

高级软件开发——云盘系统设计文档

Koorye

2023 年 10 月 3 日

目录

1	摘要	2
1.1	编写目的	2
1.2	阅读对象	2
1.3	术语解释	2
2	概要设计	3
2.1	系统架构	3
2.2	模块设计	3
2.2.1	用户服务模块	3
2.2.2	文件服务模块	6
3	详细设计	7
3.1	数据表结构	7
3.2	算法流程	9
3.2.1	登录流程	10
3.2.2	文件上传流程	11

1 摘要

1.1 编写目的

编写该文档的目的是对云盘系统进行详尽的设计，包括概要和详细设计，以便后续的开发工作顺利进行。

1.2 阅读对象

本文档适用于参与该项目的开发人员、项目经理和测试人员。

1.3 术语解释

本文档种的术语解释如表 1所示。

表 1: 术语解释

术语	解释
WebDav	一种基于 HTTP 的文件操作协议
B/S	浏览器-服务端架构
C/S	客户端-服务端结构
容灾	系统处理危机的能力
SQL	一种用于数据库交互的脚本语言
MySQL	一个开源的数据库系统
Oracle	一个商业数据库系统
SQLServer	一个由微软开发的数据库系统
RPC	一种用于远程函数调用的通道协议
MD5	一种哈希散列算法
SHA256	一种哈希散列算法
盐	拼接在密码后的一串字符，用于增加破译难度
JWT	一种分布式 token 的生成方法

2 概要设计

本章节将描述云盘系统的概要设计，包括系统架构和模板设计。

2.1 系统架构

云盘系统总体采用浏览器-服务端 (B/S) 或客户端-服务端 (C/S) 架构，而服务端内部采用微服务结构，由网关（转发）子模块、WebDav 子模块、文件存储子模块、用户存储子模块等微服务共同组成。其中每个子模块可配置单独的集群和容灾机制。

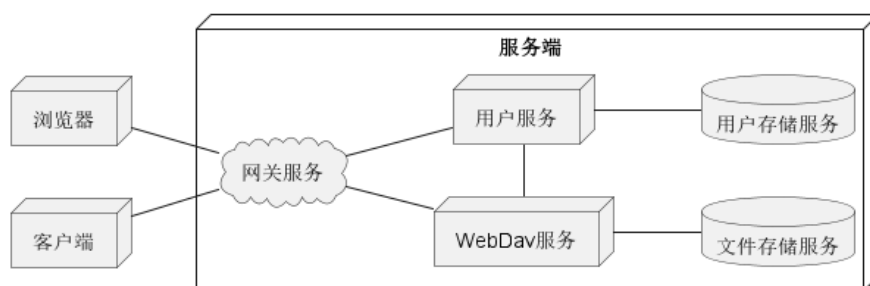


图 1: 系统架构

本系统的总体架构如图 1所示，浏览器或客户端可通过网关访端务端，网关会将请求转发给用户服务或 WebDav 服务。其中用户服务负责处理用户请求，如登录、注册等；而 WebDav 服务负责处理文件请求，如文件的上传、下载、移动、删除、预览等。

