

个人博客项目网站

软件需求规格说明

成员：于一帆、胡峻岩、邢乐凯、付召帅、许鑫

目录

1 范围	1
1.1 标识	1
1.2 系统概述	1
1.3 文档概述	2
1.4 基线	2
2 引用文件	3
3 需求	3
3.1 所需的状态和方式	3
3.2 需求概述	4
3.2.1 目标	4
3.2.2 运行环境	5
3.2.3 用户的特点	6
3.2.4 关键点	6
3.2.5 约束条件	7
3.3 需求规格	7
3.3.1 软件系统总体功能/对象结构	7
3.3.2 软件子系统功能/对象结构	9
3.3.3 描述约定	11
3.4 CSCI 能力需求	11
3.5 CSCI 外部接口需求	15
3.5.1 用户接口	16
3.5.2 管理员接口	15
3.5.3 硬件接口	16
3.5.4 软件接口	16
3.5.5 通信接口:	21
3.5.6 安全性接口:	21
3.5.7 性能接口:	22
3.6 CSCI 内部接口需求	22
3.7 CSCI 内部数据需求	25
3.8 适应性需求	28
3.9 保密性需求	28
3.10 保密性和私密性需求	29
3.11 CSCI 环境需求	29
3.12 计算机资源需求	32
3.12.1 计算机硬件需求	32
3.12.2 计算机硬件资源利用需求	32
3.12.3 计算机软件需求	33
3.12.4 计算机通信需求	35
3.13 软件质量因素	36
3.14 设计和实现的约束	36
3.15 数据	36
3.16 操作	37
3.17 故障处理	37
3.18 算法说明	37
3.19 有关人员需求	40
3.20 有关培训需求	40
3.21 有关后勤需求	41
3.22 其他需求	41

3.23 包装需求.....	42
3.24 需求的优先次序和关键程度	42
4 合格性规定.....	43
5 需求可追踪性	44
6 尚未解决的问题	45
7 注解.....	46
附录.....	46

1 范围

1.1 标识

标识号：PRJ-BLOG-001

标题：个人博客网站软件工程需求文档

缩略词语：

PRJ：项目

BLOG：博客

版本号：1.0

发行号：1.0.0

1.2 系统概述

本次开发个人博客网站项目，该项目旨在实现一个用于发布个人博客文章的网站，提供用户一个展示个人观点和分享经验的平台。用户可以在该网站上创建个人账户，创建、编辑、发布和管理自己的博客文章，并与其他用户进行互动。

1、系统的一般特性：

- 用户注册和登录功能：支持用户创建个人账户，包括注册、登录和密码管理功能，以使用户可以更方便地浏览博主的文章。
- 文章创建、编辑、发布和管理功能：支持博主创建、编辑和发布博客文章，包括文章的标题、内容、标签、分类等信息的录入和展示，以便博主可以方便地发布和管理自己的博客内容。
- 互动功能：支持用户对文章进行评论、点赞、分享等互动功能，以促进用户之间的交流和互动，增加用户参与度。
- 搜索和筛选功能：支持用户对博客文章进行搜索和筛选，以使用户可以快速找到感兴趣的文章。
- 响应式设计：具备响应式设计，以在不同终端设备上适配，包括桌面、平板和移动设备，以提供良好的用户体验。

2、系统的开发、运行和维护历史：

- 开发历史：该系统尚未开始开发，本文档将作为项目的需求基准，并在系统开发和维护过程中进行参考和更新。
- 运行和维护：系统将在项目完成后部署到指定的服务器环境，并进行运行和维护。

护，包括监控、备份、安全管理等。

3、该项目是大学生实验项目，所以并没有投资方，需方为我们的团队成员本身，用户为访问和使用博客网站的注册用户，开发方为项目的开发团队即我们小组成员，支持机构为提供系统运维和支持的机构。

4、当前和计划的运行现场尚未确定，将在项目进一步开发和部署后进行确认和更新。

5、其他相关文档如下：

- 系统设计文档：包括系统的架构、数据库设计、界面设计等详细设计文档。
- 用户手册：包括用户使用系统的详细说明和操作指南。
- 测试文档：包括系统的测试计划、测试用例、测试结果等测试相关文档。
- 运维文档：包括系统的运维管理、备份和恢复、安全管理等文档。
- 项目计划和进度文档：包括项目的计划、进度、里程碑等项目管理相关文档。

1.3 文档概述

本需求文档是为了支持"个人博客网站"项目的开发而编写的。该项目是大学软件工程课程的小组作业，旨在设计和实现一个简单的个人博客网站。本文档描述了项目的需求和功能，包括用户注册和登录、文章的创建、编辑、发布和管理、互动功能、搜索和筛选功能等。

此外，本文档还提供了项目的一般特性和概述了项目的历史，包括系统开发和运行的背景和目的。本文档还标识了项目的投资方、需方、用户、开发方和支持机构，以便了解项目的相关参与者和相关方的身份。

同时，本文档还列出了其他相关文档，包括系统设计文档、用户手册、测试文档、运维文档以及项目计划和进度文档等，以便我们小组成员和助教了解项目的需求和开发进展。

在使用本需求文档时，应注意保密性和私密性要求。本文档可能包含了一些信息，但不涉及商业机密或用户隐私数据。因此，传递、存储和使用本文档时应保护项目的安全和保密性，但无需过于专业或复杂的保密措施。

本需求文档将作为项目的基础文档，用于项目团队和相关利益相关方的参考，以确保系统能够按照需求规格进行开发和交付，并满足项目的目标和老师的期望。

1.4 基线

1、需求文档：本系统设计说明书基于前期编写的项目需求文档，包括对个人博客网站的功能需求（如用户登录、文章发布、评论管理等）、性能需求（如页面加载速度、

并发处理能力等)、安全性需求(如用户数据保护、防止 SQL 注入等)等方面的详细描述。

2、技术选型: 本系统前台采用 Vue3 作为主要框架, 后台采用 Vue2 作为主要框架, 后端使用 Spring Boot 作为主要框架, 并使用 Mysql 作为数据库

3、架构设计: 本系统采用经典的前后端分离架构, 前端和后端之间通过 API 进行交互, 前端负责页面的渲染和用户交互, 后端负责数据的处理和业务逻辑, 数据库负责数据的存储和查询。前台采用 Vue3 作为主要框架, 后台采用 Vue2 作为主要框架; 后端使用 Spring Boot 作为主要框架, 并使用 Mysql 作为数据库。

4、设计模式和设计原则: 本系统设计采用了常见的设计模式和设计原则, 如单一职责原则(SRP)、开放封闭原则(OCP)、依赖倒置原则(DIP)等, 以保证系统的可扩展性、可维护性和可重用性。设计基线应包括对所采用的设计模式和设计原则的说明, 包括其应用方式和效果。

5、项目内部的开发规范: 本系统设计遵循了项目内部的开发规范和标准, 包括命名规范、代码风格、文档格式等。设计基线应包括对这些规范和标准的遵循和应用方式的说明, 以确保设计文档与小组成员内部的开发规范一致。

2 引用文件

[1] Martin Halvey and Mark T. Keane, An Assessment of Tag Presentation Techniques, poster presentation at WWW 2007, 2007

3 需求

本章应分以下几条描述 CSCI 需求, 也就是, 构成 CSCI 验收条件的 CSCI 的特性。CSCI 需求是为了满足分配给该 CSCI 的系统需求所形成的软件需求。给每个需求指定项目唯一标识符以支持测试和可追踪性。并以一种可以定义客观测试的方式来陈述需求。如果每个需求有关的合格性方法(见第 4 章)和对系统(若适用, 子系统)需求的可追踪性(见 5.a 条)在相应的章中没有提供, 则在此进行注解。描述的详细程度遵循以下规则: 应包含构成 CSCI 验收条件的那些 CSCI 特性, 需方愿意推迟到设计时留给开发方说明的那些特性。如果在给定条中没有需求的话, 本条应如实陈述。如果某个需求在多条中出现, 可以只陈述一次而在其他条直接引用。

3.1 所需的状态和方式

CSCI (Computer Software Configuration Item) 所需的状况包括:

- 1、源代码: 描述软件系统的程序源代码。
- 2、文档: 描述软件系统的设计、需求、用户手册等文档。

3、测试用例和测试结果：描述对软件系统进行测试的用例和测试结果。

4、可执行文件：软件系统的运行版本。

而 CSCI 所需的方式包括：

- 1.版本控制：通过版本控制系统来管理 CSCI 的不同版本和变更历史。
- 2.配置控制：通过配置管理系统来管理 CSCI 的组成部分，并确保它们在开发生命周期中的每个阶段都能够正确地被识别、记录和维护。
- 3.过程控制：通过为开发过程定义一套明确的规范和过程，确保 CSCI 在各个开发阶段中遵循一致的方法和标准。
- 4.标识控制：通过标记和标识来识别和记录 CSCI 及其关联元素的状态和历史信息。

3.2 需求概述

3.2.1 目标

a.本系统的开发意图、应用目标及作用范围(现有产品存在的问题和建议产品所要解决的问题)。

开发意图：为用户提供一个发布和分享自己观点、想法和经验的平台，方便用户与他人交流和互动。

应用目标：通常包括但不限于写作、阅读、评论、分享等，

作用范围：涵盖个人博客、企业博客、教育博客等领域。

可能存在以下问题：

- 1、信息质量难以保证：由于开放性和去中心化的特性，博客网站内容质量参差不齐，有些内容可能存在误导、错误或低劣之处。
- 2、用户交互效果不佳：用户体验受到限制，如搜索引擎排名、页面加载速度、响应时间等方面，可能影响访问者的使用体验和留存率。
- 3、安全隐患较高：由于博客网站大多包含用户生成内容（UGC），因此其存在着恶意软件、安全漏洞以及垃圾邮件和水军等风险。
- 4、平台维护需求较高：博客网站需要定期更新和维护服务端与前端代码，对网络安全、系统稳定性和用户体验等方面都有较高要求。

建议产品所要解决的问题：

用户交互效果不佳的问题，解决此问题可以大幅度改善用户的体验

b.本系统的主要功能、处理流程、数据流程及简要说明。

主要功能：写作、阅读、评论、分享

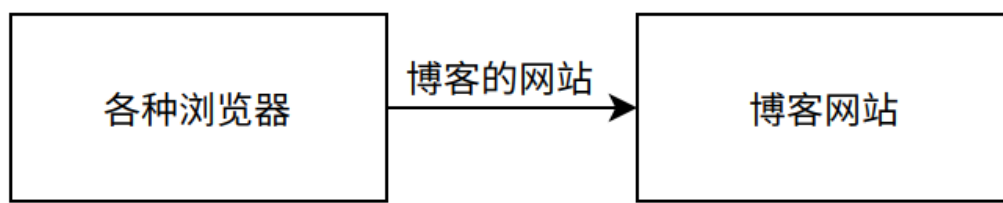
处理流程：注册，登录，发表，评论，分享

数据流程：注册的账号，密码等信息储存在数据库，发表的内容和评论储存在数据，

简要说明：

Software Chicken 个人博客网站是一个用于个人展示、分享和记录个人经验、观点、技能和兴趣的在线平台。作为一个典型的个人博客网站，只有网站拥有者可以上传和管理博客，其他注册用户只有浏览和评论的权限。这样的设计可以保证博客网站的内容质量和安全性，同时也可以为用户提供一个高品质的内容平台和社区交流平台。

c.表示外部接口和数据流的系统高层次图。说明本系统与其他相关产品的关系，是独立产品还是一个较大产品的组成部分(可用方框图说明)。



3.2.2 运行环境

硬件条件：

服务器：小型机/工作站

网络设备：网线，网卡，网络交换机

软件条件：

前端：

- 基础框架：vue3(前台) vue2(后台)
- 状态管理：pinia(前台) vuex(后台)
- 路由组件：vue-router
- 网络请求：axios
- 其他技术：详见前端项目的 package.json

后端：

- 基础框架：springboot
- ORM 框架：mybatisplus
- 权限框架：springsecurity
- 缓存中间件：redis
- 消息中间件：rabbitmq
- 搜索引擎：elasticsearch
- 对象存储：minio
- 定时任务：quartz
- 其他技术：详见后端项目的 pom.xml

前端浏览器：Microsoft Edge

3.2.3 用户的特点

个人博客的博主：可以上传和管理博客

个人博客的评论者：只有浏览和评论的权限。

3.2.4 关键点

说明本软件需求规格说明书中的关键点(例如：关键功能、关键算法和所涉及的关键技术等)。

关键功能：

后台管理网站的几个功能：

可视化展示网站数据，如访问量、用户量、文章贡献、文章浏览量等数据。

对文章的管理，包括增删改查及文章的分类、标签的增删改查。

对说说的管理，包括增删改查。

对用户评论的管理，包括增删改查

权限的管理，包括后台菜单的增删改查、前后端 api 的增删改查、和用户角色的增删改查。

关键算法和所涉及的关键技术：

前端：前台采用 Vue3 作为主要框架，后台采用 Vue2 作为主要框架

后端：使用 Spring Boot 作为主要框架，并使用 MongoDB 作为数据库

数据库：MongoDB

采用经典的前后端分离架构，前端和后端之间通过 API 进行交互，前端负责页面的渲染和用户交互，后端负责数据的处理和业务逻辑，数据库负责数据的存储和查询。前台采用 Vue3 作为主要框架，后台采用 Vue2 作为主要框架；后端使用 Spring Boot 作为主要框架，并使用 MongoDB 作为数据库。

