软件架构文档（SAD）

小组成员信息

|  |  |
| --- | --- |
| 姓名 | 学号 |
| 郭汝震 | 202200130144 |
| 崔程皓 | 202200130181 |
| 张高俊 | 202200130217 |
| 王奕澄 | 202200460093 |
| 郝嘉琦 | 202200130030 |

文档分工：

1引言,2架构背景：郝嘉琦、张高俊

3视图：崔程皓、王奕澄

4需求与架构之间的映射：郭汝震

目录

[软件架构文档（SAD） 1](#_Toc10320)

[1. 引言 3](#_Toc13223)

[1.1 目的与范围 3](#_Toc25359)

[1.2 文档结构 3](#_Toc14273)

[2. 架构背景 5](#_Toc19288)

[2.1系统概述 5](#_Toc27710)

[2.2架构需求 5](#_Toc1149)

[2.2.1 技术环境需求 6](#_Toc24864)

[2.2.2 数据采集 8](#_Toc22662)

[2.2.3 设计决策 8](#_Toc4633)

[2.2.4 功能模块 8](#_Toc28413)

[3.视图 12](#_Toc16547)

[3.1 逻辑视图 12](#_Toc17197)

[3.1.1顶层逻辑视图 12](#_Toc971)

[3.1.2用户模块逻辑视图 14](#_Toc7468)

[3.1.3文章管理模块逻辑视图 15](#_Toc25480)

[3.1.4评论管理模块逻辑视图 16](#_Toc11371)

[3.1.5管理员后台模块逻辑视图 17](#_Toc17295)

[3.2 开发试图 18](#_Toc6055)

[3.2.1顶层开发视图 18](#_Toc16901)

[3.3 运行视图 19](#_Toc13671)

[3.3.1 顶层运行视图 19](#_Toc4222)

[3.4 部署视图 20](#_Toc31089)

[3.4.1主表示 20](#_Toc16932)

[4. 需求与架构之间的映射 21](#_Toc18136)

[4.1核心功能： 21](#_Toc8197)

[4.2辅助功能： 21](#_Toc5013)

[4.3软件系统总体功能/对象结构 22](#_Toc23761)

[4.3.1总体结构 22](#_Toc29127)

[4.3.2软件子系统功能/对象结构 23](#_Toc2965)

[5.附录 24](#_Toc29708)

[5.1架构元素索引 24](#_Toc6449)

[5.2术语表 24](#_Toc19861)

[5.3缩略语 24](#_Toc18076)

# 1. 引言

## 1.1 目的与范围

本文件描述了博客系统的软件架构，涵盖设计决策、架构视图以及需求与架构组件之间的映射。该系统支持用户注册、文章发布、评论、点赞和关注功能，作为一个具有 FAST API后端的网页应用。

## 1.2 文档结构

本文档旨在全面描述系统的架构设计，涵盖从高层次概述到具体实现细节的各个方面。文档结构如下：

**1. 架构概述**

* **系统背景**：介绍系统的业务背景、核心目标和主要功能，说明系统在整体业务或技术生态中的定位。
* **技术环境**：描述系统运行的技术栈，包括硬件、软件、网络环境以及依赖的第三方服务或框架。
* **设计决策**：阐述关键架构设计的选择及其依据。

**2. 架构视图**

架构视图是描述系统架构的核心部分，展现不同维度的设计。以下是具体视图的详细说明：

**逻辑视图**

* **定义**：描述系统的功能组件及其相互关系，关注“系统做什么”。
* **内容**：
  + 顶层逻辑视图
  + 用户模块逻辑视图
  + 文章管理模块逻辑视图
  + 评论管理模块逻辑视图
  + 管理员后台模块逻辑视图
* **用途**：帮助开发人员理解系统功能边界和设计一致性。

**开发视图**

* **定义**：描述系统的代码组织、开发工具和模块化结构，关注“如何构建系统”。
* **内容**：
  + 顶层开发视图
  + 构件目录
* **用途**：指导开发团队实现代码管理和协作。

**运行视图**

* **定义**：描述系统运行时动态行为，关注“系统如何工作”。
* **内容**：
  + 关键流程的动态交互。
  + 并发处理。
  + 异常处理机制。
* **用途**：确保系统在运行时满足性能、可靠性和扩展性需求。

**部署视图**

* **定义**：描述系统物理部署结构，关注“系统运行在何处”。
* **内容**：
  + 硬件/云资源配置。
  + 网络拓扑。
  + 依赖的中间件。
* **用途**：为运维团队提供部署和运维的蓝图。

