





设计目标

云译网旨在打造一个功能强大、用户体验良好的在线翻译平台

高质量翻译

借助 LLM实现自然流畅的翻译, 确保翻译结果的准确性和专业

性。

多样化服务

提供图片识别翻译功能,满足 不同用户在不同场景下的多样 化翻译需求。

个性化需求

支持用户自定义翻译风格和专业术语翻译,并收集用户反馈 不断优化模型。



辅助翻译学习

提供翻译质量检测功能,帮助翻译人员检验自己的学习成果, 促进翻译技能的提升。



翻译拓展功能

在翻译的基础上,通过与大模型的交互,使用户能够方便地深入了解翻译内容相关的知识,增强用户对翻译内容的理解。

模块设计

翻译处理模块

包括文本翻译子模块、图片识别翻译 子模块、翻译内容拓展子模块、个性 化翻译处理子模块、翻译质量检测子 模块。

系统支持模块

包括API 管理子模块、系统设置子模块、帮助与支持子模块、日志与分析子模块









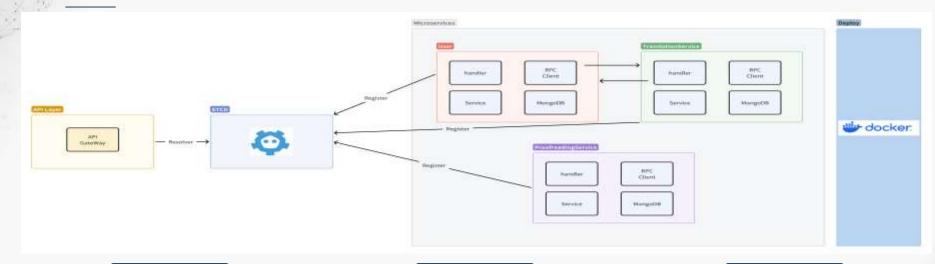
用户交互模块

包括注册登录子模块、个性化设置子 模块、在线翻译操作界面子模块、翻 译结果展示与交互子模块。

数据管理模块

包括用户数据管理子模块、 翻译数据 管理子模块、个性化设置数据管理子 模块、数据统计与分析子模块。

后端架构



API层

● 接受并处理用户请求,然后返回 相应的相应,通过ETCD进行配 置管理。

微服务层

- translationService负责翻译相 关的请求,ProofreadingService 负责文本的校对和修正工作。
- 该微服务通过RPC进行通信,实现服务之间的解耦和独立部署,提高系统的灵活性和可扩展性。
- 使用MongoDB作为数据存储, 确保数据的持久化和高效访问。

部署层

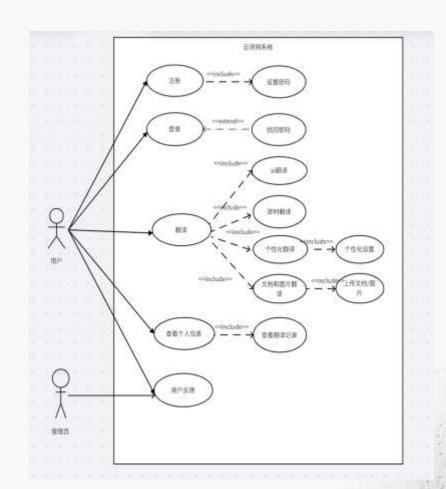
- 采用Docker容器化技术,将各个 微服务器打包成容器进行部署。
- 容器化技术使系统更加灵活,便 于系统的拓展和维护。