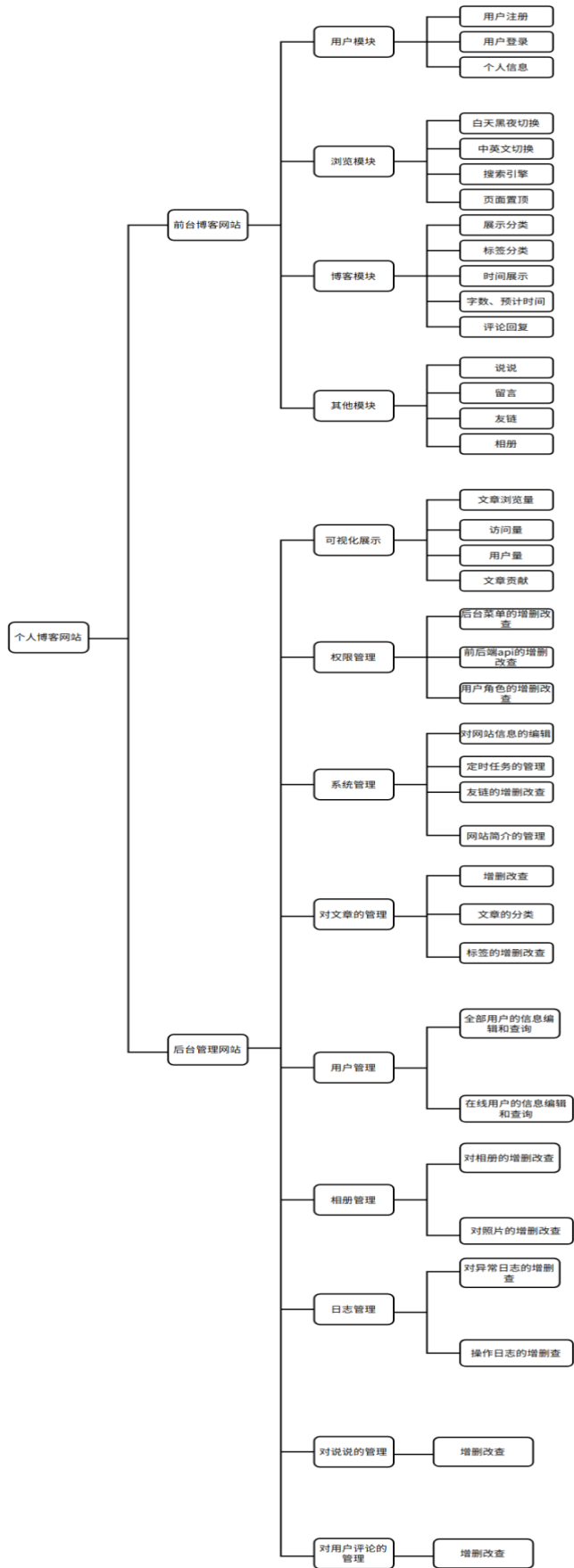
3.2.5 约束条件

进行个人博客网站开发时，可能会面临以下约束条件：

1. 经费限制：必须在预算范围内完成开发工作。
2. 开发期限：需要在指定的时间内完成开发，并交付使用。
3. 技术选型：需要根据业务需求和预算选择合适的技术和工具，例如网站框架、数据库等。
4. 政治环境：需要遵守政府相关规定，例如网络安全法等。
5. 社会文化环境：需要考虑用户对于内容、界面设计等方面的偏好和习惯，以及本地文化特色。
6. 法律环境：需要遵守相关法律法规，例如版权、隐私保护等。

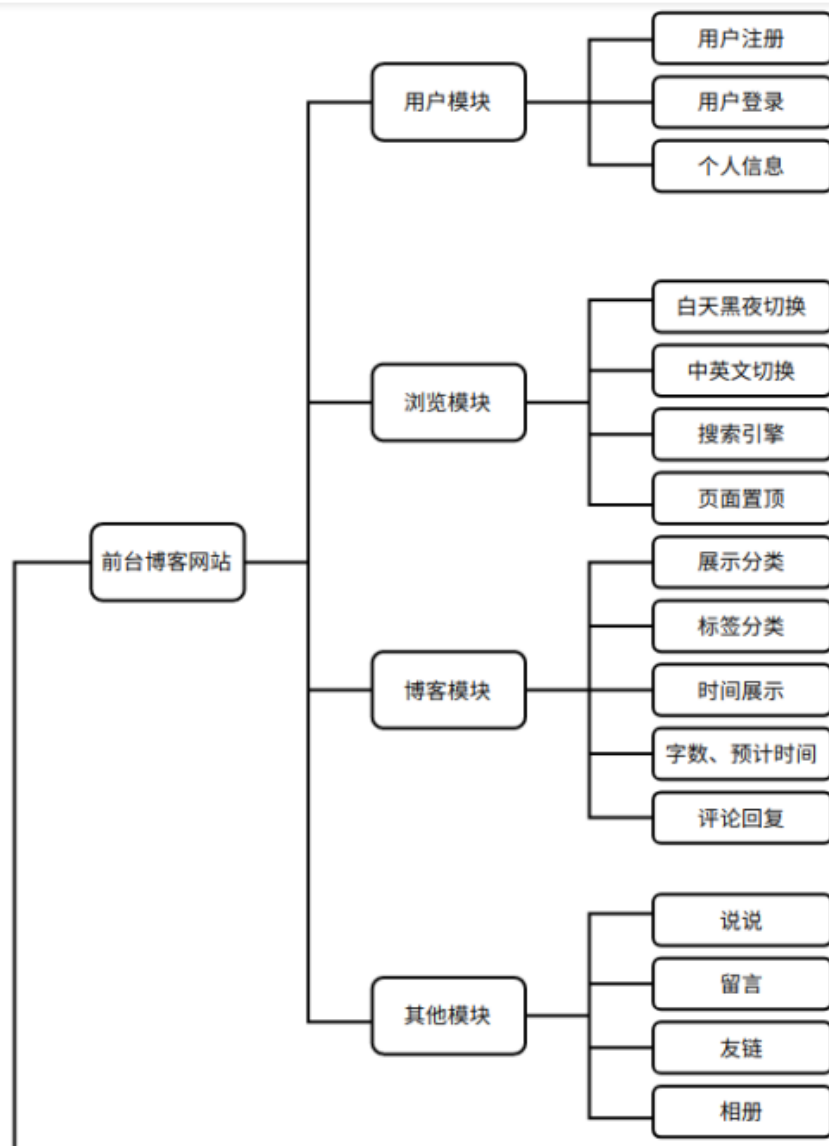
3.3 需求规格

3.3.1 软件系统总体功能/对象结构

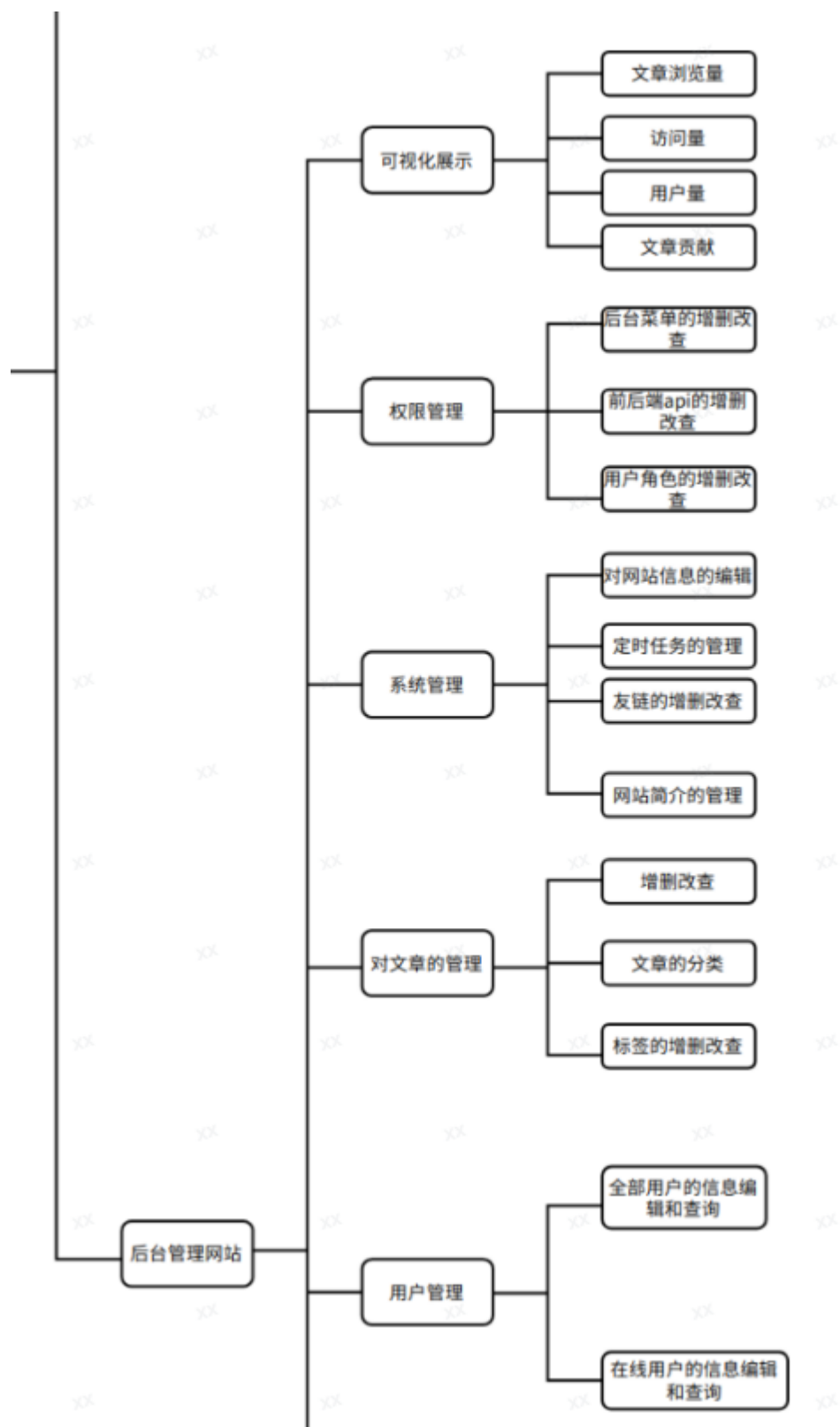


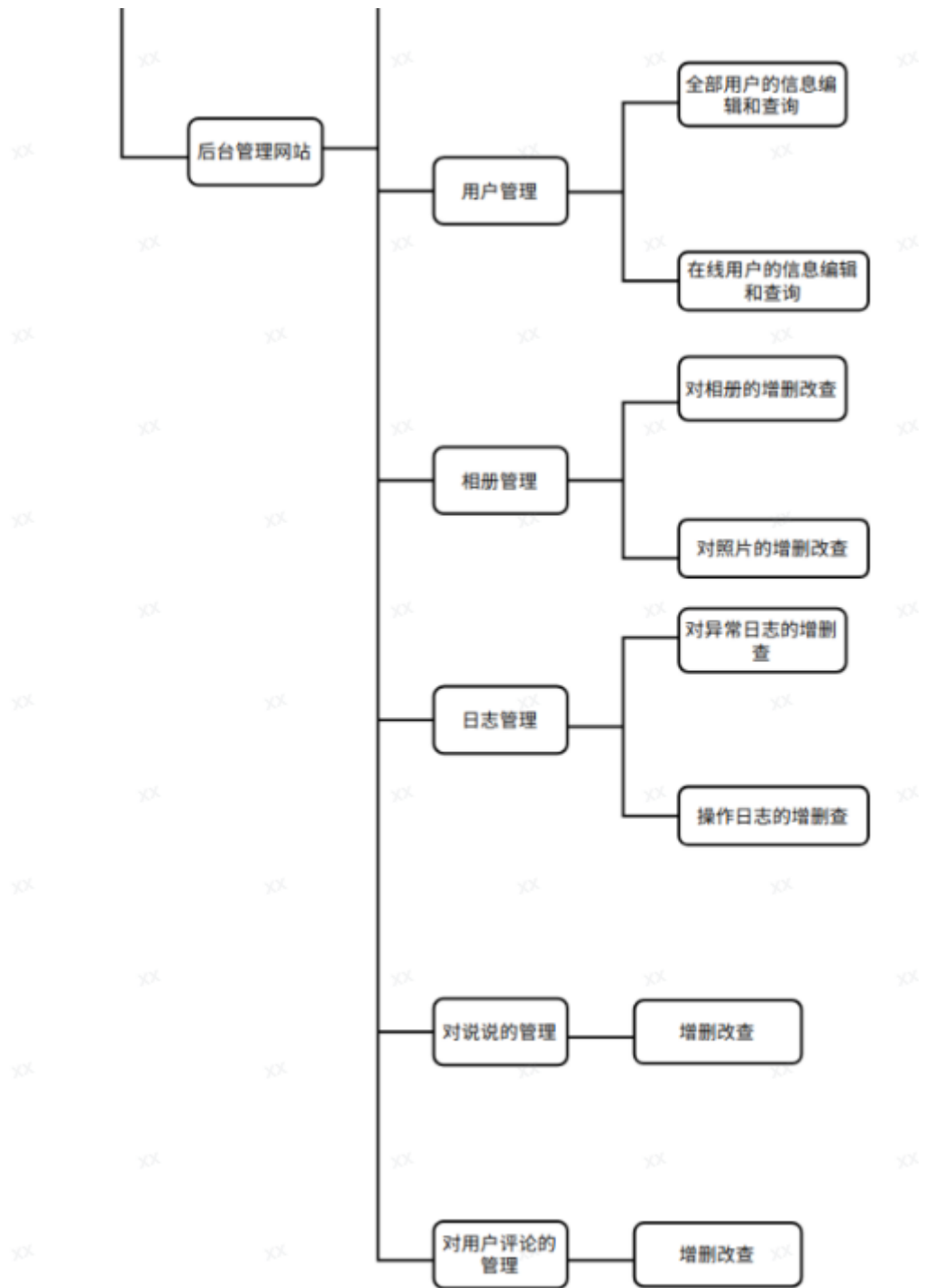
3.3.2 软件子系统功能/对象结构

前台博客网站：



后台博客网站：





3.3.3 描述约定

和国际通用的单位，符号一致

3.4 CSCI 能力需求

1、浏览模块

a. 说明

该功能旨在实现用户在线浏览相册、说说、留言板等信息。主要采用的方法和技术包括数据库设计、分页查询、权限控制等。由来和背景是方便用户快速访问和查看感兴趣的内容。

b. 输入

- 1) 访问网站时需要输入 URL 地址；
- 2) 浏览相册和说说时需要进行信息筛选和排序，可以根据不同特征进行搜索，如名称、发布时间、类别等。

c. 处理

- 1) 对输入数据进行有效性检查，如检查 URL 地址是否符合规范、关键字是否为空等；
- 2) 权限控制：根据不同的用户登录状态，展示信息的访问级别不同；
- 3) 操作的先后顺序和时间设定：根据不同的筛选条件进行查询和排序操作，显示结果输出到页面上；
- 4) 受操作影响的参数：信息文件、搜索条件等；
- 5) 相应的转换和处理方法：采用分页查询技术进行数据的查询和展示等操作；
- 6) 输出数据检查：输出相应的处理结果以供下一步操作使用。

d. 输出

- 1) 输出目的地：网页等；
- 2) 输出数据包括：相册和说说的名称、描述、发布者信息等；
- 3) 异常情况的出错信息、非法值处理等；
- 4) 有关接口：与数据库、查询系统相关的接口。

2、可视化展示模块

a. 说明

该功能旨在将用户上传的图片、视频、音频等多媒体素材进行可视化呈现。主要采用的方法和技术包括图像处理、音视频格式转码、前端可视化库等。由来和背景是增强用户的感官体验，提高页面交互性。

b. 输入

- 1) 用户上传的图片、视频、音频文件等；

c. 处理

- 1) 对输入数据进行有效性检查，如检查文件是否符合规范、大小是否符合限制等；
- 2) 文件解析和处理：根据文件类型采用相应的处理方法，如图片文件可以通过尺寸缩

