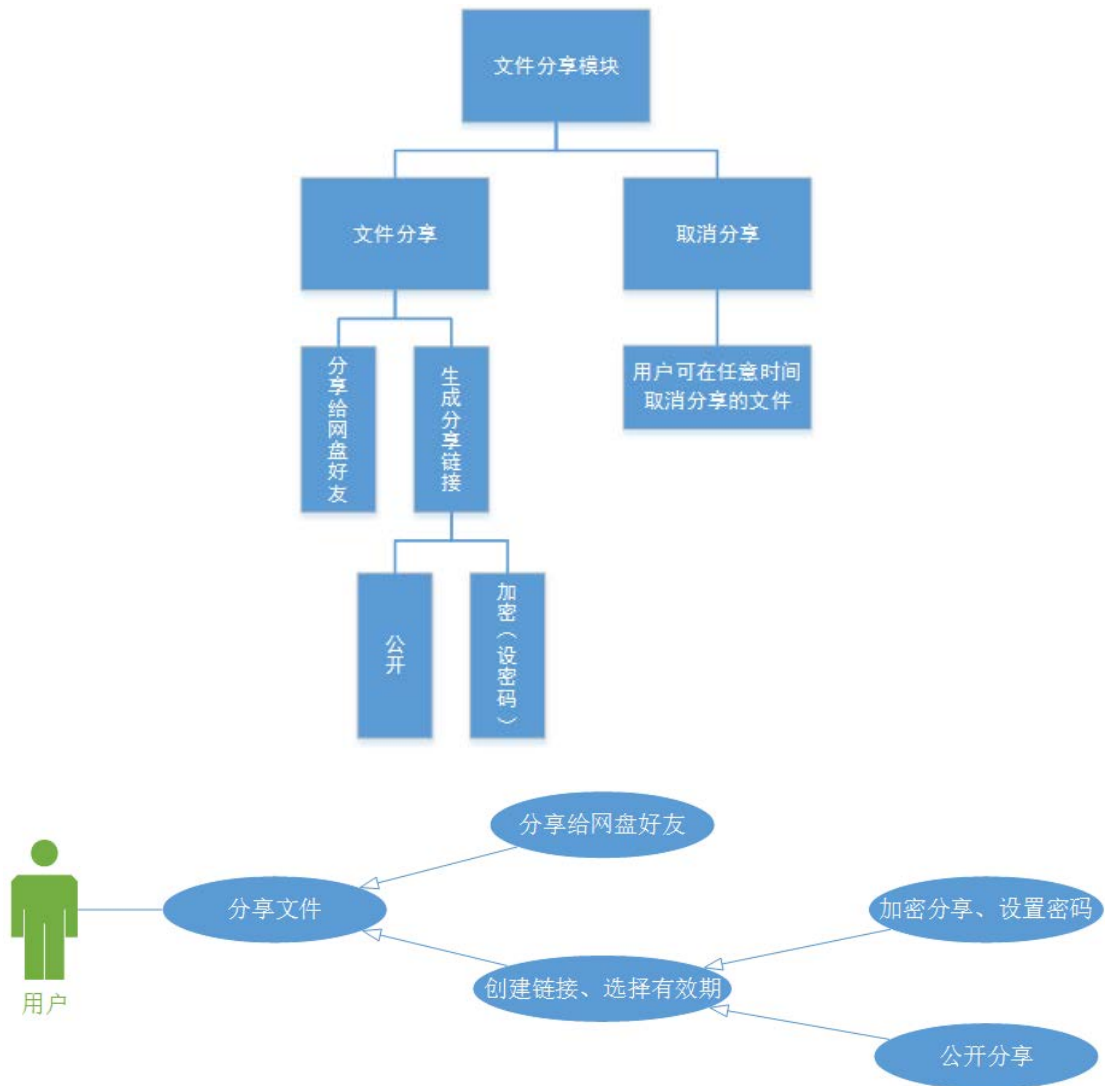
(3) 个人信息管理。可以更新个人用户的相关资料包括用户名、密码、个人介绍、Email 等个人相关的资料。增加视频/音频/图片等文件的管理。



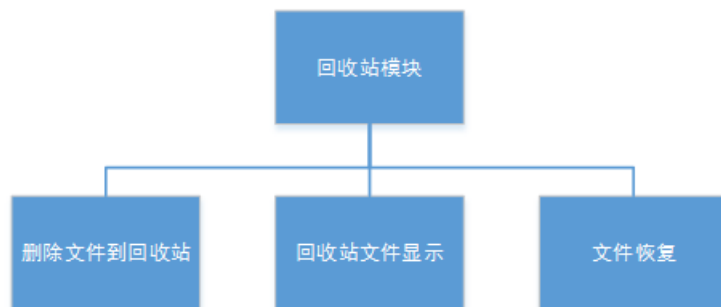
文件管理模块：



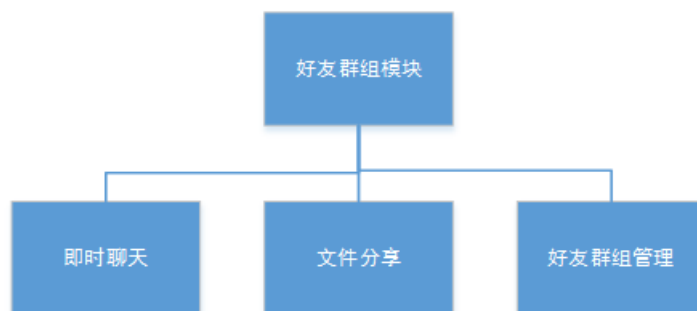
文件分享模块：



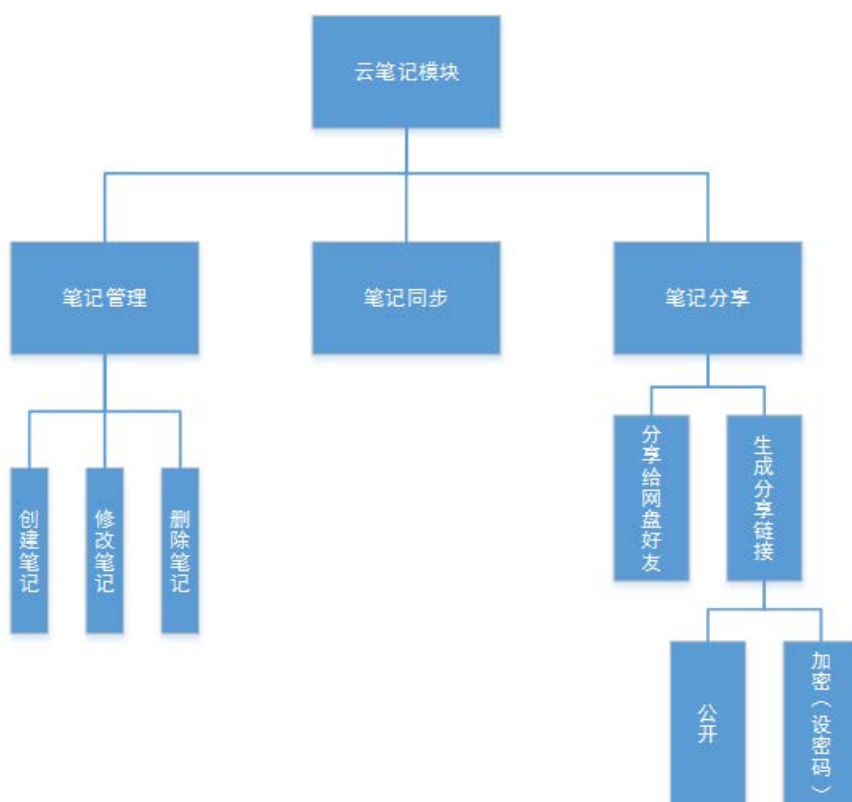
回收站模块：



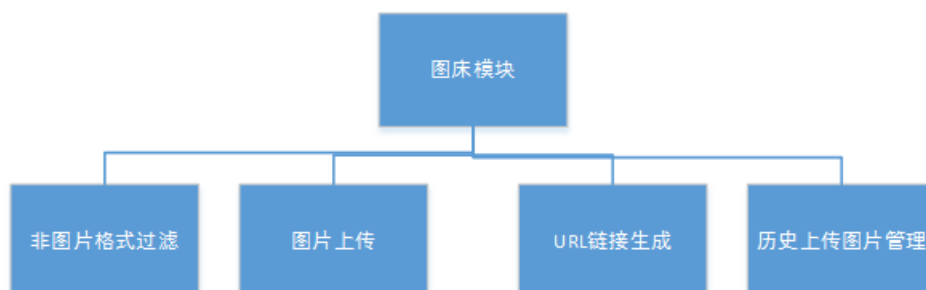
好友群组模块：

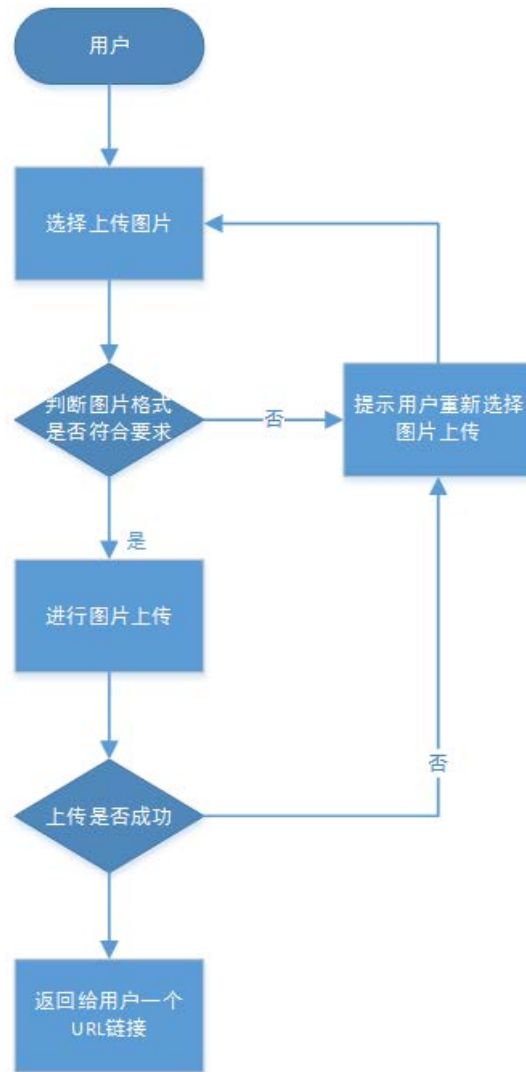


云笔记模块：



图床模块：

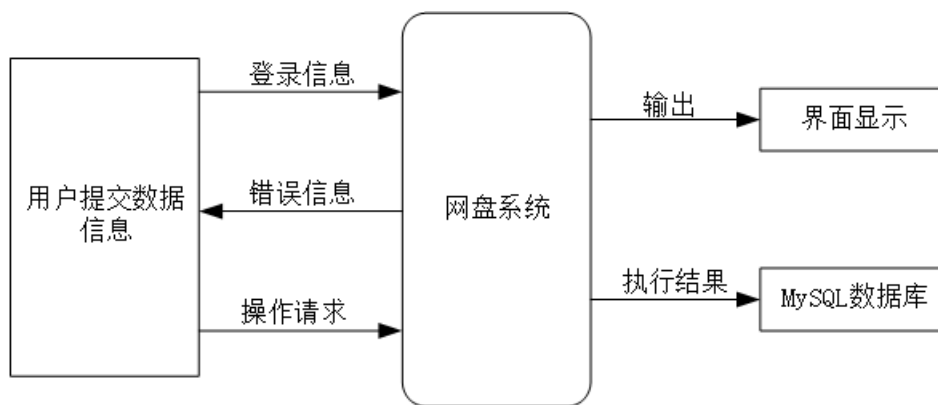




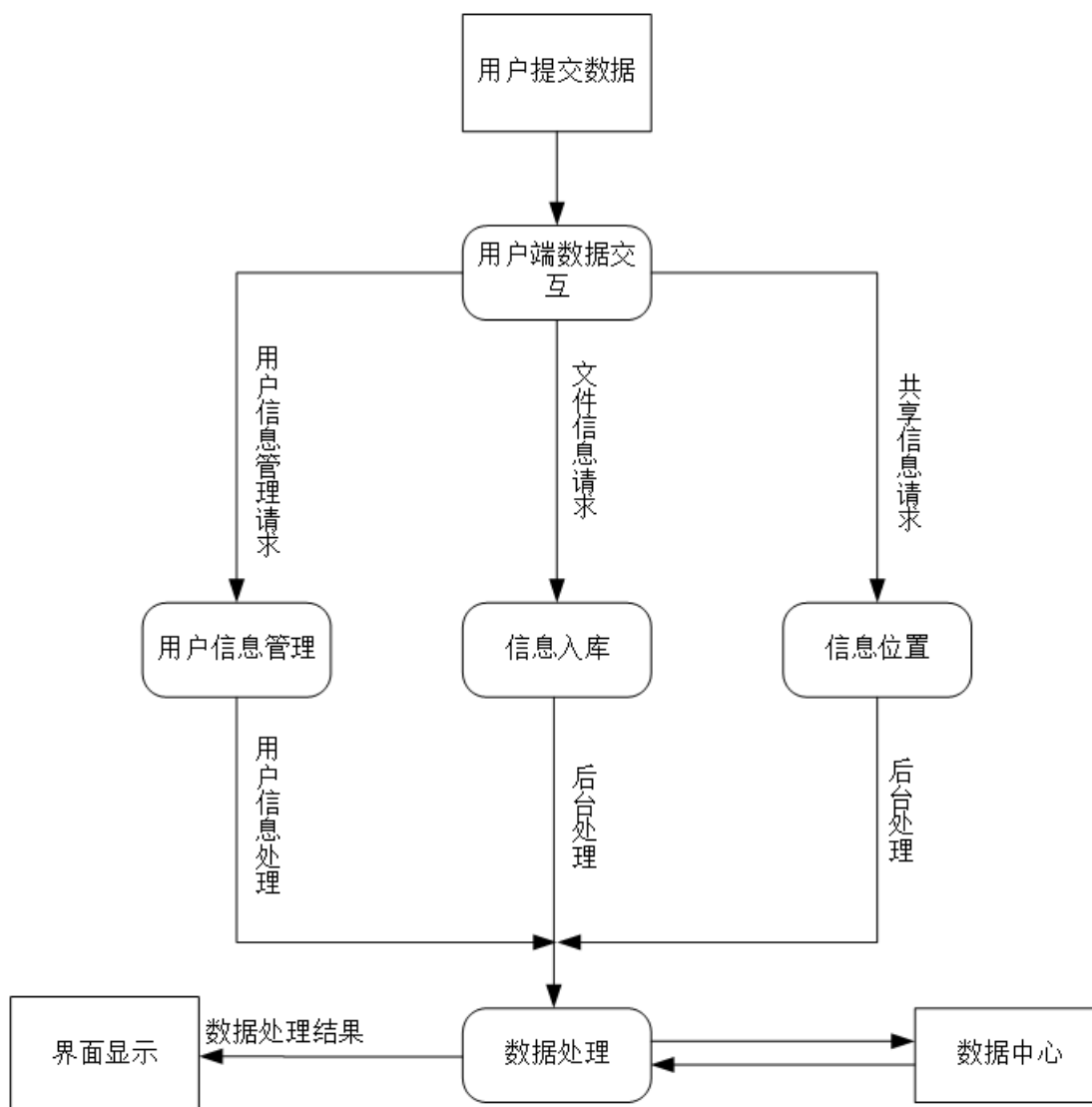
3.3.2. 数据流程图与过程流程图

数据流程分析主要包括对信息的流动、传递、处理、存储等的分析。数据流程分析的目的就是要发现和解决数据流通中的问题。现有的数据流程分析多是通过分层的数据流程图（data flow diagram，简称 DFD）来实现的。其具体的做法是：按业务流程图理出的业务流程顺序，将相应调查过程中所掌握的数据处理过程，绘制成一套完整的数据流程图。