2.2 模块设计

本章节将描述云盘系统中的主要模块设计，包括用户服务模块和文件服务模块。

2.2.1 用户服务模块

用户服务模块包括用户服务和用户存储（数据库）服务，用户服务和用户存储服务之间通过内部网络通信。

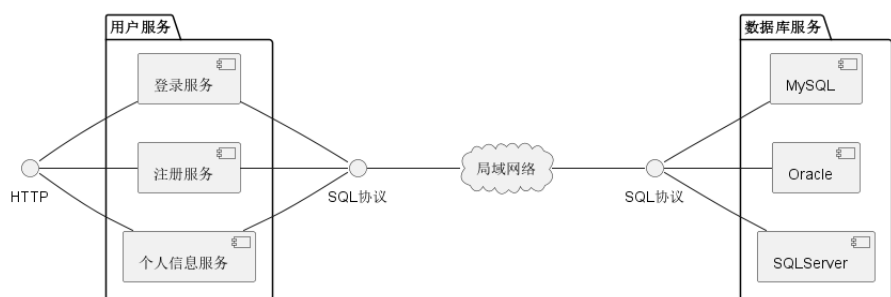


图 2: 用户服务模块

如图 2所示，用户服务中包含登录、注册、个人信息处理等业务，用户服务通过 HTTP 协议接收请求，之后通过 SQL 协议和局域网络与数据库进行交互。

考虑到与数据库的交互需要适配不同类型的数据库,如 MySQL、Oracle、SQLServer 等，逻辑较为复杂，这里提出几种设计模式来抽象为该步骤，以便之后的实现。

