

---

# xyz 云盘系统 设计文档

---

	人员	日期
拟制	宋小牛 陈泳洲 金泽文	2018-05-20
评审人	•	yyyy-mm-dd
批准	•	yyyy-mm-dd
签发	•	yyyy-mm-dd

## 摘 要

本文档是 xyz 云盘系统需求规格分析文档，由宋小牛、陈泳州和金泽文共同创建，

本文档主要分析了该软件的任务概述、总体设计、接口设计、数据结构设计、数据库设计、界面设计、出错处理设计和安全保密设计、维护设计、等关于软件多个方面的设计。

**关键词：**云盘    分布式存储    文件共享    版本更新    网络    隐私    安全    p2p  
存储冗余    内容审核    数据结构    算法

表 1   缩略词清单

缩略语	英文全名	中文解释
CentOS	Community Enterprise Operating Syste	社区事业版操作系统
CSS	Cascading Style Sheets	层叠样式表
HTML	HyperText Markup Language	超文本标记语言
HTTP	HyperText Transfer Protocol	超文本传输协议
HTTPS	Hypertext Transfer Protocol Secure	超文本传输安全协议
IOPS	Input/Output Operations Per Second	每秒读写操作的次数
IP	Internet Protocol	网际协议
MD5	Message-Digest Algorithm 5	讯息摘要演算法 5
TCP	Transmission Control Protocol	传输控制协议

# 目 录

摘要 .....	
第 1 章 引言 .....	6
1.1 编写目的 .....	6
1.2 项目背景 .....	6
1.3 术语 .....	7
第 2 章 任务概述 .....	8
2.1 目标 .....	8
2.2 开发与运行环境 .....	8
2.2.1 开发环境的配置 .....	8
2.2.2 测试环境的配置 .....	8
2.2.3 运行环境的配置 .....	8
2.3 需求概述 .....	8
2.4 条件与限制 .....	8
第 3 章 总体设计 .....	13
3.1 软件描述 .....	13
3.2 处理流程 .....	13
3.2.1 总体流程 .....	13
3.2.2 系统基本流程 .....	13
3.2.3 客户端基本流程 .....	13
3.2.4 服务器端基本流程 .....	13
3.2.5 功能 1 具体流程 .....	13
3.2.6 功能 2 具体流程 .....	14
3.2.7 功能 3 具体流程 .....	14
3.3 功能结构设计 .....	14
3.3.1 整体结构 .....	14
3.3.2 用户端结构 .....	14
3.3.3 服务器端结构 .....	14
3.3.4 后台数据库维护模块结构 .....	14

3.4 功能需求与程序代码的关系 .....	14
第 4 章 接口设计 .....	20
4.1 外部接口 .....	20
4.1.1 支付宝接口 .....	20
4.2 内部接口 .....	20
第 5 章 数据结构设计 .....	21
5.1 逻辑结构设计 .....	21
5.1.1 用户管理系统数据结构设计 .....	21
5.1.2 客户端数据结构 .....	21
5.1.3 用户端数据结构 .....	21
5.2 物理结构设计 .....	21
5.3 数据结构与程序模块的关系 .....	21
第 6 章 数据库设计 .....	22
6.1 数据库环境说明 .....	22
6.2 数据库的命名规则 .....	22
6.3 逻辑设计 .....	22
6.4 物理设计 .....	22
6.4.1 数据库产品 .....	22
6.4.2 实体属性、类型、精度 .....	22
6.5 安全性设计 .....	22
6.6 数据库管理与维护说明 .....	23
第 7 章 界面设计 .....	24
7.1 客户端界面 .....	24
7.2 服务器端界面 .....	24
7.3 登录界面 .....	24
7.4 缩略图模式界面 .....	24
第 8 章 出错处理设计 .....	27
8.1 数据库出错处理 .....	27
8.2 某模块失效处理 .....	27

---

第 9 章 安全保密设计 .....	28
9.1 服务器安全性 .....	28
9.2 数据库安全性 .....	28
9.3 网络传输安全性 .....	28
9.4 网络接口安全性 .....	28
9.5 用户信息安全性 .....	28
第 10 章 维护设计 .....	29
第 11 章 图片 .....	30
11.1 示例 .....	30
11.2 带图注的图 .....	30
第 12 章 表格 .....	32
12.1 A Simple Table .....	32
12.2 长表格 .....	32
第 13 章 算法环境 .....	34
第 14 章 代码环境 .....	36
第 15 章 引用文献标注 .....	37
15.1 著者-出版年制标注法 .....	37
15.2 顺序编码制标注法 .....	37
15.3 其他形式的标注 .....	38

## 图目录

3.1 总体流程图 .....	16
3.2 系统基本流程图 .....	17
3.3 客户端基本流程图 .....	18
3.4 服务器端基本流程图 .....	19
7.1 用户初始界面 .....	25
7.2 用户登录界面 .....	25
7.3 缩略图模式界面 .....	26
11.1 测试图片 .....	30
11.2 带图注的图片 .....	31

## 表目录

1 缩略词清单 .....	
1.1 术语表 .....	7
2.1 开发环境的配置 .....	9
2.2 测试环境的配置 .....	10
2.3 运行环境的配置 .....	11
3.1 功能需求与程序代码的关系表 .....	15
5.1 数据结构与程序代码的关系表 .....	21
6.1 用户数据表 Users 设计 .....	23
6.2 订单数据表 Orders 设计 .....	23
12.1 这里是表的标题 .....	32
12.2 长表格演示 .....	32

## 第 1 章 引言

### 1.1 编写目的

在本项目的前一阶段，也就是需求分析阶段，已经将系统用户对本系统的需求做了详细的阐述，这些用户需求已经在上一阶段中对不同用户所提出的不同功能，实现的各种效果做了调研工作，并在需求规格说明书中得到详尽得叙述及阐明。

