**概**

**要**

**设**

**计**

**文**

**档**

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布  [ ] 正在修改 | 文件标识： | 仿小红书App |
| 当前版本： | 1.0 |
| 作 者： | 舒镐 |
| 完成日期： | 2023/11/27 |

目录

[1. 定义系统目标 3](#_Toc1526912930)

[(1) 目标 3](#_Toc396032860)

[(2) 范围 3](#_Toc1072455967)

[2. 划分系统模块 3](#_Toc937188098)

[(1) 用户管理 3](#_Toc1675295988)

[(2) 文章管理 3](#_Toc1041574499)

[(3) 问题管理 3](#_Toc1603397996)

[(4) 评论管理 3](#_Toc1685316216)

[(5) 推荐功能 3](#_Toc1947822029)

[(6) 搜索功能 3](#_Toc804126535)

[(7) 私信功能 3](#_Toc840083174)

[(8) 审核功能 3](#_Toc1720410040)

[(9) 反馈和举报功能 3](#_Toc1211719072)

[3. 设计系统架构 4](#_Toc775018603)

[(1) 架构风格 4](#_Toc1249341566)

[(2) 主要组件 4](#_Toc1736083043)

[4. 定义数据结构和数据流 4](#_Toc487391912)

[(1) 数据库设计 4](#_Toc1093235326)

[(2) 数据流向 4](#_Toc136040350)

[5. 设计系统界面 4](#_Toc1507562042)

[(1) 设计软件选型 4](#_Toc1583172588)

[(2) 界面设计原则 5](#_Toc1059300186)

[6. 考虑系统性能 5](#_Toc1018792472)

[(1) 响应时间 5](#_Toc957959373)

[(2) 吞吐量 5](#_Toc738280452)

[(3) 资源利用率 5](#_Toc119044398)

[7. 安全和可靠性设计 5](#_Toc1471921829)

[(1) 数据保护 5](#_Toc1726050210)

[(2) 身份验证 5](#_Toc1516775794)

[(3) 错误验证 6](#_Toc1819879868)

[8. 进行系统集成 6](#_Toc111357255)

[(1) 持续集成 6](#_Toc1123128248)

[(2) 持续部署 6](#_Toc35207006)

[9. 系统文档编写 6](#_Toc1166146917)

[(1) 架构设计 6](#_Toc1495471497)

[(2) UI设计 6](#_Toc240845591)

[(3) 接口设计 6](#_Toc2032656989)

[(4) 数据库设计 6](#_Toc696157647)

# 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。

# 划分系统模块

* 1. 用户管理
  2. 文章管理
  3. 问题管理
  4. 评论管理
  5. 推荐功能
  6. 搜索功能
  7. 私信功能
  8. 审核功能
  9. 反馈和举报功能

# 设计系统架构

* 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，服务端为阿里云服务器。

* 1. 主要组件

前端客户端

后端服务器

数据库

# 定义数据结构和数据流

* 1. 数据库设计

包括用户表、文章表、问题表、评论表等。

* 1. 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。

# 设计系统界面

* 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。

* 1. 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。

# 考虑系统性能

## 6.1 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。

## 6.2 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。

## 6.3 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。

# 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。

* 1. 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

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。

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。

# 系统文档编写

* 1. 架构设计
  2. UI设计
  3. 接口设计
  4. 数据库设计