放、裁剪、滤镜效果等操作生成展示效果，音视频文件则需要进行格式转换和编码处理；

3) 前端展示：采用各种框架和可视化库进行页面搭建和交互设计；

4) 受操作影响的参数：上传文件、处理参数等；

5) 相应的转换和处理方法：采用图像处理库、音视频编解码库、可视化库等技术进行数据的转换和处理；

6) 输出数据检查：输出相应的处理结果以供下一步操作使用。

d. 输出

1) 输出目的地：网页等；

2) 输出数据包括：图片、视频、音频等可视化效果；

3) 异常情况的出错信息、非法值处理等；

4) 有关接口：与图像处理库、音视频编解码库、前端可视化库相关的接口。

3、权限管理模块

a. 说明

该功能旨在对用户的权限进行管理，包括用户身份验证、角色授权、资源访问控制等。主要采用的方法和技术包括认证、加密、会话管理等。由来和背景是保障系统安全性和用户信息私密性。

b. 输入

1) 用户名、密码等登录信息；

2) 角色、权限等管理信息。

c. 处理

1) 对输入数据进行有效性检查，如检查用户名、密码格式是否符合规范等；

2) 身份认证：根据不同的用户登录状态，认证用户身份并维护用户的登录会话；

3) 权限控制：采用 RBAC 等权限模型，设置不同用户的角色和权限，控制资源的访问级别；

4) 受操作影响的参数：登录、授权参数等；

5) 相应的转换和处理方法：采用密码加密、会话管理等技术进行数据的加密和认证；

6) 输出数据检查：输出相应的处理结果以供下一步操作使用。

d. 输出

1) 输出目的地：网页等；

- 2) 输出数据包括：登录是否成功、角色、权限等信息；
- 3) 异常情况的出错信息、非法值处理等；
- 4) 有关接口：与数据库、会话管理等相关的接口。

4、相册管理模块

a. 说明

该功能旨在实现对用户相册的创建、删除、搜索、浏览、点赞等操作。主要采用的方法和技术包括数据库设计、权限控制、图片上传等。由来和背景是方便用户在线管理自己的照片。

b. 输入

- 1) 创建相册时需要输入相册名称、描述、可见权限等信息；
- 2) 删除相册时需要输入相册 ID 或快捷识别码；
- 3) 搜索相册时需要输入关键字、筛选条件等；
- 4) 浏览相册时需要输入相册 ID 或快捷识别码等；
- 5) 点赞相册时需要登录用户点击相册的点赞按钮即可。

c. 处理

- 1) 对输入数据进行有效性检查，如检查相册名称是否为空、相册描述的长度是否符合限制等；
- 2) 权限控制：根据不同的用户登录状态，展示相册列表的访问级别不同；
- 3) 操作的先后顺序和时间设定：创建相册需要进行空间和数量判断；删除相册的时候需要考虑相册内照片的迁移和相册内容的清空；搜索相册需要进行信息匹配和分页查询等操作；
- 4) 受操作影响的参数：相册文件信息、删除指示参数、搜索条件等；
- 5) 相应的转换和处理方法：采用数据库存储技术、图片上传处理等技术进行相册数据的管理，采用分页查询等技术进行数据的查询、展示等操作；
- 6) 输出数据检查：输出相应的处理结果以供下一步操作使用。

d. 输出

- 1) 输出目的地：网页、数据库等；
- 2) 输出数据包括：相册名称、描述、时间戳、照片数量等；
- 3) 异常情况的出错信息、非法值处理等；
- 4) 有关接口：与数据库、图片上传相关的接口。

5、说说管理模块

a. 说明

该功能旨在实现用户在线创建、编辑、删除、搜索、点赞、评论、分享等操作。主要采用的方法和技术包括数据库设计、权限控制、富文本编辑器等。由来和背景是方便用户在线发布动态的文字、图片等内容。

b. 输入

- 1) 创建说说时需要输入类别、标题、内容、权限等信息；
- 2) 删除说说时需要输入说说 ID 或快捷识别码；
- 3) 搜索说说时需要输入关键字、筛选条件等；
- 4) 点赞、评论、分享说说需要登录用户点击对应的按钮即可。

c. 处理

- 1) 对输入数据进行有效性检查，如检查说说标题是否为空、说说内容的长度是否符合限制等；
- 2) 权限控制：根据不同的用户登录状态，展示说说列表的访问级别不同；
- 3) 操作的先后顺序和时间设定：创建说说需要进行空间和数量判断；删除说说的时候需要考虑说说内图片的迁移和内容的清空；搜索说说需要进行信息匹配和分页查询等操作；
- 4) 受操作影响的参数：说说文件信息、删除指示参数、搜索条件等；
- 5) 相应的转换和处理方法：采用数据库存储技术、富文本编辑器等技术进行说说数据的管理，采用分页查询等技术进行数据的查询、展示等操作；
- 6) 输出数据检查：输出相应的处理结果以供下一步操作使用。

d. 输出

- 1) 输出目的地：网页、数据库等；
- 2) 输出数据包括：说说类别、标题、内容、时间戳、图片数量等；
- 3) 异常情况的出错信息、非法值处理等；
- 4) 有关接口：与数据库、评论系统、分享系统、富文本编辑器相关的接口。

3.5 CSCI¹ 外部接口需求

本节描述了 Software Chicken 个人博客网站的外部接口需求，该网站具有前台博客网站和后台管理网站两个部分。前台博客网站用于用户查看和互动博客文章，而后台管理网站用于网站管理员管理博客文章、用户和其他网站功能。接口需求具体见下：

3.5.2 管理员接口

- 1、文章管理接口：通过接口完成对文章的增删改查，以及对文章的分类、标签进行增删改查的操作。
- 2、说说管理接口：通过接口完成对说说的增删改查操作。
- 3、评论管理接口：通过接口完成对用户评论的增删改查操作。
- 4、权限管理接口：通过接口完成对后台菜单的增删改查，以及对前后端 API 的增删改查操作和用户角色的增删改查。
- 5、用户管理接口：通过接口完成对全部用户和在线用户的信息编辑和查询操作。
- 6、相册管理接口：通过接口完成对相册的增删改查操作和对照片的增删改查操作。
- 7、系统管理接口：通过接口完成对网站信息的编辑操作，同时进行定时任务的管理、友链的增删改查操作、网站简介的管理。
- 8、日志管理接口：通过接口完成对异常日志的增删查操作和操作日志的增删查操作。
- 9、数据可视化接口：通过接口完成对访问量、用户量、文章贡献、文章浏览量等网站数据的可视化。

3.5.3 硬件接口

本个人博客网站没有特定的硬件要求，除了需要访问和使用 Web 浏览器和互联网连接的标准硬件。

3.5.4 软件接口

3.5.1 用户接口

- 1、注册接口：用户可以使用他们的电子邮件地址注册帐户。网站将向用户的电子邮件地址发送验证码，验证成功后，用户输入密码成功完成注册。
- 2、登录接口：用户可以使用他们的电子邮件地址和密码登录。同时用户需要完成一个滑块验证码来验证他们是真实用户。另外支持用户使用 QQ 账号直接登录。
- 3、个人资料接口：用户可以管理他们的个人资料信息，包括个人头像、昵称、个人简介和博客链接等，同时可以在个人资料注销自己的账号。
- 4、浏览接口：用户可以在白天和夜间两种模式下浏览博客文章，点击网站右侧按钮可以完成浏览模式的切换和中英文语言选项的切换。同时用户还可以使用键盘快捷键返回页面顶部或底部。
- 5、搜索接口：使用 ElasticSearch 搜索引擎查找相关的博客文章，点击结果可以直接跳转到相应的博客文章。
- 6、博客展示接口：博客文章按照三个优先级排序：置顶、推荐和普通。文章可以按标

签分类，并按时间线归档。进入博客文章后显示文章的字数和预计阅读时间。

7、说说接口：用户通过接口可以发布文字+图片形式的短篇幅动态

8、评论留言接口：通过接口用户可以在博客文章、说说动态下方进行评论和回复，同时用户可以通过接口在留言板给网站留言。

9、友链接口：通过接口展示网站拥有者朋友的个人博客链接，点击连接可以跳转到朋友的个人博客网站。

10、相册接口：通过接口浏览网站拥有者存储的相册和相册内包含的照片。

应用程序接口

即 API，在本项目中，主要调用的应用程序接口如下：

- Pinia API:

Pinia 是一个专为 Vue 3 开发的状态管理库，它提供了一组 API 来管理应用程序的状态，使得开发者可以更容易地管理应用程序的状态，提高应用程序的开发效率。以下是 Pinia API 的一些常见作用：

- 1、创建 store：通过定义一个新的 Store 类来创建一个 store 实例。
- 2、注册 store：将 store 实例注册到 Vue 应用程序中。
- 3、获取 store：通过 \$store 属性或 useStore() 函数在组件中获取 store 实例。
- 4、状态管理：通过定义 state 属性来管理应用程序的状态。可以使用 reactive 或 ref 等 Vue 3 响应式数据类型定义状态。
- 5、计算属性：可以使用 computed 属性来计算 store 中的派生状态。
- 6、动作管理：通过定义 actions 属性来管理 store 的动作。可以在 actions 中实现异步操作和提交 mutations。
- 7、提交 mutations：可以使用 commit() 函数来提交 mutations。mutations 用于同步更新状态。
- 8、订阅状态变化：可以使用 watch() 函数订阅 store 中状态的变化。

- Vuex API:

Vuex 是 Vue.js 应用程序开发的状态管理模式，它采用集中式存储管理应用的所有组件的状态，并以相应的规则保证状态以一种可预测的方式发生变化，通过使用 Vuex，我们可以将应用中的状态和行为从组件中抽离出来，以一种集中式的方式进行管理和维护。同时，通过 API 的方式，我们可以很方便地对状态进行修改和获取，提高了应用的开发效率和维护性。Vuex 具有以下 API：

- 1、state：定义应用状态的数据源，即定义应用中需要集中管理的数据，也就是定义全局变量。在组件中使用时，可以通过 \$store.state.xxx 来访问。

2、**getters**: 可以看做是 **store** 的计算属性, 提供了一种在 **store** 中实现组件之间共享的计算属性的方式。当一个 **getter** 访问其他 **getter** 时, 其他 **getter** 的返回值会被缓存起来, 只有当它所依赖的 **getter** 值发生变化时, 才会重新计算。

3、**mutations**: 只能进行同步操作, 是修改 **state** 的唯一方式, 用于更新应用中的状态。**Mutation** 必须是同步函数, 执行过程是不可逆的, 可以通过 `$store.commit('mutationName')` 触发。

4、**actions**: 可以进行异步操作, 可以用于提交 **mutation**, 而不是直接变更状态, 用于处理异步操作, 通过 `$store.dispatch('actionName')` 触发。可以在 **action** 中通过 `context.commit` 来提交 **mutation**, 也可以通过 `context.dispatch` 来触发别的 **action**。

5、**modules**: 用于对于大型的 **store** 进行拆分, 将 **store** 拆分为多个 **module**, 每个 **module** 可以有自己的 **state**、**getters**、**mutations**、**actions**。模块化使得应用的状态更加模块化、结构化, 方便维护和开发。

- **Vue-router API:**

vue-router 是 **Vue.js** 官方的路由管理器, 用于构建单页面应用程序 (SPA)。它提供了一组 **API** 来管理 **URL** 和页面之间的映射关系, 实现了前端路由功能, 让我们能够以单页应用的方式构建交互丰富、用户体验良好的 **Web** 应用程序。**vue-router API** 可以用于:

1、定义路由规则: 通过定义路由规则来将 **URL** 映射到相应的组件, 从而实现页面之间的切换。

2、导航: 提供了程式化导航和声明式导航两种方式进行页面跳转。

3、嵌套路由: 支持嵌套路由, 通过子路由和父路由的组合来实现复杂的页面结构。

4、路由传参: 支持通过 **URL** 参数或者路由元信息来传递数据。

5、路由守卫: 提供了全局路由守卫、路由独享守卫和组件内的守卫等多种路由守卫机制, 可以用于控制页面访问权限、验证用户登录状态等场景。

- **axios API:**

axios 是一个基于 **Promise** 的 **HTTP** 客户端, 用于浏览器和 **Node.js**。它可以发送异步请求, 处理请求和响应数据, 支持拦截器和取消请求等功能。**axios API** 提供了丰富的功能和灵活的配置选项, 方便开发人员进行网络请求和数据处理, **axios API** 的主要作用包括:

1、发送 **HTTP** 请求: 通过调用 `axios.get`、`axios.post` 等方法发送 **HTTP** 请求, 支持 **GET**、**POST**、**PUT**、**DELETE** 等常见的 **HTTP** 请求方法。

2、处理请求和响应数据: **axios** 可以自动将请求和响应数据转换成 **JavaScript** 对象, 方便使用和处理。

3、拦截器: 可以通过 `axios.interceptors` 对请求和响应进行拦截和处理, 例如添加请

求头、请求拦截器和响应拦截器等。

4、取消请求：通过 `axios.CancelToken` 可以取消请求，避免因为网络延迟或其他原因导致的重复请求或无用请求。

5、配置选项：可以通过 `axios.defaults` 设置全局的请求配置选项，例如请求超时时间、响应数据格式等。

6、错误处理：`axios` 支持处理 HTTP 错误，例如 404、500 等，可以在错误时进行统一处理，避免代码重复。

- **Elasticsearch API:**

Elasticsearch 是一种分布式的搜索和分析引擎，它提供了各种 API 来帮助用户进行数据检索、数据分析、聚合和可视化等操作。Elasticsearch 的 API 提供了强大的功能，可以帮助用户处理大量的数据并从中提取有用的信息，以下是一些 Elasticsearch API 的作用：

1、基本搜索 API：用户可以使用基本搜索 API 在 Elasticsearch 中搜索文档，指定查询条件、过滤器、排序和分页等参数。

2、聚合 API：用户可以使用聚合 API 来计算和统计数据，例如计算平均值、最小值、最大值、总和等，并根据特定的条件分组计算。

3、索引 API：用户可以使用索引 API 将文档存储到 Elasticsearch 中，这个 API 提供了许多参数和选项，用于控制文档的存储和索引方式。

4、更新和删除 API：用户可以使用更新和删除 API 来更新或删除现有的文档，这些 API 还提供了一些选项来控制如何更新或删除文档。

5、批量 API：用户可以使用批量 API 来执行大量的索引、更新或删除操作，这可以极大地提高处理速度和效率。

6、分析 API：用户可以使用分析 API 来分析和处理文本数据，例如将文本分解为单词、标记、词干或同义词等。

- **Rabbitmq API:**

RabbitMQ 是一个开源的消息队列系统，用于处理和传递大量的消息。它提供了一些 API 来支持消息的发送和接收，包括以下主要功能：

7、发布/订阅模式：通过 RabbitMQ 的 `exchange` 来实现，发送者将消息发送到 `exchange` 上，然后由 `exchange` 将消息分发给多个订阅者。