本阶段已在系统的需求分析的基础上，对 xyz 云盘系统进行设计。主要解决了实现该系统需求的程序模块设计问题。包括如何把该系统划分成若干个模块、决定各个模块之间的接口、模块之间传递的信息，以及数据结构、模块结构的设计等。在以下的概要设计报告中将对在本阶段中对系统所做的所有概要设计进行详细的说明，在设计过程中起到了提纲挈领的作用。

在下一阶段的详细设计中，程序设计员可参考此概要设计报告，在概要设计即时聊天工具所做的模块结构设计的基础上，对系统进行详细设计。在以后的软件测试以及软件维护阶段也可参考此说明书，以便于了解在概要设计过程中所完成的各模块设计结构，或在修改时找出在本阶段设计的不足或错误。

### 1.2 项目背景

随着互联网技术的飞速发展以及广泛应用，云计算这一技术也随之普及。云存储，是近几年在云计算的发展潮流之中诞生的，一项新兴的网络存储技术。云存储集成了网络技术和分布式文件系统等功能，是通过对不同的物理存储设备进行虚拟化映射，以形成逻辑层面统一的大存储空间的应用。

云盘系统，就是利用云存储技术，面向广大的有存储需求的客户，提供数据文件存储服务的第三方托管系统。

我们的 xyz 云盘系统，是基于分布式文件系统来设计和开发的云盘系统，是一个独立的项目。它的命名来自三位开发者名字的首字母（Xiaoniu, Yongzhou, Zewen），表明这将是三位开发者开发的完全不同于其他云盘系统的新兴的云盘系统。



## 1.3 术语

表 1.1 术语表

缩写、术语	解释
CentOS	Community Enterprise Operating System, 社区企业版操作系统
CSS	Cascading Style Sheets, 层叠样式表
HTML	HyperText Markup Language, 超文本标记语言
HTTP	HyperText Transfer Protocol, 超文本传输协议
HTTPS	Hypertext Transfer Protocol Secure, 超文本传输安全协议
IOPS	Input/Output Operations Per Second, 每秒读写操作的次数
IP	Internet Protocol, 网际协议
MD5	Message-Digest Algorithm 5, 讯息摘要演算法 5
TCP	Transmission Control Protocol, 传输控制协议

## 第 2 章 任务概述

本系统的目标是实现一个 xyz 云盘系统，包括客户端、服务器端两个部分。

客户端面向 xyz 云盘用户，为用户提供 p2p 下载、上传、分享、加密、在线解压缩等服务。

### 2.1 目标

实现 xyz 云盘系统，实现需求规格说明书中所描述的 p2p 下载、上传、分享、备份、加密、审核、在线解压缩和预览等功能，并且保证系统的健壮性、高可用性和数据安全。

### 2.2 开发与运行环境

#### 2.2.1 开发环境的配置

如表 2.1 开发环境的配置所示。

#### 2.2.2 测试环境的配置

如表 2.2 测试环境的配置所示。

#### 2.2.3 运行环境的配置

如表 2.3 运行环境的配置所示。

### 2.3 需求概述

功能需求包括：

用户的注册、登陆、退出、忘记密码。

文件的上传、下载、重命名、移动、加密、分享、搜索、在线解压缩、审核、共享文件夹等。

### 2.4 条件与限制

为了完成这个项目，xyz 云盘系统的开发应该在以下条件下展开：

表 2.1 开发环境的配置

类别	标准配置	最低配置
计算机硬件	基于 x86 结构的 CPU 内存 $\geq 8\text{G}$ 硬盘 $\geq 250\text{G}$ 主频 $\geq 2.4\text{GHz}$ 网络带宽 $\geq 100\text{Mbps}$	基于 x86 结构的 CPU 内存 $\geq 4\text{G}$ 硬盘 $\geq 100\text{G}$ 主频 $\geq 1.0\text{GHz}$ 网络带宽 $\geq 10\text{Mbps}$
计算机软件	服务器端:CentOS (version $\geq 7.4$ ) 客户端:Win10(version $\geq 1709$ )	服务器端:CentOS (version $\geq 7.1$ ) 客户端:Win10(version $\geq 1703$ )
其他	Java HotSpot VM 18.3 Tomcat (version $\geq 9$ ) Mysql(version $\geq 5.7$ ) Google Chrome(version $\geq 61$ ) 等浏览器 Ceph(version $\geq 12.2$ )	Java HotSpot VM 18.3 Tomcat (version $\geq 9$ ) Mysql(version $\geq 5.6$ ) Google Chrome(version $\geq 61$ ) 等浏览器 Ceph(version $\geq 12.0$ )

• 开发者掌握足够的开发 xyz 云盘系统的能力，比如前后端代码编写的能力、UI 设计的能力、与客户进行有效沟通的能力等。

• 开发者掌握足够的需要开发 xyz 云盘系统的软硬件环境配置，尤其是开发过程中的软硬件资源、以及运行时的足够的服务器资源。

• 开发者有足够的精力与时间进行 xyz 云盘系统的开发。

• 开发者能够负责后续的项目更新、bug 修复等事宜。

同时，xyz 云盘系统的开发具有如下的限制因素：

• 硬件资源：开发者没有足够的经费以维持足够的服务器硬件开销，尤其是庞大的硬盘开销以及网络带宽开销。

• 开发经验：开发者没有足够的相关应用的开发经验。

• 人力资源：没有足够的人员数量以及开发的时间精力。

• 用户隐私：xyz 云盘系统为了在中国能够合法的运营下去，必须遵守中国的相关法律规定，包括存储必要的用户数据以允许相关部门的合法审查，而这必然会限制用户隐私的绝对保护。