根据上面的分析我们可以对系统画出如图所示的系统顶层数据流图。



如果上图是在顶层上的细分，从整体考虑网盘系统出现的数据流。那系统第1层数据流图展示如下。



3.4. 功能需求

3.4.1. 功能划分

产品主要划分为以下模块：

1. 用户管理模块
2. 文件管理模块
3. 文件分享模块
4. 回收站模块
5. 好友群组模块
6. 云笔记模块
7. 图床模块

3.4.2. 功能描述

用户管理模块

1、用户注册：用户注册和登录验证码，该功能为用户登入系统的必备条件，用户要想在该系统中有一个权限空间，那就先对本系统进行注册，当注册成功之后系统分配给该用户一个访问权限，使该用户可以在系统中完成普通用户的各项基本功能。

注册需要填写的信息有用户名、密码、昵称、真实姓名、性别、手机号码、邮箱地址、用户头像、个人简介。其中用户名和密码是必填，其他的可以为空。

2、忘记密码：根据手机号码或邮箱地址验证就能找回密码。

3、用户信息修改：除了用户名外，其他都可以修改。

4、用户搜索：多种条件搜索有关用户，可对其加入为网络通讯录

5、会员充值、会员信息管理：系统主要面向两类用户：即普通用户和会员用户，会员用户有更多的权限及福利。

文件管理模块

1、文件上传、文件下载

2、文件/文件夹重命名和移动

3、文件分类显示、排序显示：全部、文档、图片、视频、音乐、种子、其它

4、文件搜索：按多种条件进行文件搜索，可实现 OFFICE 全文检索

5、存储备份：对用户本地目录文件同步

6、文件下载统计。该功能的设定也是为了更好地管理自己的文件，可以轻松地知道自己的文件有多少人访问和下载

7、保险箱/私人空间：密码保护，在该目录下用户必须输入密码才能访问。

文件分享模块

1、文件分享：对云盘中已有文件（单个或多个）分享，可选择链接分享，也可选择直接分享给好友。

链接分享：由服务器生成一个 url 链接，加密分享同时生成一个密码（也可自定义），公开分享只生成一个链接，同时链接的访问支持二维码扫描，方便一些场景的使用。

好友分享：直接传送给选定好友或群组。

有效期：永久、一周、一天、自定义。

2、取消分享：用户可在任意时间取消对已经分享的文件的分享。

回收站模块

1、删除文件到回收站

2、回收站文件显示

3、文件恢复：对失误删除的目录或文件能够还原。

好友群组模块

1、即时聊天：采用即时通讯技术，好友间或群组内可以自由聊天

2、文件分享：好友间或群组内可以互相分享文件

3、好友群组管理：该功能提供了用户更灵活、更方便、更亲密的交流空间，其中用户可以添加系统中的用户为自己的好友，添加成功后在我的好友管理里面可看到对方在好友列表中显示；这给用户不受时间和空间的限制，随时随地的与好友分享自己的快乐。

云笔记模块

1、笔记创建、删除、修改

2、笔记同步

3、笔记分享

图床模块

1、非图片格式过滤

2、图片上传

3、url 链接生成

4、历史图片上传管理

3.5. 性能需求

3.5.1. 数据精确度

用户注册时间精确到秒；信息处理时间精确到秒；文件操作时间精确到秒。
价格单位保留到分。

3.5.2. 时间特性

响应时间

在网络状况良好的情况下,对于页面跳转的操作,响应时间应在1~2s之内,对于高清大图的加载,响应时间不应超过0.5s。涉及与第三方社交平台交互的部分,响应时间取决于与第三方平台的数据传输速度。

更新处理时间

数据库的更新涉及网络数据传输,取决于网络传输速度。但正常情况下,数据库更新时间不应超过1s,网络状况较差的情况下,不应超过4s。采用消息中间件,异步更新,不影响主进程界面响应。

数据转换与传输时间

首次打开网站时需要进行大量的数据加载,时间较长,但不超过3s。对于高并发、频繁访问的图片,浏览一次过后存入缓存数据库,以便再次浏览时迅速加载。

界面更新时间

1. 界面加载采用异步加载的方式,优先加载文字。文字更新时间不应超过0.1s,缩略图更新时间不应超过0.2s。
2. 高清图片加载采用先低清,再提升清晰度的方式。低像素图片加载不应超过0.5s,高清图片加载不应超过1s。

3.5.3. 适应性

- (1) 文件存储具有恢复功能;
- (2) 服务端能够支持 Windows XP 以上 Windows 平台和 Linux 等运行环境;
- (3) 支持 IE 等主流浏览器兼容;
- (4) 留有与其他系统的接口。

3.6. 运行需求

3.6.1. 用户界面

用户界面尽量采取简约风格,按 Web 应用软件用户界面规范设计。各个功能点易于识别,操作简便,方便用户使用。功能模块菜单使用左侧滑动抽屉组件。

具体界面设计应遵循:

- (1) 保持一致性,界面规范应遵循 Web 前端界面设计规范;
- (2) 设计完整的对话过程:系统的每一次对话都应该有明确的次序:开始、中间处理过程、结束;
- (3) 提供简单的错误处理机制,用户输错数据有提示信息,有较好容错性能;
- (4) 操作可逆:其动作可以是单个的操作,或者是一个相对独立的操作序列;
- (5) 显示启动画面:画面简洁明快,富有现代气息,不能太花哨;
- (6) 提供控制的内部轨迹:系统应该让用户觉得是由用户在做决定,可以通过提示字符和提示消息的方式使用户产生这种感觉

3.6.2. 硬件接口

考虑到大量数据的备份等要求，需要保持与服务器硬盘的接口，这较易实现。

3.6.3. 软件接口

本系统连接的外部组件有：

服务器平台 Java EE

服务器框架 Spring + Spring MVC + Mybatis

数据库 MySQL 5.7

系统要尽量采用 MVC 结构来构建，将模型、视图、控制分开，让系统更利于维护。使得各功能模块具有各自独立的功能，具有丰富实用的接口，松散地耦合成一个整体。

3.6.4. 故障处理

硬件故障：

情况 1：服务器电源不正当关闭

处理：a . 数据库事务回滚到断电之前的状态。

b. 为服务器提供一个备用充电电源能够支持服务器进行临时数据的正常保存。

情况 2：客户端与服务器网络不能连接

处理：a. 检修本地端网络连接是否正常；

b. 查看服务器端交换机端口是否被关闭；

c. 查看端点间网络线路是否完好。

情况 3：存储空间已满

处理：a. 数据库切换到备份存储空间；

b. 进行数据检索，删除日期超过 1 年的数据。

软件故障：

情况 1：服务器死机

处理：程序自动保存内存中的临时数据；重启机器

情况 2：服务端程序报告异常

处理：保存好临时数据，关闭访问端口，关闭程序重新启动，通过临时数据初始化程序

情况 3：服务器端响应速度慢

处理：查看接收的数据信息，看是否有不法分子攻击，并作好防火墙的配置

3.7. 其它需求

3.7.1. 通信接口需求

系统的服务器端和客户端应使用 HTTP/TCP 协议进行通信，消息传输交换格式使用 JSON。

由于系统应用涉及到用户资料、聊天记录等私密信息，必须要对通信数据进行加密，不应使入侵者通过简单的截取数据报就可以获得情报信息。

3.7.2. 安全保密需求

防止网络方面的攻击。本系统应当具备防治黑客入侵随意篡改用户已经生成好的线路的技术，尤其是系统内的一些核心数据，应采取数据加密存储技术，防治不法分子盗用数据。

数据库

1. 通过网络防火墙将数据库系统与外界隔离，保证其安全稳定地运行。
2. 多机备份，减少信息丢失可能性。
3. 用户密码等关键隐私信息经过 MD5 加密，防止拖库等数据库攻击。
4. 对数据库的访问采用权限授予机制，在用户表中表示用户的权限级别，只有具备相应权限的用户具有操作权限，控制部分关键信息的增删操作。

软件使用数据

本应用存储的数据涉及用户上传的文件、专辑、分享、小组信息、用户的偏好信息等。一旦丢失将导致用户个人上传记录、收藏历史的遗失，推荐算法无法根据历史记录为用户定向推送，导致用户体验感急剧下降。因此本应用对于数据的安全性要求较高，要尽量保证用户的数据不会丢失。

1. 用户上传的文件采用多重备份的方式存储。采用分布式存储的方式，图片信息及其副本存放在不同的机器上，即使一台机器宕机，仍能保证数据的安全存储，防止由于掉电、机械故障等物理原因导致的数据丢失。

2. 用户的偏好信息由上传记录、分享记录、收藏记录、小组及好友记录构成，为了保证数据的新鲜度，本应用对用户近 30 次的使用记录进行存储。由于特殊原因造成的个别记录丢失，并不影响系统按照用户偏好记录进行推送。

用户账户数据

1. 应用仅对用户在本系统的用户名、密码等进行保存，个人信息仅与本网站功能相关，不会对用户隐私造成任何不利影响。

2. 对于用户在本系统中对于第三方社交平台的授权信息，系统将进行加密存储，严格防止用户第三方社交平台的信息泄漏。应用不会利用用户第三方社交平台进行用户指定的分享之外的任何操作。

3.7.3. 可使用性需求

软件功能设计合理，易于操作使用，用户可快速掌握软件操作。软件应正常完成所有功能需求。服务器应长时间运行无故障，故障修复时间一般控制在 3 小时内，除非出现恶性攻击、病毒破坏、系统资源不足、硬件故障、操作系统崩溃网络中断等情况，不允许出现内存泄露。

3.7.4. 可维护性需求

本系统将采用严格的软件工程的规范进行开发，并采用良好的设计模式保证系统各模块之间的低耦合及模块之间的高内聚。

本系统的所有代码将会被详细注释，对于系统所有代码，我们会生成详尽的技术文档。对于系统开发过程可能出现的报错，我们将以文档的方式详细罗列报错码及对应的报错信息。

3.7.5. 可移植性需求

系统建成后，应在现行系统上不需要做大的改动或不影响整个系统结构，就可以增加功能模块，改善软件现有功能，这就必须在系统设计时留有接口，使其具有可扩展性和维护性，这样就方便在后期的维护过程中根据用户的需求添加相应的功能，同时也不会影响系统其他功能模块的正常运行。

3.7.6. 灵活性需求

1. 采用测试驱动开发的开发模式（TDD），尽可能保证开发高效率和高准确度的完成代码编写。
2. 部署 Code Review 环节，尽可能提高每一次代码编写质量，提高工作效率。
3. 采用 Git 版本控制系统进行版本控制。
4. 软件架构采用 MVC 模式，使得界面和逻辑充分脱耦，提高灵活度。

3.7.7. 其他专门要求

灾难恢复

由于本应用为服务器—客户端模式的网络应用，用户数据保存在服务器端，因此服务端负责数据备份并保证数据的安全性。数据有备份存储，即使发生灾难造成数据丢失，系统立即通过备份数据进行恢复，并且重新备份，保证系统中始终保持至少一份数据备份。

法律限制

1. 版权限制

本网站致力于打造开放的高质量文件云存储分享系统，不涉及任何盈利目的。我们鼓励用户充分利用本应用自由地张贴和共享自己的信息。用户可以自由上传、分享文件，但这些内容必须位于公共领域内，或者用户本人拥有这些内容的使用权。同时，用户不应在自己的个人主页或社区中张贴其他受版权保护的内容。我们如果收到按下述程序提起的正式版权投诉，将会删除这些内容。用户对于其创作并在本应用平台上发布的合法内容依法享有著作权及其相关权利。

2. 合法内容限制

用户在使用本应用提供的服务过程中，必须遵循国家的相关法律法规，不得利用本平台，发布危害国家安全、色情、暴力、凶杀、恐怖等非法内容；不得利用本平台发布含有虚假、有害、胁迫、侵害他人隐私、骚扰、侵害、中伤、粗俗、或其它道德上令人反感的内容。参见相关法规。

用户使用本服务的行为若有任何违反上述条款的情形时，本应用有权直接删除该等违反条款之内容。用户影响系统总体稳定性或完整性的操作可能会被暂停或终止，直到问题得到解决。

3. 责任限制

互联网是一个开放平台，用户将图片等资料上传到互联网上，有可能会被其他组织或个人复制、转载、擅改或做其它非法用途，用户必须充分意识此类风险的存在。用户明确同意其使用本应用服务所存在的风险将完全由其自己承担；因其使用本应用服务而产生的一切后果也由其自己承担，本应用对用户不承担任何责任。

第四章 系统概要设计

4.1. 总体设计

4.1.1. 需求规定

详见《系统需求分析》部分。

4.1.2. 运行环境

硬件环境

要求操作系统为 Linux (Ubuntu), CPU 主频 2.5GHz, 内存 1GB, 带宽 1M。
系统运行开发所使用的服务器如下表所示:

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: JDK 1.8 或更高版本
Apache Tomcat 8 及以上版本
安装有 MongoDB, Redis, RabbitMQ 等应用。

4.1.3. 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。

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。

4.1.4. 结构

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

4.1.5. 人工处理过程

暂无人工处理过程。

4.1.6. 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，增强系统的异步性；
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。

4.2. 接口设计

4.2.1. 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。

4.2.2. 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。

4.2.3. 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。

4.3. 运行设计

4.3.1. 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;

