

# 武汉大学计算机学院 本科生实验报告

## 软件需求规格说明书

专 业 名 称：软件工程

课 程 名 称：软件工程

指 导 教 师：何璐璐

团 队 成 员 一：王皓霖

团 队 成 员 二：莫韬

团 队 成 员 三：周锦耀

团 队 成 员 四：余嵘

团 队 成 员 五：张开宇

团 队 成 员 六：蔡昀燃

## 郑 重 声 明

本团队呈交的实验报告，是在指导老师的指导下，独立进行实验工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本实验报告不包含他人享有著作权的内容。对本实验报告做出贡献的其他个人和集体，均已在文中以明确的方式标明。本实验报告的知识产权归属于培养单位。

团队成员签名：王皓霖、莫韬、周锦耀、余嵘、张开宇、蔡昀燃

日期： 2025.4.6

# 目 录

<b>1 引言</b>	<b>5</b>
1.1 编写目的	5
1.2 读者对象	5
1.3 软件项目概述	5
1.4 文档概述	5
1.5 参考资料	6
<b>2 软件的一般性描述</b>	<b>6</b>
2.1 软件产品与环境	6
2.2 假设与前提条件	6
<b>3 软件功能需求描述</b>	<b>7</b>
3.1 软件需求的用例模型	7
3.2 软件需求的分析模型	8
3.2.1 “诗词搜索”的用例描述	8
3.2.2 “诗词推荐”的用例描述	9
3.2.3 “诗词测验”的用例描述	11
3.2.4 “诗词创作”的用例描述	11
3.2.5 “论坛功能”的用例描述	12
3.2.6 “个人空间管理”的用例述	13
3.2.7 “飞花令游戏”的用例描述	14
<b>4 其它软件需求描述</b>	<b>24</b>
4.1 性能要求	24
4.2 设计约束	25
4.3 界面要求	25
4.4 进度要求	25
4.5 交付要求	26
4.6 验收要求	26

# 1 引言

## 1.1 编写目的

本需求规格说明书旨在明确《云舟词渡》诗词学习创作平台的系统边界、功能架构和技术实现路径。通过规范文档约束诗词资源整合标准、AI 创作辅助机制和社区交互规则，确保开发团队在云原生架构、智能算法集成与游戏化运营等维度达成技术共识。重点解决传统诗词工具检索效率低、创作门槛高、学习场景单一等痛点，为后续系统设计、测试验收及商业运营提供基线标准。

## 1.2 读者对象

核心开发团队：前端/后端工程师、AI 算法工程师、交互设计师  
项目管理团队：产品经理、技术负责人、运维主管  
关联合作方：高校诗词社团、文旅项目运营人员  
终端用户代表：00 后诗词爱好者、语文教育工作者  
质量保障团队：测试工程师、安全审计人员

## 1.3 软件项目概述

项目名称：云舟词渡——智能诗词学习创作平台 用户单位：面向诗词爱好者、教育机构及文旅产业的三类用户群体，具备：

个人用户：提供结构化诗词库（覆盖朝代/诗人/主题三维检索）

教育机构：搭载课堂辅助模块（测验/朗诵/接龙创作）

文旅单位：定制 AR 诗词场景生成器

核心价值体系：

技术融合：采用 LSTM+Attention 构建诗人风格迁移模型，实现"李白-杜甫"式 AI 对诗

场景创新：开发双屏创作工作台（左屏格律检测+右屏 AI 改写建议）

生态闭环：构建"学习-创作-竞赛-变现"循环体系（含墨豆打赏、会员订阅等）

差异化优势：

首创 OT 算法支持的多人协同接龙创作

集成 WebRTC 语音识别技术的朗诵评分系统

三维可视化诗词地图（LBS+历史事件关联）

## 1.4 文档概述

- 1) 系统背景与行业分析：解析诗词数字化现状及 00 后用户行为特征
- 2) 功能架构蓝图：划分学习/创作/社区/商业四大子系统，明确 23 个核心功能点
- 3) 技术实现方案：详述 React+SpringBoot 技术栈选型依据，图解 RESTful API 交互流程
- 4) 质量保障体系：包含压力测试指标（支持万级并发创作请求）、安全防护方案（JWT 鉴权+防 SQL 注入）
- 5) 商业运营规划：文旅分成模式、赛事增值服务、校园会员体系的财务模型
- 6) 风险应对策略：制定诗人风格迁移模型训练失败时的 Jieba 分词替代方案

## 1.5 参考资料

[1] 彭蓉.软件工程.清华大学出版社, 2024

# 2. 软件一般性描述

## 2.1 软件产品与环境

### 1. 产品定位与技术架构

《云舟词渡》是融合 AI 技术与传统文化的智能化诗词平台, 主要面向诗词爱好者提供学习、创作与互动服务。系统采用分层架构设计:

- o 前端层: 基于 Vue3 构建跨终端响应式界面, 集成可视化数据看板
- o 服务层: Spring Cloud 微服务体系实现模块化开发, 包含用户服务、AI 引擎服务、内容管理三大核心模块
- o 数据层: 采用 MySQL 集群存储结构化数据, Elasticsearch 支持全文检索, Redis 缓存热点内容

### 2. 运行环境与部署

环境类型技术规格功能定位开发环境 Docker 容器组 (Node.js+JDK+MySQL) 功能验证与单元测试测试环境阿里云 ECS (4 核 8G/CentOS 7.9) 压力测试与安全评估生产环境高可用集群

(3 节点负载均衡) 支持 10 万级并发访问, 核心组件包括:

- o AI 生成引擎: 基于 Transformer 架构训练的古诗词生成模型
- o 安全体系: RBAC 权限管理结合实时内容过滤
- o 运维监控: Prometheus+Grafana 实现全链路监控

## 2.2 假设与前提条件

### 1. 基础保障条件

需求变更率控制在迭代周期的 15%以内

AI 生成结果格律合规率 $\geq 85\%$ , 服务响应延迟 $< 500\text{ms}$

持续获得中华书局等机构的古籍数据授权

### 2. 依赖与约束

类别内容应对措施第三方服务阿里云 OCR 图像识别接口建立腾讯云备用通道合规要求符合

《网络内容生态治理规定》集成自动化审核系统人力资源需维持 5 人以上开发团队签订外包服务协议