• 安全性依赖：xyz 云盘系统依赖于 Windows、Java、CentOS、Mysql、Ceph、Tomcat、Chrome 等多个第三方开源产品，所以其安全性受到这些第三方产品的

表 2.2 测试环境的配置

类别	标准配置	最低配置
计算机硬件 (服务器端)	内存 >=8G 硬盘 >=250G 主频 >=2.4GHz 网络带宽 >=1GBps	内存 >=4G 硬盘 >= 100G 主频 >=1.0GHz 网络带宽 >=10MBps
计算机硬件 (客户端)	内存 >=2G 硬盘剩余空间 >=1G 主频 >=2.4GHz 网络带宽 >=100MBps	内存 >=1G 硬盘剩余空间 >= 100M 主频 >=1.0GHz 网络带宽 >=10MBps
计算机软件	服务器端:CentOS (version>=7.4) 客户端:Win10(version>=1709)	服务器端:CentOS (version>=7.1) 客户端:Win10(version>=1703)
其他	Java HotSpot VM 18.3 Tomcat (version>=9) Mysql(version>=5.7) Google Chrome(version>=61) 等浏览器 Ceph(version>=12.2)	Java HotSpot VM 18.3 Tomcat (version>=9) Mysql(version>=5.6) Google Chrome(version>=61) 等浏览器 Ceph(version>=12.0)

表 2.3 运行环境的配置

类别	标准配置	最低配置
计算机硬件 (服务器端)	内存 $\geq 8\text{G}$ 硬盘 $\geq 250\text{G}$ 主频 $\geq 2.4\text{GHz}$ 网络带宽 $\geq 1\text{Gbps}$	内存 $\geq 4\text{G}$ 硬盘 $\geq 100\text{G}$ 主频 $\geq 1.0\text{GHz}$ 网络带宽 $\geq 10\text{Mbps}$
计算机硬件 (客户端)	内存 $\geq 2\text{G}$ 硬盘剩余空间 $\geq 1\text{G}$ 主频 $\geq 2.4\text{GHz}$ 网络带宽 $\geq 100\text{Mbps}$	内存 $\geq 1\text{G}$ 硬盘剩余空间 $\geq 100\text{M}$ 主频 $\geq 1.0\text{GHz}$ 网络带宽 $\geq 500\text{KBps}$
计算机软件	服务器端:CentOS (version $\geq 7.4$ ) 客户端:Win10(version $\geq 1709$ )	服务器端:CentOS (version $\geq 7.1$ ) 客户端:Win10(version $\geq 1703$ )
其他	Java HotSpot VM 18.3 Tomcat (version $\geq 9$ ) Mysql(version $\geq 5.7$ ) Google Chrome(version $\geq 61$ ) 等浏览器 Ceph(version $\geq 12.2$ )	Java HotSpot VM 18.3 Tomcat (version $\geq 9$ ) Mysql(version $\geq 5.6$ ) Google Chrome(version $\geq 61$ ) 等浏览器 Ceph(version $\geq 12.0$ )

限制。

- 性能依赖：xyz 云盘系统由于是网盘软件，所以其下载、上传等功能会极大地受到用户自身硬件资源等的限制，所以性能与用户硬件的相关性极大。
- 浏览器依赖：xyz 云盘系统的客户端部署在用户的浏览器中，所以根据浏览器对 Http、Https 等协议的实现不同，其兼容性可能也会有些许偏差。

## 第 3 章 总体设计

### 3.1 软件描述

系统包括前台和后台两个部分。

前台主要功能是：初始化界面的显示、用户的登录相关操作的请求（如输入用户名密码等）、用户的文件相关操作的请求（如选中文件并上传、下载、分享等操作）的控制信息的发送以及数据文件的发送、用户操作的结果显示等。

后台主要功能是：处理用户的输入，判断其权限、其操作是否合法；对于相应的文件操作，进行相应的判断与处理，比如：对于上传以及分享的文件，进行内容审核处理；对文件进行存储冗余处理；并且将处理结果（包括操作的结果以及下载操作对应的数据文件的发送等）返回到客户端。

### 3.2 处理流程

#### 3.2.1 总体流程

总体流程图如图 3.1 所示。总体来说，客户端将用户的请求通过网络发送到服务器端，服务器端对该请求进行检查，审核等处理之后，再执行相关操作，并最终将操作的结果返回到客户端。

#### 3.2.2 系统基本流程

系统基本流程如图 3.2 所示。

#### 3.2.3 客户端基本流程

客户端基本流程如图 3.3 所示。

#### 3.2.4 服务器端基本流程

服务器端基本流程如图 3.4 所示。

#### 3.2.5 功能 1 具体流程

举个例子：交易处理流程

已登录用户在购物车中提交请求交易的 **POST** 请求，提交的表单中指明了交易中包括的所有商品、商家、付款信息、收货地址，输入输出处理系统接收到合法请求后，向商品信息系统请求数据，收到数据以后验证是否正确，然后向订单系统发起生成新订单的请求，订单系统负责更新商品信息系统、商家信息，通知商家接单，返回订单处理结果输入输出处理系统，输入输出处理系统依照结果产生 **HTML** 页面，并返回给用户。

### 3.2.6 功能 2 具体流程

此处应当有描述。

### 3.2.7 功能 3 具体流程

此处应当有一个描述。

## 3.3 功能结构设计

### 3.3.1 整体结构

此处应当有一个图和对应的描述。系统如果像微内核那样，划分成核心模块和若干个子系统，此处应当有图示及说明，然后后续几个节应当描述这几个子系统。如果系统像宏内核，那应当说明有哪些紧密联系的模块，并在后续几个节内描述这些模块。

