**概要设计说明书**

**目 录**

[1. 引言 1](#_Toc93963567)

[1.1 编写目的 1](#_Toc93963568)

[1.2 背景 1](#_Toc93963569)

[1.3 定义 1](#_Toc93963570)

[2. 总体设计 3](#_Toc93963571)

[2.1 需求规定 3](#_Toc93963572)

[2.2 运行环境 5](#_Toc93963573)

[2.3 基本设计概念和处理流程 5](#_Toc93963574)

[2.4 结构 6](#_Toc93963575)

[2.5 程序技术架构设计 6](#_Toc93963576)

[2.6 人工处理过程 6](#_Toc93963577)

[3. 接口设计 7](#_Toc93963578)

[3.1 登录模块接口 7](#_Toc93963579)

[3.2文件模块接口 7](#_Toc93963580)

[3.3 区块链模块接口 7](#_Toc93963581)

[4.运行设计 8](#_Toc93963582)

[4.1 运行模块组合 8](#_Toc93963583)

[4.2 运行控制 8](#_Toc93963584)

[4.3 运行时间 8](#_Toc93963585)

[5.系统数据结构设计 9](#_Toc93963586)

[5.1 逻辑结构设计要点 9](#_Toc93963587)

[5.2 物理机构设计要点 9](#_Toc93963588)

[6.系统出错处理设计 10](#_Toc93963589)

[6.1 出错信息 10](#_Toc93963590)

[6.2 补救措施 10](#_Toc93963591)

[6.3 系统维护设计 10](#_Toc93963592)

# 引言

## 1.1 编写目的

本说明书是重复理解系统需求分析基础上，为后续工作准备的。本手册将进行技术评审和技术可行性检查，它是详细设计和编码的重要根据，为下一步的详细设计说明提供框架同时也是用户进行交流的文档之一。软件开发小组的每一位参与开发成语应该阅读本说明，以清除产品在技术方面的要求和实现策略。预期读者为系统开发者、测试人员。

## 1.2 背景

说明：

1. 开发软件系统名称： 测试文件上传上链系统
2. 任务提出者： 林翰
3. 开发者： 林翰
4. 用户： XX
5. 本系统将使用Mysql8.0作为数据库存储系统

## 1.3 定义

采用的技术介绍

1. 后端采用技术：
   1. Java语言进行开发
   2. Spring-boot架构
   3. Nginx 服务器负载均衡
   4. Spring-cloud-eureka服务注册中心
   5. Spring-cloud-gateway网关控制
   6. Minio分布式存储架构
   7. Mysql8 数据库
2. 前端采用技术：
   1. HTML5、CSS
   2. Vue框架
3. 区块链采用技术：
   1. FISCO-BCOS3.0 rc1

# 总体设计

## 需求规定

#### 系统功能

1. **登录注册功能**
2. 注册： 用户必须认真填写个人信息以确保信息准确无误
3. 登录： 登录后的用户可以查看和修改自己的个人信息，并进行系统操作
4. 上传文件： 登录后的用户可以进行文件的上传和选择是否将文件内容进行上链
5. 个人信息： 查看自己注册信息及保存在区块链上的文件信息
6. **文件上传**
7. 文件上传： 提供文件上传保存至数据库
8. 文件下载： 提供文件从数据库内下载的功能
9. **文件信息上区块链**
10. 文件上链： 支持将文件MD5值上传至区块链
11. 文件查询： 支持查询文件是否在区块链上

#### 系统性能

1. **精度**
2. 在执行数据增删修改时，不允许因为程序的原因导致增删操作失败，且不能重复执行增删操作
3. 数据的修改保持对应的准确性
4. **时间特性要求**
5. 系统的更新处理时间应该在可接受的范围内
6. 系统的数据查询时间应该在可接受的范围内
7. 系统的数据统计时间应该在可接受的范围内
8. **可靠性**

本系统在一定时间内的实际应用率是可靠的

1. **灵活性**
2. 操作上满足鼠标和键盘任意切换的需求

#### 输入输出要求

1. 输入项： 系统要求的用户注册信息、用户名、所需要上传的文件
2. 输出项： 注册页面、登录页面、用户个人信息页面、上传文件页面、查询文件上链信息页面

#### 数据管理能力要求

1.对于用户的历史数据进行永久保存，需要对数据进行备份

2.保证数据的安全，数据库内部采用加密算法

#### 故障处理要求

1. 用户在输入一些不合理的数据时候，能够进行合理的提示信息，不能因为输入错误而导致系统的错误，或程序停止运行
2. 程序运行时，对服务器和网络通信的故障能识别并提示，当故障排除后，程序恢复正常运行
3. 数据需要有灾难备份机制，防止数据的全部丢失