4.3.2. 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；

4.3.3. 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。

4.4. 系统数据结构设计

4.4.1. 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。

4.4.2. 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录后操作，系统安全保密性较高。

4.4.3. 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。

4.5. 系统出错处理设计

4.5.1. 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。

4.5.2. 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

4.5.3. 系统维护设计

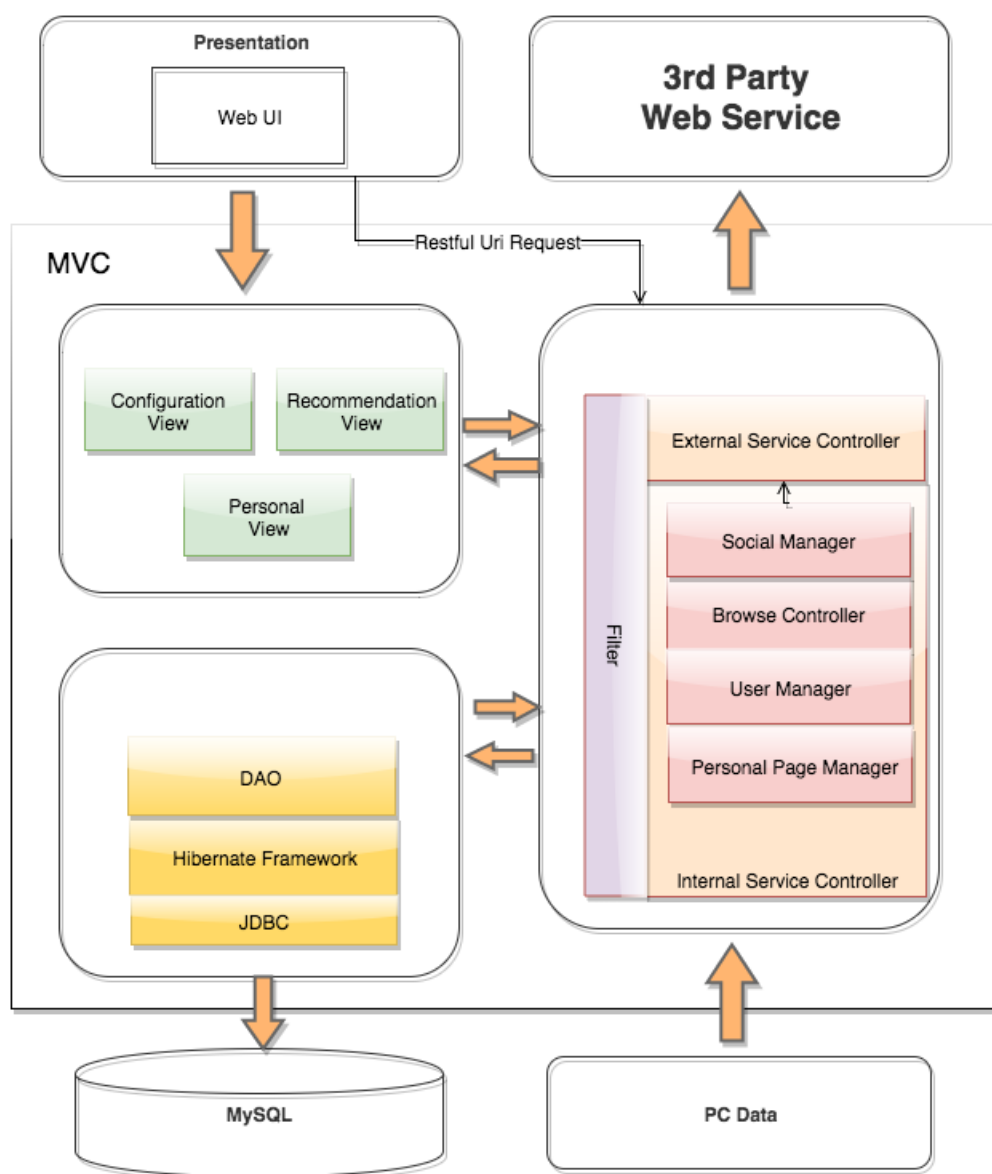
1. 本系统采用日志框架记录程序运行情况和崩溃信息，开发人员可以第一

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。

第五章 系统详细设计

5.1. 程序系统的结构



本系统由服务器端和客户端两部分组成。服务器端设计以实现各模块间的低耦合及模块内部的高内聚为导向，采用 MVC 作为总体架构；客户端向服务端请求采用 RESTful 设计风格，力求访问的灵活性；服务器端访问 MySQL 数据库进行数据交互；同时在系统层面设计了安全保障模块、缓存模块、日志和验证模块，为系统平台的安全和高效保驾护航。

数据模块

本系统作为基于 Web 的云存储平台，对于大量的用户信息、文件数据、小组信息、用户访问记录等数据的大量存储和快速访问有着较高的要求。本系统使用

MySQL 5.7 关系型数据库存储信息，通过 JDBC 完成对数据库的访问。MySQL 数据库作为一个快速、多线程编程的数据库，不仅满足本系统对于访问速度的要求，而且具有强大的查询功能，可以在同一查询中混用来自不同数据库的表，查询方便快捷。本系统支持用户上传本地图片，支持 PNG, BMP, JPEG, JPG, GIF 等常见图片格式，上传之后存入 MySQL 数据库。

模型层

模型层负责处理本系统中应用程序的数据逻辑部分，对数据对象进行存取操作。本模块通过 JDBC 访问 MySQL 数据库，采用 Mybatis 框架自动完成数据库表到 Java 类的映射。为了构建一个健壮的 J2EE 应用，本模块采用 DAO 作为接口，统一管理上层逻辑对于数据源的访问操作，使得上层逻辑无需关心数据访问的细节，同时扩展了系统的可维护性、可拓展性。

控制器层

控制器层是本系统处理用户请求的核心层，负责对用户的 HTTP 请求做出响应，处理本系统的核心业务逻辑，转发用户请求。控制器层通过 Filter 过滤来自用户的请求，检查用户请求的数据格式及用户当前的登录状态，并根据判断结果对请求进行转发。用户请求被转发到内部服务、外部服务两大模块，外部服务负责获取与第三方平台的接口，并将用户图片分享到第三方社交平台；内部服务分为用户管理、文件管理、文件分享管理、回收站管理、图床、好友群组六大部分，是本系统的核心功能逻辑。

用户管理模块：负责用户个人信息的管理

文件管理模块：负责文件上传、下载、移动、重命名等网盘文件管理功能

文件分享模块：负责文件分享相关的功能

回收站模块：负责回收站清空、回复等功能

图床模块：负责图片的上传、链接分配及上传历史记录管理等功能

好友群组模块：负责好友间和群组内的即时聊天、分享等社交功能

视图层

视图层是本系统的数据显示部分，是与用户交互的接口。负责系统的前端业务逻辑，支持 CSS, JavaScript, JSP 文件，将用户请求转发给 Controller 层，并显示经过 Controller 处理的业务。

5.2. 登陆验证模块设计说明

5.2.1. 程序描述

对于一个完整的在线网盘管理平台，它不仅要求只是一个简单和强大的操作，有一个通用的简洁美观的操作界面，良好的设计等风格。只有输入正确的密码，防止恶意用户和非授权用户访问系统。

本系统的所有功能模块都必须调用“登录验证”子模块进行身份验证，此模块是所有用户进入系统时都必须调用的模块，系统用户登录界面中的【登录】按钮触发。

进行登录界面时，需要输入账号、密码、验证码等信息均正确的情况下方可进入。

否则，界面会提示输入用户名或密码不正确，需要重新输入。

验证码是从 0-9 数字与 a-z 大小写字母组成，随机生成验证码图片，并保存到服务器会话对象中，因此验证码有时间期限。

5.2.2. 功能

登录验证功能模块的 IPO 表如下表所示：

系统名称	基于 WEB 的网盘系统	
模块名称	登录验证	
模块简述	用于对登陆系统的用户进行身份验证	
调用模块	系统的主控制模块	
输入	项目	用户账号，密码，验证码
	格式	用户账号：用户注册设置 密码： 用户注册设置 验证码：随机产生
处理	验证用户输入的验证账号、密码和验证码； 获得登录用户输入的用户账号、密码和验证码； 访问数据库的用户账号表； 查询匹配的用户名和密码，若未查询到或不匹配，返回提示错误，相反登录成功	
输出	提示信息：有效用户或无效用户	
注释	涉及的表：用户表	

5.2.3. 算法

【登录】按钮触发的处理：验证用户的合法性。

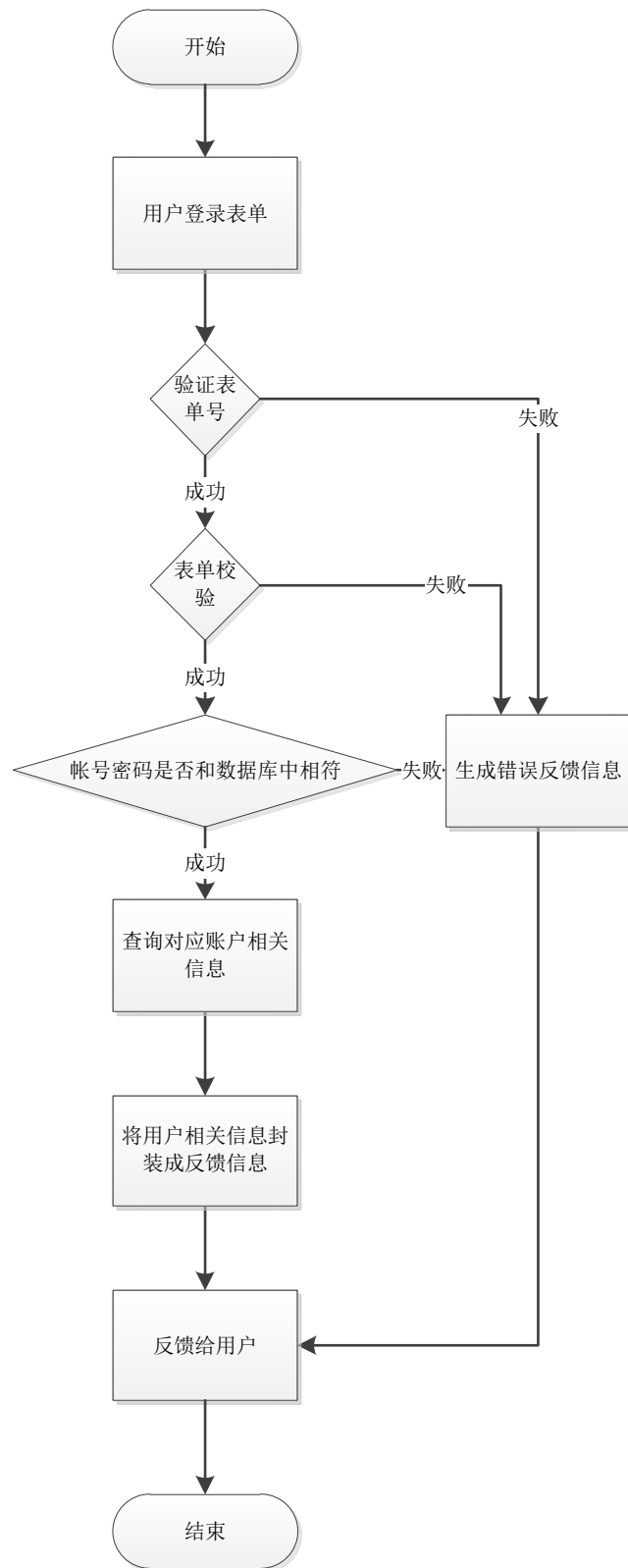
处理流程：

取得用户输入的用户名、密码和验证码；

用户名和加密密码传输到数据库并与用户表进行一致性验证；

如果用户名、密码和验证码都正确，以该用户身份进入系统总控制界面并获得相应系统权限，否则根据某项错误提示对应的错误信息，如：不存在该账号、密码错误、验证码错误、账号不能为空。

5.2.4. 流程逻辑



5.2.5. 接口

本模块是系统的启动模块，它需要用户交互输入用户名、密码；它调用主控制模块。

5.2.6. 限制条件

预先设置一个普通用户和一个会员用户。

5.2.7. 测试计划

- (1) 初次安装系统后，测试初始用户帐号和密码能否正确登录到系统。
- (2) 正确登录后，退出注册一个新的账号。
- (3) 采用刚注册的用户账号和密码登录，应该能够正确登录。
- (4) 将某一项不填看是否提示错误信息。
- (5) 将某一项填错查看错误信息。

5.3. 用户注册模块设计说明

5.3.1. 程序描述

在使用本在线网盘系统之前，用户事先注册并且登录才能实现文件存储以及共享功能。进入注册界面后，需要输入相应的信息，并且这些信息符合规范才可以注册成功。将注册成功后的信息加入到对应的数据库表中。

点击注册，则注册完成。

用户在注册成功之后，其具体信息会显示在用户的个人信息页面上。

在注册过程中，对新用户的创建主要分成两部分，第一是对新用户用户名是否与已存在用户名重复，新用户可以点击查看是否重名来进行确认，系统会对新用户名与已有的用户名进行比较；第二是注册窗口显示输入两次密码进行密码验证，以确保用户密码的准确性，如果两次密码不一致，会弹出错误信息，提示用户重新输入。

5.3.2. 功能

注册模块的 IPO 表如下表所示：

系统名称	基于 WEB 的网盘系统
模块名称	注册模块
模块简述	用于用户注册
调用模块	注册成功界面
输入	输入用户名、密码、联系方式、个人介绍等基本的个人信息

	息
处理	将对应的信息进行判断是否符合规范，当所有的信息符合规范后，将信息存储到相应的表中。
输出	注册成功
注释	涉及的表：用户表

5.3.3. 算法

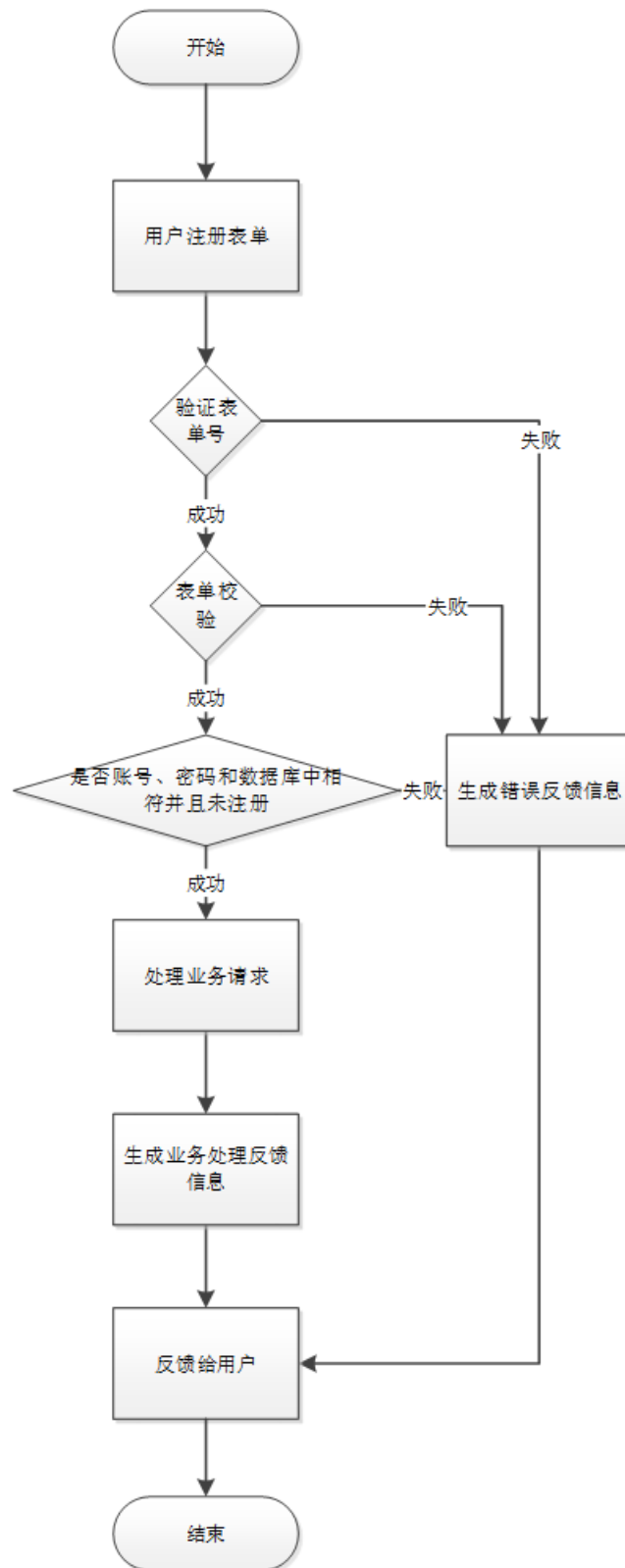
点击注册按钮，检测输入的合法性。

处理流程：

- （1）输入注册界面的所有内容。
- （2）点击注册，检测输入的合法性。
- （3）如果不合法提示错误信息，如果合法则显示注册成功。

注册时，密码采用 MD5 进行加密，将加密后的密码保存到相应表的数据库中，以增加账户的安全性。

5.3.4. 流程逻辑



5.3.5. 接口

当所有信息填写均符合规范，并点击注册时，所有相应信息会保存到相应数据表中。并提示注册成功。

5.3.6. 限制条件

只有所有信息都符合限制规范才可以注册成功。

5.3.7. 测试计划

- (1) 进行多次注册尝试，经注册后的信息进行登录，看是否会提示错误。
- (2) 输入不符合规范的信息，看是否会提示相应错误。

5.4. 用户信息管理模块设计说明

5.4.1. 程序描述

系统对用户进行分级，分为普通用户和会员用户，并对其进行权限的授予。
普通会员用户：普通会员用户可以对大部分文件列表进行浏览、上传文件、下载文件和进行用户之间点对点的文件分享的权限；会员用户拥有更大的权限和福利，在用户信息页面有区分。