### 3.3.2 用户端结构

此处应当有一个图和对应的描述。这只是举个例子。可能的内容包括用户端的具体模块、耦合情况等。

### 3.3.3 服务器端结构

此处应当有一个图和对应的描述。这只是举个例子。

### 3.3.4 后台数据库维护模块结构

此处应当有一个图和对应的描述。这只是举个例子。

## 3.4 功能需求与程序代码的关系

[此处指的是不同的需求分配到哪些模块去实现。可按不同的端拆分此表]



表 3.1 功能需求与程序代码的关系表

·	模块 1	模块 2	模块 3
需求 1	·	Y	·
需求 2	·	Y	·
需求 3	·	Y	·
需求 4	Y	·	·
需求 5	·	·	Y

注：各项功能需求的实现与各个程序模块的分配关系

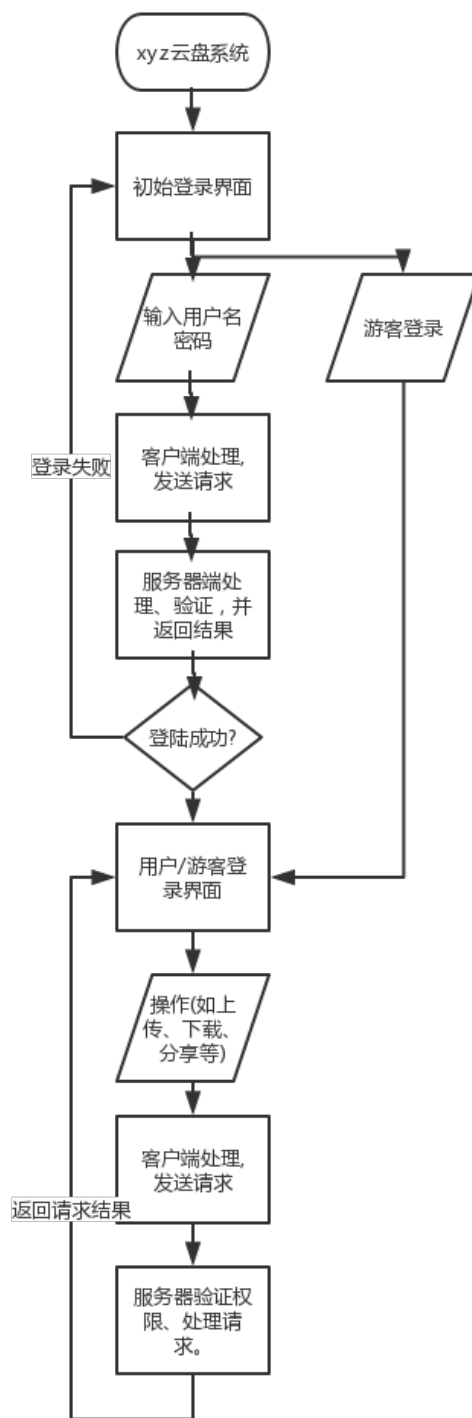


图 3.1 总体流程图

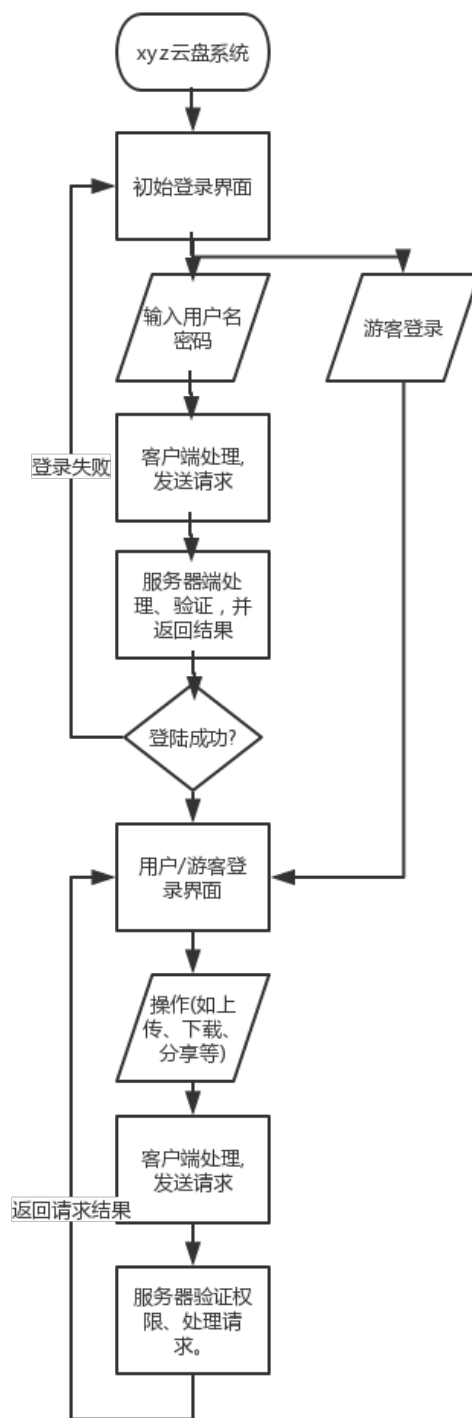


图 3.2 系统基本流程图

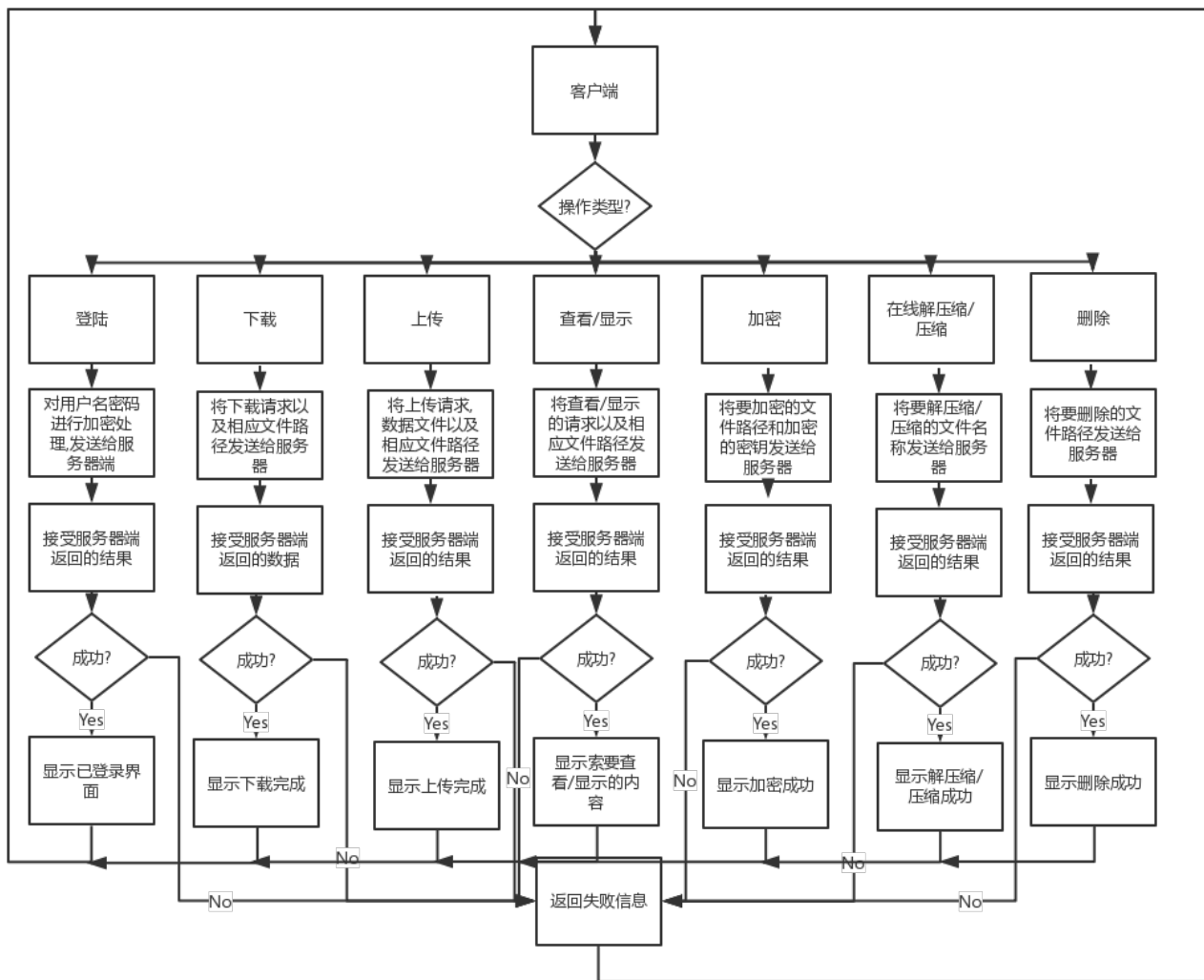


图 3.3 客户端基本流程图

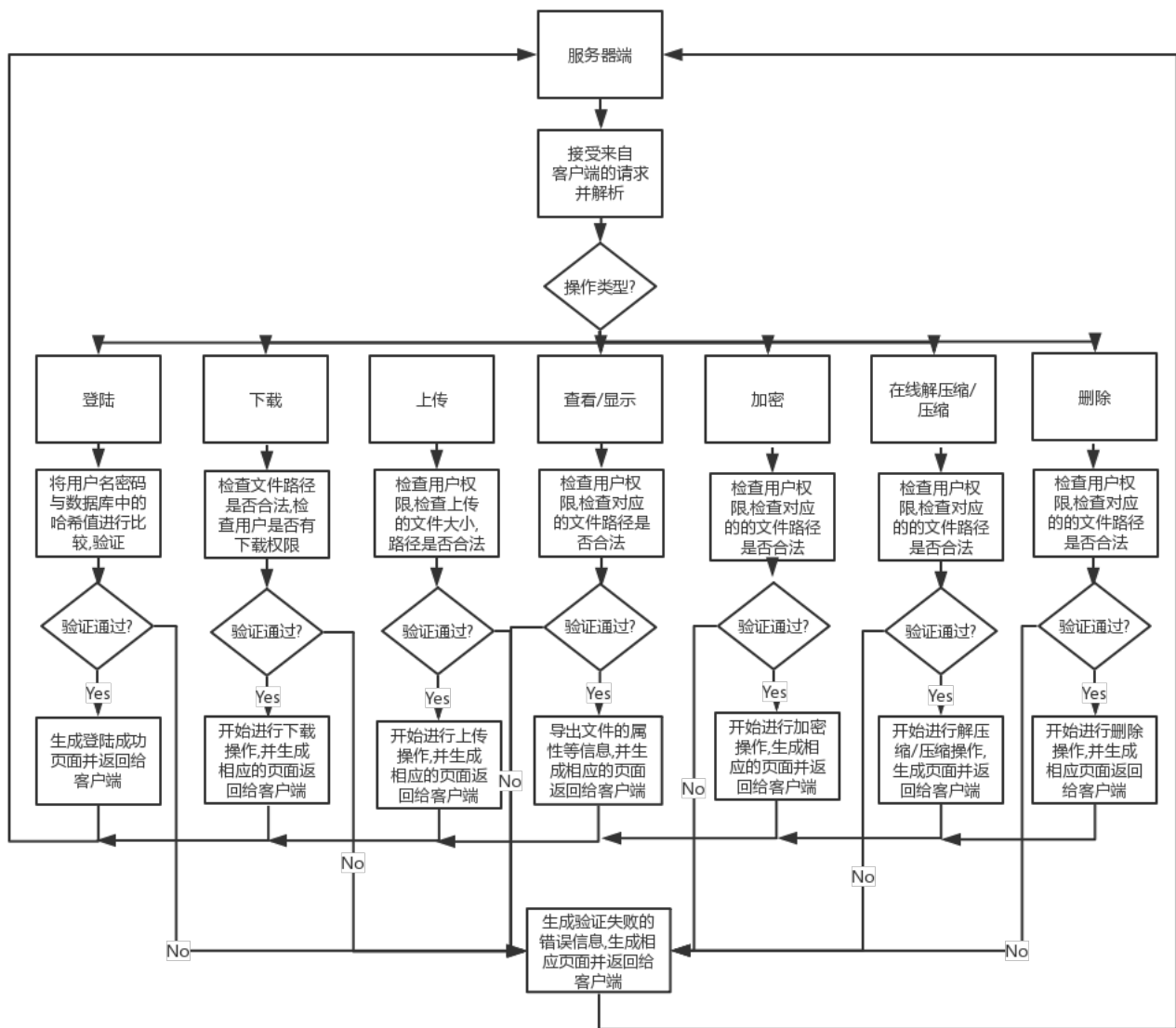


图 3.4 服务器端基本流程图

## 第 4 章 接口设计

### 4.1 外部接口

比如说需要用到支付宝等外部支付系统，接口应当如何封装。