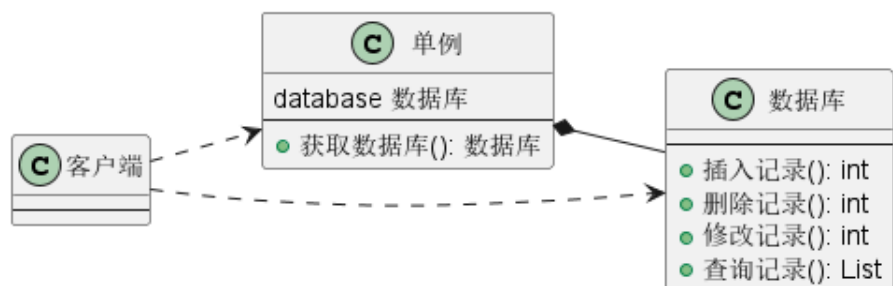


图 3: 基于单例模式的数据库服务实现

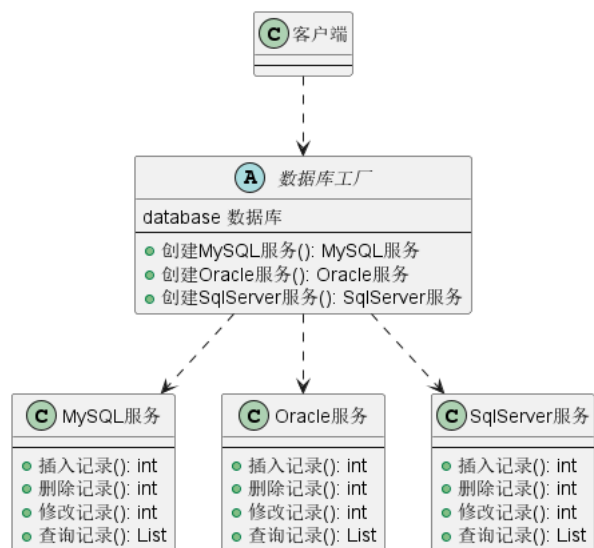


图 4: 基于工厂模式的数据库服务实现

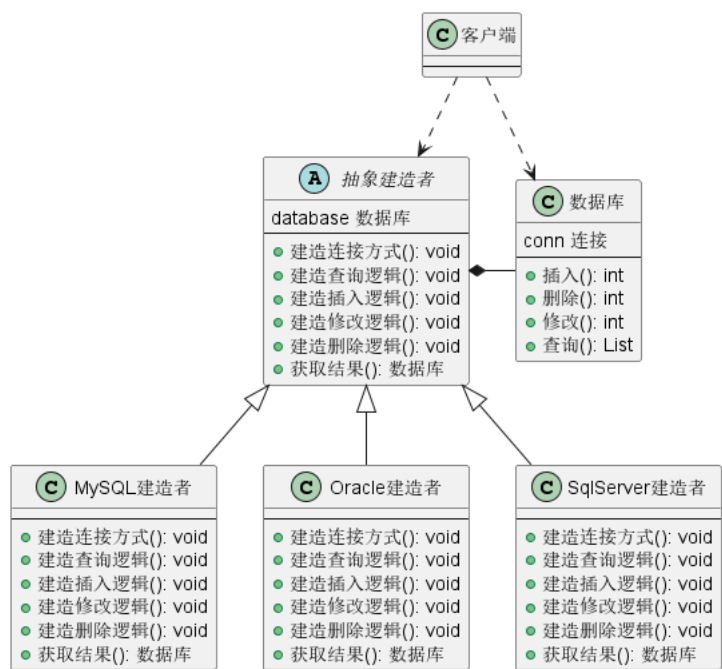


图 5: 基于建造者模式的数据库服务实现

这里提出了 3 种数据库服务的设计模式，如图 3,4,5所示，分别为单例模式、工厂模式和建造者模式：

- 单例模式。数据库服务仅为整个系统创建一个单例，保证所有线程使用相同的服务，避免不同线程造成冲突。
- 工厂模式。通过一个数据库工厂来创建基于不同数据库系统的数据库服务，保证该模块可以适配不同的数据库系统。
- 建造者模式。将数据库服务的创建逻辑分解为多个步骤，以简化操作。

事实上，在代码实现时，可以采用多个设计模式的混合策略。如工厂中的每个创建方法都采用建造者模式逐步构建；其次可以引入单例模式，如果服务已经被创建，就返回现有服务而不要创建新的服务。

2.2.2 文件服务模块

文件服务模块包括 WebDav 服务和文件存储服务，其中文件存储可以是本地的文件系统，也可以是远程的文件系统。

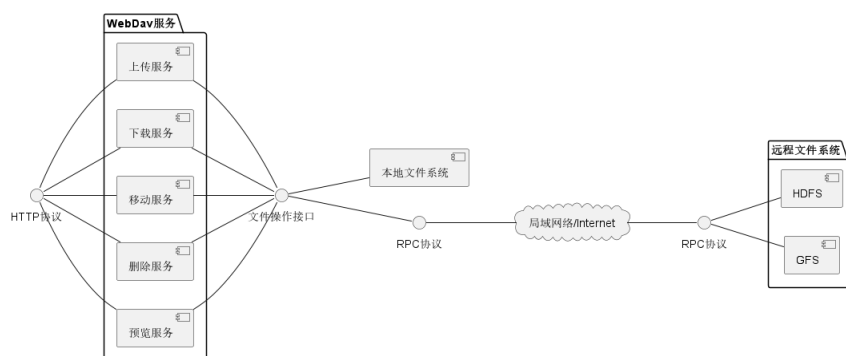


图 6: 文件服务模块

文件服务模块的架构如图 6所示，用户请求通过 HTTP 协议发送给 WebDav 服务，WebDav 处理文件的上传、下载、删除、移动、预览等请求，并于文件系统进行交互。

考虑到该模块需要支持不同的文件系统，本系统进行了高度的抽象，将本地/远程的文件系统交互抽象为一个接口，在此基础上实现本地文件系统的交互逻辑和基于 RPC 协议的远程文件系统交互逻辑。该部分同样可以采用单例、工厂、建造者等设计模式，与上一部分大致相同。

3 详细设计

本章节将描述云盘系统的详细设计，包括数据表结构和算法流程。

3.1 数据表结构

本章节将描述云盘系统的数据表结构。

表 2: 用户表

名称	类型	必填	特殊	描述
用户 ID	bigint	是	主键	用户的唯一标识符。
用户名	varchar(20)	是	-	用户的名称
密码	varchar(20)	是	-	用户的密码
手机号	varchar(15)	是	-	用户的手机号（包括区号）
电子邮箱	varchar(100)	否	-	用户的电子邮箱
生日	date	否	-	用户的生日，包括年-月-日
角色	short	是	-	用户的角色，0 表示普通用户，1 表示管理员用户

