吉首大学



码

客

之

家

概

要

设

计

文

档

|  |  |
| --- | --- |
| 成 员： | 周鑫 |
| 班 级： | 20计科一班 |
| 学 号： | 2020404101 |
| 时 间： | 2023年6月30日 |

目录

[1. 定义系统目标 2](#_Toc257572221)

[1.1目标 2](#_Toc1836769642)

[1.2范围 3](#_Toc509947469)

[2. 划分系统模块 3](#_Toc79876306)

[2.1用户管理 3](#_Toc303795567)

[2.2评论管理 3](#_Toc1323465650)

[2.3反馈和举报功能 4](#_Toc1999047571)

[3. 设计系统架构 4](#_Toc610868482)

[3.1架构风格 5](#_Toc1894744314)

[3.2主要组件 5](#_Toc2080167682)

[4. 定义数据结构和数据流 5](#_Toc344458214)

[4.1 数据库设计 5](#_Toc1840774033)

[4.2数据流向 5](#_Toc1239753949)

[5. 设计系统界面 5](#_Toc1658277649)

[5.1设计软件选型 6](#_Toc629675977)

[5.2界面设计原则 6](#_Toc164733023)

[6. 考虑系统性能 6](#_Toc561496578)

[6.2响应时间 6](#_Toc1029841528)

[6.3吞吐量 6](#_Toc1975849923)

[6.4资源利用率 6](#_Toc1570022300)

[7. 安全和可靠性设计 6](#_Toc1233225411)

[7.1数据保护 6](#_Toc1454805480)

[7.2身份验证 6](#_Toc1814381265)

[7.3错误验证 7](#_Toc38133455)

[8. 进行系统集成 7](#_Toc958851379)

[8.1持续集成 7](#_Toc697839765)

[8.2持续部署 7](#_Toc1184734088)

[9. 系统文档编写 7](#_Toc357442032)

[9.1架构设计 7](#_Toc1016471165)

[9.2模块和类设计 7](#_Toc598458270)

[9.3 UI设计 7](#_Toc1622224989)

[9.4数据库设计 7](#_Toc283007811)

[9.5接口设计 7](#_Toc1983485019)