**3. 需求映射**

* 将业务需求或用户故事映射到具体的架构组件，说明每个需求如何通过架构实现。

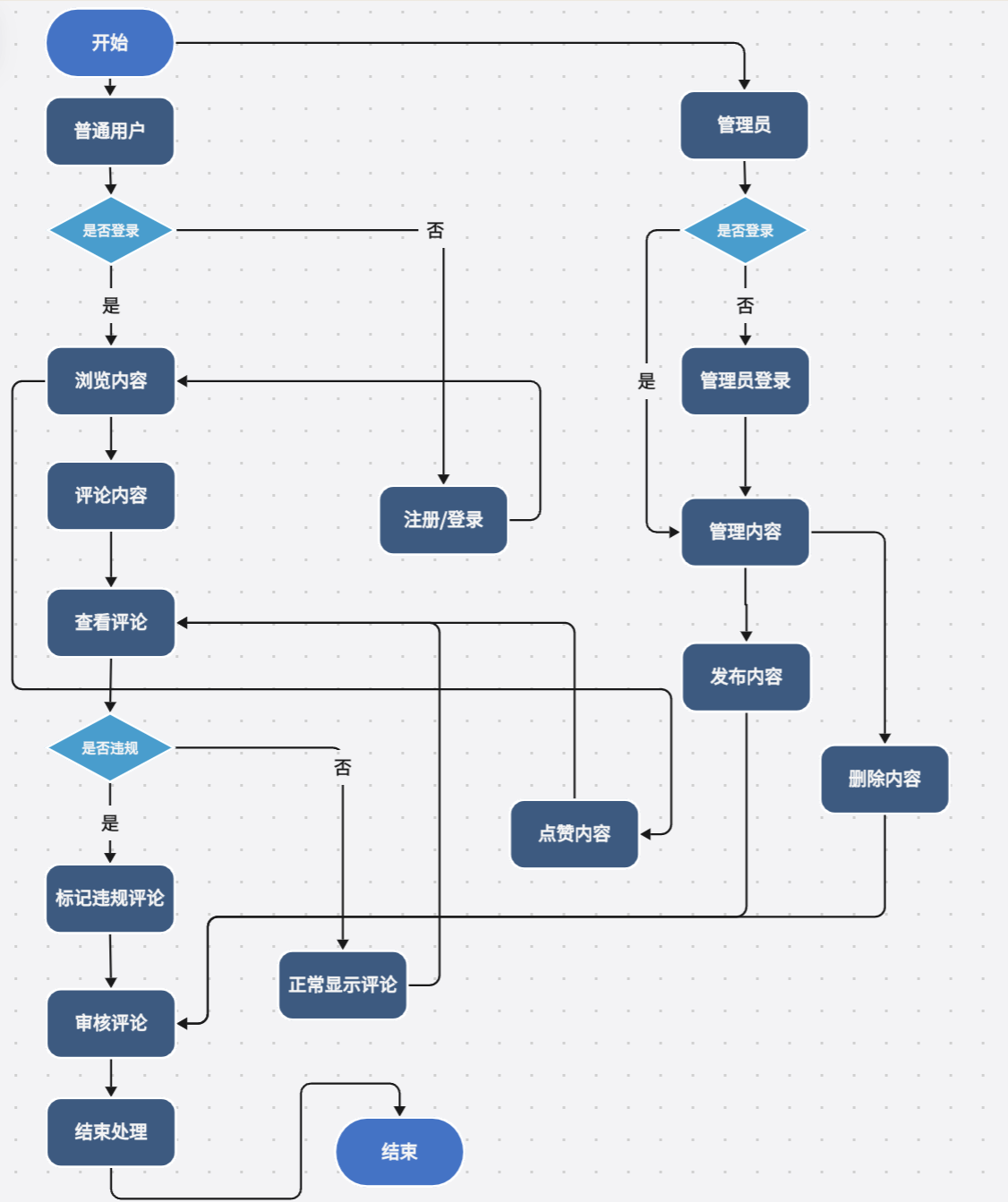
**4. 附录**

# 2. 架构背景

## 2.1系统概述

博客系统采用客户端-服务器架构，前端可部署为 Web 应用或微信小程序，后端基于 FAST API，使用python开发，数据库采用 MySQL。

## 2.2架构需求



### 2.2.1 技术环境需求

技术环境需求主要包括数据需求，涵盖系统运行所需的静态和动态数据，以及数据字典和数据采集方式。

#### 静态数据

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 设置值 | 定义 | 格式 | 类型 |
| sql | SELECT \* FROM ... | 数据库查询语句 | String | String |
| finalPath | /image/user/... | 用户上传图像存储位置 | String | String |
| maxUploadSize | 5MB | 用户上传图片的最大限制 | SizeFormat | String |
| defaultAvatar | /static/avatar.png | 新注册用户默认头像路径 | String | String |
| siteName | "My Blog" | 系统前端显示的站点名称 | String | String |

#### 动态数据

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 设置值 | 定义 | 格式 | 类型 |
| articleCount | 由系统实时统计 | 用户已发布文章的数量，更新频繁 | Integer | Int |
| commentCount | 实时增减 | 用户或全站评论总数 | Integer | Int |
| likedNum | 动态增加 | 文章或评论被点赞的次数 | Integer | Int |
| publishDate | 系统当前时间戳 | 用户发布文章或评论的时间 | yyyy-MM-dd | Date |
| sessionToken | 登录成功时生成 | 用户当前登录状态的 Token，用于身份校验 | JWT String | String |
| activeUser | 用户名列表 | 当前在线用户列表，由服务器维护 | String[] | Array |
| tempUploadPath | /tmp/image/... | 临时存储用户上传的图像，定期清理 | String | String |
| editHistory | JSON 格式内容快照 | 用户文章的历史版本数据（用于撤回或恢复） | JSON | Text |

#### 数据词典

表1：login\_info（登入信息表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 中文说明 | 数据类型 | 允许为空 | 主键 | 备注 |
| user\_name | 用户名 | varchar(20) | Not Null | Yes | 登录名，外键关联多个表 |
| account\_number | 账号 | varchar(20) | Not Null | Yes | 登录账号标识 |
| password | 密码 | varchar(20) | Not Null | No | 加密存储推荐 |

表2：user\_info（用户信息表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 中文说明 | 数据类型 | 允许为空 | 主键 | 备注 |
| user\_name | 用户名 | varchar(20) | Not Null | Yes | 外键关联 login\_info |
| sex | 性别 | varchar(10) | Not Null | No | 如 "男"/"女" |
| mail | 邮箱 | varchar(20) | Not Null | No | 电子邮件地址 |
| icon | 头像 | varchar(20) | Not Null | No | 图片文件名 |
| article\_num | 文章数量 | varchar(10) | Not Null | No | 可设为数字型 |
| comment\_num | 评论数量 | varchar(10) | Not Null | No |  |
| commented\_count | 被评论数量 | int | Not Null | No | 用户提醒用途 |
| user\_age | 用户使用时间 | date | Not Null | No | 可为 NULL 或默认值 |
| authority | 管理权限 | int | Not Null | No | 0 普通用户，1 管理 |