在此模块，用户可以修改自己的个人信息，如真实姓名、手机号、邮箱、个人介绍。除此以外，还可以修改用户密码。

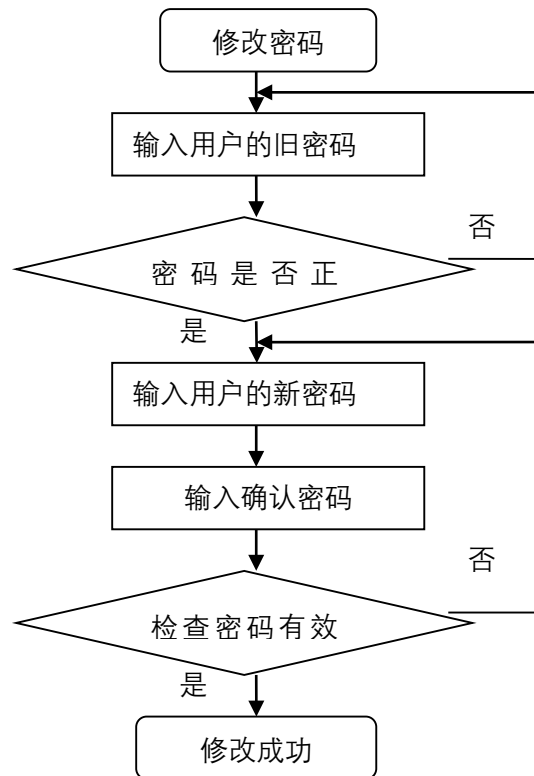
5.4.2. 功能

用户信息管理模块的 IPO 表如下表所示：

系统名称	基于 WEB 的网盘系统
模块名称	用户信息管理
模块简述	用户对个人信息的修改
调用模块	用户登录模块
输入	触发界面相应按钮进行对用户个人资料进行填写或修改
处理	根据用户 ID 查询相应的用户数据
输出	用户个人信息
注释	涉及的表：用户表

5.4.3. 流程逻辑

用户登录到主页面后可以密码修改的功能，在检验到用户输入的旧密码正确后，用户就可以输入新密码，进行修改，然后保存成功。



5.4.4. 限制条件

用户填写的用户信息表单每项应符合长度大小等规定，并做正则表达式的过滤，不符合的表单项将被提示无效。

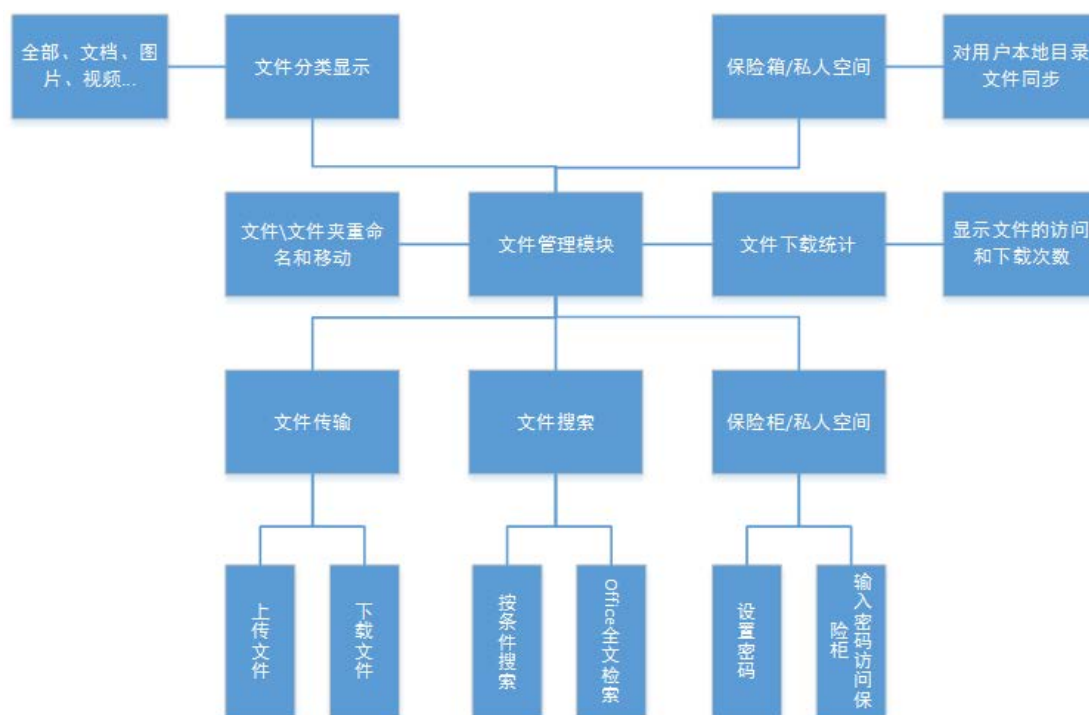
5.5. 文件管理模块设计说明

5.5.1. 程序描述

该模块是本系统的核心部分，用户可以在此模块对网盘内文件及文件夹进行各种操作，包括文件的查看、预览、编辑、上传下载、删除、移动、重命名、断点续传、搜索等功能。

5.5.2. 功能

- 1、文件上传：用户拥有初始存储空间 10GB，在限制范围内可以任意上传任意格式的文件。而会员用户拥有更大的特权，相对普通用户而言拥有更快的上传速度，一次能传更大的文件和更大的存储空间的权限。
- 2、文件下载：用户可以批量下载网盘内的文件及文件夹。系统对普通用户和会员用户一次在单位时间内能下载多少存储量的文件、一次性能下载多大存储量的文件和根据用户的权限等级能搜索浏览并下载的文件进行控制。会员用户相对普通用户而言拥有更快的下载速度，一次能载更大的文件和更多的文件下载权限。
- 3、文件/文件夹重命名和移动：用户可以批量重命名和移动文件及文件夹。
- 4、删除文件：用户可以任意删除不需要的文件到回收站中。会员用户在回收站中的文件相对普通用户的文件拥有更长的保留时间。
- 5、文件分类显示、排序显示：全部、文档、图片、视频、音乐、种子、其它。
- 6、文件搜索：按多种条件进行文件搜索，可实现 OFFICE 全文检索。
- 7、存储备份：对用户本地目录文件同步。
- 8、文件下载统计：该功能的设定也是为了更好地管理自己的文件，可以轻松地了解自己的文件有多少人访问和下载。
- 9、保险箱/私人空间：密码保护，在该目录下用户必须输入密码才能访问。
- 10、断点续传：用户可以任意暂停正在上传的文件，等需要时继续下载未完成的人而不用重新下载。



5.5.3. 算法

【文件列表】用户登录成功后页面自动跳转到用户文件列表页面，即“我

的文件”，显示用户已经保存的文件。实现过程是：登录成功转发请求到用户列表处理函数，处理函数根据拦截器设置的User对象查询数据库获得用户的所有文件，将查询结果封装成系统定义的文件类型，格式化User对象使它便于页面展示，最后将文件列表对象和格式化的User对象传到前端页面展示结果。

【文件下载】用户在文件列表页面点击下载文件时，进入服务端后台，首先需要校验用户是否有权限下载文件，如果用户是管理员或者文件是公共文件则直接判断有权限。如果是普通用户则查询数据库获得用户所拥有的文件列表。匹配是否前端传来的文件id在用户拥有的文件列表中。如果存在则用户有权限，继续后面的操作，没有则转到错误页面并提示错误信息。下载文件时读取保存在服务器的文件。形成字节流写入HTTP应答，并设置HTTP的相关Header指示浏览器对字节流的处理。

【文件上传】用户选择文件后上传文件，页面将文件的数据POST到后台处理函数，处理函数接收一个文件类型的参数，此时页面传来的文件数据不是处理函数接受的类型。因此自定义的参数解析器发挥作用。将页面传来的文件数据写到服务器本地，服务器处理函数接收参数，首先检查云盘存储空间是否足够，如果不足，前端将进行相应提示，上传失败。否则，获取文件参数生成一个自定义的文件对象，再作为被调用的Service的参数，将数据插入到数据库。至此文件上传完成。系统对上传文件的限制写在配置文件里面，主要是最大文件大小和最大在内存中的大小设置。

【文件删除】用户删除文件是同样首先进行权限的验证。用户只能删除属于自己的文件。权限验证完毕，根据前端传来的文件id删除数据库信息及删除服务器硬盘上保存的文件。如果是批量删除文件及文件夹，则获取要删除的ID列表相应地从数据库中batch delete相应数据。

【文件重命名】前端获取需要重命名文件ID和对应的新名称，再在数据库中更新源文件的名称。同时，前端将立即AJAX更新目录列表，保证数据的即时性。

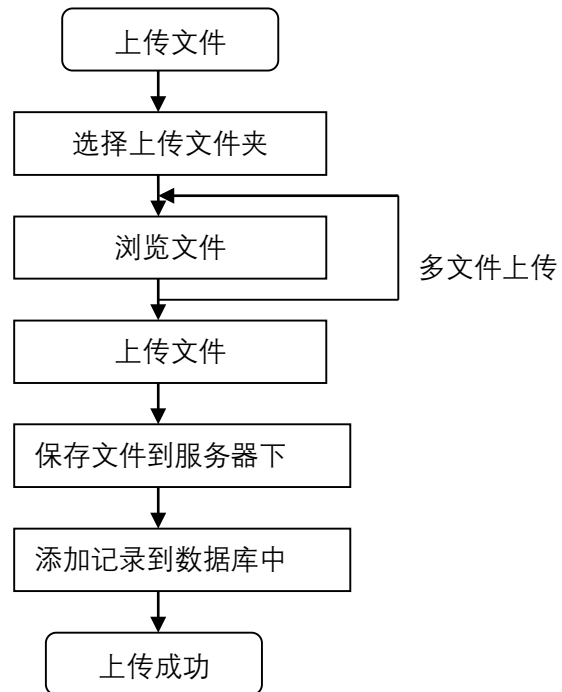
【文件移动】前端获取需要移动的源文件ID列表和目的文件夹ID，再在数据库中更新源文件列表所属的父文件夹ID。同时，前端将立即AJAX更新目录列表，保证数据的即时性。

【断点上传】用户准备上传文件时，向服务器传递文件名称、大小等信息，开始上传后，前端界面接受服务端上传进度信息并同步显示。服务器根据文件大小对剩余文件进行分块，传递剩余文件第一块直至完成剩余文件的传输、最后结束。

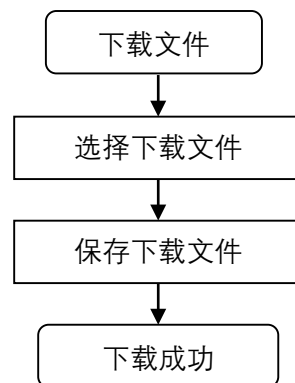
5.5.4. 流程逻辑

(1) 上传文件流程图如下图所示：

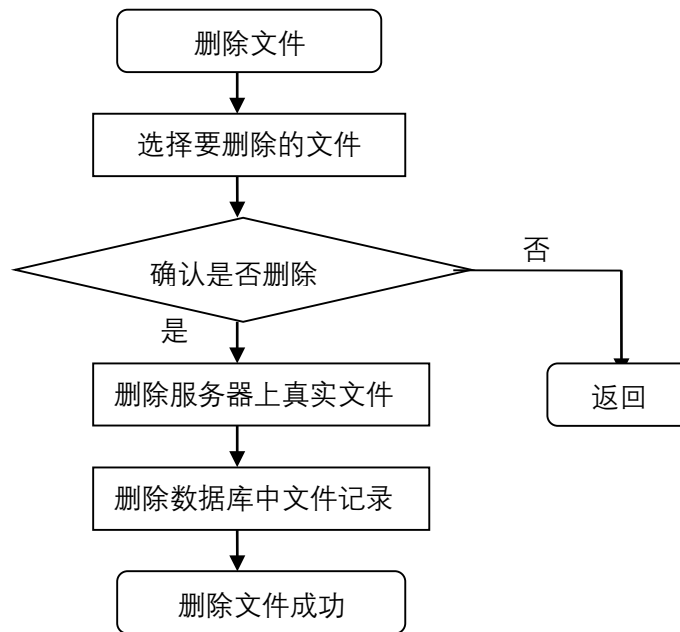
用户上传文件时，用户先选择上传到哪个文件夹，然后点击浏览文件，选择上传文件，顺利上传，当然在上传的过程中，可以点击取消，放弃正在上传的文件。而且还可以进行多文件上传。