#### 4.1.1 支付宝接口

详细讲述不同的接口（查询状态、支付交易、获取回执等）

### 4.2 内部接口

内部模块/系统之间的交互的接口。

## 第 5 章 数据结构设计

### 5.1 逻辑结构设计

#### 5.1.1 用户管理系统数据结构设计

讲述本系统内需要什么数据结构。这指的是程序运行过程中维护的数据结构。只是举个例子，此处应和 3.3 一致。

#### 5.1.2 客户端数据结构

#### 5.1.3 用户端数据结构

### 5.2 物理结构设计

各数据结构无特殊物理结构要求。（如果有，比如说 hadoop 等，应当具体说明）

### 5.3 数据结构与程序模块的关系

[此处指的是不同的数据结构分配到哪些模块去实现。可按不同的端拆分此表]

表 5.1 数据结构与程序代码的关系表

·	模块 1	模块 2	模块 3
结构 1	·	Y	·
结构 2	·	Y	·
结构 3	·	Y	·
结构 4	Y	·	·
结构 5	·	·	Y

注：各项数据结构的实现与各个程序模块的分配关系

## 第 6 章 数据库设计

### 6.1 数据库环境说明

本系统的数据系统采用 MySQL/PostgreSQL/Microsoft SQL Server 数据库系统。

其中 xxx 模块因为 xxx 而需要用到 Hadoop 架构。