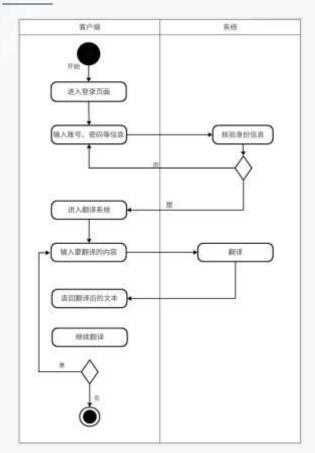
需求分析与功能设计

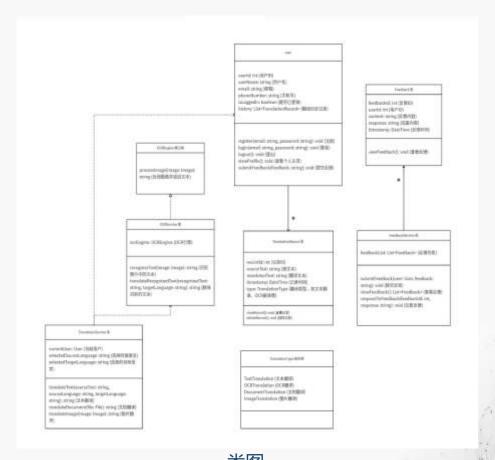
功能设计

- 注册和登录
- 支持文本翻译和图片文字提 取翻译,提供即时、准确、 个性化的大模型翻译
- 翻译学习、拓展功能
- 查看个人信息、翻译记录
- 收集用户反馈,不断迭代优化化