表3：article\_info（文章信息表）

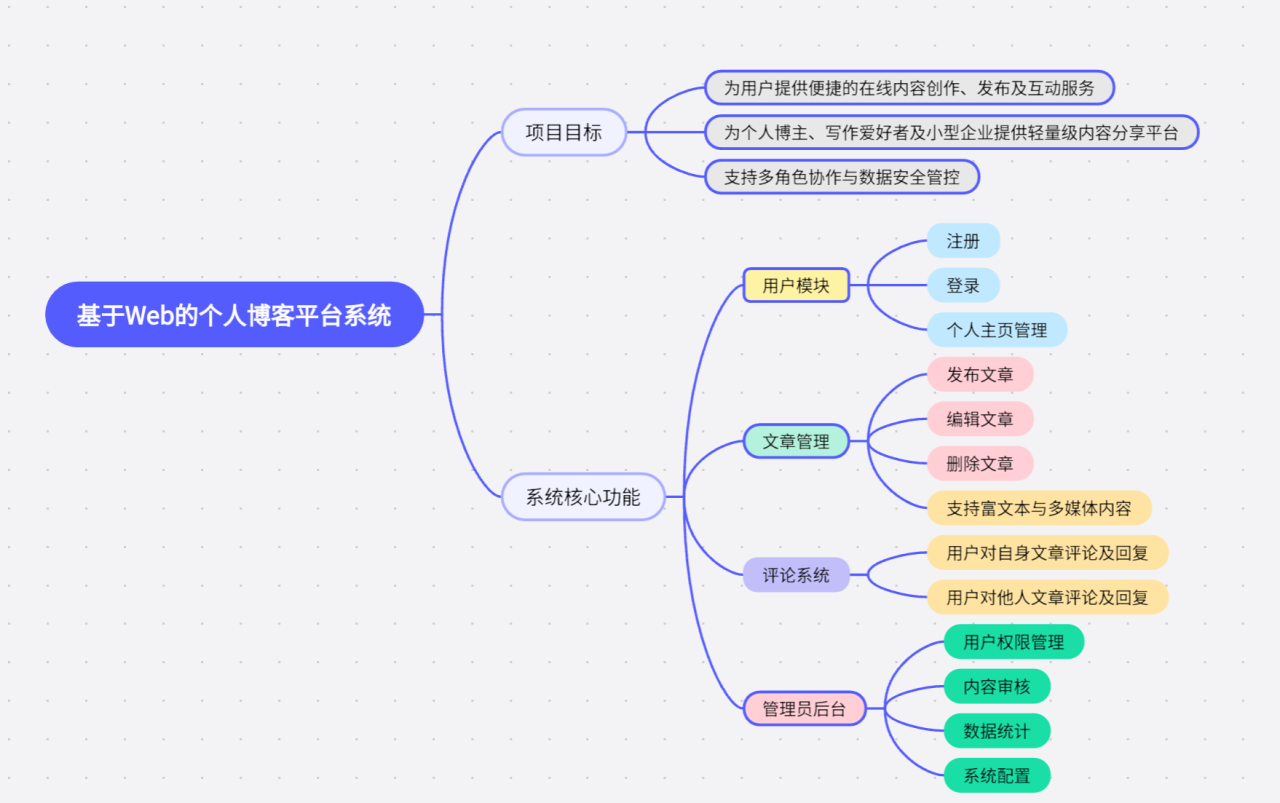
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 中文说明 | 数据类型 | 允许为空 | 主键 | 备注 |
| user\_name | 用户名 | varchar(20) | Not Null | Yes | 外键关联 login\_info |
| icon | 头像 | varchar(20) | Not Null | No |  |
| article\_name | 文章名 | varchar(20) | Not Null | No |  |
| picture | 图片路径 | varchar(200) | Null | No | 可多个路径拼接 |
| article\_content | 文章内容 | varchar(1000) | Not Null | No | 长文本建议使用 TEXT |
| useful\_num | 被点赞数 | int | Not Null | No |  |
| published\_date | 发布日期 | date | Not Null | No |  |

表4：comment\_info（评论信息表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 中文说明 | 数据类型 | 允许为空 | 主键 | 备注 |
| user\_name | 用户名 | varchar(20) | Not Null | Yes | 外键关联 login\_info |
| icon | 头像 | varchar(20) | Not Null | No |  |
| comment\_name | 评论名 | varchar(20) | Not Null | No |  |
| picture | 图片路径 | varchar(200) | Null | No | 多图可用分隔符拼接 |
| comment\_content | 评论内容 | varchar(500) | Not Null | No | 建议考虑 TEXT 类型 |
| useful\_num | 被点赞数 | varchar(10) | Not Null | No | 类型建议改为 int |
| published\_date | 发布日期 | date | Not Null | No |  |

表5：admin\_data（管理数据表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 中文说明 | 数据类型 | 允许为空 | 主键 | 备注 |
| user\_num | 用户数 | varchar(20) | Not Null | No | 统计字段 |
| icon\_data | 头像数据 | varchar(20) | Not Null | No | 图像相关统计信息 |
| comment\_num | 评论数 | varchar(20) | Not Null | No | 总评论数统计 |
| article\_num | 文章数 | varchar(20) | Not Null | No |  |
| run\_date | 运行日期 | date | Not Null | No | 系统状态统计时间点 |



### 2.2.2 数据采集

博客系统主要通过以下方式采集与生成数据：

1. 用户生成内容（UGC）：用户在平台发表的文章、评论和互动行为。

2. 系统日志：自动收集访问日志、错误日志以用于监控与分析。

### 2.2.3 设计决策

前端：响应式设计，支持桌面。

后端：FAST API。

数据库：MySQL，优化读写性能，并实施定期备份。

### 2.2.4 功能模块

表2.2-1 博客系统功能模块描述

|  |  |  |  |
| --- | --- | --- | --- |
| 功能模块 | 功能 | 功能描述 | 优先级 |
| 用户管理模块 | 注册与登录 | 支持用户注册、登录、登出及密码找回功能，确保账户安全与可管理性。 | 1 |
| 文章管理模块 | 发布与编辑文章 | 允许用户创建、编辑、删除和查看文章，支持Markdown语法与图片上传。 | 1 |
| 管理后台模块 | 后台审核与统计 | 管理员可审核内容、管理用户，并查看系统统计报表。 | 6 |

