

# 需求分析报告

## 一、项目概述

本项目旨在开发一个去中心化的分布式局域网匿名聊天Web应用，用户可以在局域网内进行匿名聊天，并通过该应用享受联机小游戏和智能聊天机器人等额外功能。

## 二、项目目标

- 聊天室管理：** 提供用户创建、查看和进入聊天室的功能。
- 消息发送与接收：** 允许用户在聊天室中发送和接收文字、静态图片、静态表情、文件等格式的消息。
- 联机游戏：** 提供用户在聊天室内共同进行联机游戏的功能。
- 智能聊天机器人：** 在聊天室内提供智能聊天机器人功能，能够对用户的消息做出智能回复。
- 消息记录与存储：** 实现聊天室消息的记录功能，并管理消息存储空间，确保数据的持久性和隐私性。
- 用户界面与用户体验：** 提供友好的用户界面，确保良好的用户体验。
- 系统扩展与维护：** 构建一个可扩展、可维护的系统，方便后续添加新功能和处理需求变更。

## 三、功能需求

### 1. 用户管理

- 用户注册与登录：** 用户可以通过简单的注册流程创建账户并登录系统。
- 匿名登录：** 允许用户不通过实名注册，使用随机生成的匿名ID登录。
- 用户权限管理：** 实现管理员、普通用户的权限区分，管理员可以管理聊天室和用户。

### 2. 聊天室管理

- 创建聊天室：** 用户可以创建新的聊天室并作为其管理员，设置聊天室名称和主题。聊天室应当有带密码的加密聊天室和开放聊天室两种选择。
- 查看聊天室：** 用户可以浏览当前活跃的聊天室列表，查看聊天室的主题和参与人数。
- 加入聊天室：** 用户可以选择并加入聊天室，参与讨论。
- 退出聊天室：** 用户可以随时退出聊天室。

### 3. 消息发送与接收

- 文字消息：** 用户可以发送和接收纯文本消息。
- 静态图片和表情：** 用户可以发送和接收静态图片和表情。
- 动态图片和表情：** 用户可以发送和接收GIF等动态图片和表情。
- 音频和视频：** 用户可以发送和接收语音消息和短视频。
- 消息撤回：** 用户可以在一定时间撤回已发送的消息，但并不保证在任何用户处都成功撤回。

## 4. 联机游戏

- **游戏选择：** 提供多种小游戏供用户选择，如棋类游戏、扑克牌游戏、益智游戏等。
- **游戏邀请：** 用户可以邀请聊天室内的其他用户共同参与游戏。
- **游戏记录：** 保存用户的游戏记录和成绩。

## 5. 智能聊天机器人

- **基本对话：** 聊天机器人可以参与基本的文字对话。
- **信息查询：** 机器人可以回答常见问题或提供信息查询服务（如天气、新闻等）。
- **娱乐功能：** 机器人可以进行趣味对话或讲笑话等。

## 6. 消息记录与存储

- **消息存储：** 实现消息的持久化存储，用户可以查看历史消息记录。
- **消息搜索：** 提供搜索功能，用户可以通过关键字搜索聊天记录。
- **消息备份与恢复：** 定期备份消息数据，提供数据恢复功能。

## 7. 用户界面与用户体验

- **界面设计：** 提供简洁、直观、易于操作的用户界面。
- **主题切换：** 用户可以选择不同的主题模式（如暗黑模式、亮色模式）。
- **实时通知：** 实时推送新消息通知，确保用户不会错过重要消息。

## 8. 系统扩展与维护

- **模块化设计：** 采用模块化设计，方便后续功能扩展和维护。
- **插件支持：** 提供插件机制，允许第三方开发者扩展功能。
- **日志管理：** 实现系统日志记录和管理，便于问题排查和系统维护。
- **性能监控：** 监控系统性能，及时发现和解决性能瓶颈。

# 四、性能需求

---

## 1. 高可用性

- **在线用户支持：** 系统应能支持局域网内的用户同时使用，最低能稳定支持50名用户在线，且根据局域网条件可扩展至更大规模。
- **高并发处理：** 系统需具备高并发处理能力，确保在局域网内高峰期依旧可以快速响应用户请求。目标是支持每秒100次以上的消息发送与接收请求。

## 2. 低延迟

- **消息延迟：** 消息发送和接收的延迟应控制在50ms以内，确保用户可以实时进行交流。
- **页面加载时间：** 用户界面的页面加载时间应控制在1秒以内，优化用户体验。

### 3. 数据一致性

- 消息一致性：** 确保消息在分布式系统中的一致性，防止消息丢失和重复。
- 数据同步：** 在多设备登录情况下，用户数据和聊天记录应实时同步，保证数据的一致性。

### 4. 资源利用率

- 系统资源优化：** 优化服务器资源利用率，包括CPU、内存和存储空间，确保高效运行。
- 网络带宽管理：** 进行合理的网络带宽管理，防止网络拥塞，确保数据传输的稳定性。

## 五、可维护性需求

### 1. 模块化设计

- 功能模块化：** 系统各功能模块应独立开发，确保各模块可以单独维护和更新。例如，用户管理、聊天室管理、消息处理、游戏管理和聊天机器人等模块应独立实现。
- 接口标准化：** 各模块之间的接口应标准化，确保模块间通信的简便性和一致性。

### 2. 日志管理

- 详细日志记录：** 系统应记录详细的操作日志，包括用户操作日志和系统运行日志，便于问题排查和分析。
- 日志存储和检索：** 日志数据应持久化存储，并提供便捷的检索功能，支持按时间、事件类型等条件进行查询。

### 4. 性能监控与优化

- 性能监控：** 实时监控系统性能，包括CPU、内存、网络 and 磁盘使用情况，及时发现性能瓶颈。
- 性能优化：** 定期进行性能分析和优化，确保系统持续高效运行。

### 5. 文档与培训

- 开发文档：** 提供详细的开发文档，包括系统架构、模块设计、接口规范等，方便开发人员进行维护和扩展。
- 用户手册：** 提供用户手册，帮助用户了解和使用系统功能。