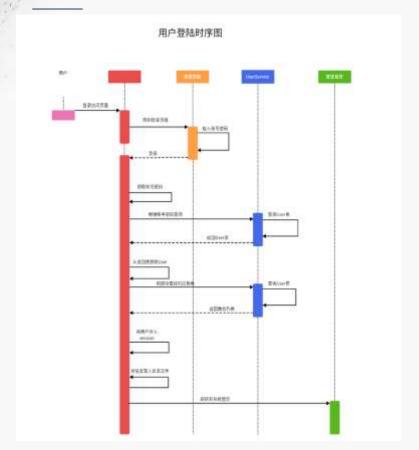
活动图与类图

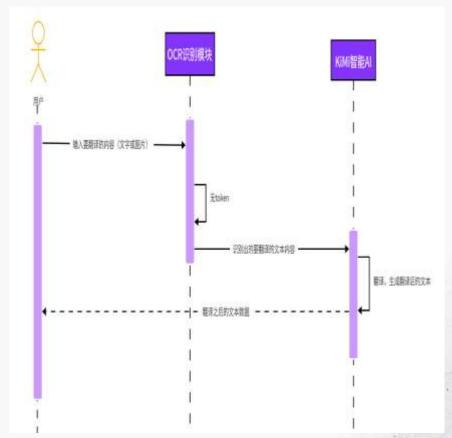






时序图

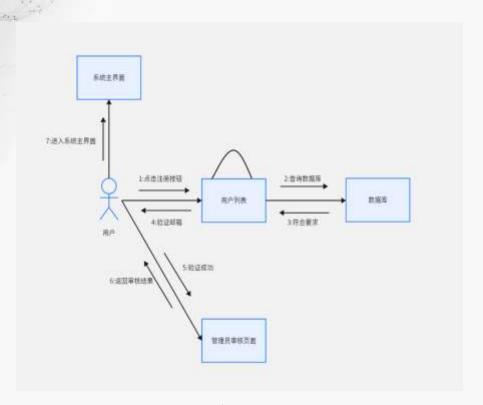


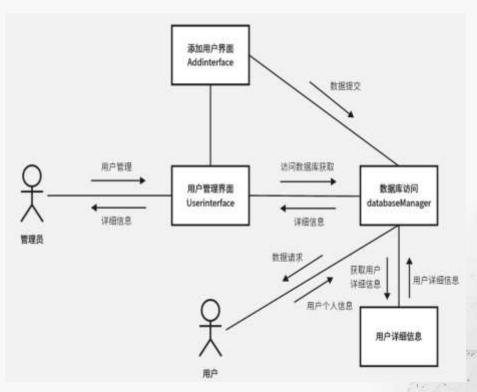


登录模块

翻译模块

协作图



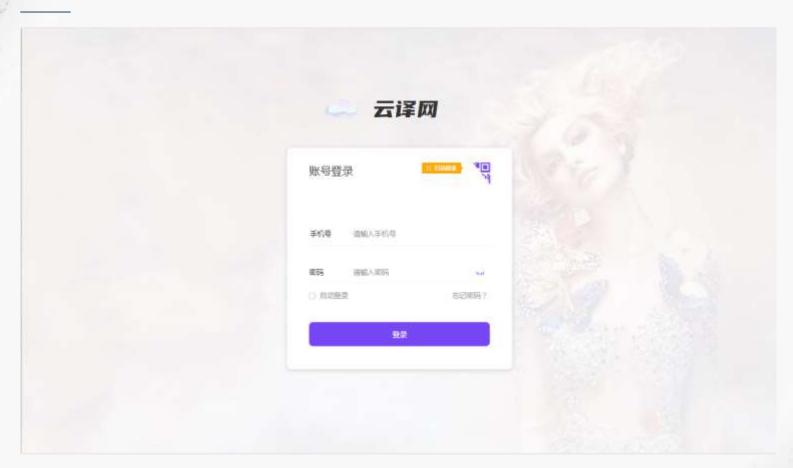


用户注册

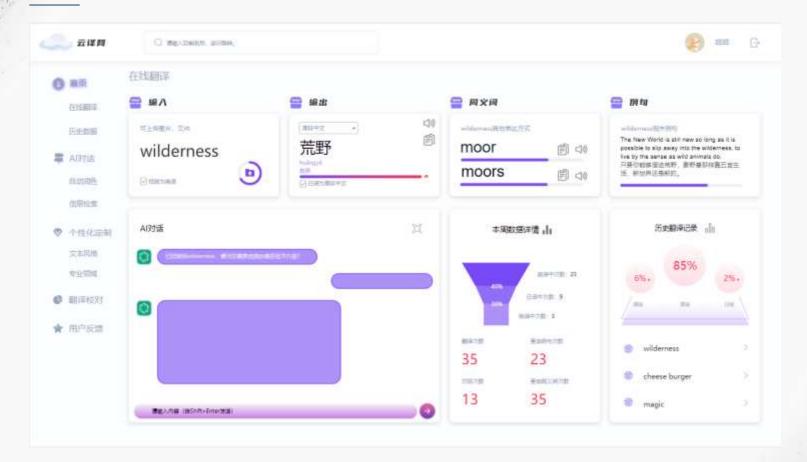
用户管理