(2) 下载文件流程图如下图所示：
选中下载文件，点击下载，保存下载文件到选中的路径，即可进行下载。

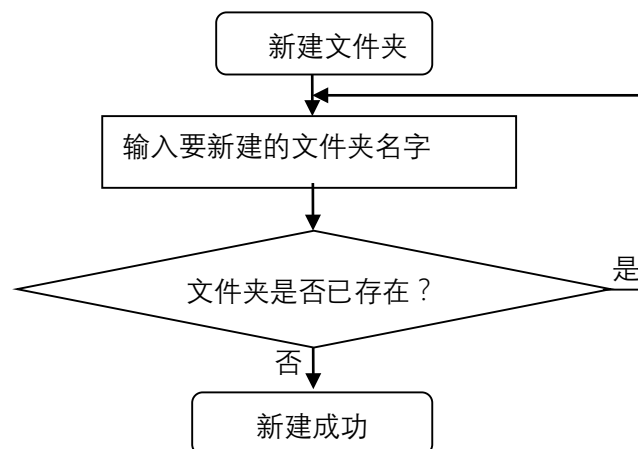


(3) 删除文件流程图如下图所示：
用户选择要删除的文件，确认删除后进行删除。



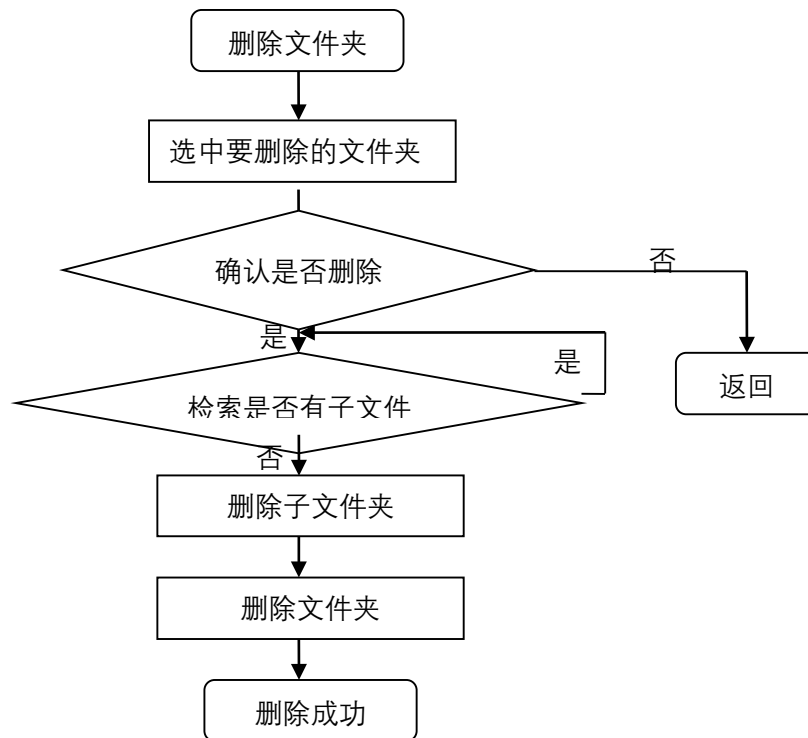
(4) 新建文件夹流程图如下所示：

选中在哪个文件夹下新建，然后输入要新建文件夹的名字，如果文件夹不存在，则新建成功，如果存在则提示文件夹已经存在。删除文件夹时，默认删除文件夹下的所有文件。



(4) 删除文件夹流程图如下图所示：

选中要删除的文件夹，点击按钮，查询该文件夹如果有子文件夹则提示用户是否确认删除子文件夹，如果确认则一并进行删除并删除文件夹下的所有文件。



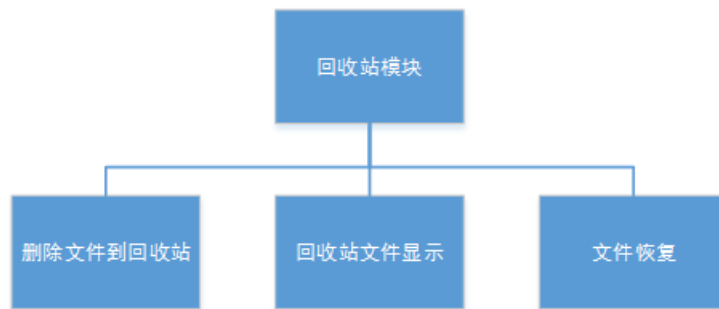
5.6. 回收站模块设计说明

5.6.1. 程序描述

当用户删除选定的网盘内文件后，实际上并未在服务器删除，服务器仍存有文件的备份，并且被删除文件被转移到回收站中，如果用户因为误操作将文件删除，可以通过回收站功能恢复。当文件被添加到回收站后，有效期为 30 天，若超过 30 天，回收站内文件将被清空无法恢复。而会员用户拥有半年的有效期。

5.6.2. 功能

- 1、删除文件到回收站
- 2、回收站文件显示
- 3、文件恢复：对失误删除的目录或文件能够还原
- 4、回收站清空：一键清空回收站所有文件。



5.6.3. 算法

【文件恢复】用户将网盘内文件删除，实际上是只在数据库中更新文件状态为“已删除”，实际上服务器上仍存有备份，当用户点击文件恢复时，前台向后台传递要恢复的文件 ID 列表，服务器将指定 ID 的文件状态重新恢复到原来存储的网盘位置。

【回收站清空】前台传递回收站中所有文件的 ID，后台操作数据库删除文件 ID 对应的文件记录。

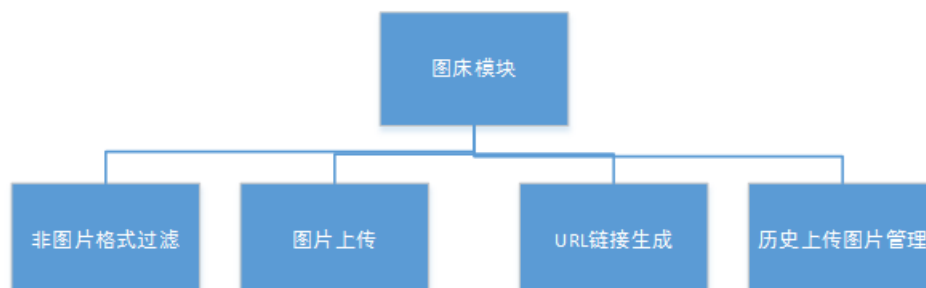
5.7. 图床模块设计说明

5.7.1. 程序描述

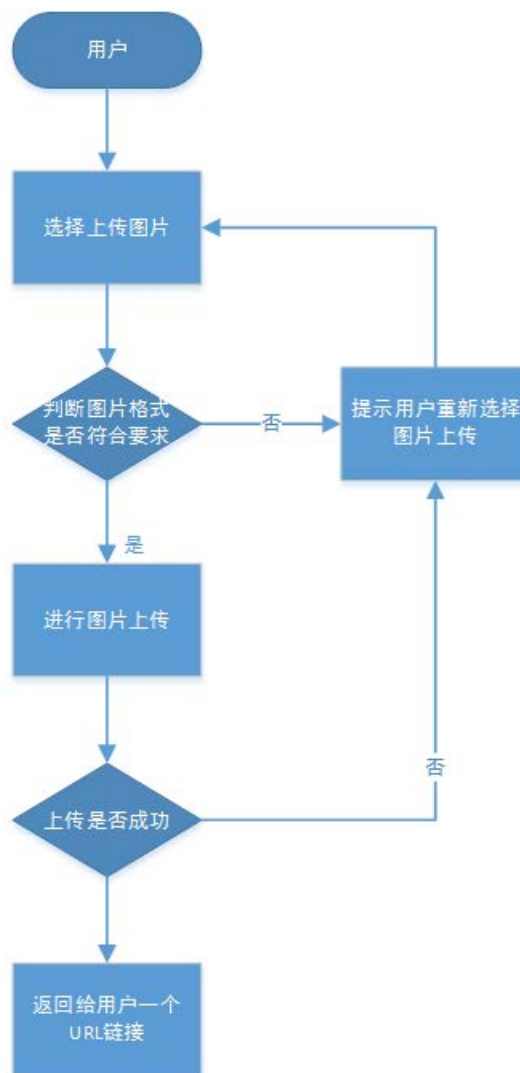
设计本程序旨在设计一个图床，让用户能够通过这个进行图片的上传、访问以及下载。通过上传可以获得 url 链接，用户可以通过此 url 访问及下载。

5.7.2. 功能

1. 非图片格式文件过滤：本模块仅限图床功能，用户只能上传图片格式的文件
2. 图片上传
3. 返回 URL 链接，方便再次访问及下载
4. 历史上传图片管理



5.7.3. 流程逻辑



5.7.4. 接口

调用方法: `upload()`

参数: `MultipartFile`, `description`

功能: 上传单张图片、修改文件名、返回 url 链接给用户

5.7.5. 限制条件

1. 用户输入的格式必须是图片，否则提示用户重新上传；
2. 获取图片失败，提示用户重新上传。

第六章 数据库设计

6.1. 外部设计

6.1.1. 标识符和状态

数据库名称: bits_cloud_disk

6.1.2. 使用它的程序

访问此数据库的所有应用程序: Bits Cloud Disk

6.1.3. 约定

表名、字段名必须使用小写字母或数字，禁止出现数字开头，禁止两个下划线中间只出现数字。

视图名称均以 “view_” 为前缀。

存储过程均以 “sp_” 为前缀。

所有表均以自增 id 为主键，无外键。

6.2. 结构设计

6.2.1. 概念结构设计

经过需求分析后，我们将需求分析得到的用户需求抽象为信息结构（即概念模型）的过程就是概念结构设计。概念结构是各种数据模型的共同基础，它比数据模型更独立于机器，更抽象，从而更加稳定，它是整个数据库设计的关键。它通过 ER 图的形式来展现。

主要实体及其属性如图所示。

user_folder	
folder_id	INT(11)
user_id	INT(11)
parent_id	INT(11)
folder_name	VARCHAR(255)
create_time	TIMESTAMP
modify_time	TIMESTAMP
delete_time	TIMESTAMP
Indexes	

user_file	
file_id	INT(11)
user_id	INT(11)
parent_id	INT(11)
origin_id	INT(11)
file_name	VARCHAR(200)
file_type	VARCHAR(100)
file_status	TINYINT(2)
create_time	TIMESTAMP
modify_time	TIMESTAMP
delete_time	TIMESTAMP
Indexes	

origin_file	
origin_file_id	INT(11)
file_md5	CHAR(32)
file_size	BIGINT(20)
file_type	VARCHAR(100)
file_url	VARCHAR(255)
file_count	INT(11)
file_status	TINYINT(2)
create_time	TIMESTAMP
modify_time	TIMESTAMP
Indexes	

friend	
id	INT(11)
friend_type	VARCHAR(100)
user_id	INT(11)
friend_id	INT(11)
create_time	TIMESTAMP
Indexes	



user	
user_id	INT(11)
username	VARCHAR(32)
password	VARCHAR(32)
nickname	VARCHAR(32)
realname	VARCHAR(32)
gender	TINYINT(4)
phone	VARCHAR(11)
email	VARCHAR(100)
iconimg	VARCHAR(180)
info	VARCHAR(255)
level	INT(2)
is_vip	TINYINT(1)
memory_size	BIGINT(20)
used_size	BIGINT(20)
private_status	TINYINT(1)
private_pass	VARCHAR(32)
created_time	TIMESTAMP
last_login	TIMESTAMP
Indexes	

note	
id	INT(11)
user_id	INT(11)
title	VARCHAR(128)
content	TEXT
is_delete	TINYINT(1)
create_time	TIMESTAMP
update_time	TIMESTAMP
Indexes	

image	
id	INT(11)
name	VARCHAR(50)
md5	CHAR(32)
url	VARCHAR(255)
description	VARCHAR(100)
Indexes	

6.2.2. 逻辑结构设计

用户表 User: 用户相关信息

```
DROP TABLE IF EXISTS `user`;
CREATE TABLE `user` (
  `user_id` INT(11) UNSIGNED PRIMARY KEY NOT NULL AUTO_INCREMENT COMMENT '用户ID, 主键',
  `username` VARCHAR(32) UNIQUE NOT NULL COMMENT '用户名',
  `password` VARCHAR(32) NOT NULL COMMENT '密码',
  `nickname` VARCHAR(32) UNIQUE DEFAULT '请更换昵称' COMMENT '用户昵称',
  `realname` VARCHAR(32) DEFAULT '' COMMENT '真实姓名',
  `gender` TINYINT(4) DEFAULT 0 COMMENT '性别 (0:未知, 1:男, 2:女)',
  `phone` VARCHAR(11) DEFAULT '' COMMENT '手机号码',
  `email` VARCHAR(100) DEFAULT '' COMMENT '邮箱地址',
  `iconimg` VARCHAR(180) DEFAULT 'avatar.jpg' COMMENT '用户头像',
  `info` VARCHAR(255) DEFAULT '' COMMENT '个人简介',
  `level` INT(2) UNSIGNED DEFAULT 1 COMMENT '用户等级',
  `is_vip` TINYINT(1) NOT NULL DEFAULT 0 COMMENT '0:不是vip, 1:是vip',
  `memory_size` BIGINT(20) UNSIGNED DEFAULT 1073741824 COMMENT '用户默认内存(16)',
  `used_size` BIGINT(20) UNSIGNED DEFAULT 0 COMMENT '用户已用内存',
  `private_status` TINYINT(1) DEFAULT 0 COMMENT '是否开启私密空间 (0:未开启, 1:已开启)',
  `private_pass` VARCHAR(32) COMMENT '私密空间密码',
  `created_time` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '账号创建时间',
  `last_login` TIMESTAMP NOT NULL DEFAULT '1970-01-02 00:00:00' ON UPDATE CURRENT_TIMESTAMP COMMENT '账号最近登录时间'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci AUTO_INCREMENT=10000 COMMENT='用户信息表';
```

名	类型	长度	小数点	不是 null	
user_id	int	11	0	<input checked="" type="checkbox"/>	🔑1
username	varchar	32	0	<input checked="" type="checkbox"/>	
password	varchar	32	0	<input checked="" type="checkbox"/>	
nickname	varchar	32	0	<input type="checkbox"/>	
realname	varchar	32	0	<input type="checkbox"/>	
gender	tinyint	4	0	<input type="checkbox"/>	
phone	varchar	11	0	<input type="checkbox"/>	
email	varchar	100	0	<input type="checkbox"/>	
iconimg	varchar	180	0	<input type="checkbox"/>	
info	varchar	255	0	<input type="checkbox"/>	
level	int	2	0	<input type="checkbox"/>	
is_vip	tinyint	1	0	<input type="checkbox"/>	
memory_size	bigint	20	0	<input type="checkbox"/>	
used_size	bigint	20	0	<input type="checkbox"/>	
private_status	tinyint	1	0	<input type="checkbox"/>	
private_pass	varchar	32	0	<input type="checkbox"/>	
created_time	timestamp	0	0	<input checked="" type="checkbox"/>	
last_login	timestamp	0	0	<input checked="" type="checkbox"/>	

原始文件表 OriginFile: 真实存储在服务器的文件的相关信息

```
CREATE TABLE `origin_file` (
  `origin_file_id` INT(11) UNSIGNED PRIMARY KEY NOT NULL AUTO_INCREMENT COMMENT '原始文件ID, 主键',
  `file_md5` CHAR(32) NOT NULL UNIQUE COMMENT 'md5值',
  `file_size` BIGINT(20) UNSIGNED NOT NULL COMMENT '文件大小(字节)',
  `file_type` VARCHAR(100) NOT NULL DEFAULT 'UNKNOWN' COMMENT '文件的真实类型',
  `file_url` VARCHAR(255) NOT NULL UNIQUE COMMENT '文件保存路径',
  `file_count` INT(11) UNSIGNED DEFAULT 1 COMMENT '引用计数/文件拥有者数量',
  `file_status` TINYINT(2) UNSIGNED DEFAULT '1' COMMENT '文件状态[1=正常][2=不允许上传][3=未经允许的上传][4=已屏蔽]',
  `create_time` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '添加时间',
  `modify_time` TIMESTAMP NOT NULL DEFAULT '1970-01-02 00:00:00' ON UPDATE CURRENT_TIMESTAMP COMMENT '文件修改时间'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci AUTO_INCREMENT=10000 COMMENT='源文件(多用户共享一份)';
```

名	类型	长度	小数点	不是 null	
file_id	int	11	0	<input checked="" type="checkbox"/>	🔑1
user_id	int	11	0	<input checked="" type="checkbox"/>	
parent_id	int	11	0	<input checked="" type="checkbox"/>	
origin_id	int	11	0	<input checked="" type="checkbox"/>	
file_name	varchar	200	0	<input checked="" type="checkbox"/>	
file_type	varchar	100	0	<input checked="" type="checkbox"/>	
file_status	tinyint	2	0	<input type="checkbox"/>	
create_time	timestamp	0	0	<input checked="" type="checkbox"/>	
modify_time	timestamp	0	0	<input checked="" type="checkbox"/>	
delete_time	timestamp	0	0	<input checked="" type="checkbox"/>	