### 3. 风险管理

需求变更: 建立 CCB 变更控制委员会, 设置需求缓冲池

技术风险: 预留 15%研发预算用于算法优化与性能调优

运维风险: 设计自动伸缩架构, 支持实时资源扩容

# 3 软件功能需求描述

## 3.1 软件需求的用例模型

软件用例模型图如下图所示。

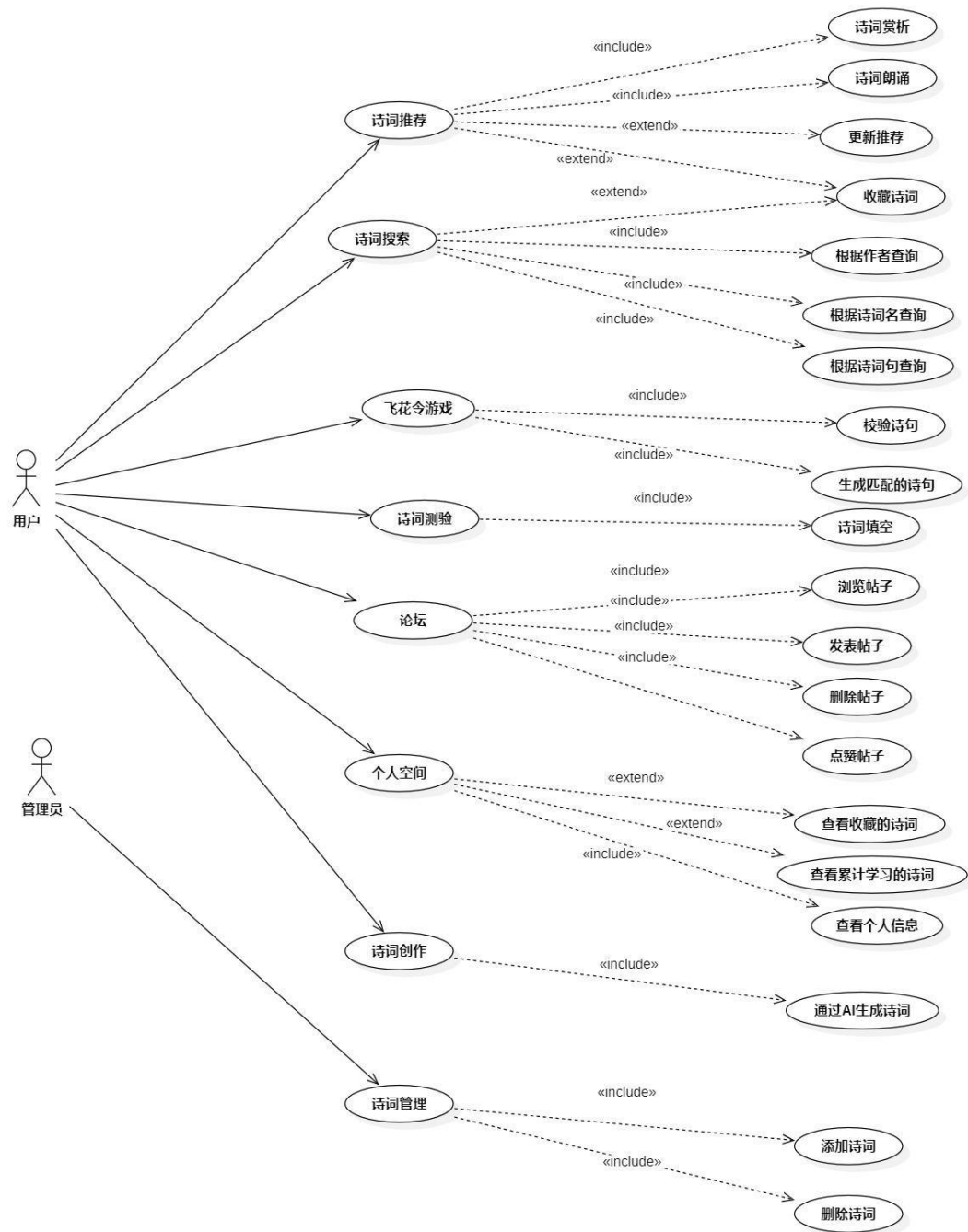
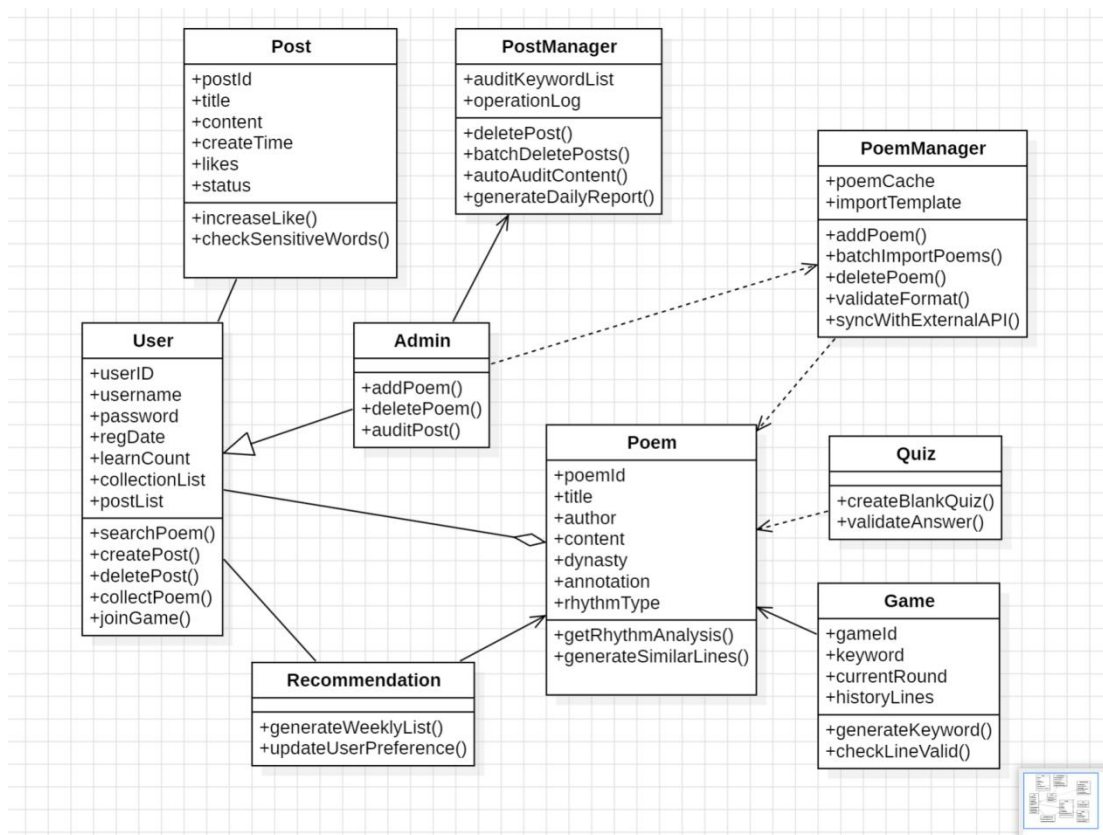


图 3.1.1 软件需求用例模型

类图展示。



### 1. 用户 (User) 类

作为平台的基础实体，User类通过userId唯一标识用户身份，username和password字段实现账户体系的安全登录，其中password采用BCrypt加密存储以防止泄露。regDate记录注册时间用于计算用户活跃周期，learnCount动态统计学习过的诗词总数，为学习报告生成提供数据支撑。collectionList以链表结构存储用户通过collectPoem()方法收藏的诗词对象，支持按朝代或作者分类浏览；postList则通过createPost(title, content)方法关联用户发布的社区帖子，实现内容溯源。核心方法searchPoem(keyword)整合全文检索技术，支持作者名（如“杜甫”）、标题（《春望》）或诗句片段（“国破山河在”）的模糊匹配，返回结果按相关性排序。joinGame(keyword)方法调用飞花令引擎，实时验证用户输入诗句是否符合关键字和格律要求。