### 6.2 数据库的命名规则

是否允许单词缩写，允许的单词缩写有哪些。

表名是单数还是复数。关联表如何命名。字符数限制等。

字段是否带上前缀（如 integer 类型则加上 i 前缀等）。

### 6.3 逻辑设计

是否需要满足某一种范式。

画个实体的逻辑关系表/图在此处。

### 6.4 物理设计

#### 6.4.1 数据库产品

用哪家数据库，是否分布式等。

#### 6.4.2 实体属性、类型、精度

##### 6.4.2.1 客户数据表设计

##### 6.4.2.2 订单数据表设计

### 6.5 安全性设计

备份和容灾设计。



表 6.1 用户数据表 Users 设计

字段名	类型	大小	说明	备注
ID	char	64	用户的唯一标识符	主键
pw	char	512	用户的登录密码	.

注：用户数据表 Users 设计

表 6.2 订单数据表 Orders 设计

字段名	类型	大小	说明	备注
ID	char	64	订单的唯一标识符	主键
user	char	64	对应用户	外键，来自 xx 表

注：订单数据表 Orders 设计

## 6.6 数据库管理与维护说明

对于数据库的维护，随时对数据库中的信息加以调试和保存备份。同样需要个工作人员进行系统的分析和用户的反馈，对系统进行升级以及功能的完善。同时保证系统安全有序的运行。