用户文件表 UserFile：用户网盘目录中的文件信息

```
CREATE TABLE `user_file` (
  `file_id` INT(11) UNSIGNED PRIMARY KEY NOT NULL AUTO_INCREMENT COMMENT '用户文件ID, 主键',
  `user_id` INT(11) UNSIGNED NOT NULL COMMENT '文件所属者ID',
  `parent_id` INT(11) UNSIGNED NOT NULL COMMENT '父级文件夹ID',
  `origin_id` INT(11) UNSIGNED NOT NULL COMMENT '源文件ID',
  `file_name` VARCHAR(255) NOT NULL COMMENT '文件名称',
  `file_type` VARCHAR(100) NOT NULL DEFAULT 'UNKNOWN' COMMENT '文件类型 (用户可以修改后缀名, 但不会影响真实文件类型)',
  `file_status` TINYINT(2) UNSIGNED NOT NULL DEFAULT 1 COMMENT '文件状态, [1=正常][2=回收站][3=已删除][4=未分享][5=已分享][6=私密]',
  `create_time` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '文件创建时间',
  `modify_time` TIMESTAMP NOT NULL DEFAULT '1970-01-02 00:00:00' ON UPDATE CURRENT_TIMESTAMP COMMENT '文件修改时间',
  `delete_time` TIMESTAMP NOT NULL DEFAULT '1970-01-02 00:00:00' ON UPDATE CURRENT_TIMESTAMP COMMENT '文件删除时间',
  CONSTRAINT `file_id_FK` FOREIGN KEY (`file_id`) REFERENCES `origin_file` (`origin_file_id`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci AUTO_INCREMENT=10000 COMMENT='用户文件信息表';
```

名	类型	长度	小数点	不是 null	
file_id	int	11	0	<input checked="" type="checkbox"/>	🔑1
user_id	int	11	0	<input checked="" type="checkbox"/>	
parent_id	int	11	0	<input checked="" type="checkbox"/>	
origin_id	int	11	0	<input checked="" type="checkbox"/>	
file_name	varchar	200	0	<input checked="" type="checkbox"/>	
file_type	varchar	100	0	<input checked="" type="checkbox"/>	
file_status	tinyint	2	0	<input type="checkbox"/>	
create_time	timestamp	0	0	<input checked="" type="checkbox"/>	
modify_time	timestamp	0	0	<input checked="" type="checkbox"/>	
delete_time	timestamp	0	0	<input checked="" type="checkbox"/>	

用户目录表 UserFolder：用户网盘中的目录层级表

```
DROP TABLE IF EXISTS `user_folder`;
CREATE TABLE `user_folder` (
  `folder_id` INT(11) UNSIGNED PRIMARY KEY NOT NULL AUTO_INCREMENT COMMENT '文件夹ID',
  `user_id` INT(11) UNSIGNED NOT NULL COMMENT '文件夹所属者ID',
  `parent_id` INT(11) UNSIGNED NOT NULL COMMENT '父级文件夹ID',
  `folder_name` VARCHAR(255) NOT NULL COMMENT '文件夹名称',
  `create_time` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '文件夹创建时间',
  `modify_time` TIMESTAMP NOT NULL DEFAULT '1970-01-02 00:00:00' ON UPDATE CURRENT_TIMESTAMP COMMENT '文件夹修改时间',
  `delete_time` TIMESTAMP NOT NULL DEFAULT '1970-01-02 00:00:00' ON UPDATE CURRENT_TIMESTAMP COMMENT '文件夹删除时间',
  CONSTRAINT `folder_id_FK` FOREIGN KEY (`folder_id`) REFERENCES `origin_folder` (`origin_folder_id`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci AUTO_INCREMENT=10000 COMMENT='文件夹信息表';
```

名	类型	长度	小数点	不是 null	
folder_id	int	11	0	<input checked="" type="checkbox"/>	🔑1
user_id	int	11	0	<input checked="" type="checkbox"/>	
parent_id	int	11	0	<input type="checkbox"/>	
folder_name	varchar	255	0	<input checked="" type="checkbox"/>	
create_time	timestamp	0	0	<input checked="" type="checkbox"/>	
modify_time	timestamp	0	0	<input checked="" type="checkbox"/>	
delete_time	timestamp	0	0	<input checked="" type="checkbox"/>	

好友表 Friend：用户的好友及群组关系

```
CREATE TABLE `friend` (
  `id` INT(11) UNSIGNED PRIMARY KEY NOT NULL AUTO_INCREMENT COMMENT '主键',
  `friend_type` VARCHAR(100) NOT NULL DEFAULT '1' COMMENT '好友类型[1:个人好友][2:群组好友]',
  `user_id` INT(11) UNSIGNED NOT NULL COMMENT '用户ID',
  `friend_id` INT(11) UNSIGNED NOT NULL COMMENT '好友ID',
  `create_time` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '好友关系建立时间',
  CONSTRAINT FK_UserId_UserId FOREIGN KEY(user_id) REFERENCES user(user_id), # 设置 UserId 为外键
  CONSTRAINT FK_FriendId_UserId FOREIGN KEY(friend_id) REFERENCES user(user_id) # 设置 FriendId 为外键
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci AUTO_INCREMENT=10000 COMMENT='好友表';
```

名	类型	长度	小数点	不是 null	
id	int	11	0	<input checked="" type="checkbox"/>	🔑 1
friend_type	varchar	100	0	<input checked="" type="checkbox"/>	
user_id	int	11	0	<input checked="" type="checkbox"/>	
friend_id	int	11	0	<input checked="" type="checkbox"/>	
create_time	timestamp	0	0	<input checked="" type="checkbox"/>	

分享表 Share：文件分享相关信息

```
CREATE TABLE `share` (
  `id` INT(11) UNSIGNED PRIMARY KEY NOT NULL AUTO_INCREMENT COMMENT '自增主键',
  `user_id` INT(11) UNSIGNED NOT NULL COMMENT '文件分享者ID',
  `file_id` text not null COMMENT '文件ID',
  `share_type` INT not null COMMENT '分享类型，共享，好友共享，私密共享',
  `share_pass` varchar(30) COMMENT '分享密码，如果设的话，表示共享',
  `share_time` datetime not null COMMENT '分享时间',
  `share_deadline` datetime COMMENT '分享有效时间',
  `share_link` INT not null COMMENT '分享连接',
  `thumb_up` INT COMMENT '点赞次数',
  `thumb_down` INT COMMENT '反对次数',
  CONSTRAINT UserID_ShareFile_FK FOREIGN KEY () References UserInfo(UserID) -- 设置 UserID 为外键
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci AUTO_INCREMENT=10000 COMMENT='文件分享表';
```

名	类型	长度	小数点	不是 null	
id	int	11	0	<input checked="" type="checkbox"/>	🔑 1
user_id	int	11	0	<input checked="" type="checkbox"/>	
file_id	int	11	0	<input checked="" type="checkbox"/>	
share_type	tinyint	4	0	<input checked="" type="checkbox"/>	
share_pass	varchar	30	0	<input type="checkbox"/>	
share_time	timestamp	0	0	<input checked="" type="checkbox"/>	
share_deadline	timestamp	0	0	<input checked="" type="checkbox"/>	
share_link	varchar	255	0	<input checked="" type="checkbox"/>	
thumb_up	int	11	0	<input type="checkbox"/>	
thumb_down	int	11	0	<input type="checkbox"/>	

笔记表 Note：云笔记存储相关信息

```
DROP TABLE IF EXISTS `note`;
CREATE TABLE `note` (
  `id` INT(11) UNSIGNED PRIMARY KEY NOT NULL AUTO_INCREMENT COMMENT '主键',
  `user_id` INT(11) UNSIGNED NOT NULL COMMENT '用户ID',
  `title` VARCHAR(128) NOT NULL COMMENT '笔记标题',
  `content` TEXT NOT NULL COMMENT '笔记内容',
  `is_delete` TINYINT(1) NOT NULL DEFAULT '0' COMMENT '是否删除[0:未删除][1:已删除]',
  `create_time` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
  `update_time` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '更新时间'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci AUTO_INCREMENT=10000 COMMENT='云笔记表';
```

名	类型	长度	小数点	不是 null	
id	int	11	0	<input checked="" type="checkbox"/>	🔑 1
user_id	int	11	0	<input checked="" type="checkbox"/>	
title	varchar	128	0	<input checked="" type="checkbox"/>	
content	text	0	0	<input checked="" type="checkbox"/>	
is_delete	tinyint	1	0	<input type="checkbox"/>	
create_time	timestamp	0	0	<input checked="" type="checkbox"/>	
update_time	timestamp	0	0	<input checked="" type="checkbox"/>	

图片表 Image：图片存储相关信息

```
DROP TABLE IF EXISTS `image`;
CREATE TABLE `image` (
  `id` INT(11) UNSIGNED PRIMARY KEY NOT NULL AUTO_INCREMENT COMMENT '主键',
  `name` VARCHAR(50) NOT NULL COMMENT '图片名称',
  `md5` CHAR(32) NOT NULL UNIQUE COMMENT 'md5值',
  `url` VARCHAR(255) NOT NULL UNIQUE COMMENT '图片保存路径',
  `description` VARCHAR(100) DEFAULT NULL COMMENT '图片描述'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci AUTO_INCREMENT=10000 COMMENT='图床表';
```

名	类型	长度	小数点	不是 null	
id	int	11	0	<input checked="" type="checkbox"/>	🔑 1
name	varchar	50	0	<input checked="" type="checkbox"/>	
md5	char	32	0	<input checked="" type="checkbox"/>	
url	varchar	255	0	<input checked="" type="checkbox"/>	
description	varchar	100	0	<input type="checkbox"/>	

6.2.3. 物理结构设计

数据库服务器：

- 服务器：localhost via UNIX socket
- 服务器类型：MySQL
- 服务器版本：5.5.49-0ubuntu z0.16.04.1 - (Ubuntu)
- 协议版本：10
- 用户：root@localhost
- 服务器字符集：UTF-8 Unicode (utf8)

网站服务器：

- Web 应用服务器
- Apache/2.4.7 (Ubuntu)
- 数据库客户端版本：MySQL 5.7

6.3. 运用设计

6.3.1. 数据字典设计

数据字典是关于数据的信息的集合，也就是对数据流图中包含的所有元素的定义的集合。数据流图和数据字典共同构成系统的逻辑模型。

根据数据流图和系统要求列出以下主要数据字典：

名称：user

别名：用户表

简述：记录用户的信息

定义：用户表 = 用户 ID + 用户名 + 密码 + 昵称 + 真实姓名 + 性别 + 手机号 + 邮箱地址 + 用户头像 + 个人简介 + 用户等级 + 用户是否为 VIP + 用户默认内存 + 用户已用内存 + 是否开通私密空间 + 私密空间密码 + 账号创建时间 + 账号最近登录时间

名称: origin_file
别名: 原始文件表
简述: 真实存储在服务器的文件的相关信息
定义: 原始文件表 = 原始文件 ID + 文件 md5 值 + 文件大小 + 文件真实类型 + 文件保存路径 + 文件拥有者数量 + 文件状态 + 添加时间 + 文件修改时间

名称: user_file
别名: 用户文件表
简述: 用户网盘目录中的文件信息
定义: 用户文件 ID + 用户文件 ID + 父级文件夹 ID + 源文件 ID + 文件名 + 文件类型 + 文件状态 + 文件创建时间 + 文件修改时间 + 文件删除时间

名称: user_folder
别名: 用户目录表
简述: 用户网盘级中的目录层级表
定义: 文件夹 ID + 文件夹所属者 ID + 父级文件夹 ID + 文件夹名称 + 文件夹创建时间 + 文件夹修改时间 + 文件夹删除时间

名称: share
别名: 分享表
简述: 文件分享相关信息
定义: 文件分享者 ID + 文件 ID + 分享类型 + 分享密码 + 分享时间 + 分享有效时间 + 分享链接 + 点赞次数 + 反对次数

名称: friend
别名: 好友表
简述: 用户的好友及群组关系
定义: 好友类型 + 用户 ID + 好友 ID + 好友关系建立时间

名称: image
别名: 图片表
简述: 图片存储相关信息
定义: 图片名称 + 文件 md5 值 + 图片保存路径 + 图片描述

名称: note
别名: 笔记表
简述: 云笔记存储相关信息
定义: 用户 ID + 笔记标题 + 笔记内容 + 是否删除 + 创建时间 + 更新时间

6.3.2. 安全保密设计

本系统应当具备防治黑客入侵随意篡改用户数据的技术，尤其是系统内的一些核心数据，应对数据库采取一系列的安全保密技术，防治不法分子盗用数据。

本数据库系统采用安全的用户名加口令方式登录。使用者在使用此数据库时，由于不同的访问者的登陆账号不同，且需要密码验证，故相当于对于数据的访问设置了权限，所以数据可以说是安全的。用户名的权限限制为只能进行基本的增删改查数据功能。具体的安全保护措施如下：

数据库

1. 通过网络防火墙将数据库系统与外界隔离，保证其安全稳定地运行。
2. 多机备份，减少信息丢失可能性。
3. 用户密码等关键隐私信息经过 MD5 加密，防止拖库等数据库攻击。
4. 对数据库的访问采用权限授予机制，在用户表中表示用户的权限级别，只有具备相应权限的用户具有操作权限，控制部分关键信息的增删操作。

软件使用数据

本应用存储的数据涉及用户上传的文件、专辑、分享、小组信息、用户的偏好信息等。一旦丢失将导致用户个人上传记录、收藏历史的遗失，推荐算法无法根据历史记录为用户定向推送，导致用户体验感急剧下降。因此本应用对于数据的安全性要求较高，要尽量保证用户的数据不会丢失。

1. 用户上传的文件采用多重备份的方式存储。采用分布式存储的方式，图片信息及其副本存放在不同的机器上，即使一台机器宕机，仍能保证数据的安全存储，防止由于掉电、机械故障等物理原因导致的数据丢失。

2. 用户的偏好信息由上传记录、分享记录、收藏记录、小组及好友记录构成，为了保证数据的新鲜度，本应用对用户近 30 次的使用记录进行存储。由于特殊原因造成的个别记录丢失，并不影响系统按照用户偏好记录进行推送。

用户账户数据

1. 应用仅对用户在本系统的用户名、密码等进行保存，个人信息仅与本网站功能相关，不会对用户隐私造成任何不利影响。

2. 对于用户在本系统中对于第三方社交平台的授权信息，系统将进行加密存储，严格防止用户第三方社交平台的信息泄漏。应用不会利用用户第三方社交平台进行用户指定的分享之外的任何操作。

6.3.3. 优化

分析并优化数据库的“时一空”效率，尽可能地“提高处理速度”并且“降低数据占用空间”。

- (1) 分析“时一空”效率的瓶颈，找出优化对象（目标），并确定优先级。
- (2) 当优化对象（目标）之间存在对抗时，给出折衷方案。
- (3) 给出优化的具体措施，例如优化数据库环境参数，对表格进行反规范化处理等。

