#### AI聊天室web架构设计

1. 架构总体概述

该 AI 聊天室 Web 架构采用分层设计，主要分为展示层、业务逻辑层和数据层。这种分层架构模式使系统各部分职责明确，降低了模块间的耦合度，便于开发、维护和扩展，是现代 Web 应用开发中广泛采用的架构方式。

1. 架构风格选择：分层架构设计
2. 展示层：处理用户界面和交互
3. 业务逻辑层：处理核心业务规则和流程
4. 数据访问层：负责数据持久化和检索
5. 详细架构设计
6. 展示层
7. 展示用户界面：构建并呈现注册登录页面、聊天窗口以及个人信息展示区域等界面给用户。
8. 处理用户交互：监听用户在界面上的操作，如点击按钮、输入文字、滚动页面等，并做出相应处理。
9. 调用 API 接口：与业务逻辑层进行通信，将用户操作请求和相关数据发送到业务逻辑层，并接收业务逻辑层返回的响应数据，以更新界面。

（二）业务逻辑层

用户功能模块：

1. 注册账号：对用户输入的注册信息进行合法性验证，包括用户名长度、格式，密码强度，邮箱或手机号格式等。验证通过后，将用户信息传递给数据层存储，同时可进行生成用户唯一标识、初始化默认设置等操作。
2. 登录账号：接收用户输入的登录凭证，与数据层存储的用户信息比对验证。验证通过则生成登录会话，记录登录状态，加载个性化界面配置和数据；验证失败返回错误提示给展示层。
3. 选择 AI Agent：从数据层获取可用的 AI Agent 列表信息并展示给用户，记录用户选择，在后续交互中将用户请求发送给所选 AI Agent。
4. 收藏 Agent：接收用户收藏 AI Agent 的操作请求，将相关信息存储到数据层，并在用户下次登录时快速展示收藏列表。
5. 查看聊天历史记录：根据用户请求从数据层检索相关聊天历史记录，可按时间顺序、对话主题等整理排序，进行预处理（如过滤敏感信息）后传递给展示层展示。
6. 与 Agent 聊天：接收展示层传递的用户输入内容，进行预处理（如去除多余空格、转换编码等）后发送给选定的 AI Agent，等待回复。对接收到的回复进行格式调整、内容审核等处理，再传递给展示层展示给用户，同时将聊天记录存储到数据层。

管理员功能模块：

1. 查看模型调用次数：定期从数据层统计 AI 模型的调用次数信息，可按时间（每日、每周、每月）、用户、AI Agent 等维度进行统计分析，将结果提供给管理员。
2. 查看用户调用次数：从数据层获取每个用户对 AI 服务的调用次数数据，分析用户使用行为和活跃度，为运营策略制定提供参考。
3. 编辑 Agent 列表：对系统中的 AI Agent 列表进行增删改操作。添加新 Agent 时将相关信息录入数据层；删除 Agent 时清理相关数据和处理关联关系；修改 Agent 信息时及时更新数据层记录并通知相关模块。

（三）数据层

1. 用户信息存储：数据库存储用户信息，包括用户名、加密密码等基本信息。
2. 聊天信息存储：存储聊天信息。每条聊天记录作为一个文档保存，包含聊天双方标识（用户和 AI Agent）、聊天时间、聊天内容等信息。
3. Agent 信息存储：存储 AI Agent 的详细信息，如名称、参数等，用于向用户展示可用 Agent 列表和为业务逻辑层调用提供配置信息。
4. 模型调用数据存储：记录每次 AI 模型被调用的相关数据，如调用用户标识、调用的 AI Agent 标识返回结果等，为分析 AI 模型使用情况、用户行为提供数据支持。
5. 架构设计图