## 第 7 章 界面设计

### 7.1 客户端界面

### 7.2 服务器端界面

由于服务器端由管理员使用命令行进行交互，因此无界面图

### 7.3 登录界面

### 7.4 缩略图模式界面

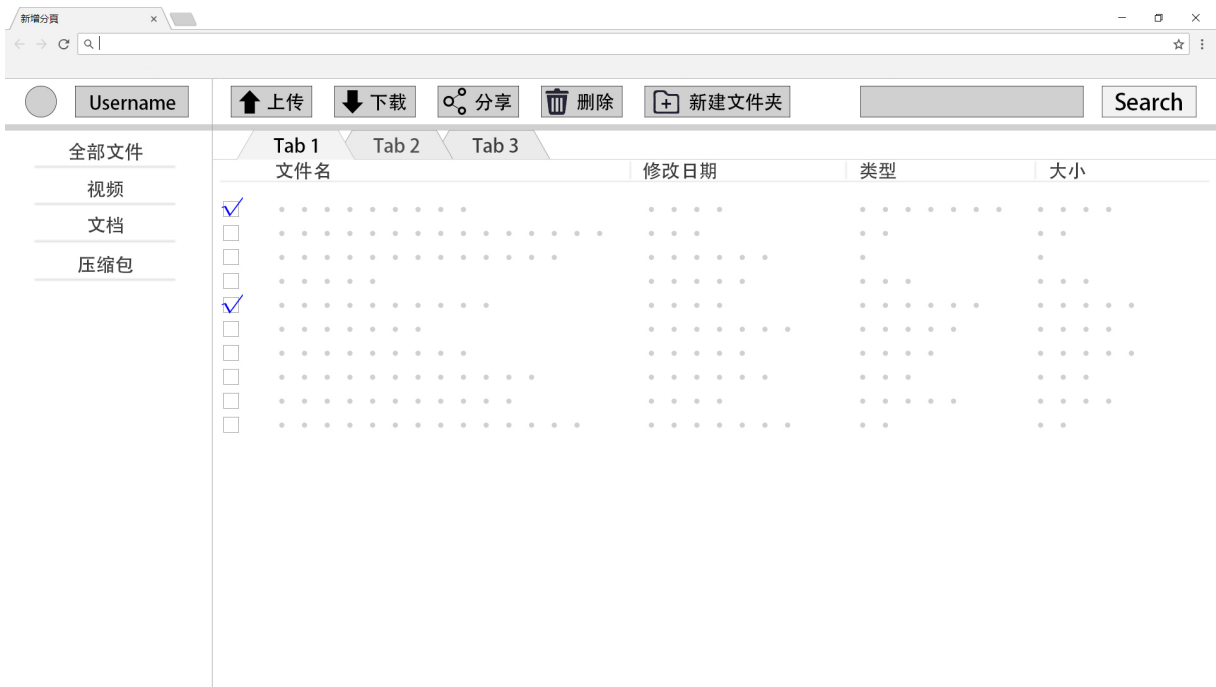


图 7.1 用户初始界面



图 7.2 用户登录界面



图 7.3 缩略图模式界面

## 第 8 章 出错处理设计

### 8.1 数据库出错处理

本云盘系统仅针对于用户信息以及其权限建立了数据库，其十分轻量，改变量比较小，故采用同步的，完全备份的方式。当数据库出错时，切换至有备份的另一台服务器继续提供该功能，无需暂停在线服务，待主服务器恢复后切换回来提供服务。

### 8.2 某模块失效处理

模块失效时依据失效模块的功能取决我们的处理方式：若核心模块如登陆模块、数据库连接与请求、底层文件系统或者 HTTP 请求模块失效，则应暂停整个系统的服务，在客户端提示维护信息，并由该模块的开发人员为主，在其他开发人员支持下调整失效模块，同时注意调整与该模块相关的其他模块的接口，测试完成后将系统整体上线。若非核心模块如传输、分享，则先关闭失效模块提供的服务以及相关接口提供的服务，其余不相关的服务维持状态，之后由该模块的开发人员为主，在其他开发人员支持下调整失效模块，同时注意调整与该模块相关的其他模块的接口，测试完成后将要修改的所有模块一起上线。

## 第 9 章 安全保密设计

### 9.1 服务器安全性

系统运行所在的服务器为 CentOS 系统，定期使用 yum 包管理器进行软件升级以及进行系统版本升级，保证及时更新安全补丁。服务器禁止 root 用户登陆；普通用户的用户名、密码的长度、字符集均做限制；管理员使用公钥私钥登录服务器。服务器运行防火墙软件，禁止非 HTTP/HTTPS/ssh 的端口链接

### 9.2 数据库安全性

数据库只允许 localhost 链接。对于部分固定的信息，数据库只提供只读权限，同时禁止删除数据库等危险操作

### 9.3 网络传输安全性

本系统全站使用 HTTPS 安全连接，保证服务器于浏览器之间传输是安全加密的。

### 9.4 网络接口安全性

所有 API 都对 SQL 注入进行过滤，保证安全性

### 9.5 用户信息安全性

用户使用用户名与密码进行验证，忘记密码可以通过邮箱找回密码用户输错密码时延迟提示，连续 3 此密码错误则要求输入验证码，降低暴力破解可能性用户密码在数据库中不保存明文，而是保存加盐的散列值，即使数据斜率也不能得到用户密码

## 第 10 章 维护设计

维护设计主要为数据库的维护功能，包括数据库的日常备份、压缩、维护，以及文件的备份与恢复。数据库的具体备份、压缩功能的实现主要交由 MySQL 的功能来完成。备份的主要策略为：每次操作由主数据库完成之后，备份数据库向主数据库 **fetch** 更新。备份的恢复方式为将备份数据库的内容 **copy** 到主数据库。对于文件系统的备份，由 **ceph** 这一高可用的文件系统来完成，其内部已有 **replication** 以及灾害恢复的功能，无需在外部另作备份。

## 第 11 章 图片

本章展示图片相关用法。

### 11.1 示例



图 11.1 测试图片

### 11.2 带图注的图





图 11.2 带图注的图片

注：the solid lines represent the time histogram of the spontaneous activities of an old monkey cell(gray) and a young monkey cell (black). The bin-width is 1

## 第 12 章 表格

### 12.1 A Simple Table

表 12.1 这里是表的标题

a	b
c	d

注：这里是表的注释

### 12.2 长表格

表 12.2 长表格演示

名称	说明	备注
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC
AAAAAAAAAAAAA	BBBBBBBBBBBBB	CCCCCCCCCCCCCCC

续下页

表 12.2 长表格演示 (续)

[illegible]

## 第 13 章 算法环境

模板中使用 `algorithm2e` 宏包实现算法环境。关于该宏包的具体用法，请阅读宏包的官方文档。

```
Data: this text

Result: how to write algorithm with LATEX2ε

1 initialization;
2 while not at end of this document do
3   read current;
4   if understand then
5     go to next section;
6     current section becomes this one;
7   else
8     go back to the beginning of current section;
9   end
10 end
```