用例规约：发布与编辑文章

|  |  |
| --- | --- |
| **项目** | **描述** |
| 用例名称 | 发布与编辑文章 |
| 功能简述 | 用户创建新文章或编辑已有文章，并保存到系统中 |
| 用例编号 | BSYS.001 |
| 执行者 | RegisteredUser |
| 前置条件 | 用户已登录，拥有发布或编辑权限 |
| 后置条件 | 文章内容保存成功，生成文章ID并可供阅读 |
| 涉众利益 | 用户希望便捷地发布和更新内容；系统需要保证数据一致性 |
| 基本路径 | 1. 用户点击“新建文章”或“编辑”按钮。2. 系统展示编辑器界面。3. 用户输入或修改文章标题和内容，可插入图片。4. 用户点击“保存”或“发布”。5. 系统校验内容并保存至数据库，返回成功提示 |
| 扩展路径 | 2a. 内容不符合格式要求，系统提示错误并阻止保存。4a. 图片上传失败，提示并允许重试 |
| 字段列表 | 文章ID、用户ID、标题、内容、标签列表、创建时间、更新时间 |
| 设计规则 | 编辑器与后端解耦，使用FASTAPI接口；Markdown渲染遵循安全规范 |
| 未解决的问题 | 如何兼容多媒体嵌入，如视频或代码高亮 |
| 备注 | 支持草稿箱功能，防止数据丢失 |

**用例规约：评论与回复**

|  |  |
| --- | --- |
| **用例名称** | **评论与回复** |
| 功能简述 | 用户对文章或评论进行评论和回复 |
| 用例编号 | BSYS.002 |
| 执行者 | RegisteredUser |
| 前置条件 | 用户已登录，目标文章或评论存在 |
| 后置条件 | 评论内容保存成功，并计入评论列表 |
| 涉众利益 | 用户可表达观点并讨论；系统需防止垃圾评论 |
| 基本路径 | 1.用户在文章底部输入评论内容 2.点击“提交评论” 3.系统校验并保存评论，刷新评论列表 |
| 扩展路径 | 2a.评论内容存在敏感词，提示修改后提交 3a.验证失败，提示重新登录 |
| 字段列表 | 评论ID、文章ID、父评论ID、用户ID、内容、创建时间 |
| 设计规则 | 评论分页加载；支持富文本及表情 |
| 未解决的问题 | 评论审核策略需进一步优化 |
| 备注 | 可考虑使用异步加载提升体验 |