第七章 系统测试

7.1. 测试概要

7.1.1. 测试用例

7.1.1.1. 功能性

主要功能有 5 个模块：

用户管理模块

文件分享模块

回收站模块

图床模块

文件管理模块

7.1.1.2. 易用性

对用户的输入数据进行正确性校验，并提供相应的易理解的信息提示，帮助用户输入合法的信息。

界面设计美观，按钮名称易懂，用词准确，与同一界面上的其他按钮易于区分，能望文知意。

7.2. 测试环境

操作系统：Windows 8.1(x64)，JDK1.8

数据库：MySQL 5.7

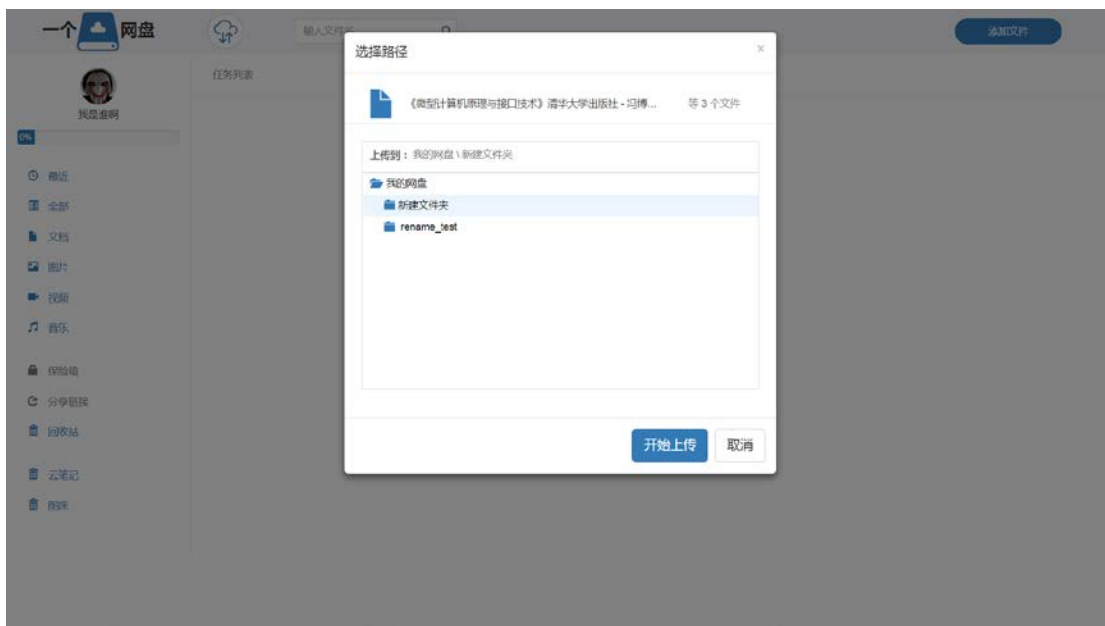
IDE：IntelliJ IDEA，Tomcat 8.5

语言：Java

7.3. 测试结果

7.3.1. 文件上传

测试说明：上传不同文件名（包含中文和不包含中文）、大小的文件，观察是否能正常显示、上传。

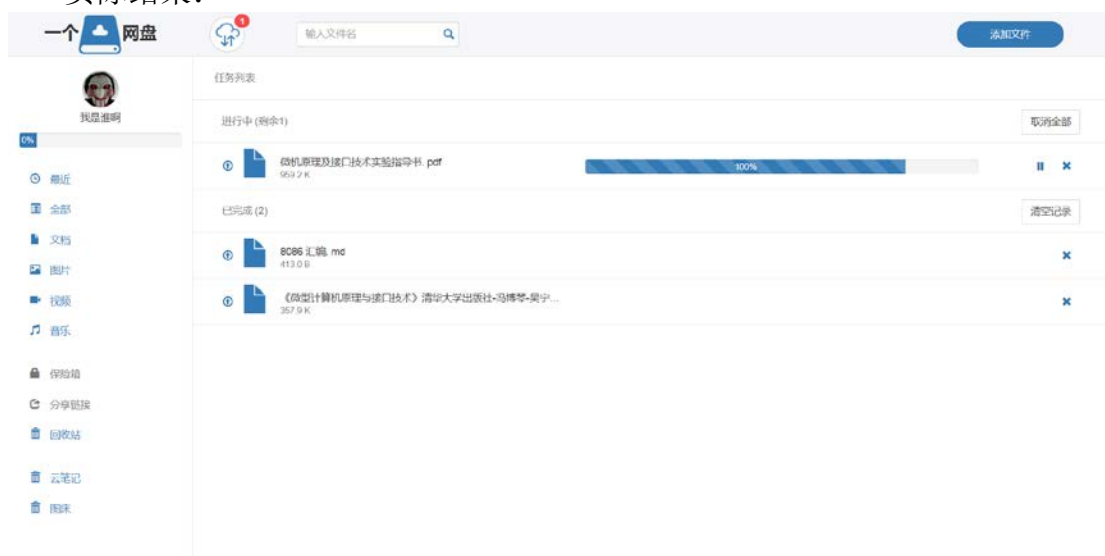


测试数据：

文件路径\文件名	文件大小
C:\Users\cloud\Desktop\test\test1, pdf	413B
C:\Users\cloud\Desktop\test\test2, pdf	959. 2KB
C:\Users\cloud\Desktop\test\中文测试\8086 汇编, pdf	413B
C:\Users\cloud\Desktop\test\中文测试\微机原理及接口技术实验指导书, pdf	959. 2KB

预计测试结果：文件正常加入到上传列表，无显示异常等错误。

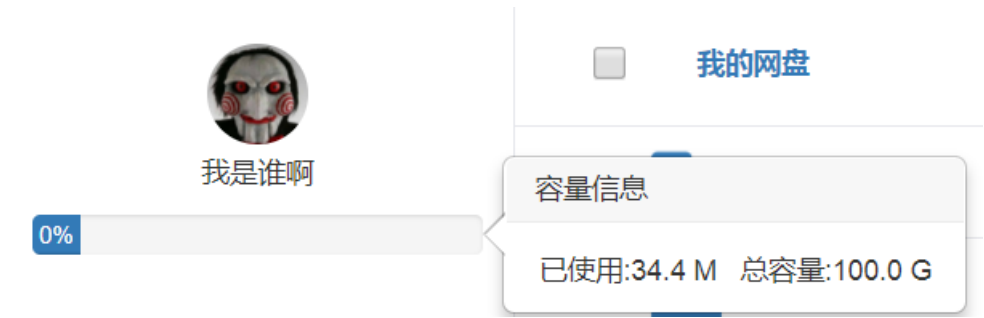
实际结果：



结论：测试成功

7.3.2. 容量信息显示

测试说明：上传任意文件，观察已使用容量改变量是否与上传文件大小相符。

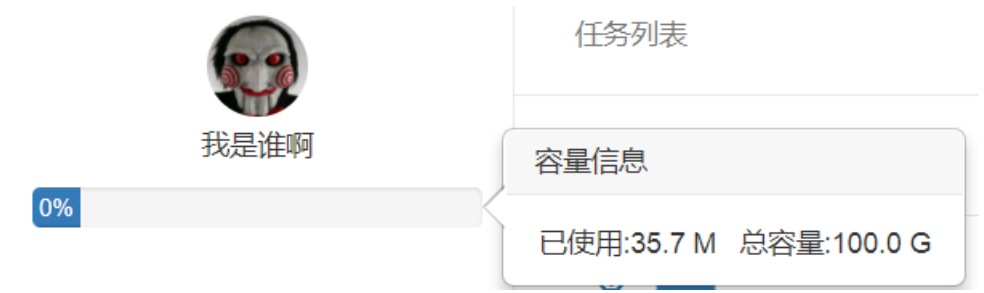


测试数据：

文件路径\文件名	文件大小
C:\Users\cloud\Desktop\test\test3.png	1.3MB

预计测试结果：上传文件前后已使用容量发生改变，改变量与上传文件大小相符。

实际结果：



结论：测试成功

7.3.3. 缩略图显示

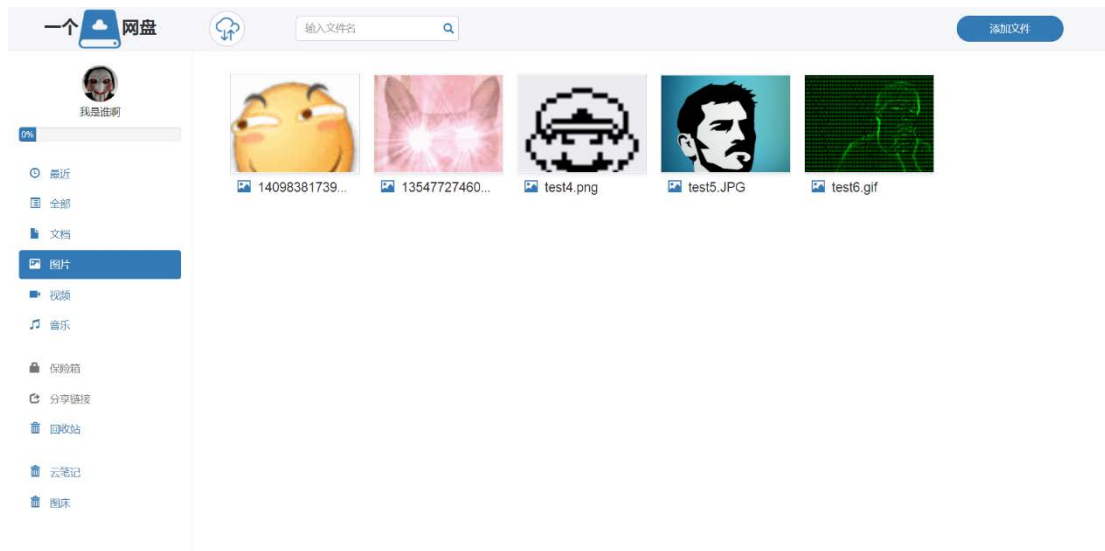
测试说明：上传三种最常用格式的图片（jpg、png、gif），观察图片界面能否正常显示缩略图，gif 动图能否自动播放。

测试数据：

文件路径\文件名	文件大小
C:\Users\cloud\Desktop\test\test4.png	813KB
C:\Users\cloud\Desktop\test\test2.jpg	959.2KB
C:\Users\cloud\Desktop\test\test6.gif	962KB