8、路由模式：通过 RabbitMQ 的 `exchange` 和 `queue` 的绑定关系来实现，发送者将消息发送到指定的 `exchange` 上，然后由 `exchange` 将消息分发给与之绑定的 `queue`。

9、工作队列模式：通过 RabbitMQ 的 `queue` 来实现，多个消费者从同一个 `queue` 中接收消息，每个消息只能被其中一个消费者处理。

另外 RabbitMQ 还提供了一些高级功能，如消息确认机制、消息持久化、消息优先级等，可以通过 API 来配置和使用这些功能。

- **Minio API:**

MinIO 是一个基于对象存储的开源云存储服务，它提供了与 Amazon S3 兼容的 API 接口，可以用来存储和检索大量的非结构化数据，例如图片、视频、音频、日志等。

MinIO API 提供了以下功能：

- 1、对象存储：使用 MinIO API 可以将数据存储为对象，并将其保存在 MinIO 服务器上。对象存储可以扩展到非常大的数据集，并且可以自动进行备份和恢复。
- 2、对象检索：使用 MinIO API 可以从 MinIO 服务器中检索存储的对象。检索支持按对象名称、标记、元数据、类型等多种方式进行搜索。
- 3、数据保护：MinIO API 提供了一些功能来保护存储在 MinIO 服务器上的数据。例如，可以使用访问控制策略来限制对特定对象的访问，还可以使用加密来保护数据的隐私性。
- 4、元数据管理：使用 MinIO API 可以向对象添加元数据，例如对象名称、大小、类型、修改时间等。这些元数据可以用来帮助组织和检索数据集。
- 5、分布式存储：MinIO API 支持分布式存储，它可以在多个服务器之间分配数据，并提供了一些功能来确保数据的完整性和可用性。

- **Quartz API:**

Quartz 是一个开源的任务调度框架，提供了很多灵活的配置选项和管理工具，可以用来创建基于时间的定时任务。其 API 的作用是实现任务的调度和管理。通过 Quartz API，我们可以创建和配置定时任务，指定任务的触发时间和执行逻辑，管理任务的状态和执行记录等。Quartz API 提供了丰富的类和方法，可以满足不同场景下的任务调度需求。

数据库接口

使用 Mysql 作为主要的数据库，存放持久化数据，如用户信息、博客内容、评论和讨论等数据。使用 Redis 作为缓存数据库，提高运行效率。同时使用 ORM 框架 Mybatisplus 来。数据库应该提供以下接口：

- 用户信息接口：创建用户、获取用户信息、更新用户信息等。
- 博客接口：创建博客、获取博客列表、获取单篇博客、编辑博客、删除博客等。
- 评论接口：创建评论、获取评论列表、回复评论等。
- 留言接口：创建留言、获取留言列表、回复留言等。
- 统计接口：获取用户访问量、博客阅读量、评论和讨论量等数据。

服务器接口

为了确保系统的稳定性和可用性，我们的网站使用云服务提供商来部署该系统。服务器应该提供以下接口：

服务器状态接口：获取服务器状态信息，如 CPU、内存、磁盘等。

日志接口：获取服务器日志信息，如访问日志、错误日志等。

监控接口：获取服务器监控信息，如系统负载、网络流量、响应时间等。

3.5.5 通信接口：

网站应使用标准的 Web 通信协议（如 HTTP 和 HTTPS）在客户端和服务端之间进行接口的通信，接口参数可以使用 JSON 或 XML 格式进行传输，同时，可以使用 JWT（JSON Web Token）或其他认证方式进行用户身份验证，确保接口安全性。网站还应设计为响应式并可访问于不同设备和屏幕尺寸上，如桌面、笔记本电脑、平板电脑和智能手机。

3.5.6 安全性接口：

网站应采取适当的安全措施来保护用户的个人数据和网站的机密信息，包括但不限于以下方面：

1、用户认证接口

- 登录认证接口：使用 JWT（JSON Web Token）进行身份验证，确保只有经过身份验证的用户才能访问网站内容。
- 注册接口：对用户注册信息进行验证和过滤，确保注册信息的合法性，如限制用户名长度、密码复杂度等。
- 修改密码接口：对用户提交的密码进行加密处理，确保密码的安全性。

2、输入验证接口

- 对用户提交的表单数据进行验证和过滤，防止恶意脚本和 SQL 注入攻击。
- 对上传的文件进行类型和大小的限制，防止恶意文件上传。

3、数据库安全接口

- 对数据库中的用户密码进行加密处理，确保用户密码的安全性。
- 对数据库的访问权限进行限制，只允许有权限的用户进行数据库操作。
- 对数据库进行备份，以防数据丢失。

4、系统安全接口

- 对服务器进行安全配置，如使用防火墙、关闭不必要的端口、定期更新系统补丁等。

- 对网站进行日志记录，记录用户行为，便于后续的追踪和监控。
- 使用 HTTPS 协议进行数据传输加密，防止数据被窃取和篡改。
- 对网站进行定期的漏洞扫描，发现漏洞及时修复。

3.5.7 性能接口：

从以下几个方面考虑：

- 1、页面加载时间：衡量网站打开速度的一个重要指标，需要测量网站在不同网络环境下的加载速度，可以使用前端性能优化工具，如 WebPageTest 和 GTmetrix 等。
- 2、数据库响应时间：数据库响应时间对网站的性能有很大的影响，需要对数据库进行优化，如使用缓存、索引和查询优化等技术。
- 3、网络请求时间：衡量网站前后端交互的效率，需要优化网络请求的速度，如使用 CDN 加速、压缩资源、合并请求等技术。
- 4、并发用户数：测试网站在不同并发用户数下的性能表现，可以使用压力测试工具，如 Apache JMeter 等。

基于以上几个方面，可以设计如下的性能接口：

- 1、获取页面加载时间：该接口返回网站在不同网络环境下的加载速度，可以根据不同网络环境返回不同的加载时间。
- 2、获取数据库响应时间：该接口返回数据库的响应时间，可以根据不同的数据库操作类型返回不同的响应时间。
- 3、获取网络请求时间：该接口返回网站前后端交互的时间，可以根据不同的请求类型返回不同的时间。
- 4、获取并发用户数：该接口返回网站在不同并发用户数下的性能表现，可以根据不同的并发用户数返回不同的响应时间和请求成功率。

这些接口可以帮助网站管理员监测网站的性能表现，及时发现问题并进行优化。

3.6 CSCI 内部接口需求

1、模块间通信接口

- 数据传输格式：使用 JSON 格式进行数据传输。
- 数据内容：明确需要传输的数据，如用户信息、博客信息、评论信息等。对于每个数据类型，需要明确其数据结构，包括字段名、数据类型、长度等。
- 接口协议：前后端的通信使用 HTTP 协议。在前端部分，可以使用 Axios 作为网络请求工具，在后端部分，可以使用 Spring Boot 的 Web 框架作为 HTTP 服务提供

者。同时，后端也需要与消息中间件 RabbitMQ 进行通信，使用 AMQP 协议进行消息传输。对于搜索引擎 Elasticsearch 的通信，可以使用 Elasticsearch 提供的 RESTful API 进行通信。对于对象存储 MinIO，可以使用 MinIO 提供的 API 进行通信。

2、数据库接口

- 数据库表结构：需要创建用户信息表、博客信息表、评论信息表、相册信息表、日志信息表、角色信息表、权限信息表等。
- 数据类型：需要使用适当的数据类型存储不同类型的数据，如字符串、整数、日期、布尔值等。
- 索引：可以为一些经常被查询的字段创建索引，如用户表中的邮箱、博客表中的标题和标签等。同时需要注意索引的维护和更新。
- 交互方式：可以使用 ORM 框架（MyBatisPlus）进行数据库交互，通过实体类映射数据库表结构，简化 SQL 操作。可以通过手写 SQL 语句实现高效的数据查询和操作。同时，需要注意数据库的连接池和事务管理等。

3、缓存接口

- 缓存数据结构：对于缓存数据的存储，考虑使用 Redis 作为缓存中间件，采用键值对存储数据，如博客文章、用户信息等，可以使用字符串类型存储；对于复杂的数据结构，如评论、回复等，可以使用 Redis 的哈希表或列表进行存储。
- 缓存策略：为了保证缓存的高效性，需要制定缓存淘汰策略。可以考虑使用 LRU（Least Recently Used，最近最少使用）算法或 LFU（Least Frequently Used，最不经常使用）算法，也可以根据业务需求自定义缓存淘汰策略。
- 缓存更新方式：在数据更新时，需要及时更新缓存中的数据。可以考虑采用主动更新和被动更新相结合的方式。主动更新指在数据更新后主动清除缓存中对应的数据，以保证缓存中的数据与数据库中的数据一致；被动更新指在缓存数据过期或缓存中没有对应的数据时，再去查询数据库并将查询到的数据重新缓存。同时，在写入缓存时，需要注意缓存的过期时间，以防止缓存数据过期而无法使用。

4、消息队列接口

消息队列接口是一个用于异步通信的重要接口，可以将不同模块之间的通信解耦并提高系统性能。

- 消息格式：在消息队列接口中，明确消息的格式非常重要，因为不同的模块可能使用不同的消息格式进行通信。因此，需要明确使用哪种消息格式，如 JSON、XML 等。通常，JSON 是一个非常流行的消息格式，因为它易于阅读和编写，并且具有广泛的支持。
- 消息内容：在消息队列接口中，需要明确需要传输哪些消息内容。这通常包括消息类型、消息内容等。消息类型可以用于标识消息的目的，例如，一个模块可能需要

向另一个模块发送一个任务请求消息。消息内容则可以包含任务的详细信息，例如任务的 ID、执行时间等。

- 消息队列协议：在消息队列接口中，需要明确使用哪种协议进行消息传输。不同的消息队列系统使用不同的协议，如 AMQP、STOMP 等。在选择协议时，需要考虑系统的性能要求、可靠性和扩展性等因素。例如，AMQP 是一种可靠的消息队列协议，适用于需要高可靠性和安全性的应用程序，而 STOMP 则是一种轻量级的消息队列协议，适用于需要高性能和低延迟的应用程序。

5、安全接口

- 用户认证：明确用户认证方式，如使用用户名密码、第三方授权等。Software Chicken 个人博客网站提供注册功能，要求用户提供邮箱和密码进行认证。此外，还提供了第三方登录功能，如使用 QQ 等第三方账号进行认证。
- 授权：明确用户授权方式，如角色授权、权限授权等。您的博客网站可以通过角色授权和权限授权来控制用户对不同资源的访问权限。将管理员和普通用户分别赋予不同的角色，同时针对不同资源设置不同的权限，以控制用户对资源的访问。
- 加密：明确数据加密方式，如使用对称加密、非对称加密等。可以使用 HTTPS 协议来加密传输数据，防止网络传输过程中的数据被窃听和篡改。此外，您还可以对一些敏感数据进行加密处理，如用户密码等。
- 防火墙：明确如何设置防火墙，如白名单、黑名单等。您的博客网站可以通过设置网络防火墙来限制对网站的访问，以保护网站免受恶意攻击和入侵。可以通过设置 IP 白名单，只允许白名单中的 IP 访问网站。此外，还可以设置黑名单，防止一些已知的恶意 IP 对网站进行攻击。

6、性能接口

- 响应时间：系统对请求的响应时间是一个重要的性能指标，通常需要明确定义对于不同的请求类型或者模块，对应的响应时间的上限。对于一个需要查询数据库的请求，需要在 1 秒内返回结果。
- 吞吐量：吞吐量是系统能够同时处理的请求数量，通常需要根据业务场景进行评估并定义。对于个人博客网站需要支持每秒处理 100 个用户下单请求。
- 并发量：并发量是指系统同时处理的请求数量，通常需要根据系统负载测试结果进行评估。对于个人博客网站需要支持同时连接 1000 个用户。

针对上述指标，通常需要进行性能测试以验证系统是否能够满足要求，并根据测试结果进行调整和优化。同时，也需要关注系统的硬件资源，如 CPU、内存、网络带宽等，以保证系统的性能表现。

7、日志接口

- 日志格式：明确日志记录的格式，如 JSON、XML 等。例如，可以使用 JSON 格

式记录日志信息，包括时间戳、日志等级、错误信息、请求路径、用户 IP 等。

- 日志内容：明确需要记录哪些内容，如错误信息、调试信息等。例如，在记录错误日志时，需要记录错误堆栈信息、异常类型等；在记录访问日志时，需要记录请求路径、请求参数等。
- 日志等级：明确日志的等级，如 DEBUG、INFO、ERROR 等。不同等级的日志适用于不同的场景，例如，在开发和测试阶段使用 DEBUG 等级记录详细的调试信息，而在生产环境中使用 ERROR 等级记录错误信息。

3.7 CSCI 内部数据需求

Software Chicken 个人博客网站的数据库设计如下：

1、用户表（User），字段如下：

- ID：主键 ID，自增
- Email：用户邮箱
- Password：加密后的用户密码
- Nickname：用户昵称
- Avatar：用户头像链接
- Introduction：个人简介
- BlogUrl：博客链接
- Status：用户状态（正常、冻结等）

2、博客表（Blog），字段如下：

- ID：主键 ID，自增
- Title：博客标题
- Content：博客内容
- CreateTime：博客创建时间
- UpdateTime：博客更新时间
- IsTop：是否置顶
- IsRecommend：是否推荐
- ReadCount：阅读量
- Type：博客类型（原创、转载等）
- CoverImage：博客封面图片

- UserId: 作者 ID (外键关联 User 表)
- CategoryId: 博客分类 ID (外键关联 Category 表)

3、博客分类表 (Category) , 字段如下:

- ID: 主键 ID, 自增
- Name: 分类名称
- Description: 分类描述

4、标签表 (Tag) , 字段如下:

- ID: 主键 ID, 自增
- Name: 标签名称

5、博客标签关联表 (BlogTag) , 字段如下:

- ID: 主键 ID, 自增
- BlogId: 博客 ID (外键关联 Blog 表)
- TagId: 标签 ID (外键关联 Tag 表)

6、评论表 (Comment) , 字段如下:

- ID: 主键 ID, 自增
- Content: 评论内容
- CreateTime: 评论创建时间
- UserId: 评论用户 ID (外键关联 User 表)
- BlogId: 博客 ID (外键关联 Blog 表)
- ParentId: 父级评论 ID (用于评论的回复)

7、说说表 (Moments) , 字段如下:

- ID: 主键 ID, 自增
- Content: 说说内容
- CreateTime: 说说创建时间
- Image: 说说图片
- UserId: 说说用户 ID (外键关联 User 表)

8、友链表 (FriendLink) , 字段如下:

- ID: 主键 ID, 自增
- Name: 友链名称

- Url: 友链 URL
- Description: 友链描述

9、留言表 (Message) , 字段如下:

- ID: 主键 ID, 自增
- Content: 留言内容
- CreateTime: 留言创建时间
- UserId: 留言用户 ID (外键关联 User 表)