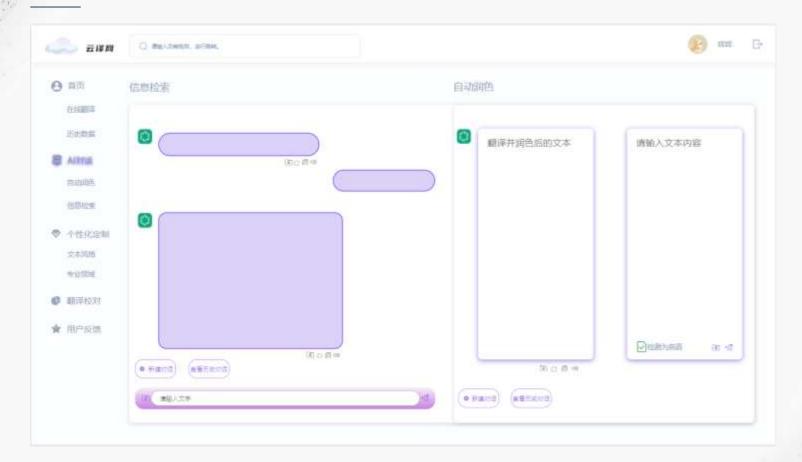
登录



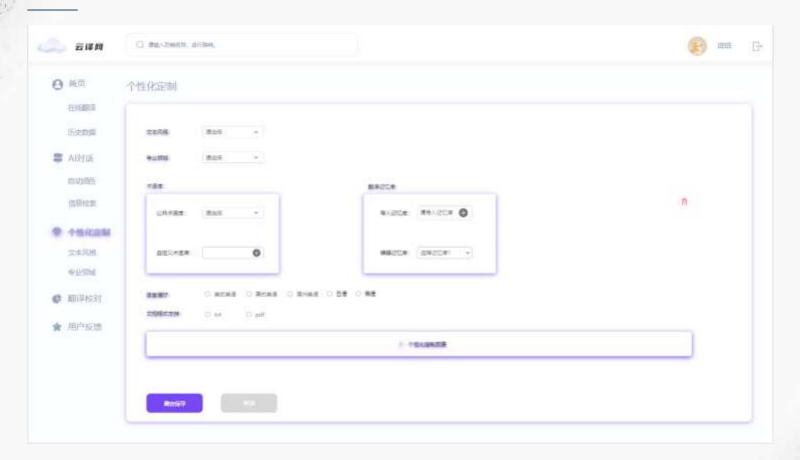
首页



AI对话



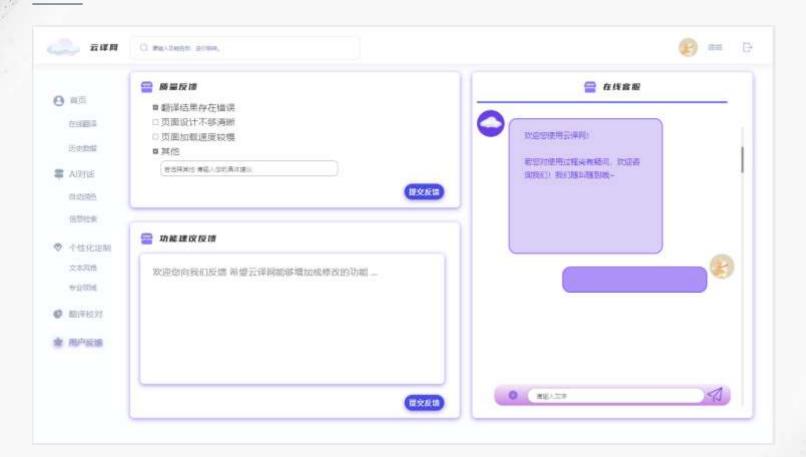
个性化定制



翻译校对



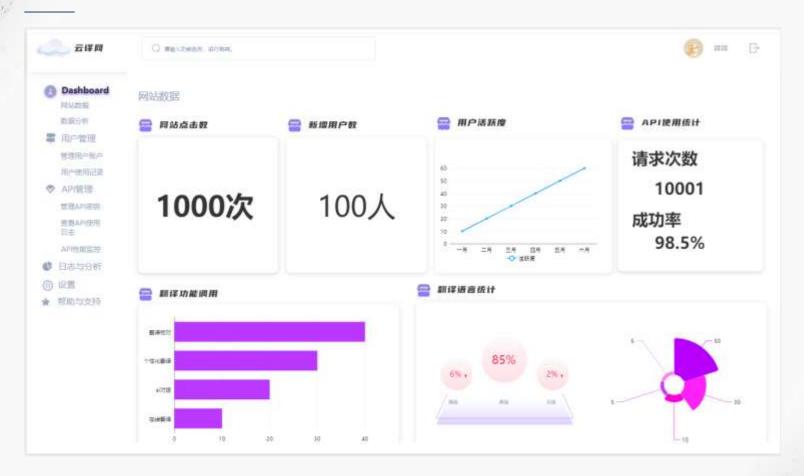
用户反馈



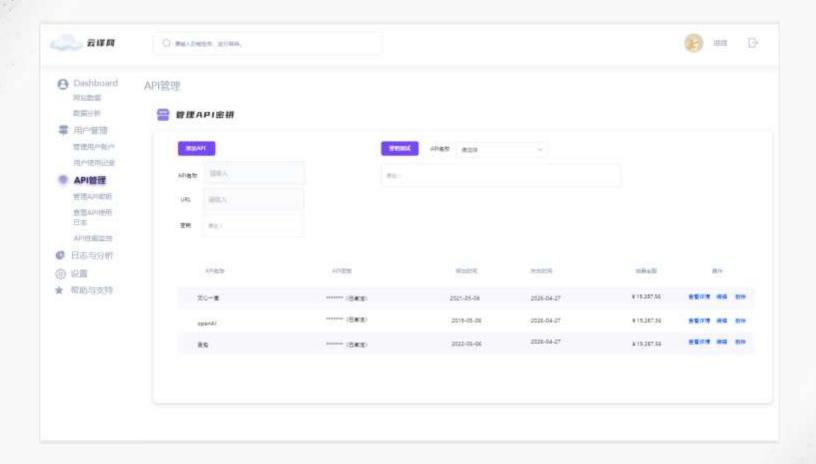
个人主页



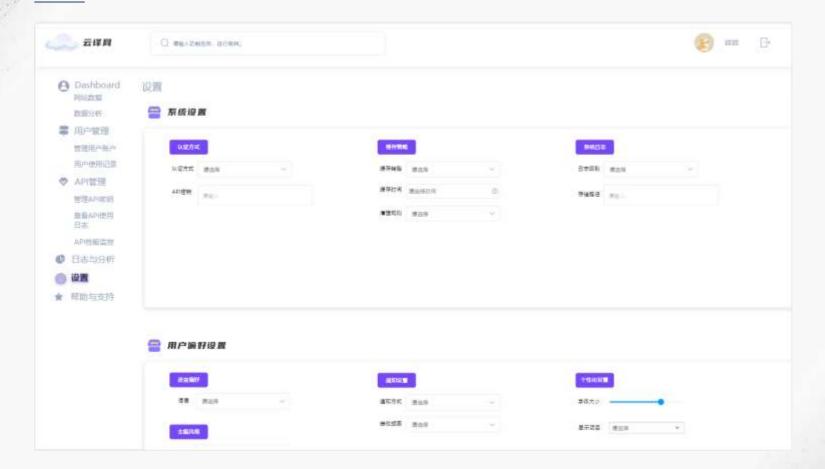
后台管理系统: 仪表盘



后台管理系统: API管理界面