# 3.视图

## 3.1 逻辑视图

### 3.1.1顶层逻辑视图

#### 3.1.1.1主表示

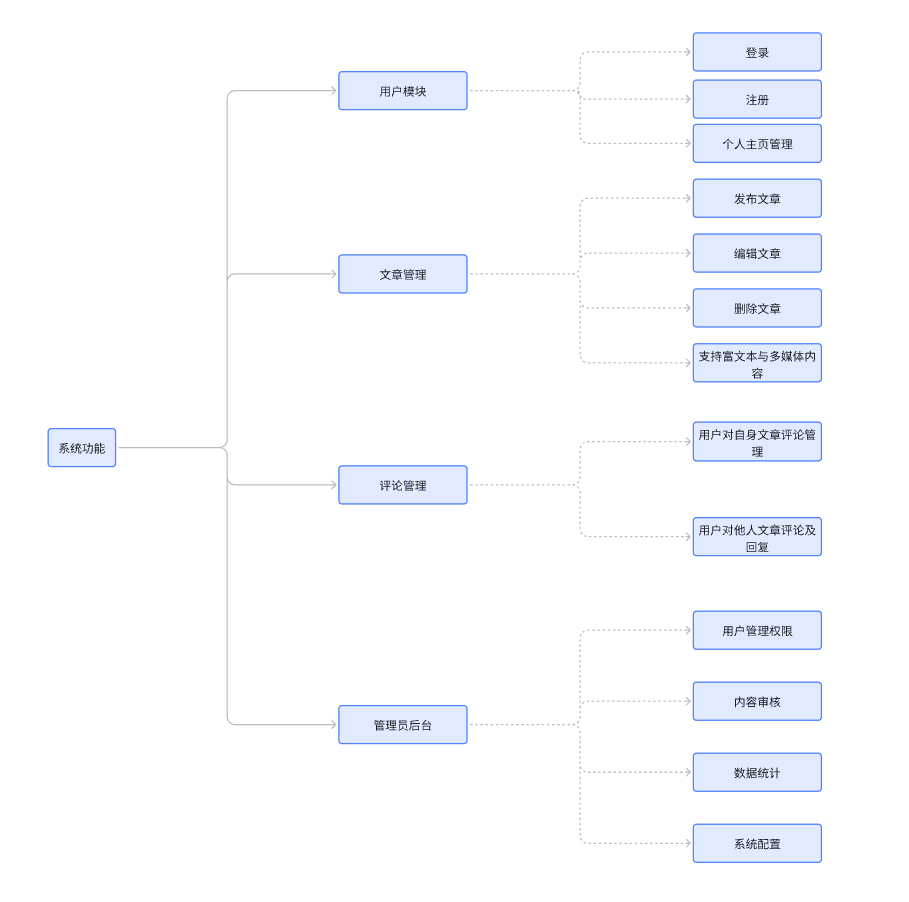


图3.1.1-1 逻辑流程图

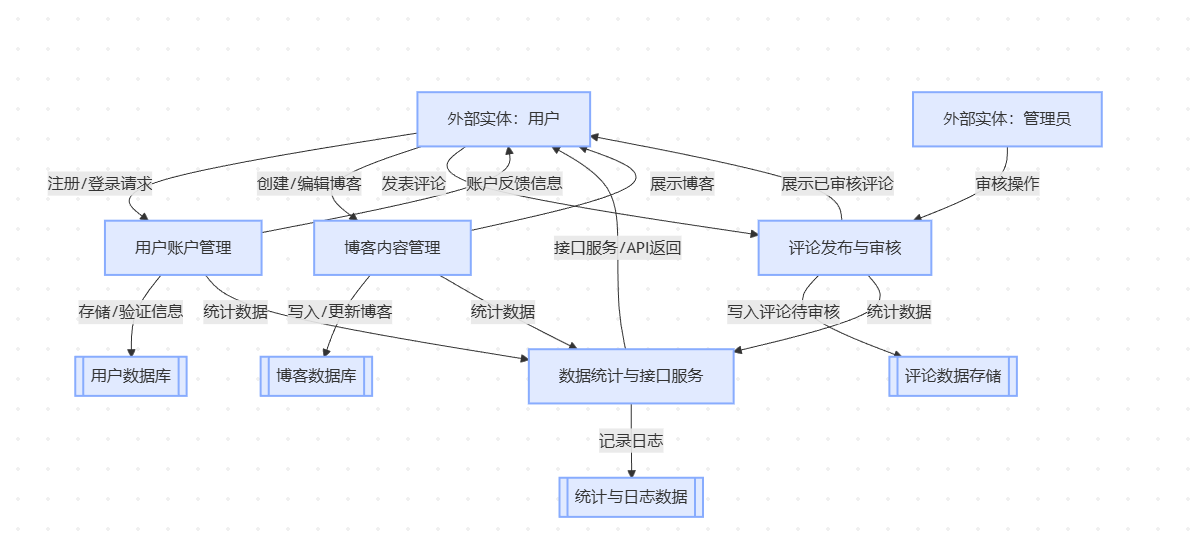


图3.1.1-2 数据流程图

#### 3.1.1.2构件目录

A. 构件及其特性

| 构件 | 描述 |
| --- | --- |
| 用户模块 | 提供用户登录、注册、个人主页管理等基础功能支持 |
| 文章管理模块 | 支持文章的发布、编辑、删除及多媒体分享 |
| 评论管理模块 | 提供用户对文章进行评论、回复以及对自身评论进行管理的功能 |
| 管理员后台模块 | 提供管理员进行用户管理、内容审核、数据统计与系统设置的功能 |

B. 关系及其特性

使用关系：

用户模块与文章管理模块、评论管理模块紧密协作，用户需要先登录才能进行文章发布与评论操作。

管理员后台模块具有最高权限，可对用户模块、文章模块、评论模块中的数据与操作进行监管与控制。

组成关系：

系统主界面调用用户模块进行身份识别，并根据权限级别跳转到相应的功能模块。

各模块由若干功能组件（如文章发布、评论回复、数据统计等）构成，每个组件通过统一接口对接系统主流程。

C. 元素接口

硬件接口：

无特殊硬件依赖。

软件接口：

用户模块与数据库系统接口进行用户数据的持久化存储。

管理员后台模块通过Axios API接口与前端管理界面交互。

各功能模块间采用统一的数据接口标准进行调用（如 JSON 格式的数据传输）。

D. 元素行为

用户管理模块：

处理用户登录、注册请求，校验用户身份，提供个人信息展示与修改接口。

文章管理模块：

接收并处理用户上传的文章内容，支持富文本编辑及媒体上传功能。

评论管理模块：

实时处理评论内容的发布、编辑与展示，维持评论区的互动逻辑。

管理员后台模块：

以权限控制为核心，记录用户行为日志，分析内容数据，提供可视化管理功能。

### 3.1.2用户模块逻辑视图

#### 3.1.2.1主表示

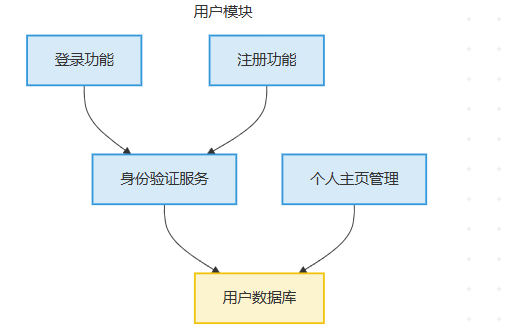


图3.1.2-1用户模块逻辑图

#### 3.1.2.2构件目录

A. 构件及其特性

| 构件名称 | 描述 |
| --- | --- |
| 登录功能 | 提供用户登录界面及身份验证入口，接受用户名和密码，调用认证服务完成验证。 |
| 注册功能 | 实现新用户注册功能，收集必要信息（如用户名、邮箱、密码），并将数据写入用户数据库。 |
| 个人主页管理 | 用户登录后可查看与修改个人资料，包括头像、简介、密码等；提供权限校验及数据更新功能。 |
| 身份验证服务（Auth Service） | 处理登录/注册请求中的验证逻辑，包括校验用户凭证、生成会话令牌、错误反馈处理等。 |

#### 3.1.2.3相关视图

父视图：总视图

### 3.1.3文章管理模块逻辑视图

#### 3.1.3.1主表示

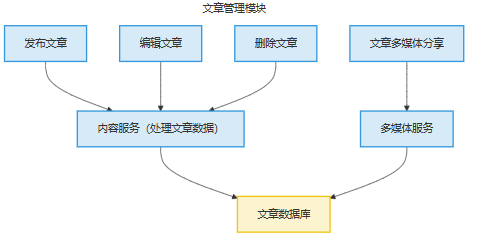


图3.1.3-1文章管理模块逻辑图

#### 3.1.3.2构件目录

A. 构件及其特性

| 构件名称 | 描述 |
| --- | --- |
| 发布文章 | 提供文章的创建功能，支持标题、内容、标签、分类、封面图上传等字段的录入与校验。 |
| 编辑文章 | 支持对已发布文章内容进行修改，调用内容服务模块完成数据更新。 |
| 删除文章 | 实现文章的逻辑删除或物理删除操作，确保用户权限校验。 |
| 文章多媒体分享 | 支持图片、音视频等媒体的上传、预览与插入，便于富媒体内容创作。 |