算法 13.1: 算法示例 1

```

input : A bitmap  $Im$  of size  $w \times l$ 
output: A partition of the bitmap

1 special treatment of the first line;
2 for  $i \leftarrow 2$  to  $l$  do
3   special treatment of the first element of line  $i$ ;
4   for  $j \leftarrow 2$  to  $w$  do
5      $\text{left} \leftarrow \text{FindCompress}(Im[i, j - 1]);$ 
6      $\text{up} \leftarrow \text{FindCompress}(Im[i - 1, j]);$ 
7      $\text{this} \leftarrow \text{FindCompress}(Im[i, j]);$ 
8     if left compatible with this then //  $\text{O}(\text{left}, \text{this}) == 1$ 
9       if  $\text{left} < \text{this}$  then  $\text{Union}(\text{left}, \text{this});$ 
10      else  $\text{Union}(\text{this}, \text{left});$ 
11    end
12    if up compatible with this then //  $\text{O}(\text{up}, \text{this}) == 1$ 
13      if  $\text{up} < \text{this}$  then  $\text{Union}(\text{up}, \text{this});$ 
14      // this is put under up to keep tree as flat
        as possible
15      else  $\text{Union}(\text{this}, \text{up});$ 
16      // this linked to up
17    end
18  end
19  foreach element  $e$  of the line  $i$  do  $\text{FindCompress}(p);$ 
20 end

```

算法 13.2: 算法示例 2

## 第 14 章 代码环境

模板中使用 listings 宏包实现代码环境。详细用法见宏包的官方说明文档。

以下是代码示例，可以在文中任意位置引用??。

代码 14.1 示例代码

```
1 #include <stdio.h>
2
3 int main( )
4 {
5     printf("hello, \world\n");
6     return 0;
7 }
```

## 第 15 章 引用文献标注

### 15.1 著者-出版年制标注法

<code>\citestyle{ustcauthoryear}</code>	
<code>\cite{knuth86a}</code>	$\Rightarrow$ ?
<code>\citet{knuth86a}</code>	$\Rightarrow$ ?
<code>\citet[chap.~2]{knuth86a}</code>	$\Rightarrow$ ?, chap. 2
<code>\citep{knuth86a}</code>	$\Rightarrow$ (?)
<code>\citep[chap.~2]{knuth86a}</code>	$\Rightarrow$ (?, chap. 2)
<code>\citep[see][ ]{knuth86a}</code>	$\Rightarrow$ (see ?)
<code>\citep[see][chap.~2]{knuth86a}</code>	$\Rightarrow$ (see ?, chap. 2)
<code>\citet*{knuth86a}</code>	$\Rightarrow$ ?
<code>\citep*{knuth86a}</code>	$\Rightarrow$ (?)
<code>\citet{knuth86a,tlc2}</code>	$\Rightarrow$ ??
<code>\citep{knuth86a,tlc2}</code>	$\Rightarrow$ (??)
<code>\cite{knuth86a, knuth84}</code>	$\Rightarrow$ ??
<code>\citet{knuth86a, knuth84}</code>	$\Rightarrow$ ??
<code>\citep{knuth86a, knuth84}</code>	$\Rightarrow$ (??)

### 15.2 顺序编码制标注法

`\citestyle{ustcnumerical}`

---

<code>\cite{knuth86a}</code>	$\Rightarrow$	<code>[?]</code>
<code>\citet{knuth86a}</code>	$\Rightarrow$	<code>?]</code>
<code>\citet[chap.~2]{knuth86a}</code>	$\Rightarrow$	<code>? , chap. 2]</code>
<code>\citep{knuth86a}</code>	$\Rightarrow$	<code>[?]</code>
<code>\citep[chap.~2]{knuth86a}</code>	$\Rightarrow$	<code>[?] chap. 2</code>
<code>\citep[see][ ]{knuth86a}</code>	$\Rightarrow$	<code>see[?]</code>
<code>\citep[see][chap.~2]{knuth86a}</code>	$\Rightarrow$	<code>see[?] chap. 2</code>
<code>\citet*{knuth86a}</code>	$\Rightarrow$	<code>?]</code>
<code>\citep*{knuth86a}</code>	$\Rightarrow$	<code>[?]</code>
<code>\citet{knuth86a,tlc2}</code>	$\Rightarrow$	<code>? ?]</code>
<code>\citep{knuth86a,tlc2}</code>	$\Rightarrow$	<code>[? ?]</code>
<code>\cite{knuth86a, knuth84}</code>	$\Rightarrow$	<code>[? ?]</code>
<code>\citet{knuth86a, knuth84}</code>	$\Rightarrow$	<code>? ?]</code>
<code>\citep{knuth86a, knuth84}</code>	$\Rightarrow$	<code>[? ?]</code>
<code>\cite{knuth86a, knuth84,tlc2}</code>	$\Rightarrow$	<code>[? ? ?]</code>

### 15.3 其他形式的标注

<code>\citealt{tlc2}</code>	$\Rightarrow$	<code>?</code>
<code>\citealt*{tlc2}</code>	$\Rightarrow$	<code>?</code>
<code>\citealp{tlc2}</code>	$\Rightarrow$	<code>?</code>
<code>\citealp*{tlc2}</code>	$\Rightarrow$	<code>?</code>
<code>\citealp{tlc2, knuth86a}</code>	$\Rightarrow$	<code>??</code>
<code>\citealp[pg.~32]{tlc2}</code>	$\Rightarrow$	<code>? pg. 32</code>
<code>\citenum{tlc2}</code>	$\Rightarrow$	<code>?</code>
<code>\citetext{priv.\ comm.}</code>	$\Rightarrow$	<code>[priv. comm.]</code>
<code>\citeauthor{tlc2}</code>	$\Rightarrow$	<code>?</code>
<code>\citeauthor*{tlc2}</code>	$\Rightarrow$	<code>?</code>
<code>\citeyear{tlc2}</code>	$\Rightarrow$	<code>?</code>
<code>\citeyearpar{tlc2}</code>	$\Rightarrow$	<code>?</code>