#### 其他专门要求

1. 本系统设计用户文件上链保存，因此系统要保证数据的一致性、完整性、准确性的要求达到99%左右

2） 本系统基于FISCO BCOS区块链采用联盟链机制，大大提升区块链的稳定，和数据安全性，能高效的对订单进行处理

## 运行环境

#### 2.2.1 支持软件

1） 支持Chrome浏览器

2） 支持的数据库： Mysql8

3） 本系统的开发工具： Vscode、Intelij IDEA 2.1.0、Navicat Premium 15、remix ide

#### 权限控制

通过网关模块对于用户访问不同模块的权限控制

## 图示 描述已自动生成基本设计概念和处理流程

## 日程表 描述已自动生成结构

## 图示 描述已自动生成程序技术架构设计

## 2.6 人工处理过程

定期备份数据库，维护数据的完整性、一致性

# 3. 接口设计

## 3.1 登录模块接口

对于登录模块接口设计需要满足一下几点要求。用户登录时，对于用户的输入密码进行校验。并对多次输入错误的用户进行限制。用户注册时，需要对用户填写的个人信息进行校验，由于系统的特殊原因，用户的个人信息唯一所以接口需要对此进行验证。对于用户输入的密码等私密信息在进行数据库保存时进行加密处理。

## 3.2文件模块接口

满足用户对于文件上传需求。需要满足多文件上传，并对文件提取MD5等加密算法的唯一码，并根据用户的个人信息生成文件对应唯一的文件id，且同一用户同一文件生成的文件id相同。处理完文件信息后对于文件原本内容进行储存。

## 3.3 区块链模块接口

用户选择上链时，能将文件进行上链存储。由于区块链特殊性，本系统选择将文件的MD5进行上传，并采用KV存储方式，key为文件的id，value为文件的MD5值。并支持用户进行查询，提供文件原本和用户信息进行查询，或提供文件id查询文件是否上链。

# 4.运行设计

## 4.1 运行模块组合

**前端：**

前端界面在有输入时启动后端模块。通过用户选择和输入读入，将调用网络传输模块，将数据通过网络送到服务器，并等待接受服务器的返回信息。接收到回调后随即调用数据输出模块，对于回调信息进行处理，产生对应的输出。

**后端：**

微服务程序的接受网络数据模块保存运行状态。接到数据后，调用数据处理模块进行业务处理，与数据库完成交互进行增删改查，完成后调用网络发送模块，将信息返回到客户机上。

调用登录模块注册时，向数据库内写入用户名和密码，并生成全局唯一的用户id。调用登录模块时，通过用户的登录获取token，进行权限控制。调用文件模块时，进行文件上传感觉登录模块查询用户信息，将用户id和文件信息存入数据库，根据用户信息和文件信息生成文件id。调用区块链模块，通过文件模块获取文件的id和MD5值将其上链，并显示上链成功的交易信息如交易哈希和当前区块数。

## 4.2 运行控制

运行控制严格按照个模块间函数调用关系来运行。在各事务中心模块中，需对运行控制进行正确的判断，选择正确的运行控制路径。

## 4.3 运行时间

在软件的需求中，对于运行时间的要求必须对做出的操作有及时的反应，操作延迟小于2ms内。

# 5.系统数据结构设计

## 5.1 逻辑结构设计要点

图示

描述已自动生成1）建立数据库的视图表格

## 5.2 物理机构设计要点

各个模块阐述需要什么表和对业务数据进行表的拆分，及对应的库

# 6.系统出错处理设计

## 6.1 出错信息

**登录模块：**

登录输入错误的密码。 提示用户：用户名密码错误

注册时注册以有用户。 提示用户：当前用户已存在

**文件模块：**

上传时文件存储失败，服务器磁盘不够。 提示用户：请联系管理员

上传时文件类型不符合需求。 提示用户：文件类型错误

**区块链模块：**

上链未达成共识，继续递归调用，超过3次。 提示用户：请联系管理员

**网络错误或访问不存在页面**：

提示404界面

## 6.2 补救措施

当数据库遭到损坏入用户非法使用数据库、硬盘损坏、病毒侵入等。可以通过下面方法进行补救：

1. 定期维护数据库，根据信息更新修改数据库信息，保持数据库最近更新。可使用mysql的数据库维护功能机制。如根据事务日记文件和数据库备份机制以及回滚机制进行数据库维护。
2. 使用降效技术：将磁盘祖册磁盘阵列（RAID阵列）即镜像磁盘，当主硬盘奔溃时，镜像硬盘可以代替主硬盘运行工作

## 6.3 系统维护设计

必须按照软件产品设计的步骤进行开发，并充分考虑可维护性，详细编写各阶段的文档资料，以利于移植或二次开发。

定期维护数据库，根据信息更新数据库信息，保持数据最近最新。定期为数据进行Backup，维护管理数据库死锁问题和维护数据库内数据的一致性等。

设计三种日志：系统运行日志、操作日志、出错日志