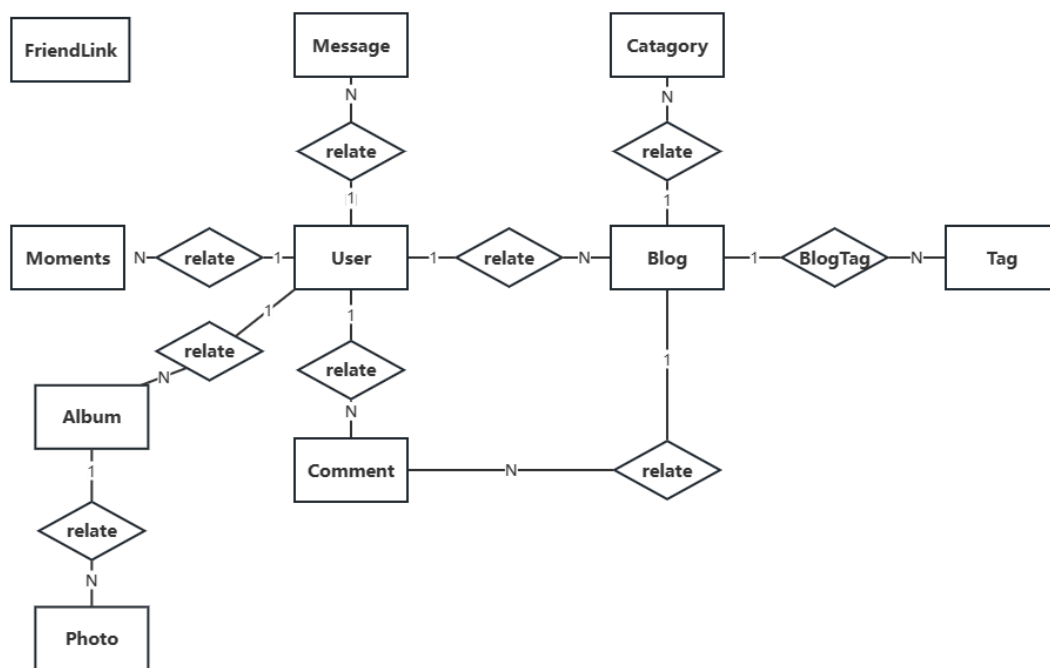
10、相册表 (Album) , 字段如下:

- ID: 主键 ID, 自增
- Name: 相册名称
- Description: 相册描述
- UserId: 相册用户 ID (外键关联 User 表)

11、照片表 (Photo) , 字段如下:

- ID: 主键 ID, 自增
- Name: 照片名称
- Description: 照片描述
- Url: 照片 URL
- CreateTime: 照片创建时间
- AlbumId: 相册 ID (外键关联 Album 表)

数据库 ER 图如下:



3.8 适应性需求

博客网站的适应性需求包括以下方面：

- 1、响应式设计：博客网站需要适应不同设备、分辨率和屏幕尺寸，使用户能够在各种平台上获得良好的浏览体验。
- 2、浏览器兼容性：博客网站需要兼容主流的浏览器和操作系统，确保用户能够正常访问和使用网站。
- 3、多语言支持：为了满足全球范围内的用户需求，博客网站有时需要提供不同语言版本，以便于更多人阅读与交流。
- 4、安全性：博客网站需要具备一定的安全保护机制，例如防止黑客攻击、保护用户隐私信息等。
- 5、技术更新与维护：随着技术的快速发展和用户需求变化，博客网站需要及时更新技术、修复代码缺陷，并提供周全的技术支持和维护。

3.9 保密性需求

博客网站的保密性需求包括以下方面：

- 1、用户隐私保护：博客网站需要保证用户个人隐私信息的安全性，如用户名、密码、电子邮件地址和收货地址等。
- 2、数据安全性：博客网站需要通过技术手段防止敏感数据泄漏和遭受攻击，如加密传输和存储、访问控制、防火墙等措施。

- 3、作者身份验证：为确保博客内容来源的真实性和可信度，在需要时要对作者身份进行验证。
- 4、内部信息保密性：如企业内部博客，需要严格保护公司的商业秘密和关键业务信息不被外泄。
- 5、禁止未经授权的复制或传播：博客所有权归属明确，禁止未经授权的复制或任何形式的传播。

3.10 保密性和私密性需求

博客网站的保密性和私密性需求包括以下方面：

- 1、用户隐私保密：博客网站需要确保用户个人隐私信息的保密性，如用户名、密码、电子邮件地址和收货地址等，以及用户发布在博客上的个人信息和文章。
- 2、加密传输和存储：博客网站需要采取技术手段保证数据的加密传输和存储，防止敏感数据被窃取或篡改。
- 3、限制访问权限：博客网站需要支持管理员对文章、评论或其他内容进行访问权限的限制。例如，博客管理员可以将某些文章或内容设为私密，只有特定的用户或群体才能访问。
- 4、身份验证机制：博客网站可以通过身份验证机制来确认访问者身份，以控制访问特定的文章或内容。例如，博客网站可以要求用户登录后方可查看特定的博客文章或信息。
- 5、数据备份与恢复：博客网站需要进行定期的数据备份，以避免数据因各种原因而丢失。同时，博客网站还需要配备适当的数据恢复技术，以尽快地恢复损失的数据。

3.11 CSCI 环境需求

Software Chicken 个人博客网站的 CSCI 环境需求包括计算机硬件、操作系统、数据库、Web 服务器和其他相关软件：

1、计算机硬件：

Software Chicken 个人博客网站的 CSCI 环境需求中的计算机硬件部分需要包括服务器配置和客户端配置。

服务器配置：

- CPU：至少 8 核 CPU 或更高配置，以满足处理请求的需要。
- 内存：至少 16GB RAM 或更高配置，以确保服务器可以同时处理多个用户请求。
- 存储：至少 500GB 硬盘空间或更高配置，以存储博客、评论、照片等数据。建议使用 SSD 存储以提高访问速度。

客户端配置：

- CPU：建议至少 2 核 CPU 或更高配置，以保证浏览博客时的流畅度。
- 内存：建议至少 4GB RAM 或更高配置，以确保可以同时打开多个标签页，同时运行其他应用程序。
- 存储：建议至少 256GB 硬盘空间或更高配置，以存储浏览器缓存、临时文件和其他应用程序数据。建议使用 SSD 存储以提高访问速度。

需要注意的是，实际需要的硬件配置可能会因为网站的访问量、数据量等因素而有所不同，因此应该根据实际情况进行调整。

2、操作系统：

在 Software Chicken 个人博客网站的 CSCI 环境需求中，操作系统是至关重要的组成部分。以下是操作系统的详细要求：

服务器端操作系统：

- 建议使用最新版本的 Linux 或 Windows Server 操作系统；
- Linux 操作系统的推荐版本包括 CentOS、Ubuntu 等，建议选择稳定版本；
- Windows Server 操作系统的推荐版本包括 Windows Server 2022 等，建议选择稳定版本；
- 操作系统必须支持所需的数据库管理系统、Web 服务器软件、应用程序和其他系统软件；

客户端操作系统：

- 建议使用最新版本的 Windows、Mac OS 或 Linux 操作系统；
- Windows 操作系统的推荐版本包括 Windows 10 等，建议选择稳定版本；
- Mac OS 操作系统的推荐版本包括 macOS Big Sur 等，建议选择稳定版本；
- Linux 操作系统的推荐版本包括 Ubuntu 等，建议选择稳定版本；
- 客户端操作系统必须支持所需的 Web 浏览器和其他应用程序。

3、数据库：

数据库是个人博客网站的核心组件之一，用于存储用户、博客、评论等数据。以下是 Software Chicken 个人博客网站的 CSCI 环境需求的数据库部分：

- 数据库类型：MySQL 或 PostgreSQL。这两个数据库都是流行的开源关系型数据库，支持高可靠性、高可扩展性和高性能，因此可以满足个人博客网站的需求。
- 数据库版本：建议使用最新版本，以便获取最新的功能和性能优化。
- 数据库服务器：应该在单独的服务器上运行，以便提高性能和安全性。建议使用

至少 8 核 CPU、16GB RAM、500GB 硬盘空间或更高配置的服务器。

- 数据库备份：应该定期备份数据库以避免数据丢失。建议使用自动化备份工具，例如 `mysqldump` 或 `pg_dump`。
- 数据库安全性：应该采取适当的安全措施，例如限制对数据库的访问、使用防火墙、加密敏感数据等。同时，应定期更新数据库软件和操作系统以避免安全漏洞的风险。

4、Web 服务器：

为了保证个人博客网站的正常运行，需要满足以下 Web 服务器的 CSCI 环境需求：

- Apache 服务器或 Nginx 服务器，建议使用最新版本；
- 必须支持 HTTP 和 HTTPS 协议；
- 必须支持动态网页请求，如 PHP 或 Python 等脚本语言；
- 必须支持静态文件请求，如 HTML、CSS、JS 等；
- 必须支持 URL 重写功能，以实现美观的 URL 路径；
- 必须支持 gzip 压缩，以提高网站访问速度；
- 必须支持反向代理，以实现负载均衡和缓存等功能；
- 必须支持 HTTPS 证书配置，以保证网站的安全性；
- 必须支持 IP 限制和访问日志记录，以保护网站安全。

5、其他相关软件：

- 编程语言：建议使用 Java 等主流编程语言来进行开发。Java 是一种面向对象的编程语言，拥有强大的生态系统和丰富的库。Java 可用于开发各种类型的 Web 应用程序，具有高效、稳定和安全的特性。Java 也是许多大型企业和组织的首选编程语言。
- Web 框架：建议使用 Spring Boot 等主流 Web 框架，具体根据开发人员的熟练度来决定。Spring Boot 是一个用于创建独立的、生产级别的 Spring 应用程序的框架。它提供了各种特性和工具，包括自动配置、内嵌式服务器、安全性、监控、度量和测试等，让开发人员更容易地构建和部署 Web 应用程序。
- 前端技术：建议使用 Vue3、Vue2、pinia、vuex、vue-router、axios 等主流前端技术。Vue.js 是一个轻量级的 JavaScript 框架，用于构建交互式的 Web 应用程序。Vue.js 具有易学易用、高效、灵活、可组合、易于测试等优点。pinia 是一个状态管理库，vuex 是 Vue.js 的官方状态管理库，vue-router 是 Vue.js 的官方路由库，axios 是一个用于发送 HTTP 请求的 JavaScript 库，这些工具可用于提高开发效率和构建高质量的 Web 应用程序。
- 版本控制：建议使用 Git 进行版本控制。Git 是一种分布式版本控制系统，可用于

跟踪和管理代码变更。它具有易于使用、高效、可靠、安全等特性，能够支持多个开发人员的协同工作，使得开发人员可以更轻松地管理代码。

以上是个人博客网站的 CSCI 环境需求，具体要求根据实际情况进行调整。

3.12 计算机资源需求

3.12.1 计算机硬件需求

本项目开发共需要 1 台服务器，服务器详细需求信息如下：

处理器：

- 8 核 CPU 或更高配置

存储器：

- 16 GB RAM 或更高配置
- 500 GB 硬盘空间或更高配置

输入/输出设备：

- 键盘、鼠标、显示器

辅助存储器：

- 备份存储器，建议使用云备份或外部硬盘

通信/网络设备：

- 网络接口卡 (NIC) 用于网络连接
- 路由器和交换机，用于连接服务器和客户端设备

其他所需设备：

- 服务器机箱，主板和电源供应器
- 服务器风扇和散热器，确保服务器的正常运行和降温
- 服务器管理卡 (IPMI) 用于远程管理和监控服务器

特征：

- 服务器配置建议至少 8 核 CPU、16GB RAM、500GB 硬盘空间或更高配置
- 硬件设备需要稳定可靠，以确保网站运行稳定，并满足用户的访问需求
- 硬件需要支持最新的操作系统和软件，以满足网站未来扩展和更新的需求

3.12.2 计算机硬件资源利用需求

- 处理器能力：Software Chicken 个人博客网站需要最多使用 8 个处理器，但实际上，平均使用量很少超过 50%，因此最大使用率应该在 60% 以下。
- 存储器容量：Software Chicken 个人博客网站需要至少 16 GB 的内存容量，但实际上，平均使用量很少超过 60%，因此最大使用率应该在 80% 以下。
- 输入/输出设备能力：Software Chicken 个人博客网站需要支持多个用户同时访问，因此需要一个可以支持高并发访问的网络适配器。网站的输入/输出设备能力需要最大化以确保响应时间短。
- 辅助存储器容量：Software Chicken 个人博客网站需要一个至少 500 GB 硬盘容量的服务器来存储网站的数据和文件。平均使用率应该在 70% 以下，以确保在需要时有足够的空间进行扩展。
- 通信/网络设备能力：Software Chicken 个人博客网站需要支持高并发访问，因此需要一个高速、可靠的网络连接。服务器需要支持千兆以太网和无线局域网，以满足多种用户需求。
- 测量资源利用的条件：为了确保 Software Chicken 个人博客网站的最佳性能，需要使用监控工具来实时跟踪和记录处理器、存储器、输入/输出设备、辅助存储器和网络设备的利用率。将使用这些工具来确定我的服务器是否足够强大以处理网站的流量和资源需求，并根据需要进行升级或扩展。

3.12.3 计算机软件需求

1、操作系统：建议使用 Linux 或 Windows Server 操作系统的最新版本，以确保系统的稳定性和安全性。

- Linux 操作系统：Ubuntu 22.04 LTS，文档参考：<https://ubuntu.com/>
- Windows Server 操作系统：Windows Server 2022，文档参考：<https://docs.microsoft.com/en-us/windows-server/>

2、数据库管理系统：需要使用 MySQL 或 PostgreSQL 数据库的最新版本来存储和管理网站的数据，确保数据的完整性和可靠性。

- MySQL 数据库管理系统：MySQL 8.0，文档引用可参考官方文档：<https://dev.mysql.com/doc/>
- PostgreSQL 数据库管理系统：PostgreSQL 14，文档引用可参考官方文档：<https://www.postgresql.org/docs/>

3、通信/网络软件：需要使用网络通信软件如 TCP/IP 协议等，以便用户可以通过 Internet 访问个人博客网站，保证网站的可访问性。

- Apache：一个流行的 Web 服务器软件，支持 TCP/IP 协议，用于向用户提供网站内容。最新版本可以在 Apache 官方网站 (<https://httpd.apache.org/>) 上找到，并包含详细文档和使用指南。

- Nginx: 一个常用的 Web 服务器软件, 支持 TCP/IP 协议。最新版本可以在 Nginx 官方网站 (<https://nginx.org/>) 上找到, 并包含详细文档和使用指南。

- OpenSSH: 一个用于安全远程访问的软件, 支持 TCP/IP 协议。最新版本可以在 OpenSSH 官方网站 (<https://www.openssh.com/>) 上找到, 并包含详细文档和使用指南。