#### 3.1.3.3相关视图

父视图：总视图

兄弟视图：用户模块视图

### 3.1.4评论管理模块逻辑视图

#### 3.1.4.1主表示

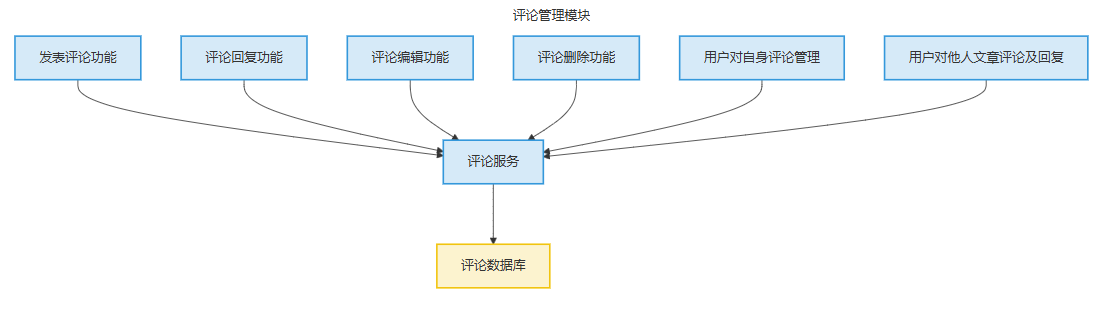


图3.1.4-1评论管理模块逻辑图

#### 3.1.4.2构件目录

1. 构件及其特性

| 构件名称 | 描述 |
| --- | --- |
| 发表评论功能 | 用户可在文章下发表评论，支持富文本格式与表情符号。 |
| 评论回复功能 | 支持用户对他人评论进行回复，构建评论层级结构。 |
| 评论编辑功能 | 用户可编辑自己已发布的评论（在限定时间或条件下）。 |
| 评论删除功能 | 用户可删除自己的评论，管理员可删除所有评论。 |
| 用户对自身评论管理 | 用户可查看、编辑、删除自己的评论内容。 |
| 用户对他人文章评论及回复 | 用户在浏览他人文章时可进行评论互动。 |

#### 3.1.4.3相关视图

父视图：文章管理模块视图

### 3.1.5管理员后台模块逻辑视图

#### 3.1.5.1主表示

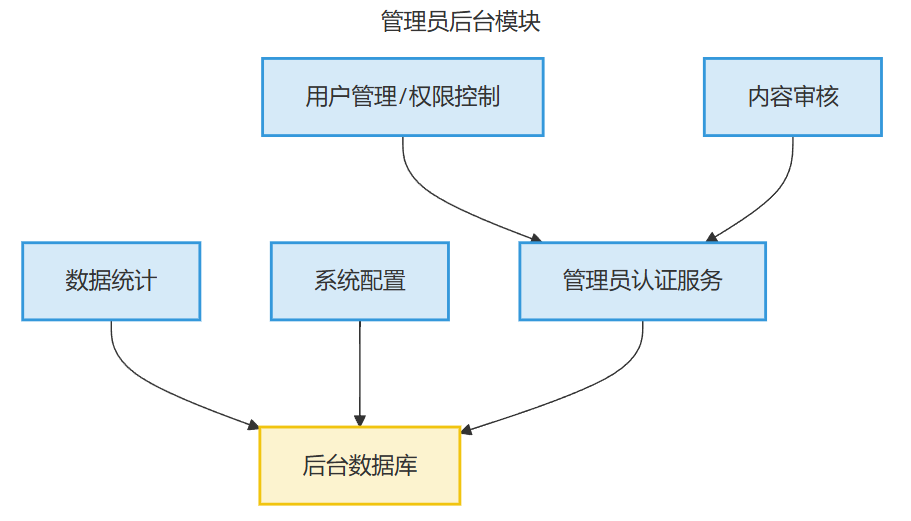


图3.1.5-1管理员后台模块逻辑图

#### 3.1.5.2构件目录

1. 构件及其特性

| 构件名称 | 描述 |
| --- | --- |
| 用户管理 | 提供对平台用户的增删改查、权限设置、封禁等管理功能 |
| 内容审核 | 审核用户发布的文章与评论，支持人工审核与违规内容标记 |
| 数据统计 | 提供平台运营数据分析与可视化展示，如用户活跃度、内容发布量等 |
| 系统配置 | 支持后台管理员对平台基础配置项（如角色权限、接口限流等）的动态调整 |
| 管理员认证服务 | 提供后台管理员的身份验证与权限校验，确保后台操作安全 |

#### 3.1.5.3相关视图

父视图：总视图

兄弟视图：用户视图

## 3.2 开发试图

### 3.2.1顶层开发视图

#### 3.2.1.1主表示

#### 3.2.1.2构件目录

A. 构件及其特性

| 构件 | 描述 |
| --- | --- |
| 前端(Vue.js) | 基于Vue.js框架构建，负责用户交互逻辑的实现和视图的渲染。主要包括用户界面组件、用户模块、文章管理模块和评论模块的客户端实现。 |
| 后端(FastAPI) | 采用Python语言和FastAPI框架开发，提供Axios API接口，处理所有业务逻辑、数据校验和与数据层的交互。核心服务包括用户服务、文章服务、评论服务及后台管理服务。 |
| MySQL数据库 | 作为主要的持久化存储，使用关系型数据库MySQL存储用户信息、文章数据、评论内容、标签分类等核心业务数据。 |
| Redis缓存 | 高性能键值存储数据库，用于缓存热点数据（如频繁访问的文章、用户信息）和会话管理，以减轻数据库压力并提高系统响应速度。 |

B. 关系及其特性

前后端分离：前端应用和后端应用是独立开发和部署的。它们之间通过定义好的 FAST API 进行通信 。这种分离提高了开发灵活性和可维护性。

API驱动：后端提供 API 接口，前端或其他客户端通过调用这些 API 来获取数据或执行操作。FastAPI 能够自动生成交互式 API 文档，方便前后端协作。

服务与数据分离：后端业务逻辑服务与数据存储（MySQL, Redis）分离，通过标准接口访问数据层。

模块化设计：系统功能被划分为用户模块、文章管理模块、评论模块和后台管理模块等 。每个模块在前后端都有相应的实现，模块内部高内聚，模块间低耦合。

版本控制：所有项目代码均使用 Git 进行版本控制，并托管在GitHub仓库中，支持团队协作和代码历史追溯。

C. 元素接口

前端-后端接口：

协议：HTTP/HTTPS

格式：Axios API，数据交换格式为JSON。

主要接口：用户认证API、文章管理API、评论API、管理员操作API等。

后端-MySQL数据库接口：

通过Python的数据库适配器（如mysqlclient）和ORM工具与MySQL进行交互。

执行标准的SQL语句或ORM操作进行数据读写。

后端-Redis缓存接口：

通过Python的Redis客户端库（如redis-py）与Redis服务器进行连接和操作。

执行缓存的SET、GET、DELETE等命令。

组件间接口：

前端组件通过Props和Events进行通信。

后端服务间的调用可以通过内部API及消息队列。

## 3.3 运行视图

### 3.3.1 顶层运行视图

#### 3.3.1.1主表示

本系统采用典型的 B/S 架构，运行时由三大核心组成部分协作完成：前端 Vue.js 客户端、后端 FastAPI 服务、MySQL 数据存储系统。下图展示了主要运行时组件及其交互关系：