用户表结构如表 2所示，用户表中包含唯一标识符、用户名、密码，以及手机号、电子邮箱等描述信息。

表 3: 文件表

名称	类型	必填	特殊	描述
文件 ID	bigint	是	主键	文件的唯一标识符
文件名	varchar(100)	是	-	文件的名称
文件类型	varchar(20)	是	-	文件类型，即后缀名
文件大小	long	是	-	文件大小，以 KB 为单位
hash 值	char(64)	是	-	哈希值，用于检测文件已经存在
创建时间	datetime	是	-	文件创建的时间
本地路径	varchar(200)	是	-	存储在本地文件系统中的路径或对象存储的 key
所属用户 ID	bigint	是	外键	所属用户的唯一标识符

文件表结构如表 3 所示，文件表中包含唯一标识符、文件名、类型、大小、hash 值、创建时间、本地路径和所属用户 ID 的外键信息。

表 4: 共享表

名称	类型	必填	特殊	描述
共享 ID	bigint	是	主键	分享的唯一标识符
属性	short	是	-	共享属性，0 表示私人，1 表示公共
密码	varchar(8)	否	-	分享链接的密码，如果有
截止时间	datetime	是	-	分享链接的有效期限
共享文件 ID	bigint	是	外键	共享文件的唯一标识符
接收用户 ID	bigint	否	外键	接收用户的唯一标识符，如果有

共享表结构如表 4 所示，共享表中包含唯一标识符、属性、密码、截止时间和文件 ID、用户 ID 的外键信息。

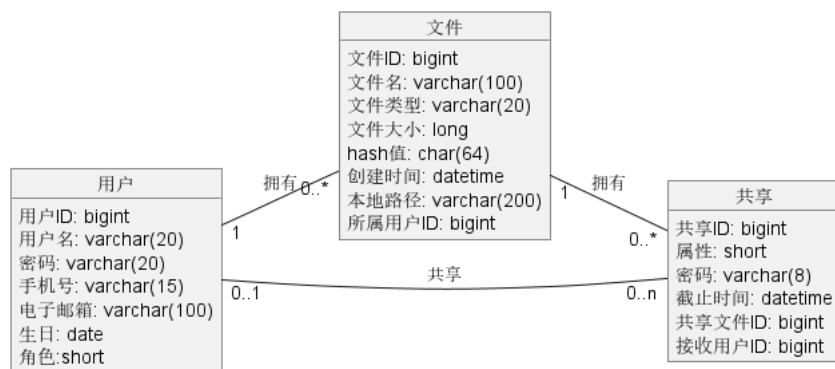


图 7: 数据表结构

数据表结构和关系如图 7 所示，其中用户可以拥有 0 至多个文件，一个文件只能属于一个用户，如果不同用户拥有的文件完整相同，将为其创建链接。一个共享链接只能指定一个文件，当模式为私人时，该共享只属于一个用户；否则，该文件属于所有公共用户。