1. 定义系统目标

1.1目标

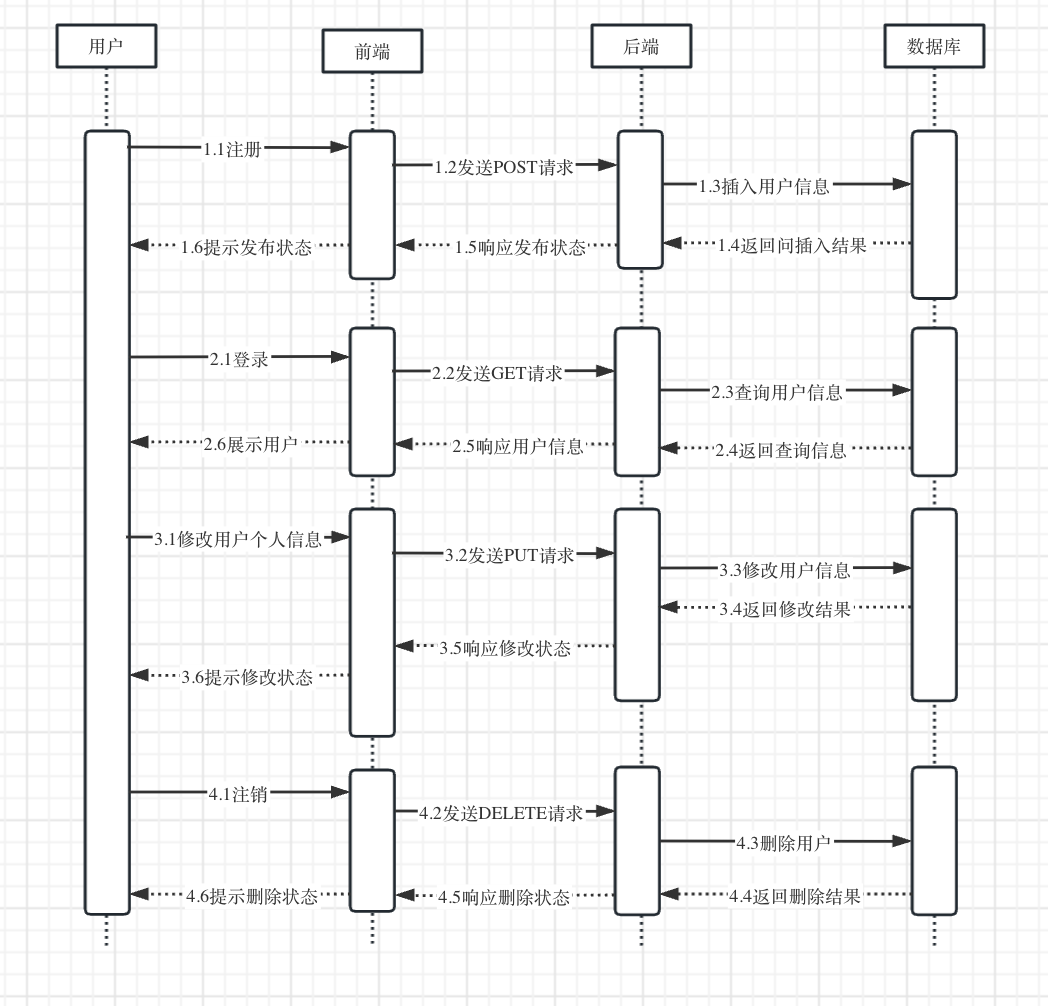
打造内容优质，资源丰富的计算机知识技术交流平台。

1.2范围

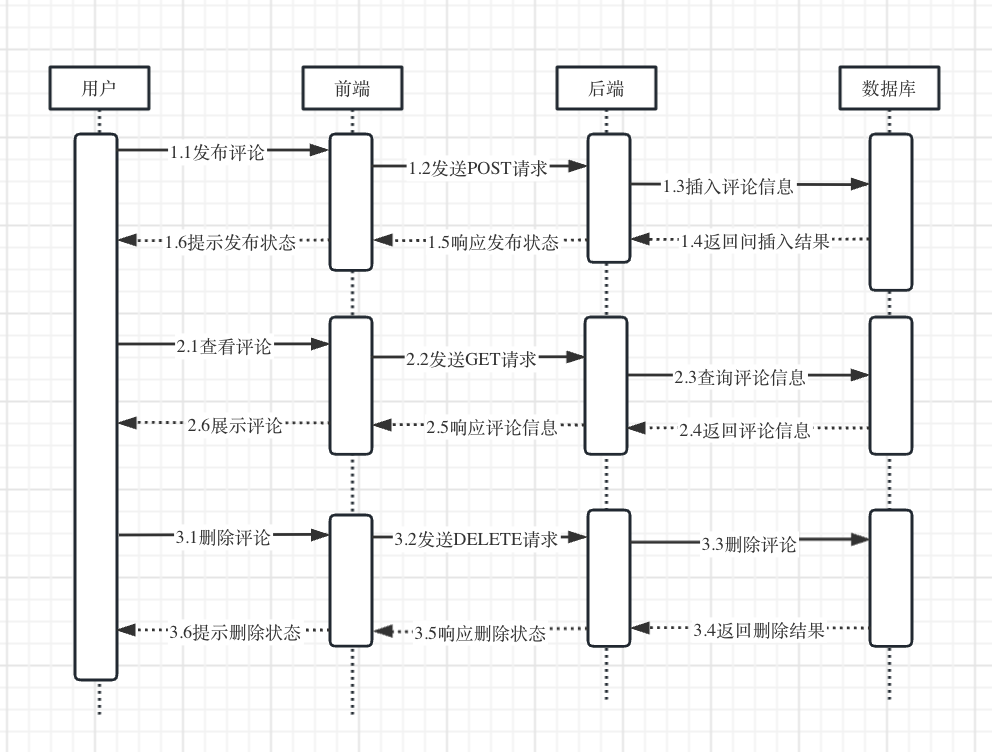
设计和实现博客平台的核心功能，包括用户认证、博客文章管理、评论功能等。

1. 划分系统模块

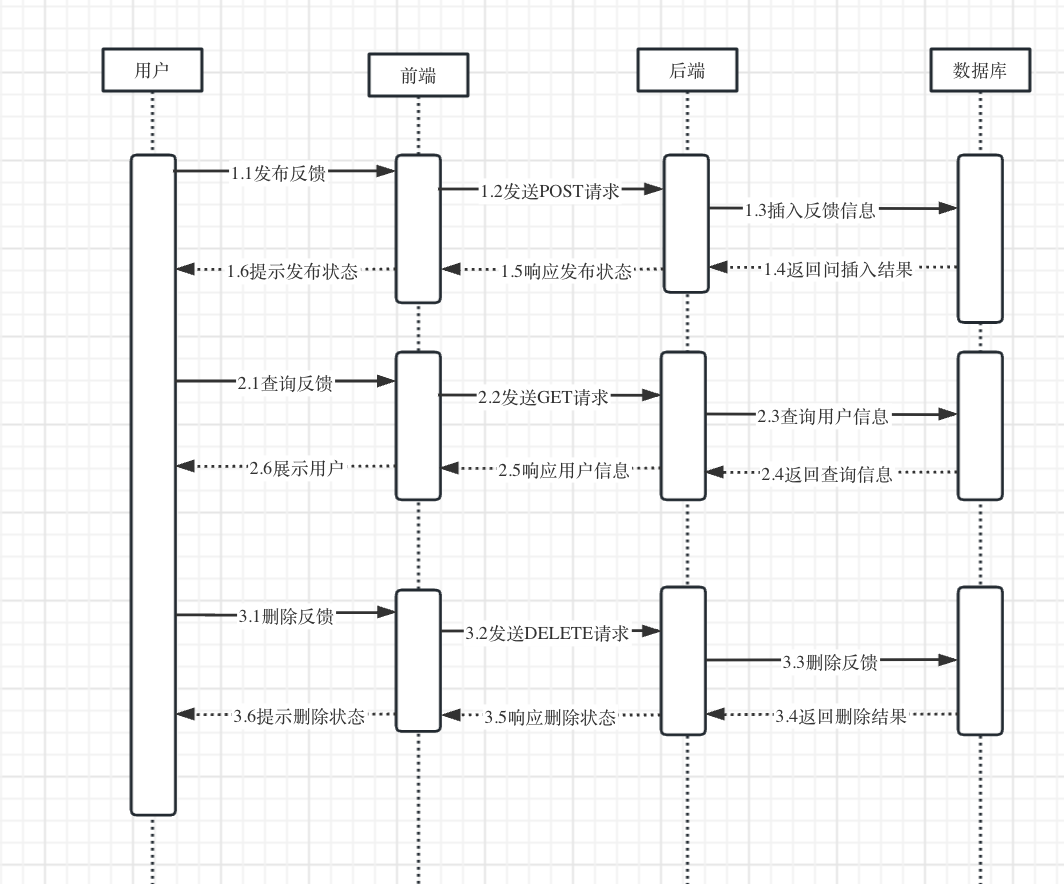
2.1用户管理



2.2评论管理



2.3反馈和举报功能



1. 设计系统架构

3.1架构风格

采用B/S（浏览器-服务器）架构模式，客户端为浏览器，服务端为云服务器。同时首次开发的应用为单体架构的应用。

3.2主要组件

前端客户端

后端服务器

运维测试部署

数据库

1. 定义数据结构和数据流

4.1 数据库设计

4.1.1数据库概念结构设计

1. 案例的局部E-R模型设计
2. 案例的全局E-R模型设计

4.1.2数据库逻辑结构设计

1. 实体关系模式设计
2. 联系关系模式设计

4.1.3数据库物理结构设计

1. 确定物理结构
2. 设计数据库表

4.2数据流向

用户通过前端界面发送请求到后端服务器，后端服务器根据请求处理数据，并将结果返回给前端。

1. 设计系统界面

5.1设计软件选型

工具和功能丰富，界面布局直观，样式和视觉效果良好，支持协作与版本管理，支持各种输出与导入。