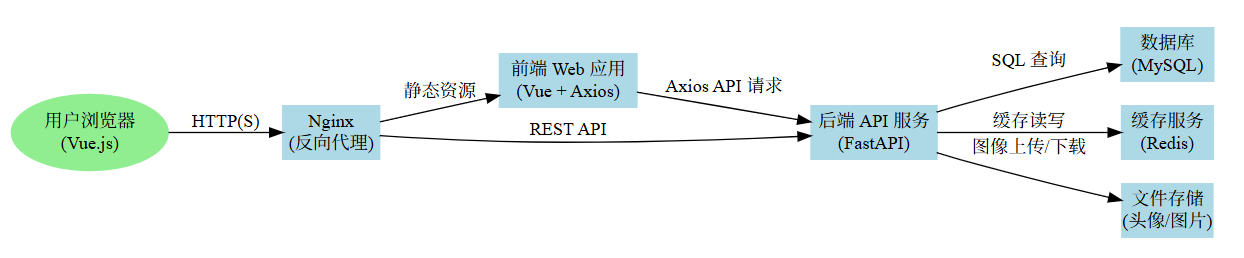


图3.3.1-1系统运行视图

#### 3.3.1.2构件目录

| **构件名称** | **描述** | **技术实现** |
| --- | --- | --- |
| 用户浏览器 | 系统访问入口，加载前端页面并与后端通信 | Chrome/Edge等浏览器 |
| 前端Web应用 | 实现用户界面与交互逻辑 | Vue.js + Axios |
| 后端API服务 | 提供业务逻辑处理与数据访问接口 | FastAPI (Python) |
| 缓存服务 | 提升性能，缓存热数据和登录状态 | Redis |
| 数据库服务 | 存储用户、博客、评论等业务数据 | MySQL 5.7+ |
| 静态资源存储 | 存储上传的头像、文章图片等文件 | 本地或云对象存储 |
| Nginx反向代理 | 统一入口，分发前端资源与后端请求 | Nginx 1.18+ |

## 3.4 部署视图

### 3.4.1主表示

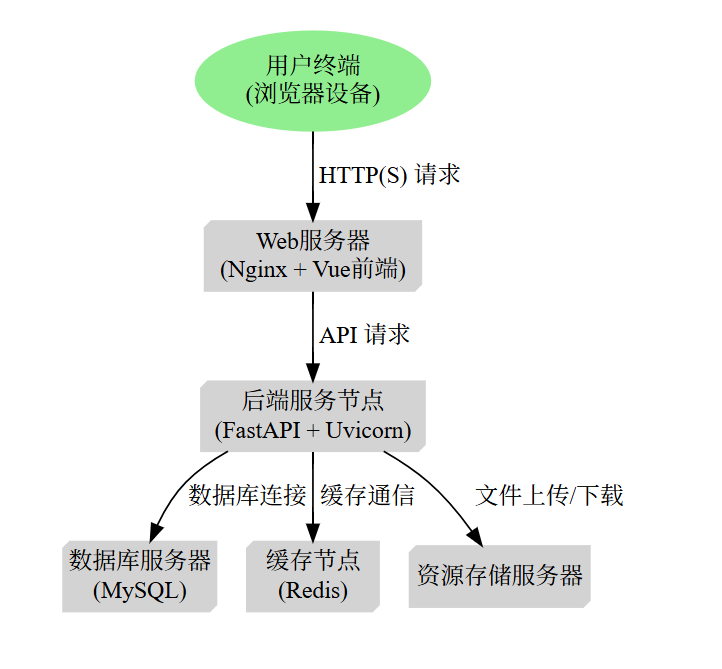
****

图3.4-1 部署图

#### 3.4.1.2构件目录

| **部署节点** | **部署内容** | **说明** |
| --- | --- | --- |
| 用户终端 | 浏览器运行前端页面 | 通过公网访问，无需安装 |
| Web服务器 | Nginx、前端构建后静态资源 | 用于前端资源托管与反向代理 |
| 后端服务节点 | FastAPI 应用、Uvicorn 运行环境 | 提供 API 接口，处理业务逻辑 |
| 数据库服务器 | MySQL 数据库 | 存储系统核心数据（用户、博客、评论等） |
| 缓存服务器 | Redis 服务 | 提供数据缓存、状态管理 |
| 文件资源服务器 | 存储上传图像、用户头像等静态文件 | 可为本地目录或云存储，如 Aliyun OSS、MinIO 等 |

# 需求与架构之间的映射

## 4.1核心功能：

1. 用户登录：根据用户账号和密码匹配账号，并实现登录功能。
2. 用户自我管理：用户通过软件提供的接口创建账号、修改个人信息、发布文章、删除文章等功能。
3. 文章数据统计：记录文章的阅读量、点赞数。对个人账号总体文章数据进行统计和分析。
4. 管理员管理：管理员对全站的用户账号进行批量创建和删除、管理员对违规文章的审查和删除，对账号的封禁。
5. 评论管理：用户和管理员针对自己的评论（管理员为全站评论）进行发表、修改、删除等操作的功能。