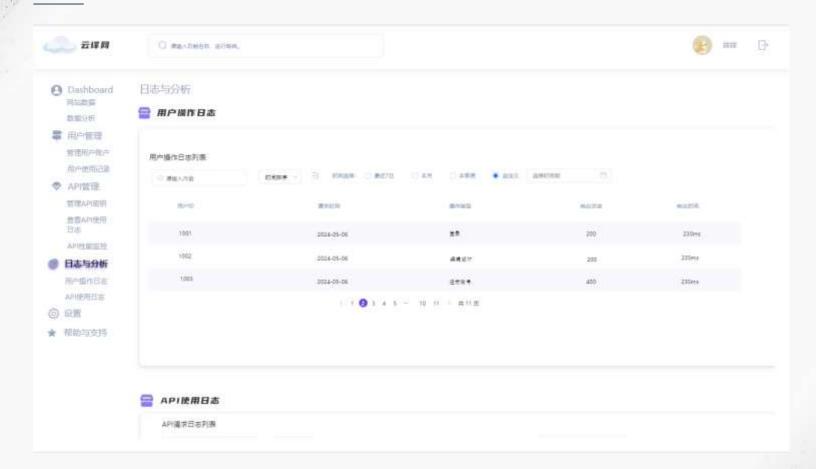
后台管理系统: 设置



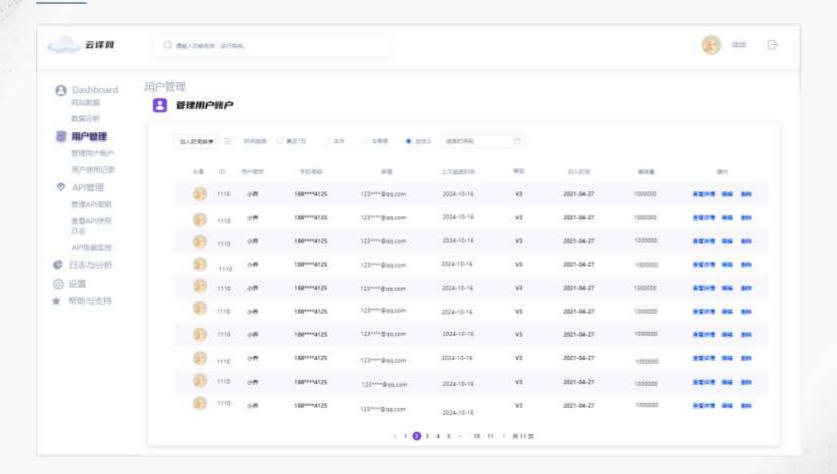
后台管理系统:帮助与支持



后台管理系统: 日志与分析



后台管理系统: 用户管理





数据库设计

数据库选择

主要使用 MongoDB, 相 比于传统的 Mysql 数据库, MongoDB 有着更强大的 灵活性, 其JSON 文档结构 能更自然地表示嵌套对象和 数组,适合存储个性化翻译 内容和用户反馈等复杂数据。 用户模型、翻译记录等数据 结构可能随业务变化而调整, MongoDB 的灵活性也使 数据结构的改变更便捷。

