**<基于Axum的轻量博客引擎>**

**系统测试文档**

**团队：菜鸟驿站 小组**

日期：2025年6月18号

目录

[**1.引言** 4](#_Toc201102034)

[**1.1. 编写**目的 4](#_Toc201102035)

[**1.2. 项目范围** 4](#_Toc201102036)

[**1.3. 术语定义** 5](#_Toc201102037)

[**1.4. 引用文件** 5](#_Toc201102038)

[**2. 总体描述** 5](#_Toc201102039)

[**2.1. 产品背景** 5](#_Toc201102040)

[**2.2. 产品描述** 6](#_Toc201102041)

[**2.2.1 系统接口** 6](#_Toc201102042)

[**2.2.2 用户界面** 7](#_Toc201102043)

[**2.2.3 硬件接口** 8](#_Toc201102044)

[**2.2.4 软件接口** 8](#_Toc201102045)

[**2.2.5 通信接口** 8](#_Toc201102046)

[**2.2.6 内存约束** 8](#_Toc201102047)

[**2.3. 产品功能** 8](#_Toc201102048)

[**2.5. 应用目标与作用范围** 9](#_Toc201102049)

[**2.6. 约束** 9](#_Toc201102050)

[**2.7. 假设与依赖关系** 9](#_Toc201102051)

[**3. 具体需求** 10](#_Toc201102052)

[**3.1外部接口** 10](#_Toc201102053)

[**3.2. 功能需求** 10](#_Toc201102054)

[**3.2.1 用例描述（Use Case）** 10](#_Toc201102055)

[**3.2.2 顺序图** 11](#_Toc201102056)

[**3.2.3 分析类图（逻辑建模）** 11](#_Toc201102057)

[**3.2.4 功能模块划分** 12](#_Toc201102058)

[**3.3. 性能需求** 12](#_Toc201102059)

[**3.3.1. 精度需求** 12](#_Toc201102060)

[**3.4 设计约束** 13](#_Toc201102061)

[**3.4.1 标准依从性** 13](#_Toc201102062)

[**3.5软件系统属性** 13](#_Toc201102063)

[**3.5.1 观感需求** 13](#_Toc201102064)

[**3.5.2 易用性需求** 13](#_Toc201102065)

[**3.5.3可执行性需求** 14](#_Toc201102066)

[**3.5.4安全性** 14](#_Toc201102067)

[**3.5.5 系统完整性** 14](#_Toc201102068)

[**3.5.6 可扩展性与可维护性** 14](#_Toc201102069)

**变更履历**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **版本** | **修订章节与内容** | **变更原因** | **修订人** | **审核人** | **修订日期** |
| **1** | **1.0** | **全文** | **建立初稿** | **杜嘉康** | **杜嘉康** | **2025-05-18** |
| **2** | **1.1** | **性能需求添加** | **内容修订** | **覃昶** | **杜嘉康** | **2025-06-18** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**说明：“变更原因”主要是分为：**

1. **建立初稿**
2. **内容修订**
3. **正式发布**

**1.引言**

**1.1. 编写**目的

本测试报告为“基于Axum的轻量博客引擎”项目的测试报告，目的在于总结测试阶段的测试以及分析测试结果，描述系统是否符合需求。预期参考人员包括用户、测试人员、开发人员、项目管理者、其他质量管理人员和需要阅读本报告的高层经理。

**1.2. 背景**

本系统为一个基于 Rust 编程语言框架开发的轻量级博客引擎，旨在满足技术博主、独立创作者及学习者对内容管理的核心需求。通过部署该系统，用户可以自由发布博客内容、管理分类与标签、浏览归档记录、进行评论互动等。

系统以高性能、高安全性、低资源消耗为目标，采用现代化 Web 框架 Axum 构建服务端 API，结合 SeaORM 实现数据库访问，所有功能模块围绕 RESTful 设计，方便后期前端集成或移动端接入。

该系统适合作为初中型博客平台的技术选型模板，也适合作为学习 Rust Web 开发的入门与实战项目。

**1.3. 术语定义**

|  |  |
| --- | --- |
| **术语** | **定义** |
| Rust | 一门注重性能和内存安全的系统编程语言，由 Mozilla 研究团队主导开发，适用于高并发、嵌入式、WebAssembly 和后端服务等场景。 |
| Axum | Rust 社区中用于构建 Web 应用的高性能异步框架，基于 Tokio 异步运行时和 Tower 服务抽象，设计简洁、类型安全、模块化强。 |
| SeaORM | Rust 中的异步 ORM 框架，支持自动实体生成、灵活查询构建和数据库迁移，适用于中大型项目的持久层开发。 |
| JWT | 一种开放标准（RFC 7519），用于以安全、紧凑的方式在客户端与服务器之间传输身份认证信息。 |
| SQLite | 嵌入式轻量级关系型数据库，支持 ACID 事务，适用于资源受限的场景或开发阶段的本地存储。 |

**1.4. 引用文件**

[1] Steve Klabnik, Carol Nichols. *The Rust Programming Language*（Rust官方权威书籍）  
[2] The Axum Book. https://docs.rs/axum  
[3] SeaQL. *SeaORM 官方文档*：https://www.sea-ql.org/SeaORM/  
[4] SQLite. *SQLite Documentation*：https://www.sqlite.org/docs.html

**2. 测试概要**

**2.1. 测试用例设计**

**2.2. 测试环境与配置**

|  |  |
| --- | --- |
| **项目** | **配置描述** |
| 操作系统 | Windows 10/11（64位） |
| 数据库 | 默认：SQLite3（单文件部署） 或 MySQL 8.0（生产环境推荐） |
| 后端语言环境 | Rust ≥ 1.70，支持 Cargo 包管理 |
| 依赖框架 | Axum（HTTP框架）、SeaORM（数据库ORM）、Serde（序列化）等 |

**2.3. 测试方法和工具**

系统采用前后端分离架构，后端基于 Rust 的 Axum 框架构建异步 Web 服务，前端可基于 React/Vue 等任意技术栈接入。系统架构分层如下：

**3. 测试结果及缺陷分析**

**3.1测试执行情况与记录**

测试组织、测试时间、测试版本

**3.2. 覆盖分析**

**3.3. 缺陷的统计与分析**

**4. 测试结论与建议**

**4.1测试结论**

**4.2 建议**