5.2界面设计原则

采用响应式设计，适应不同设备的屏幕大小,使用统一的设计风格、颜色方案和字体样式，以提供一致的用户体验。提供友好的错误提示，帮助用户快速解决问题。

1. 考虑系统性能

6.2响应时间

优化前端界面和后端服务器的响应时间，提升用户体验。

6.3吞吐量

确保服务器能够处理大量用户并发请求，提供稳定的性能。

6.4资源利用率

有效管理服务器资源，避免资源浪费和性能下降。

1. 安全和可靠性设计

7.1数据保护

采用加密算法保护用户密码和敏感数据的存储。

7.2身份验证

采用加密算法保护用户密码和敏感数据的存储。

7.3错误验证

合理处理异常情况，提供友好的错误提示和恢复机制。

1. 进行系统集成

8.1持续集成

使用版本控制系统（如Git）将代码变更提交到共享仓库，并通过自动化的构建工具（如Jenkins、Travis CI、Github Actions）实现自动化构建、编译和测试等过程。持续集成可以帮助团队及时发现和解决代码集成引入的问题，提高开发效率和代码质量。

8.2持续部署

在持续部署中，通过自动化构建和测试过程的集成，团队可以自动化地将经过验证的软件版本推送到生产环境，减少了人工操作和潜在的错误。持续部署可以加快软件交付的速度，实现快速迭代和反馈。

1. 系统文档编写

9.1架构设计

9.2模块和类设计

9.3 UI设计

9.4数据库设计

9.5接口设计