4、实用软件: 需要使用开发工具, 如 Eclipse、Visual Studio Code 等, 以便开发人员可以编写和调试网站的代码。同时需要使用 Web 服务器如 Apache 或 Nginx, 以便能够将网站文件提供给访问者。

- Eclipse: <https://www.eclipse.org/downloads/>

- Visual Studio Code: <https://code.visualstudio.com/download>

5、输入和设备模拟器: 需要使用输入和设备模拟器来模拟网站在不同操作系统和设备上的表现, 以确保网站的兼容性和可用性。例如, 可以使用浏览器开发者工具来模拟不同分辨率的设备访问网站。

常见的输入和设备模拟器包括:

- Chrome DevTools (<https://developers.google.com/web/tools/chrome-devtools/>): Chrome 浏览器自带的开发者工具, 可以模拟不同分辨率、不同设备、不同网络环境等。

- Firefox Developer Tools (<https://developer.mozilla.org/en-US/docs/Tools>): Firefox 浏览器自带的开发者工具, 也可以模拟不同分辨率、不同设备、不同网络环境等。

- Safari Developer Tools (<https://developer.apple.com/safari/tools/>): Safari 浏览器自带的开发者工具, 可以模拟不同分辨率、不同设备、不同网络环境等。

- Microsoft Emulator (<https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/>): 微软提供的虚拟机, 可以模拟不同版本的 Windows 操作系统和 IE 浏览器。

- BrowserStack (<https://www.browserstack.com/>): 提供各种浏览器和设备的云端测试服务, 可以模拟不同的设备和浏览器。

需要注意的是, 输入和设备模拟器是为了测试和优化网站的兼容性和可用性, 但最终还是要在真实的设备和浏览器上进行测试, 以确保网站的表现和用户体验。

6、测试软件: 需要使用测试工具, 如 Selenium、JUnit 等, 以确保网站的各个模块和功能能够正常工作。同时, 也需要使用性能测试工具, 如 Apache JMeter 等, 以测试网站的负载能力和响应时间等性能指标。

- Selenium: Selenium 是一个自动化测试框架, 用于测试 Web 应用程序的功能和用户界面。最新版本是 Selenium 4, 可以从官方网站 (<https://www.selenium.dev/>) 下载, 并且官方也提供了详细的文档和教程。

- **JUnit:** JUnit 是一个 Java 语言的单元测试框架，用于测试 Java 应用程序的各个单元。最新版本是 JUnit 5，可以从官方网站 (<https://junit.org/junit5/>) 下载，并且官方也提供了详细的文档和教程。
- **Apache JMeter:** Apache JMeter 是一个性能测试工具，用于测试 Web 应用程序的负载能力和响应时间等性能指标。最新版本是 JMeter 5.4.1，可以从 Apache 官方网站 (<https://jmeter.apache.org/>) 下载，并且官方也提供了详细的文档和教程。

7、生产用软件：需要使用生产用软件，如 Git 等版本控制工具，以确保网站的代码和文件都能够进行版本管理和备份，同时也可以协同多人进行开发。

Git 是目前最流行的分布式版本控制系统之一，可以轻松地管理和跟踪网站的代码变化。Git 的最新版本为 2.31.1，可以从官方网站 <https://git-scm.com/> 下载安装，同时也提供了详细的文档和教程。

3.12.4 计算机通信需求

- **连接的地理位置:** 个人博客网站的服务器应该放置在互联网服务提供商 (ISP) 的数据中心内，该数据中心应位于一个稳定的地理位置，如城市中心、商业中心等。
- **配置和网络拓扑结构:** 服务器需要一个高速的以太网连接到互联网。网络拓扑结构应该是一种高度可靠的、分层的、冗余的架构，包括至少两个边界路由器、交换机、防火墙等网络设备，以确保高可用性和故障恢复性。
- **传输技术:** 传输技术应该支持 TCP/IP 协议，确保数据传输的可靠性和安全性。此外，HTTPS 应该用于加密数据传输，以确保传输的机密性。
- **数据传输速率:** 服务器应具有足够的带宽来支持网站的流量，确保用户可以快速访问网站。带宽的需求取决于预计的网站访问量。
- **网关:** 网关应该支持高速、安全的数据传输，并具有负载均衡和故障转移功能，以确保高可用性。
- **要求的系统使用时间:** 个人博客网站应该随时可用，因此需要 24/7 的系统支持和监控。
- **传送/接收数据的类型和容量:** 个人博客网站需要支持传送和接收各种数据类型，包括文本、图像、视频等。数据容量应该适合使用场景，避免过度耗费带宽和服务器资源。
- **传送/接收/响应的时间限制:** 服务器需要快速响应用户请求，以确保用户体验良好。传输、接收和响应的时间限制应该根据预计的网站访问量和数据传输速率进行优化。
- **数据的峰值:** 服务器应该能够处理突发的高访问量，以确保网站的可用性和稳定性。

- 诊断功能：服务器应该具有实时监控和诊断功能，以便及时发现和解决故障。

3.13 软件质量因素

- 1.功能性：博客平台需要具备基本的博客功能，如发布文章、评论、搜索、分类、标签等。
- 2.可靠性：博客平台需要能够产生正确、一致的结果，并能够在不间断的情况下运行。
- 3.可维护性：博客平台需要易于更正和修改，以便进行修复和改进。
- 4.可用性：博客平台需要易于访问和操作，并提供用户友好的界面和功能。
- 5.灵活性：博客平台需要能够适应不同的需求和变化，例如新的功能或设计更改。
- 6.可移植性：博客平台需要能够在不同的环境下运行，并且易于修改以适应新环境。
- 7.可重用性：博客平台的代码和功能应该能够在其他应用中进行重用。
- 8.可测试性：博客平台需要易于进行测试，以确保其功能的正确性和可靠性。
- 9.易用性：博客平台需要易于学习和使用，并提供适当的文档和帮助。

3.14 设计和实现的约束

- a. 数据库和其他软件配置项：博客平台需要使用适当的数据库和其他必要的软件配置项来支持其功能。这可能包括使用特定版本的数据库软件或使用特定的操作系统。
- b. 设计或实现标准的使用：博客平台的设计和实现需要符合适当的标准和规范，例如 Web 应用程序开发标准。
- c. 灵活性和可扩展性：博客平台需要具备一定的灵活性和可扩展性，以便在未来增长和变化的情况下进行支持。例如，博客平台需要支持添加新的功能和插件，并且需要易于扩展和修改。

3.15 数据

- 1.输入数据：博客平台需要能够接受用户输入的文章内容、评论、标签、分类等信息。
- 2.输出数据：博客平台需要能够输出已发布的文章、评论、标签、分类等信息，并能够在用户请求时进行搜索和排序。
- 3.数据管理能力：博客平台需要能够管理大量的文章、评论、标签、分类等数据，并能够支持数据的备份和恢复。
- 4.处理量和数据量：博客平台需要能够处理大量的数据和请求，并能够支持多用户同

时访问和操作。

3.16 操作

对于一个人博客平台，以下是一些可能的操作方面的要求：

- 1.常规操作：博客平台需要支持用户进行常规操作，例如创建、编辑、发布和删除文章、评论，添加和编辑标签和分类等。
- 2.特殊操作：博客平台可能需要支持一些特殊操作，例如高级搜索、多种排序方式等，以提高用户体验和效率。
- 3.初始化操作：博客平台需要支持初始化操作，例如安装、配置和初始化数据库等，以确保系统的正常运行。
- 4.恢复操作：博客平台需要支持数据备份和恢复操作，以便在发生数据丢失或系统崩溃时能够快速恢复系统。

3.17 故障处理

- a. 软件故障：软件故障是指由于代码错误、内存泄漏、配置问题等软件方面的原因导致系统无法正常工作。
- b. 错误信息：在发生故障时，系统应该提供详细的错误信息，以便管理员和用户可以迅速定位问题。例如，系统可以提供错误代码、错误消息、错误位置、异常堆栈等信息。
- c. 补救措施：为了处理软件故障，可以采取以下措施：
 - 软件更新：如果故障是由于软件错误导致的，则需要修复软件错误并发布更新版本。
 - 配置更改：如果故障是由于配置错误导致的，则需要更改配置并重新启动系统。
 - 数据恢复：如果故障导致数据丢失，则需要从备份中恢复数据。
 - 降级处理：如果故障无法立即修复，则可以采取降级处理措施，例如关闭某些功能或减少系统的负载等。

综上所述，对于一个小型博客平台，故障处理方面的要求包括及时提供详细的错误信息，并采取合适的措施进行处理，以确保系统的可靠性和稳定性。

3.18 算法说明

列举四个最常用的算法

1.文章评分算法

```
def calculate_article_score(upvotes, downvotes, views):
```



```

"""

计算文章得分

:param upvotes: 赞数
:param downvotes: 踩数
:param views: 浏览量
"""

# 使用 Reddit 算法计算得分
score = upvotes - downvotes
order = math.log10(max(abs(score), 1))
sign = 1 if score > 0 else -1 if score < 0 else 0
return round(order + sign * views / 100000, 7)

```

2. 字数统计算法

```

def count_words(text):
    """

    统计文本中的单词数

    :param text: 文本
    """

    # 使用正则表达式匹配单词
    words = re.findall(r'\b\w+\b', text)
    return len(words)

```

3. 图片缩放算法

```

def resize_image(image, width=None, height=None):
    """

    缩放图片到指定的宽度或高度

    :param image: 原始图片
    :param width: 缩放后的宽度
    :param height: 缩放后的高度
    """

    # 计算缩放比例
    w, h = image.size
    aspect_ratio = w / h

```

```

if width and height:
    target_ratio = width / height
    if aspect_ratio > target_ratio:
        width = round(height * aspect_ratio)
    else:
        height = round(width / aspect_ratio)
elif width:
    height = round(width / aspect_ratio)
elif height:
    width = round(height * aspect_ratio)
# 缩放图片
return image.resize((width, height), Image.ANTIALIAS)

```

4. 标签云算法 ²[1]

```

def generate_tag_cloud(tags, min_font_size=10, max_font_size=50):
    """
    生成标签云

    :param tags: 标签列表，每个标签为一个字典，包含 name 和 count 两个属性
    :param min_font_size: 最小字号
    :param max_font_size: 最大字号
    """
    # 根据标签数量计算字号范围
    num_tags = len(tags)
    if num_tags == 0:
        return ""
    elif num_tags == 1:
        font_size = max_font_size
    else:
        counts = [tag['count'] for tag in tags]
        count_min = min(counts)
        count_max = max(counts)
        if count_max == count_min:
            font_size_range = [max_font_size] * num_tags

```

```

else:
    font_size_range = [(max_font_size - min_font_size) * (count - count_min) /
                        (count_max - count_min) + min_font_size for count in counts]

# 生成 HTML 标签云
html = ""
for i, tag in enumerate(tags):
    html += f'<a href="#" style="font-size: {font_size_range[i]}px">{tag["name"]}</a>'

return html

```

3.19 有关人员需求

人员数量：本项目需要至少 1 名网站管理员和 1 名网站编辑。另外，根据网站的发展情况和访问量等因素，可能需要进一步增加人员。

技能等级：需要熟悉前端和后端开发技术，有一定的数据库管理经验，以及对网站运营和维护有相关的经验和技能。可以进一步说明开发人员需要掌握哪些编程语言和框架，管理员需要具备哪些管理和维护技能，以及其他相关职位的技能要求

责任期：所有项目人员的责任期为项目启动后的整个开发周期。

培训需求：新加入项目团队的人员需要接受相关培训，包括但不限于网站开发技术和网站管理等方面的知识。

其他信息：本个人博客网站的设计目标是可以支持同时存在 1000 个注册用户，同时可以容纳 10000 个未注册用户的访问。为了支持新用户的学习和使用，本网站需要提供内在帮助功能，包括但不限于用户指南、在线文档、教程视频等。此外，还需要提供技术支持和培训服务，以保证用户能够顺利使用网站。为了减少人为错误造成的影响，本网站需要在设计和开发中考虑人员在能力和局限性方面的因素。例如，在页面设计中需要注明错误消息的颜色和持续时间，关键指示器或关键的物理位置以及听觉信号的使用等要求，以帮助用户更快地发现和纠正错误。此外，在用户权限设计和开发过程中，需要避免人员在极端情况下的错误操作，例如误删除用户信息、文章等重要数据。

3.20 有关培训需求

1、在 CSCI 中包含的培训软件：开发人员需要提供一个易于使用、交互性好的后台管理系统，以便网站管理员可以方便地进行文章管理、网站配置、评论审核等操作。开发人员还需要提供用户手册和技术支持文档，以帮助管理员和终端用户更好地了解和和使用网站的各项功能。

2、培训的目标用户：目标用户包括网站管理员和终端用户。网站管理员需要接受系统的详细培训，了解网站后台的各项功能和操作方法，掌握基本的网站管理知识，例如文章发布、评论审核等。终端用户需要得到一些简单的培训，例如如何注册、如何发表评论、如何浏览文章等。

3、培训方式：开发人员需要提供多种培训方式，例如视频教程、在线文档、在线支持等。视频教程可以提供图文并茂的操作指南，使管理员和用户更加容易地理解网站的各项功能和操作方法；在线文档和在线支持可以为管理员和用户提供了随时随地的技术支持，帮助他们解决使用过程中遇到的问题。

4、培训时长和频次：针对不同的用户，开发人员需要制定不同的培训计划。对于网站管理员，需要提供一定时长的详细培训，以确保他们能够熟练地使用网站的各项功能。对于终端用户，需要提供简单的培训，以帮助他们快速上手。此外，开发人员还应该根据用户反馈不断改进培训计划，提高培训效果。

3.21 有关后勤需求

1、系统维护：个人博客网站应该保证 24/7 的在线运行，因此需要确保系统的稳定性和可靠性。系统管理员需要进行定期的系统维护和升级，包括但不限于操作系统、数据库、服务器等方面的维护。

2、软件支持：为了保证个人博客网站的正常运行，需要对使用的软件进行支持。对于博客网站使用的开源软件或第三方组件，需要确保其能够与整个系统无缝衔接，不会引起冲突或安全漏洞。同时，需要对软件进行及时更新和升级。

3、系统运输方式：个人博客网站的运输方式应该是通过网络进行的，因此需要保证网络连接的稳定和可靠，避免网络瘫痪或中断导致用户无法访问博客网站。同时，需要对网络进行优化和调整，以确保访问速度和响应时间。

4、供应系统的需求：个人博客网站需要确保其系统能够长期运行，因此需要考虑系统所需的电力、冷却、网络等方面的供应需求。在建设和维护网站时，需要确保这些资源的可用性和稳定性。

5、对现有设施的影响：在建设个人博客网站时，需要考虑现有设施的使用情况和限制。例如，建设服务器机房时需要考虑机房的大小、环境控制等因素，以确保服务器的安全和稳定运行。