预计测试结果：图片缩略图正常显示，gif 格式动图自动播放

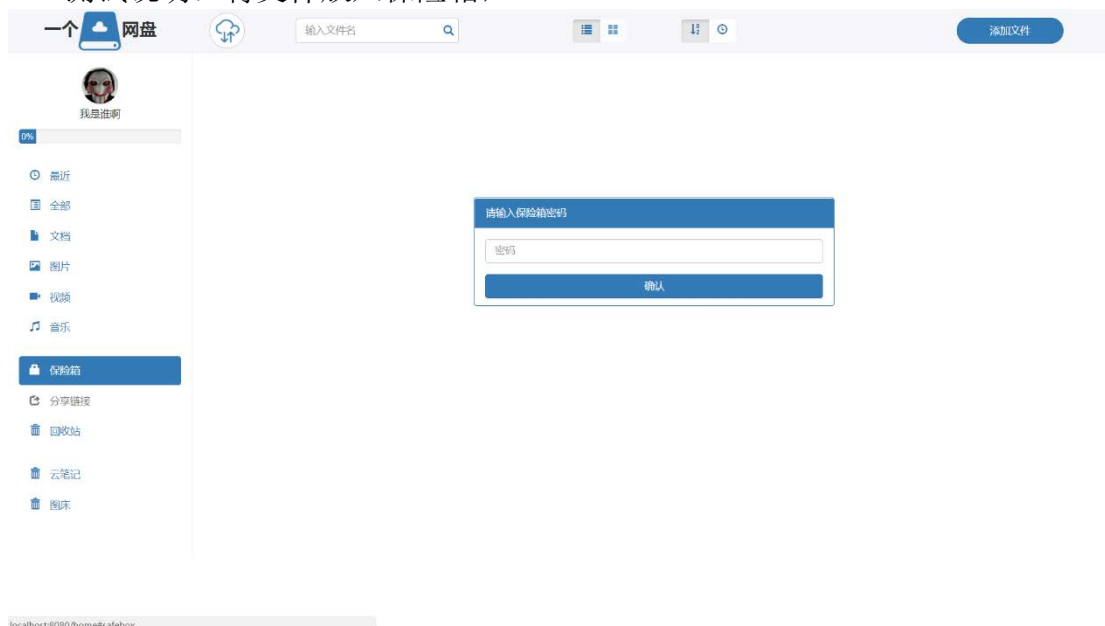
实际结果：



结论：测试成功

7.3.4. 保险箱

测试说明：将文件放入保险箱，



测试数据：无

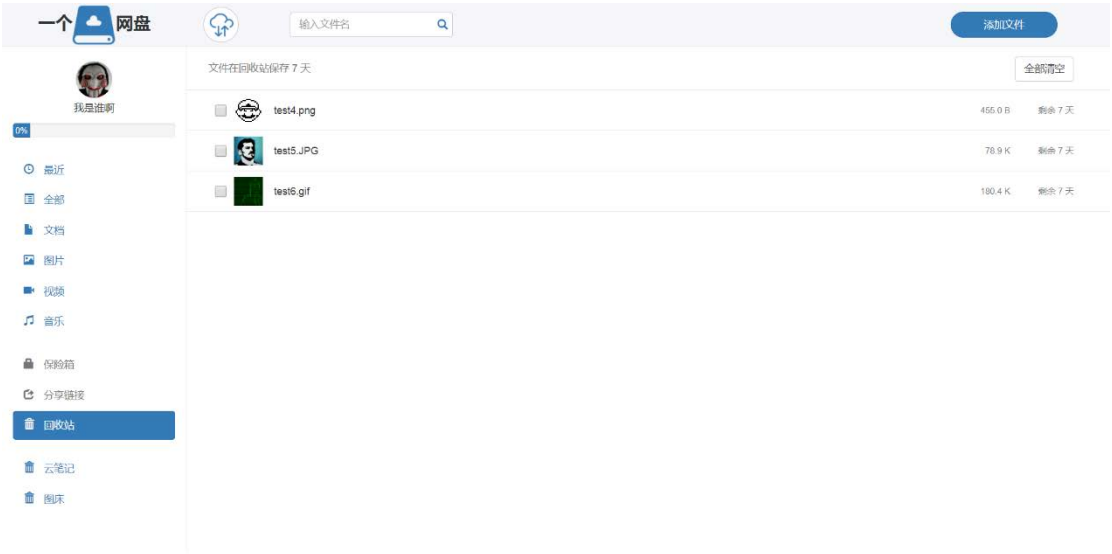
预计测试结果：使用错误密码或不输入密码无法访问保险箱，输入正确密码能正常访问。

实际结果：使用错误密码或不输入密码无法访问保险箱，输入正确密码能正常访问。

结论：测试成功

7.3.5. 回收站测试

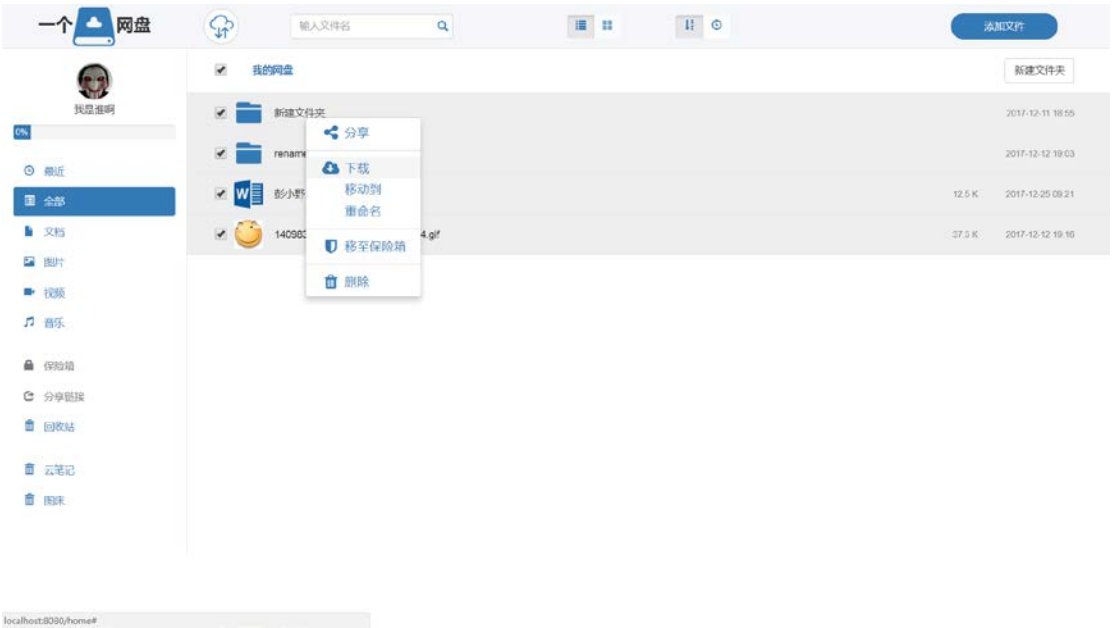
测试说明：将文件放入回收站、从回收站中恢复文件



测试数据：无
预计测试结果：无异常
实际结果：无异常
结论：测试成功

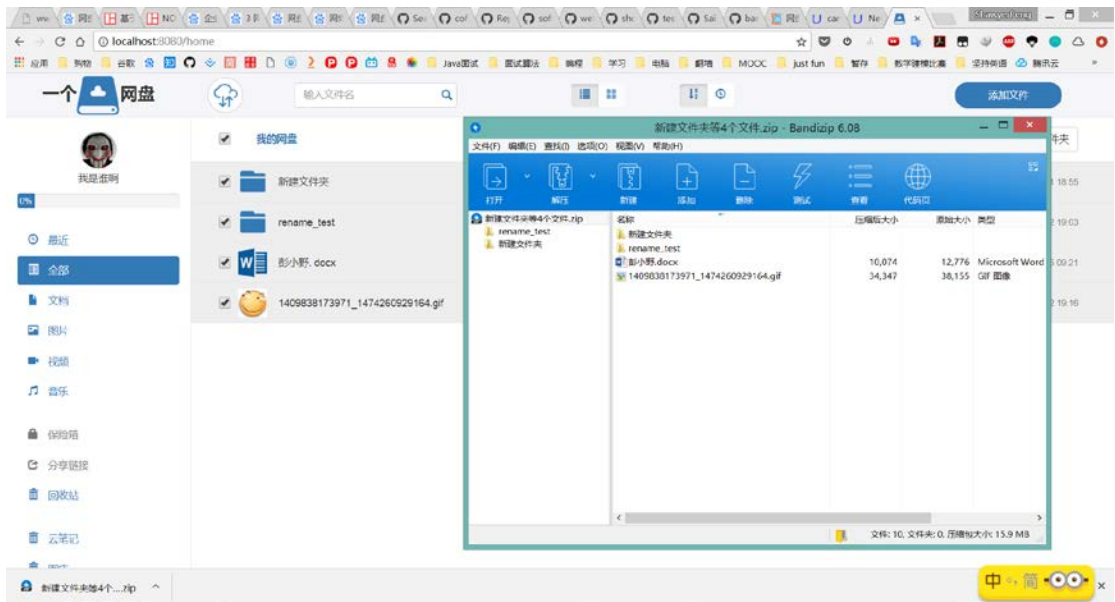
7.3.6. 下载测试

测试说明：下载文件本地



测试数据：无
预计测试结果：无异常

实际结果:



结论：测试成功

7.4. 测试结论

该系统通过三层结构进行设计，并采用在开发过程中的分层测试的方式，极大地消除了系统完成后可能出现的难找的异常，为整个测试和修复工作减少了大量的时间。

测试过程中出现了文件上传完成而网盘目录未更新的情况，需要刷新才能更新显示，这是由于此部分忽略了操作后 AJAX 异步更新数据，实属设计开发过程中的疏漏，测试完成后将对该错误进行排查和解决。

第八章 关键实现技术

1. Spring + Spring MVC + Mybatis 框架: Java Web 开发三大框架。其中 Spring 充当系统容器, 提供控制反转、依赖注入等功能, 负责管理整个系统程序的运行; Spring MVC 提供 RESTful API 支持; Mybatis 封装 JDBC 操作, 方便操作数据库, 并且支持事务处理等特性。
2. 基于 RabbitMQ 的异步任务系统: 本应用中的邮件发送、短信发送等功能全由消息队列实现, 将耗时任务并行化, 减轻服务器的压力, 保障消息发送的可靠性。
3. 基于 Redis 的分布式缓存系统: 本应用还将 Redis 用作缓存代替 Memcached, 减少数据库读取的次数, 减轻数据库负载。
4. 基于关键词分析的内容过滤算法: 这是一款用于用户文件或聊天内容是否具有不安全不健康内容的技术。系统基于搜索引擎的关键词检索过滤算法, 并在多线程环境下运行, 充分利用索引查询的高效性, 大大提高了违规过滤的效率。
5. 通信安全保障: 后端服务器为前端提供的 RESTful API 接口, 完全基于 HTTPS 交互。接口调用基于 Token 签名机制, 减小了明文的暴露次数, 保证数据安全的访问。本系统还采用 Apache Shiro 框架, 提供认证、授权、加密、会话管理等功能, 保障应用的安全性。
6. Bootstrap + jQuery 前端框架: 前后端完全分离, 前台为单页面应用, 界面简介美观, 用户体验性好。
7. Quartz: 定时任务框架, 用来定时同步数据库和用户文件夹内的文件。 关键实现技术

第九章 小结

9.1. 个人完成的程序模块和任务清单

学号	姓名	完成的程序模块	参与的任务
1502040228	彭小野	系统后端主要开发人员以及前端	需求分析、概要设计、详细设计、数据库设计等文档的书写，设计说明书汇总整理及任务分配
1509020236	刘刻	文件分享模块、表单校验	需求分析、详细设计、可行性分析文档的书写
1506010233	周懂贤	用户登录、注册	需求分析、系统测试文档的书写
1506010234	刘云浩	回收站模块	系统测试文档的书写，UML、数据流图、流程图的绘制
1506010235	吴仕权	图床模块	需求分析、详细设计文档的书写

9.2. 个人总结

9.2.1. 彭小野

本次课程设计我负责的主要模块是文件管理，可行性分析报告的书写，参与需求分析中文件分享部分和详细设计的书写。

由于时间有限，在该系统的设计上还有所欠缺，针对一些边边角角的异常未进行处理，整个系统也只完成了一部分功能，但是对于完成了的功能，至少保证了其运行的稳定性，保证数据库操作的合法性和完整性，也完成了常用的表单校验和防止表单重复提交等网站安全工作。关于网页美化方面，由于时间原因，我们主要着重于后端开发，只完成了基本的前端工作，未进行有计划性的美化工作。

由于技术上的有限性，在数据库设计，网站前端设计方面还有所欠缺，这方面的内容应该在以后的学习中有所侧重。

为了完成本次课程设计，组员们又认真学习了《软件工程》课本中相关的知识点，学习力结构化分析方法解题的步骤和特点，学习了数据流图中的图形表示、源和宿的确定、数据流和文件以及加工的定义，进一步学习了分层书库流图的画法，分层数据流图的审查，包括分层数据流图额一致性和完整性审查方法，此外还学习了数据字典。在学习过程中，对理论性的学习不断增强，重新学习到更多的知识，真正实现了温故而知新。

在学习数据流图的绘画过程中，使用 Visio 工具练习了数据流图的绘画方法，从刚开始的对工具的不熟悉，一步步熟练掌握了工具的使用方法，会根据系统进行功能分析，从而实现顶层数据流图的绘画，0 层数据流图的绘画，分层数据流图的绘画，并不断进行完善，最终达到自己可以画出完整的数据流图。正是这样，将所学的知识应用于实践，才最终实现了对本网盘系统的数据流图的绘画。

在参与代码实现过程中，不仅要熟悉 MySQL 工具的使用，还要熟悉 IntelliJ IDEA 开发工具的使用，用代码实现系统的功能会用到各种技术，包括数据库架

构的设计、接口的实现、代码实现与数据库的连接等等，这些都需要好好学习，在系统实现过程中，只有边学习边应用才能更好地掌握知识，学习到更多的东西。

在这次的实验当中，队员们深深地感受到了小组的作用。很多部分的完成凭借一个人的力量是很难完成的，但是通过小组的分工之后每个人的压力就大大减轻了，可以快速高效的完成这次的任务。同时，许多个人难以解决的问题通过小组成员的共同努力下也迎刃而解了，所以这就是小组的力量。

我们在一起讨论的时候，也发现了许多的困难，例如大家时间安排上经常会发生冲突，很大的影响了效率，还有就是在讨论的过程中经常遇见意见不同意的情况出现。

通过这次课程设计，我们了解到本身还有很多不足之处，对理论知识的掌握不够透彻，平时缺乏把理论知识应用于实践，从而减弱了动手能力。在代码编写方面，自己所掌握的编程语言还不够应用于实践，而且编码的水平有待提高。同使我也收获了很多知识，学会了数据流图的画法、数据字典的确定、系统功能模块的划分、对系统的测试方法等等，学习到了各种知识点。在今后的学习过程中，队员们会既学习好理论知识，又要注意将理论知识应用于实践，这样才能更好地掌握知识，提高动手能力，增强个人能力。当然和同伴的合作很重要，要具有极高的合作意识，善于与人合作，这样才能更快的学习到新知识，更快的提高自己。

9.2.2. 刘刻

工作内容：本次课程设计我主要负责的模块是文件分享，可行性分析报告的书写，参与需求分析中文件分享部分和详细设计的书写。

完成情况：文件分享模块暂时未实现，完成可行性分析报告的书写，参与需求分析和详细设计书写

问题：编写程序能力不强，文件分享功能还未实现，文档书写经验不足，太过依赖模板。

解决方法：查询资料，请教别人

收获和体会：此次课设过程中，认识到了文档的重要性，也认识到了自己能力的缺陷。遇到问题时团队一起解决，增强了团队合作能力和意识。加深了对软件工程的理解，增强了自学能力，解决问题的能力。

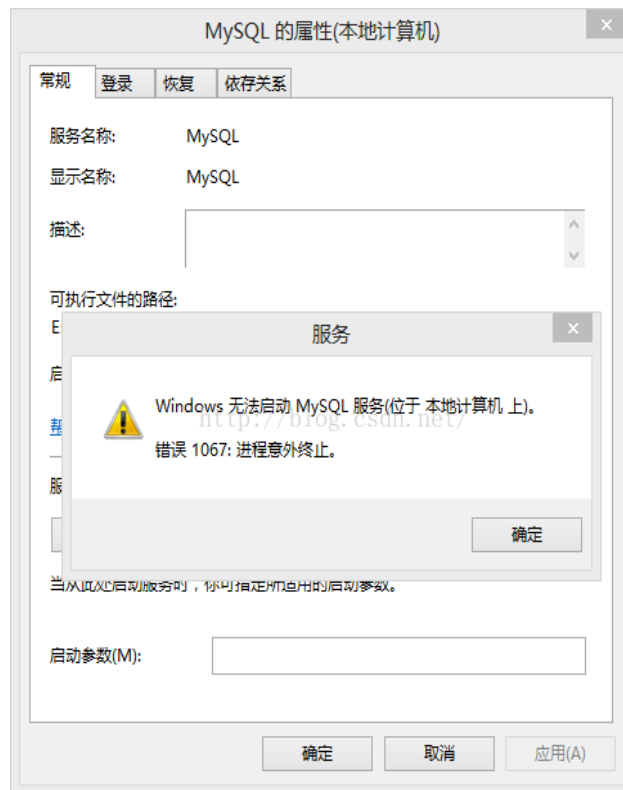
努力改进方向：加强程序编写能力，提高文档书写能力，扩展专业知识，锻炼创新能力和团队合作意识。

9.2.3. 刘云浩

工作内容：回收站处理逻辑代码、文件相关数据库表设计、Visio UML 图绘制、黑盒测试、相关文档撰写。

完成情况：因系统部分功能还未实现，黑盒测试只完成一部分。其它内容全部完成。

遇到的问题及解决办法：PRO1：启动 MySQL 服务，出现错误 1067：



SOL1: 重新安装 MySQL

PRO2: 进行黑盒测试时无法访问系统

SOL2: 关闭 VPN

心得体会: 通过本次软件开发, 我的解决问题的能力得到提高, 同时感受到团队合作的重要性, 通过合理的任务分配, 能够大大提高开发效率, 降低出错概率。

9.2.4. 吴仕权

工作内容: 图床模块的实现, 参与需求分析、详细设计文档的书写

自己完成情况: 图床模块功能基本实现 文档书写部分与其他成员整合后基本完成。

遇到的问题: 数据库方面知识不牢固, 图床部分功能不够完善; 文档编写经验不够, 文档书写不够正式。

解决方法: 查阅书籍, 请问一些比较了解的人。

收获与体会: 在课程设计过程中不免遇到各种各样的困难, 通过与同学间的探讨, 查阅资料, 解决困难, 增强了自己的自学能力和发现问题、解决问题的能力, 提高了自己与他人进行协调沟通的能力。也让我知道了什么叫做付出才有回报。

这次课程设计给了我一个锻炼自己, 强化专业知识的机会, 使我对软件工程有了更深的理解, 也让我真真切切的感受到个人与团队的关系和团队的巨大力量

今后努力与改进方向: 努力学习自己专业的必须知识, 扩增自己的知识面, 加强与队伍的沟通协作能力, 多练习文档的编写, 提高文档的编写能力。

9.2.5. 周懂贤

工作内容：用户注册，用户登录功能的实现，系统测试文档的书写。

自己的完成情况：用户注册，用户登录功能基本实现，文档书写部分与其他成员整合后基本完成。

遇到的问题：用户注册，用户登录部分遇到一系列的问题，对数据库方面的知识比较陌生，文档书写缺乏经验，写的不规范。

解决方法：网上查询相关资料，查阅书籍，请教这方面了解的同学。

收获与体会：课程设计已经做了这么多天了，虽然成果虽然不是很理想，但是收获还是有的。这期间学到了很多新的知识，一些框架的基本用法，也能编写出一个简单的页面，在期间学习了 UML 图的绘制，软件工程开发的一些基本工具的使用，软件开发文档编写的方法和实践，人性化的界面设计是需要认真努力做得，要充分考虑用户的感受和体验。提高了发现问题、解决问题的能力，强化了自己的专业知识，认识到了团队合作的重要性。

今后努力与改进方向：软件开发中的各种文档编写能力还需要提升，在以后学习和实践中认真总结和完善，参考他人的软件工程项目，重视团队的协作，提高自己的技术。

第十章 致谢

在本次软件工程实践报告设计过程中，感谢学校给了我们学习的机会。首先感谢戚荣志老师认真负责地教授我们软件工程理论知识，得以让我们在课程实践中拥有足够的基础顺利完成一个项目。同时，也要感谢余宇峰老师，在这一周的课程实践中，老师从选题指导、报告框架到细节修改，都给予了细致的指导，提出了很多宝贵的意见与推荐，老师以其严谨求实的治学态度、高度的敬业精神、兢兢业业、孜孜以求的工作作风对我们产生了重要影响。

感谢所有授我以业的老师，没有这些年知识的积淀，我们小组没有这么大的动力和信心完成这次实践。感恩之余，诚恳地请各位老师对我组的课程报告多加批评指正，使我及时完善报告的不足之处。

最后，我要向百忙之中抽时间对本文进行审阅的老师表示衷心的感谢。

参考文献

- [1] 杨岳湘, 邓文平, 邓劲生, 等. 基于云存储的网盘系统架构及关键技术研究[J]. 电信科学, 2012(10):68-72.
- [2] 王峰, 江峰, 李朝阳. 虚拟桌面及关键技术分析云计算时代的数据库研究[J]. 电信技术, 2011(1):24-26.
- [3] 刘意先, 耿鹏涛. 基于 Web 的网盘系统的设计与实现 [J]. 电子设计工程, 2014, 22(08):5-7.
- [4] 谢金星. 基于云存储的网盘系统设计与实现 [D]. 湖南大学, 2016.