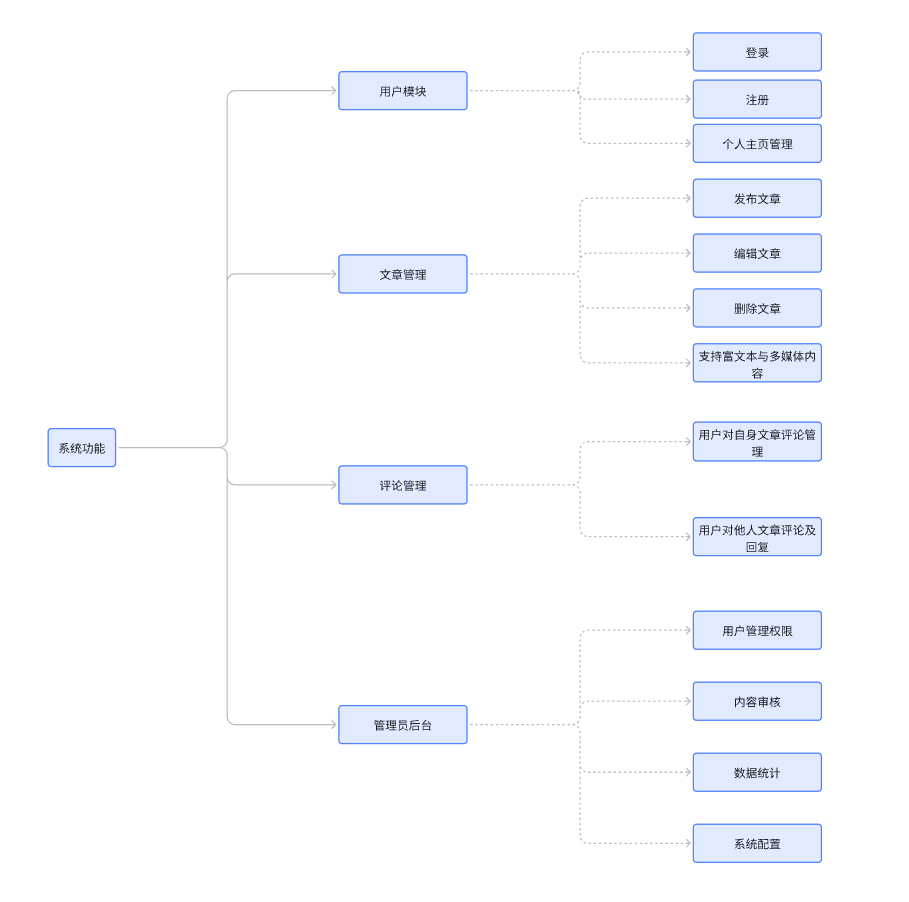
## 4.2辅助功能：

1. 文章查询功能：根据文章名查询对应文章内容。
2. 用户查询功能：根据用户名查找用户及其文章。

## 4.3软件系统总体功能/对象结构

### 4.3.1总体结构

对软件系统总体功能/对象结构进行描述，包括结构图、流程图或对象图。



如图，布劳阁个人博客系统的整体系统功能分为用户模块、文章管理模块、评论管理模块、管理员后台模块四个部分。其中用户模块主要负责给用户提供登录账号、注册账号、以及个人主页管理的功能。这个模块这是用户可以正常使用布劳阁个人博客系统的前提。文章管理模块主要给用户提供发布文章、编辑文章、删除文章等功能同时，布劳阁支持富文本和多媒体内容。这个模块是布劳阁个人博客系统的核心模块，用户主要通过这个模块提供的功能对自己的博客内容进行编辑或者发表博客、删除曾经发表过的博客。评论管理模块主要涉及用户管理自己文章的评论管理以及对他人的文章发表评论。管理员后台模块主要提供管理员权限下对用户账号的管理，对文章内容的审核，对全站账号数量、文章数量，对系统进行配置的操作。

### 4.3.2软件子系统功能/对象结构

1. 用户交互与管理模块

用户交互与管理模块为系统提供基础身份验证与个性化信息管理服务，是博客平台正常运行的前置条件与支撑平台。通过标准化的身份认证流程与用户数据管理机制，确保用户访问的安全性与系统资源的有序分配。该模块与文章、评论、后台管理等模块保持高频交互，是系统结构的核心构件之一。

（1）身份认证

a) 登录/注册机制

提供用户名、密码及邮箱校验的标准身份认证服务。

b) 权限识别与分发

用户登录后系统依据身份角色（普通用户、管理员）分配不同的权限视图与功能访问路径。

（2）个人信息管理

a) 用户主页编辑

支持用户个性化编辑个人资料、头像、签名信息，并提供可视化的展示界面。

b) 活动追踪与展示

聚合用户发布、评论、收藏等行为数据，提供历史记录与交互统计接口。

2. 内容管理与互动模块

内容管理与互动模块是博客系统的核心功能板块，涵盖文章发布、富文本编辑、评论交流等完整的内容生命周期流程。系统在设计上采用松耦合、统一接口架构，提升内容处理的实时性与扩展性。

（1）文章管理子模块

a) 内容创建与编辑

支持Markdown及富文本格式，用户可进行文本输入、图像嵌入、视频链接等多媒体处理。

b) 内容持久化存储

系统通过数据库接口完成文章内容的结构化存储，并进行时间戳与用户ID绑定。

c) 文章浏览与分享

自动生成文章详情页，支持分类、标签与分享链接生成，提升内容传播效率。

（2）评论互动子模块

a) 评论发布与管理

提供实时评论发布接口，用户可进行一级评论与多层级回复，增强文章互动性。

b) 评论内容过滤与管理

系统内置关键词过滤机制，管理员可进行内容审核与评论删除操作，维护社区秩序。

c) 用户交互增强

支持点赞等功能，增强用户间交互粘性。

3. 管理与监控模块

后台管理模块作为高权限控制中心，服务于平台运维管理、内容监管与用户行为分析等任务，是保障系统稳定运行的重要支撑。

（1）用户与内容管理

a) 用户状态监控

可对用户行为记录进行查询，发布频率、违规内容等，便于行为审查与数据归档。

b) 内容审核与日志管理

支持对文章与评论内容的审核操作，并同步记录行为日志，生成可视化数据报告。

（2）系统数据统计

a) 活跃度统计与趋势分析

定期生成文章发布数量、用户活跃度、访问来源等多维度数据报表，辅助平台运营决策。

b) 异常操作监控

通过系统日志对异常访问行为（如暴力破解尝试、频繁操作等）进行预警与记录。

（3）接口与权限配置

后台系统采用Axios与前端管理控制台进行接口通信，提供权限管理控制台，支持对接口访问进行粒度化配置。

# 5.附录

## 5.1架构元素索引

略。

## 5.2术语表

略。

## 5.3缩略语

略。