3.2 算法流程

本章节将描述详细一些主要部分的算法流程。

3.2.1 登录流程

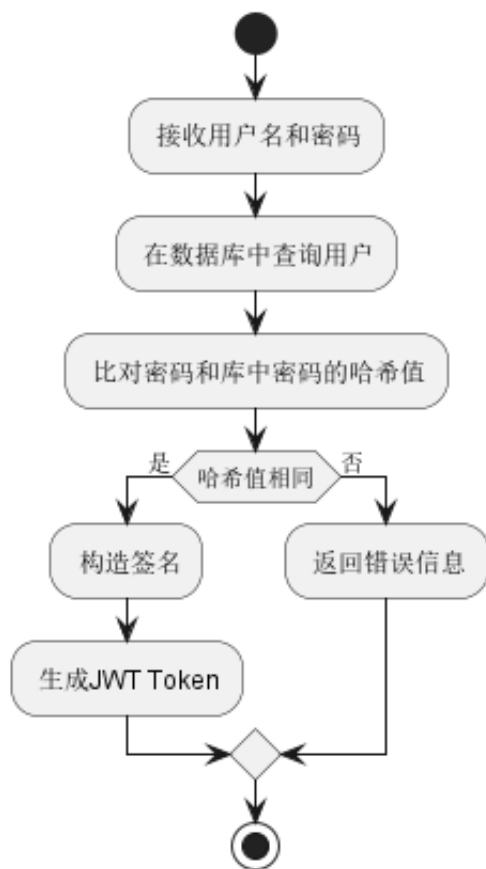


图 8: 登录流程

登录流程如图 8所示，图中展示了一些细节：

- 密码不能采用明文保存，而是采用 MD5/SHA256 等哈希算法加密后保存，之后在比对密码时也需要比对其哈希值，避免密码原文被泄露。
- 在比对通过后，将返回一个 JWT Token，该 token 不需要在服务端存储，而是可以直接将一些用户信息，如盐、用户名、过期时间等信息写入并加密。当服务器收到一个 token 后，会尝试进行解码，只有解码成功才能拿到该用户的操作权限。

3.2.2 文件上传流程

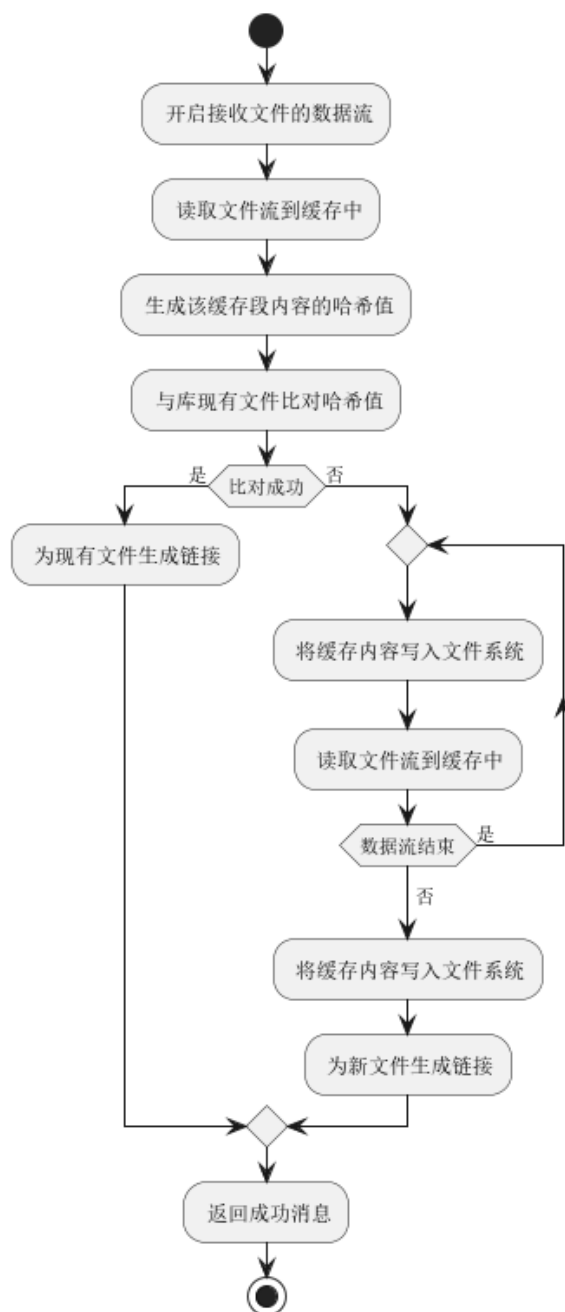


图 9: 上传流程

文件上传流程如图 9所示，图中展示了一些细节：

- 文件上传时首先会进行比对，具体来说，当读到文件的第一个数据流之后，就将其哈希值与库中保存的哈希值进行比对。如果有相同的值，则视为该文件已经存在，此时只需要对现有文件生成链接，避免文件的重复上传。
- 当文件不存在时，才需要写入文件系统并持续接收数据流直到关闭。