数据库设计

● 分为用户表 (userInfo)、 性翻译表 (termLibrary)、用户反 馈表 (feedback)、翻译 和对话记录表 (Translation Conversat ion Records)、登录记录 表 (loginRecords)、翻 译 校 对 记 录 表 (Translation Proofread Records) 。

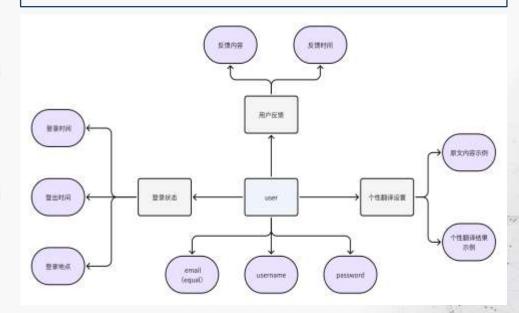
数据结构设计: 用户

用户(userInfo+termLibrary+feedback+loginRecords)

```
"_id": "ObjectId",
                                        // 用户在 MangaDB 中的唯一标识符
     "username": "String",
                                        // 用户名。该加唯一索引
     "password_hash": "String",
                                        // 密码砂带值
     "email": "String",
                                        // 用户邮箱、版加维一索引
     "created at": "Date",
     "updated_at": "Date",
                                        // 銀后費架的形
     "term library": [
                                        77 用户的个性研译记录。旅客数组
        "original_content": "String",
                                        7/ 原文内容
        "translation": "String"
                                        // 个性翻译内容
     "feedbacks":
                                        // 用户反馈被套数钳
        "feedback text": "String",
                                        77. 反部内容
        "created at": "Date"
                                        II Nothering
     "login_records": [
                                        // 原证记录能套数组
21
        "login_time": "Date",
22
                                        // 便证的间
        "login_location": "location"
                                        77 使灵地点
        "logout time": "Date"
                                        71 使出时间
27 3
```

设计介绍

- 通 过 将 userInfo+termLibrary+feedback +loginRecords嵌套起来,避免跨集合(collection) 查询。
- 考虑为 email 或者 username 添加唯一索引,以防 止重复注册。



数据结构设计: 翻译和对话记录

设计介绍

- 将翻译内容和对话内容进行结合,有利于大模型根据上下文来提供体验更好的翻译结果。
- 将多轮对话 dialog 设计为嵌套数组,方便保存多轮对话信息。
- 为 user_id 和 created_at 建立复合索引,优化查询效率。

翻译和对话记录(Translation_Conversation_Records)

```
" id": "ObjectId",
                                           // 關釋记录的唯一标识符
    "user_id": "ObjectId",
                                           // 关联用户ID, 引用 Users 集合
     "original content": "String",
                                           // 原文内容
     "translated content": "String",
                                           // 译文内容
     "dialogs": [
                                            // 多轮对话旅舞数组
        "text": "String".
                                           11 X115D199
        "timestamp": "Date"
                                           // Briging
11
     "original_language": "String",
12
                                           // 原语声
     "target_language": "String",
                                              White Elbride
     "created_at": "Date"
                                               19/18/11/11
15 }
```

数据结构设计: 翻译校对记录

设计介绍

- 通过使用嵌套数组checks来存储校对结果,允许对同一条翻译记录进行多种 类型的校对检查,提高了数据结构的灵活性。
- 关联用户且记录校对时间和各项检查得分,为用户提供了可操作的历史记录, 帮助用户追踪和改进他们的翻译质量。

翻译校对记录集合(Translation_Proofread_Records)

```
"_id": "ObjectId",
                                        // 校对记录的唯一标识符
    "user id": "ObjectId",
                                        // 关联用户ID, 引用 Users 集合
3
    "original content": "String",
                                        // 原文内容
    "translated_content": "String",
                                          译文内容
   "proofread_time": "Date",
                                        // 校对时间
    "checks": [
                                        // 校对内容的嵌套数组
8
        "type": "String",
                                        // 检查类型, 如 "grammar" 或 "content"
        "result": "String"
                                        // 检查结果, 得分
10
11
12
13
```