6、对现有设备的影响：建设个人博客网站需要考虑现有设备的使用情况和限制。例如，网站的访问量会影响网络设备和服务器的负载，需要根据实际情况调整设备的配置和数量，以确保博客网站的正常运行。

3.22 其他需求

1、安全性需求：包括保护用户数据的机制、防止恶意攻击的措施、对系统漏洞的修补

和更新等。

2、性能需求：包括响应时间、吞吐量、并发用户数等指标，需要保证网站的性能能够满足用户需求。

3、可维护性需求：包括代码可读性、可扩展性、易维护性等，需要保证网站能够方便地进行维护和更新。

4、可靠性需求：需要保证网站稳定可靠，能够长时间运行不崩溃或出现错误。

5、可用性需求：需要保证网站易用，符合用户的使用习惯和期望，提供良好的用户体验。

6、兼容性需求：需要保证网站能够在不同的浏览器、操作系统和设备上正常运行。

7、界面设计需求：需要保证网站的界面设计符合现代化设计风格，美观大方，符合用户审美观点。

3.23 包装需求

1、CSCI 需要在一个易于传输和交付的形式中交付，例如，压缩文件或打包文件。这将有助于在传输过程中减少文件丢失或损坏的风险。

2、所有 CSCI 应该打上适当的标签，以便于确定其内容和版本信息，同时标记包含特定模块或功能的 CSCI。例如，每个模块或组件应该有自己的标签。

3、CSCI 需要在适当的存储介质中交付，例如云存储或物理存储设备。

4、有关 CSCI 的任何必要的交付文档也需要被包含在交付物中，例如用户手册或操作指南。这些文档应该按照相应的标准进行格式化，并包括所有相关的信息。

5、所有 CSCI 的交付物应该被包装在防静电包装材料中，以防止电子元件在传输过程中被损坏。

6、CSCI 的交付物应该包含相关的版权和许可证信息，以确保其合法性。

3.24 需求的优先次序和关键程度

1、安全性需求：在网络应用程序中，安全性是至关重要的，它是保护用户数据免受未经授权的访问、数据泄露或恶意攻击的关键要素。因此，安全性需求应该是最高优先级的需求。

2、人员培训需求：为确保系统的正常运行和维护，需要对相关人员进行培训和支持，以便他们能够了解和应对新的技术和工具。因此，人员培训需求应该是高优先级的需求。

3、计算机资源需求：个人博客网站需要在服务器上运行，因此需要确定合适的计算机

资源以保证网站的顺畅运行。这是一个重要的需求，但相对于其他需求，它的优先级可能稍低。

4、后勤需求：系统的后勤需求涉及到系统维护、软件支持、系统运输方式、供应系统的需求、对现有设施的影响、对现有设备的影响等。这些需求虽然不如安全性和培训需求紧急，但是对系统的长期可持续性和可靠性具有重要作用，应该被视为中等优先级的需求。

5、保密性需求：虽然个人博客网站中不一定包含敏感信息，但是保密性仍然很重要。因此，对于保密性需求的重视程度取决于该网站所涉及的敏感程度。如果有一些敏感信息需要被保护，则这个需求可能需要比计算机资源需求更高的优先级。

6、适应性需求：适应性需求是指应用程序应能够适应各种不同的硬件和软件环境。在个人博客网站中，适应性需求相对较低，因为用户端的硬件和软件环境多数情况下都是相似的。

7、接口需求：接口需求指的是应用程序与其他系统或软件之间的接口。在个人博客网站中，接口需求相对较低，因为该系统很少与其他系统进行交互。

4 合格性规定

1、兼容性：系统应在主流的现代 Web 浏览器中正常运行，包括但不限于最新版的 Chrome、Firefox、Safari、Edge 等，确保网站在不同浏览器中的页面展示一致性。

2、响应式设计：系统应采用响应式设计，能够自适应不同屏幕尺寸和设备类型，包括桌面电脑、平板和手机等，保证在不同设备上都能够正常浏览和使用。

3、功能完整性：系统应按照需求文档中规定的功能要求进行开发，所有功能应能够正常运行，并保证其稳定性和可靠性。

4、安全性：系统应具备必要的安全措施，包括但不限于用户身份认证、密码加密存储、防止跨站脚本攻击（XSS）、防止 SQL 注入等，以保护用户信息和系统数据的安全。

5、性能：系统应具备良好的性能，包括但不限于页面加载速度、响应时间、并发处理能力等，以保证用户体验和系统的高效运行。

6、可维护性：系统应具备良好的代码结构和文档说明，方便后续的维护和升级工作，包括但不限于清晰的代码注释、模块化的代码设计、合理的命名规范等。

7、文档完整性：系统设计、开发、测试和维护过程中生成的文档应完整、准确、清晰地记录系统的设计、实现、测试和维护信息，包括但不限于设计文档、用户手册、技术文档等。

每种需求制定的方法如下：

- 1、CSCI 能力需求：规定系统应满足特定的功能和性能要求，例如支持用户注册和登录、发布和编辑博客文章、评论和点赞功能等。
- 2、CSCI 外部接口需求：规定系统与外部系统或服务的接口要求，例如与第三方登录、支付、地图等服务的集成要求，以确保系统与外部系统的正常通信和数据交互。
- 3、CSCI 内部接口需求：规定系统内部各个组件之间的接口要求，例如前端与后端的接口规范、数据库与应用服务器之间的接口要求等，以确保系统内部各组件之间的协调和合作。
- 4、CSCI 内部数据需求：规定系统内部数据的要求，包括数据格式、数据结构、数据存储和处理要求等，以确保系统内部数据的合法性、一致性和完整性。
- 5、适应性需求：规定系统应能够适应不同的环境和条件，例如不同浏览器、不同设备、不同网络环境等，以确保系统在不同情况下的稳定运行和用户体验。
- 6、保密性需求：规定系统对用户信息和系统数据的保密要求，包括用户隐私数据的保护、敏感信息的加密存储和传输、访问权限的控制等，以确保系统数据的安全性和保密性。
- 7、环境需求：规定系统的运行环境要求，例如操作系统、服务器配置、数据库配置等，以确保系统在特定的运行环境中能够正常运行。
- 8、计算机资源需求：规定系统对计算机资源的需求，例如处理器、内存、存储空间等，以确保系统在资源限制的情况下能够正常运行和满足性能要求。
- 9、方法指定：规定系统开发、测试、维护等过程中应采用的具体方法和工具，例如软件开发方法学、测试框架、版本控制工具等，以确保系统的开发和维护过程的规范性和可控性。

5 需求可追踪性

	ylQMw1	Sheet1				
	A	B	C	D	E	F
1	需求ID	需求描述	设计文档	编码	单元测试	集成测试
2	REQ001	用户注册功能	Y	Y	Y	Y
3	REQ002	用户登录功能	Y	Y	Y	Y
4	REQ003	文章发布功能		Y	Y	Y
5	REQ004	文章编辑功能		Y	Y	Y
6	REQ005	文章删除功能		Y	Y	Y
7	REQ006	文章搜索功能		Y	Y	Y
8	REQ007	文章评论功能		Y	Y	Y
9	REQ008	友链管理功能			Y	Y
10	REQ009	留言管理功能			Y	Y
11	REQ010	说说管理功能				Y

点击图片可查看完整电子表格

在上面的表格中，每一行代表一个需求，每一列代表需求的不同阶段，如设计文档、编码、单元测试、集成测试、系统测试和验收测试。在每个阶段完成后，相应的标记 "Y" 表示该需求已经在该阶段得到满足。通过这个需求可追踪性矩阵，可以清楚地了解每个需求在整个开发过程中的状态，从而确保项目按照需求规定的要求进行开发和测试，并及时追踪和验证需求的实现情况。

6 尚未解决的问题

- 1、用户权限管理：项目中可能需要实现用户权限管理功能，包括用户角色、权限的定义和管理，以及权限的授权和验证。目前尚未确定具体的用户权限管理方案。
- 2、数据安全性保障：项目中可能涉及用户的个人信息、文章内容等敏感数据，需要确保数据的安全性保障，包括数据的存储、传输和处理方面的安全措施。目前尚未完善数据安全性保障措施。
- 3、性能优化：项目中可能需要处理大量的用户请求和数据操作，需要对系统性能进行优化，以确保系统的响应速度和并发处理能力。目前尚未进行系统性能优化工作。
- 4、用户界面设计：项目中的用户界面设计可能需要进一步完善，包括用户界面的布局、样式、交互等方面的优化，以提供更好的用户体验。目前尚未完成用户界面设计工作。
- 5、测试与验证：项目需要进行全面的测试与验证，包括功能测试、性能测试、安全性测试等，以确保系统的稳定性、安全性和功能完整性。目前尚未完成全面的测试与验证工作。
- 6、部署与运维：项目需要在生产环境中进行部署和运维，包括系统的安装、配置、监

控和维护等方面的工作。目前尚未完成系统的部署与运维准备工作。

以上是一些本项目尚未解决的问题的例子，具体的未解决问题可能因项目的具体情况而有所不同，需要根据实际情况进行进一步的分析和补充。

7 注解

1 CSCI (Computer Software Configuration Item) :CSCI 是计算机软件配置项 (Computer Software Configuration Item) 的缩写，通常指具有独立版本控制和发布过程的软件模块。CSCI 是在软件工程中使用的术语，用于管理分布在多个计算机系统软件、文档和数据等内容。通过对 CSCI 进行管理，可以帮助开发人员更好地组织和协调软件开发流程，提高软件质量和可重用性。

2 标签云生成：即词云可视化，将网站关键词进行视觉化描述，用于汇总用户生成的标签或一个网站的文字内容。签一般是独立的词汇，常常按字母顺序排列，其重要程度又能通过改变字体大小或颜色来表现，所以标签云可以灵活地依照字序或热门程度来检索一个标签。

附录

附录 A 项目 E-R 图

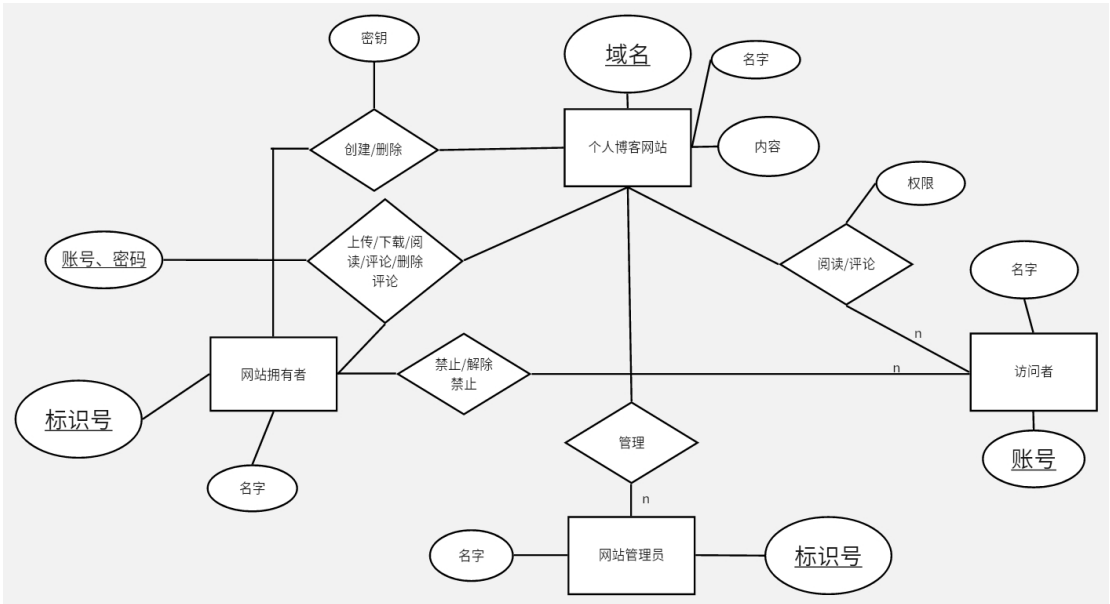
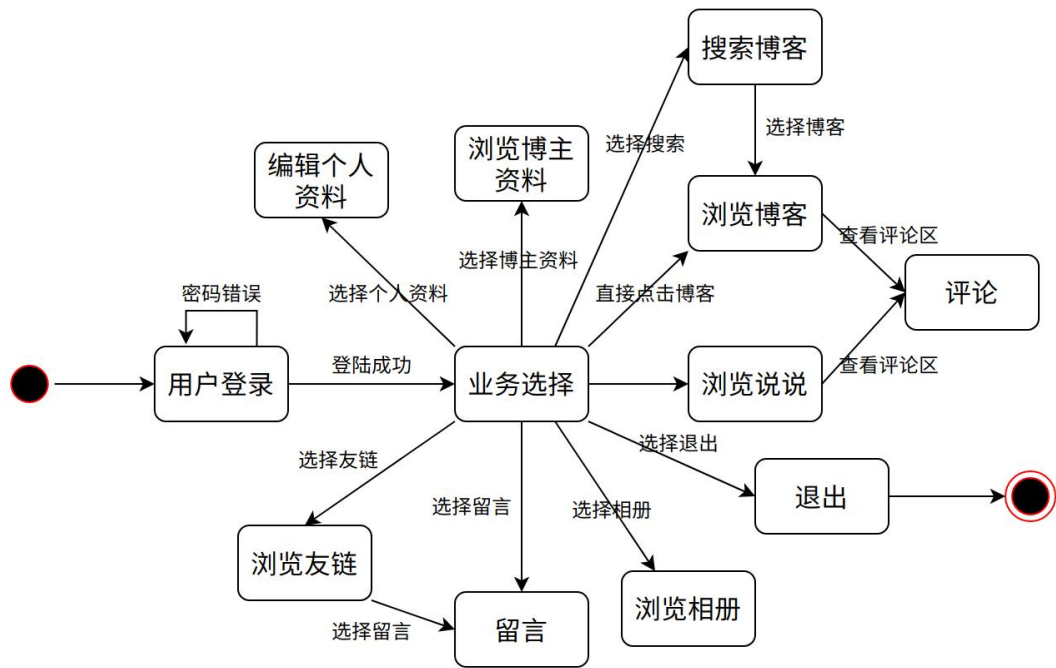


