确保RustChatGPT项目在发布前经过充分的测试，达到预期的质量和稳定性标准

RustChatGPT测试计划

摘要

彭铭琨、詹晓芹

小组成员

**修订历史记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 日期 | AMD | 修订者 | 说明 |
| V1.0 | 20240703 | A | 彭铭琨 | 新增测试计划 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

（A-添加，M-修改，D-删除）

目录

[修订历史记录 2](#_Toc20426)

[（A-添加，M-修改，D-删除） 2](#_Toc9498)

[1. 测试策略 5](#_Toc927)

[测试目标 5](#_Toc12706)

[测试范围 5](#_Toc23570)

[测试类型 5](#_Toc19400)

[2. 测试用例 6](#_Toc19010)

[功能测试用例 6](#_Toc22003)

[性能测试用例 6](#_Toc3509)

[安全性测试用例 6](#_Toc17760)

[3. 测试环境 7](#_Toc22014)

[硬件环境 7](#_Toc26599)

[软件环境 7](#_Toc17938)

[网络环境 7](#_Toc26765)

[4. 预期的测试结果 7](#_Toc8214)

## 1. 测试策略

### 测试目标

确保RustChatGPT的所有功能按照需求规格说明书的要求正确实现，并且在不同的环境下都能稳定运行。

### 测试范围

测试范围包括：

* 主页、登录注册页面、聊天界面
* 用户管理功能
* 异步操作处理
* 数据持久化

### 测试类型

* **单元测试**：测试各个功能模块的单独组件。
* **集成测试**：测试模块间的交互是否正确。
* **系统测试**：测试整个系统的功能和性能。
* **回归测试**：在每次更新后重新测试以确保新修改未引入新的缺陷。
* **安全性测试**：验证应用程序是否存在安全漏洞。

## 测试用例

**功能测试用例**

|  |  |  |  |
| --- | --- | --- | --- |
| **测试用例编号** | **测试用例描述** | **测试步骤** | **预期结果** |
| TC001 | 测试主页加载 | 打开应用程序 | 应用程序成功加载主页 |
| TC002 | 测试用户注册 | 1. 点击注册按钮<br>2. 输入用户名和密码<br>3. 点击提交 | 用户成功注册并跳转到主页 |
| TC003 | 测试用户登录 | 1. 点击登录按钮<br>2. 输入用户名和密码<br>3. 点击提交 | 用户成功登录并跳转到主页 |
| TC004 | 测试聊天功能 | 1. 登录应用<br>2. 在聊天输入框中输入消息<br>3. 点击发送 | 系统返回相应的回复并显示在聊天窗口中 |

## **性能测试用例**

|  |  |  |  |
| --- | --- | --- | --- |
| **测试用例编号** | **测试用例描述** | **测试步骤** | **预期结果** |
| PT001 | 测试应用启动时间 | 启动应用程序 | 应用启动时间在5秒以内 |
| PT002 | 测试API响应时间 | 1. 登录应用<br>2. 输入消息并发送 | API响应时间在2秒以内 |

## **安全性测试用例**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| | **测试用例编号** | | --- | | **测试用例描述** | **测试步骤** | **预期结果** |
| ST001 | 测试SQL注入 | 1. 在登录页面输入SQL注入语句<br>2. 提交表单 | 系统应拒绝请求并显示错误信息 |
| ST002 | 测试XSS攻击 | 1. 在聊天输入框中输入XSS脚本<br>2. 点击发送 | 系统应拒绝请求并显示错误信息 |

## 

## 3. 测试环境

### 硬件环境

* CPU：Intel i5 或以上
* 内存：8GB 或以上
* 硬盘：500GB 或以上

### 软件环境

* 操作系统：Windows 10, macOS Catalina, Ubuntu 20.04
* 开发工具：Rust编译器，Cargo包管理器
* 测试工具：JUnit, Selenium

### 网络环境

* 互联网连接：稳定的网络连接，带宽不低于10Mbps

## 4. 预期的测试结果

1. 所有功能测试用例应通过，确保每个功能模块按照需求规格说明书的要求正确实现。
2. 性能测试结果应满足预期，启动时间和API响应时间应在可接受范围内。
3. 安全性测试应通过，确保系统不存在SQL注入和XSS攻击等安全漏洞。
4. 集成测试和系统测试应验证模块间交互的正确性和系统整体的稳定性。
5. 回归测试应确保每次更新后系统功能依然正常，无新的缺陷引入。