图 1 项目 E-R 图

附录 B 状态图

前台用户状态图：



后台博主状态图：

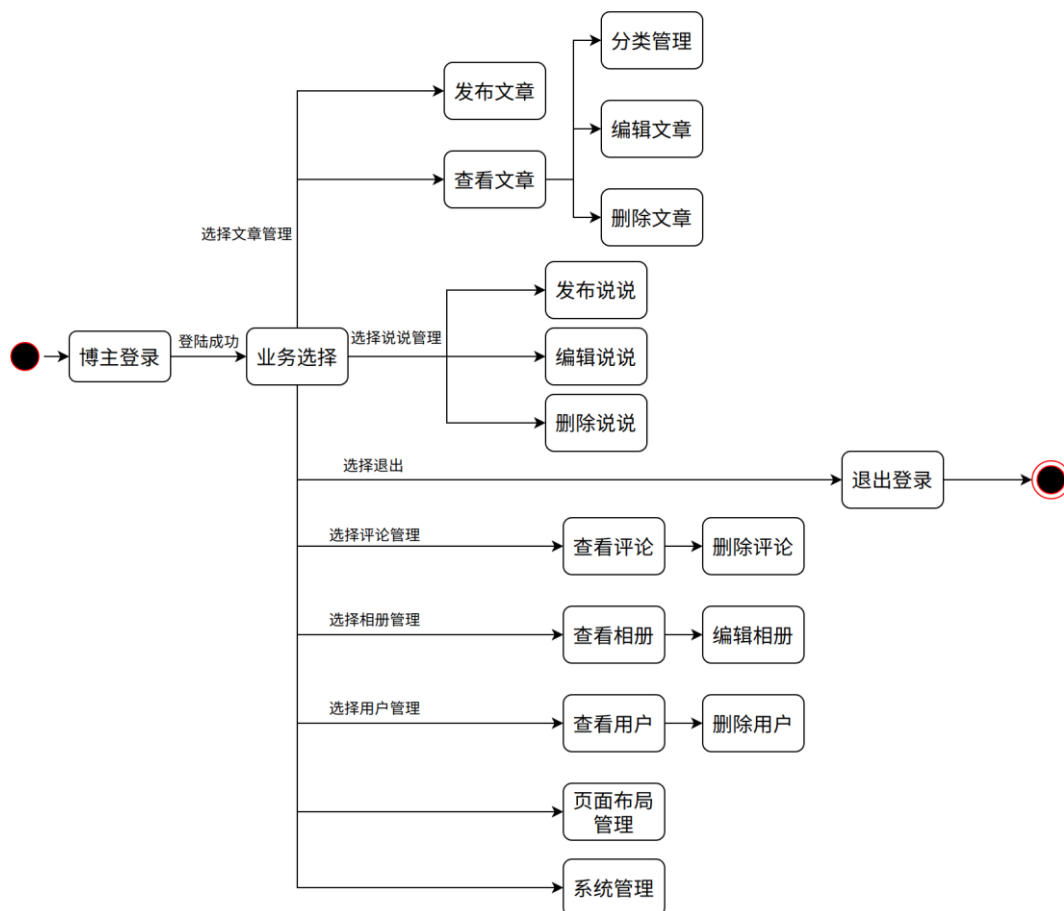


图 2 前台用户状态图

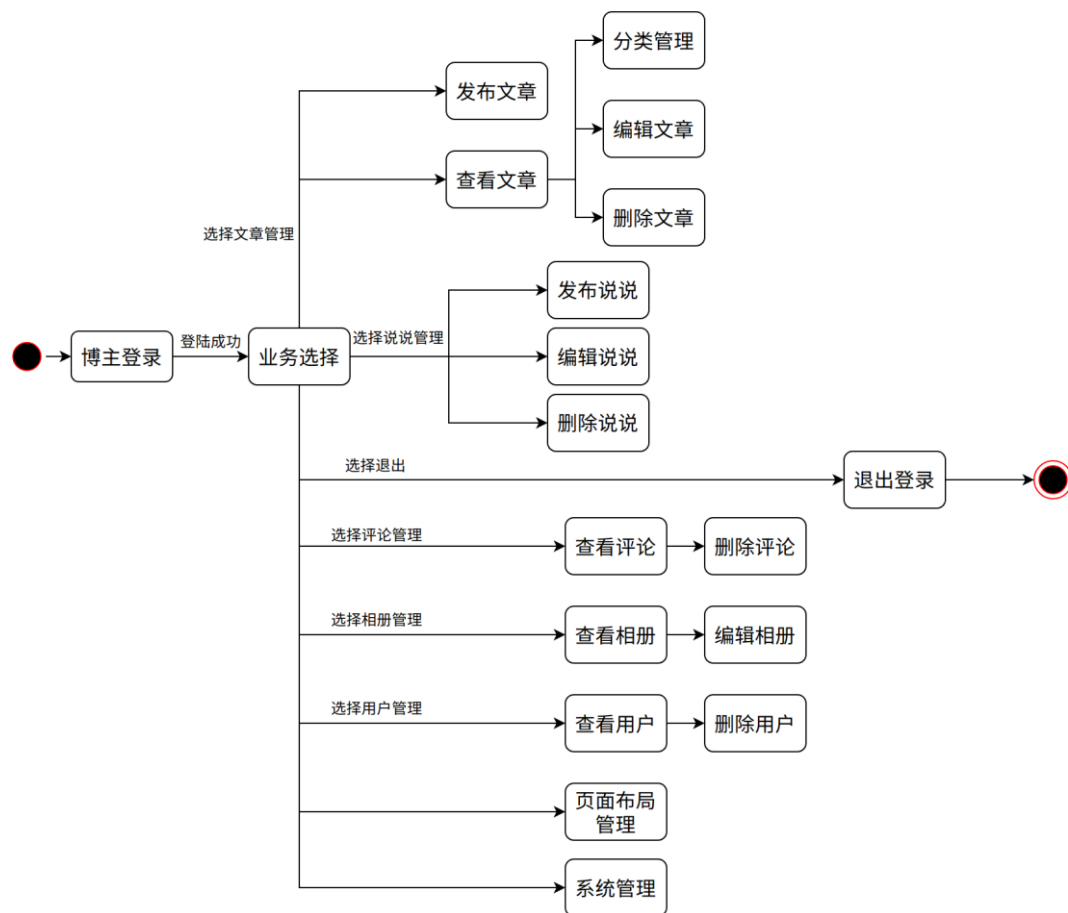


图 3 后台博主状态图

附录 C Petri 图



图 4 用户注册 Petri 网



图 5 用户登录 Petri 网

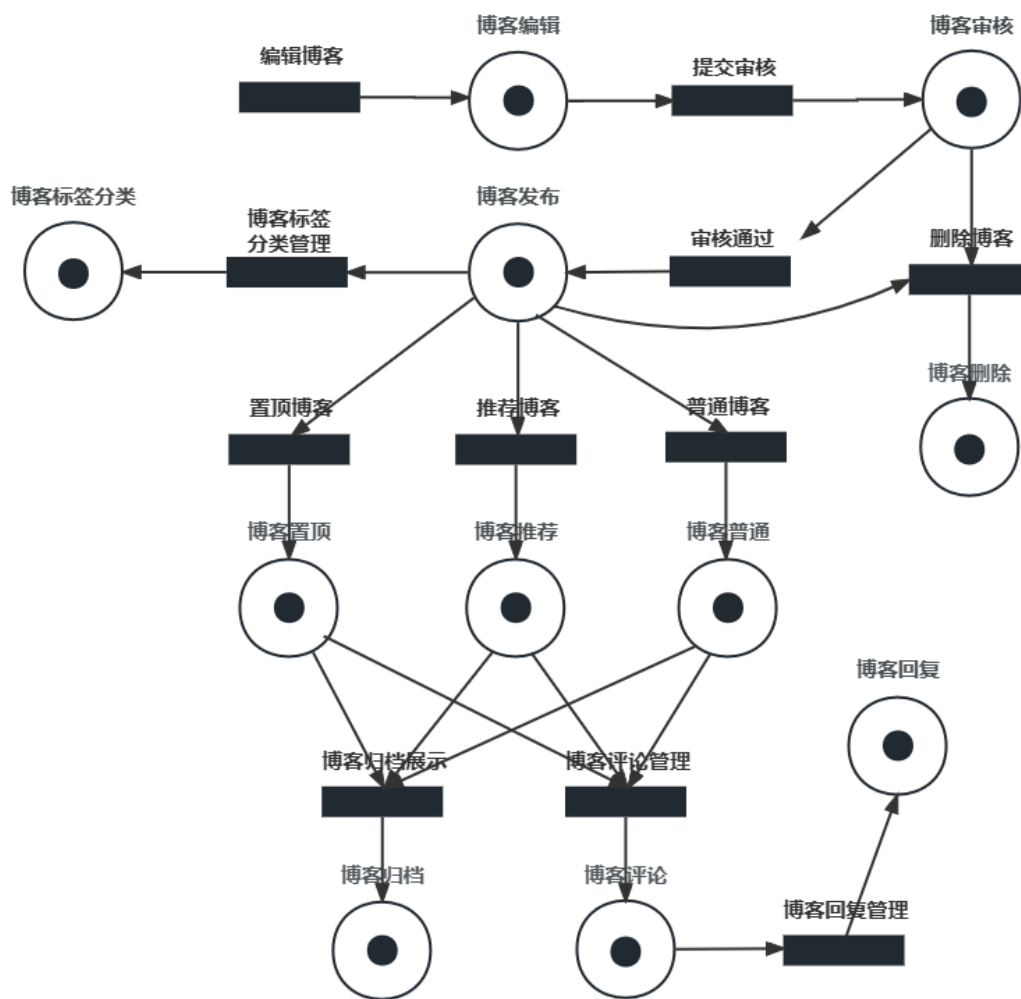


图 6 博客 Petri 网

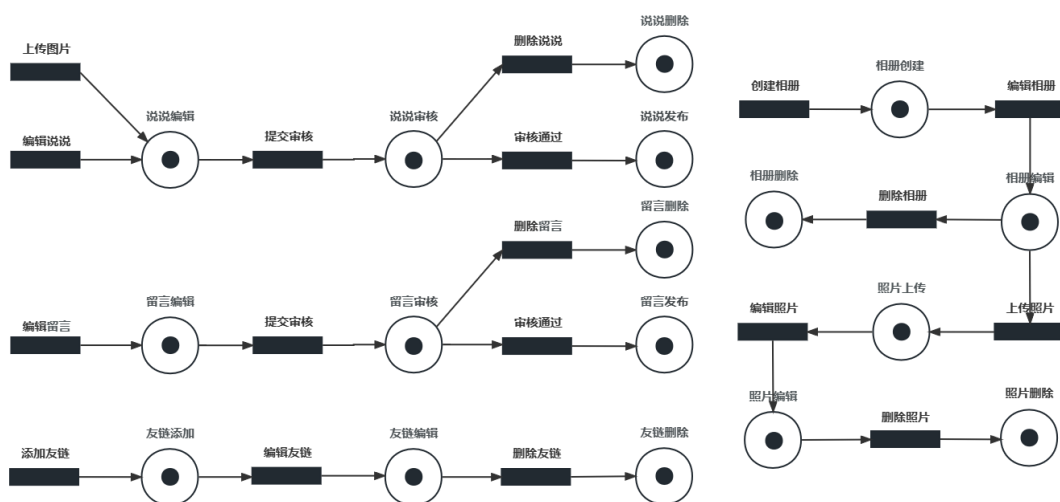


图 7 其他模块 Petri 网

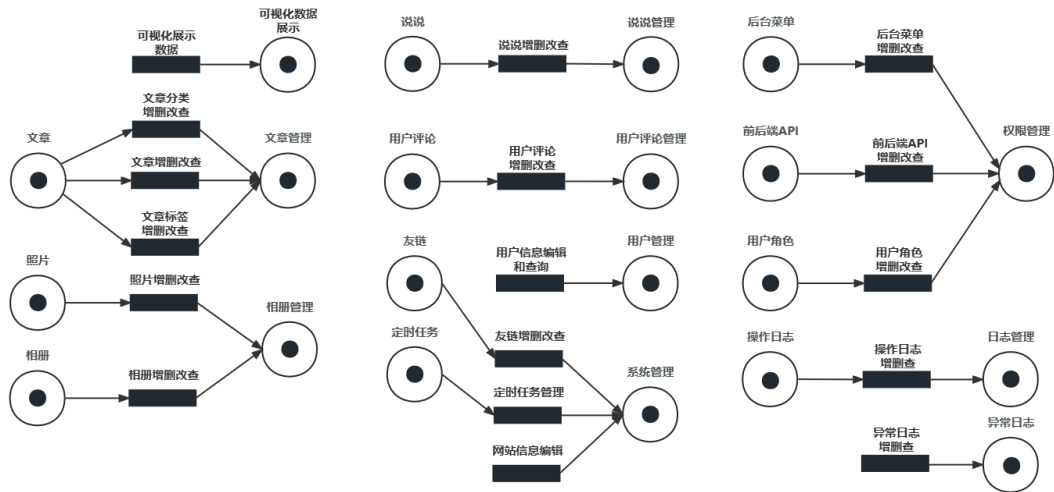


图 8 后台管理 Petri 网

附录 D 项目数据流图

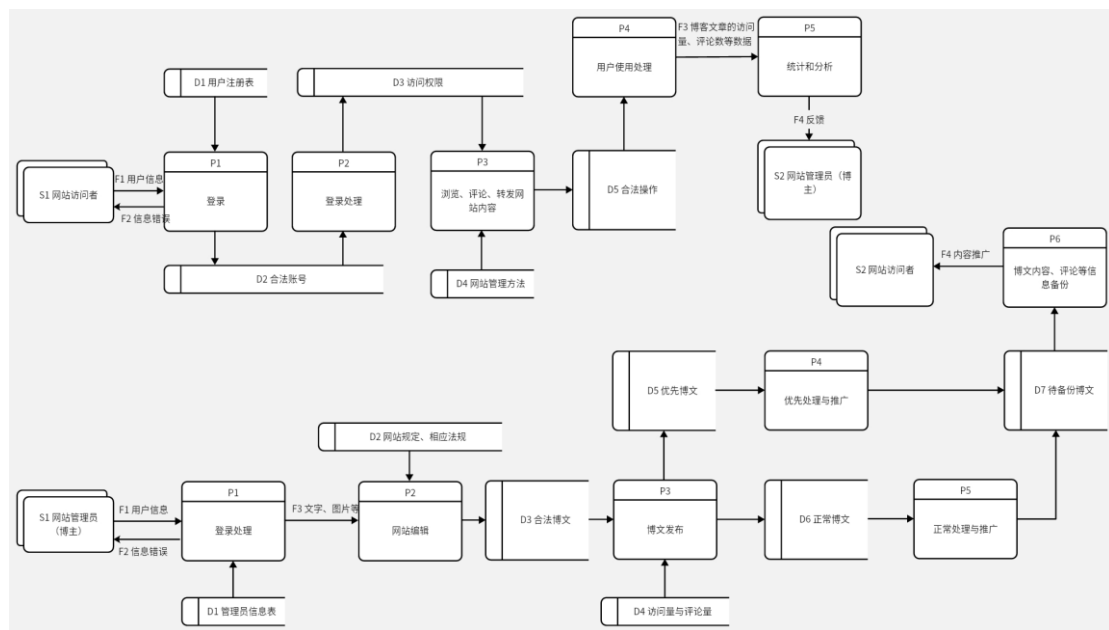


图 9 项目数据流图