### 2. 管理员 (Admin) 类

继承自User的管理员类，在保留基础功能的同时扩展了高阶权限操作。addPoem(title, author, content)方法通过JPA持久化框架将新诗词写入MySQL数据库，自动校验作者是否存在于Author表以避免数据孤岛；deletePoem(poemId)方法采用软删除机制，仅标记isDeleted状态而保留历史数据。auditPost(postId)方法结合本地敏感词库（如“暴力”“欺诈”等800+关键词）和第三方内容安全API，对帖子实施双重审核，违规内容将触发Post对象的状态status状态变更为“blocked”。管理员独有的batchImportPoems(csvPath)方法可解析CSV文件，批量导入诗词时自动跳过格式不规范的记录，并在日志中生成错误报告。

### 3. 诗词 (Poem) 类

诗词实体以poemId为主键，title存储标准标题（如《水调歌头·明月几时有》），author字段通过外键关联Author表的详细信息。content字段保存全文内容并标注换行符以保持原诗格式，dynasty采用枚举值限定唐宋元明清等朝代归属，annotation存储历代名家注释（如《唐诗三百首》版本批注）。核心方法getRhythmAnalysis()

基于声调数据库分析平仄规则，对用户输入诗句生成类似“第二字应为仄声”的纠错提示；generateSimilarLines(keyword)方法通过TF-IDF算法计算语义相似度，例如输入“春风”可推荐“春风得意马蹄疾”等关联诗句。属性rhythmType标记诗体类型（如“七律”“词牌名”），控制格律校验的规则集选择。

#### 4. 帖子（Post）类

帖子对象通过postId唯一标识，title限制50字符以内以避免过长标题，content支持Markdown格式渲染诗词引用块（如> 床前明月光）。createTime记录发帖时间用于社区按时间线排序，likes采用Redis缓存实现高并发点赞计数，status状态机管理“待审核/已发布/已删除”生命周期。方法increaseLike()原子性地更新点赞数，同时触发Kafka消息通知发帖用户；checkSensitiveWords()方法使用AC自动机算法扫描文本，识别“兼职”“代购”等广告关键词时返回匹配位置列表。删除操作通过status变更而非物理删除，保留数据审计轨迹。

#### 5. 测验（Quiz）类

作为控制类，Quiz通过createBlankQuiz(poemId)从指定诗词中随机挖空（如隐藏《静夜思》的“明”字），并生成干扰选项（“月/星/灯”）。validateAnswer(userInput)方法采用Levenshtein距离算法实现模糊匹配，允许用户答案存在1-2字符误差（如“疑是地上霜”与“疑是地土霜”均判对）。属性difficultyLevel控制题目难度（初级仅需填词，高级需补全整句），maxAttempts限制每道题最多尝试3次以防止暴力破解。测验结果存储于QuizResult关联表，用于生成用户能力分析报告。

#### 6. 推荐（Recommendation）类

推荐引擎通过generateWeeklyList(userId)分析用户行为日志，使用协同过滤算法计算相似用户群组，例如常读李白作品的用户会收到《行路难》《蜀道难》等推荐。属性preferenceWeights存储用户对不同朝代（唐诗权重0.7，宋词0.3）和题材（边塞诗权重0.6）的偏好值。updateUserPreference()方法每晚定时任务分析当日浏览记录，例如用户连续查看3首杜甫作品，则提升“现实主义”标签的权重值。冷启动阶段调用getHotPoems(topN)从访问量最高的1000首诗词中随机选取。

#### 7. 飞花令（Game）类

游戏实例由gameId唯一标识，keyword从预设词库（含“花”“月”“山”等200+高频字）随机选取，currentRound记录当前轮次（最多10轮）。方法generateKeyword()采用加权随机算法，避免连续出现生僻字；checkLineValid(line)调用诗词库的全文索引，验证用户输入是否真实存在且包含关键字。属性timeLimit设定每轮30秒倒计时，超时自动结束回合；historyLines存储已接龙的诗句，防止重复使用。游戏结果通过WebSocket实时推送至对战双方界面。

#### 8. 诗词管理（PoemManager）类

控制类PoemManager的诗emCache属性采用LRU策略缓存最近访问的1000首诗词，降低数据库查询压力。方法addPoem()执行多级校验：标题长度≤50字符、作者存在于权威名录、正文至少包含4句完整诗行。batchImportPoems()方法解析CSV时自动转换繁体字为简体，并修复常见的OCR识别错误（如“同”→“同”）。syncWithExternalAPI()每小时访问中华诗词网API，通过Jaccard相似度算法去重后增量更新本地库。删除操作前调用checkReference()确认该诗未被用户收藏，否则



转为逻辑删除。

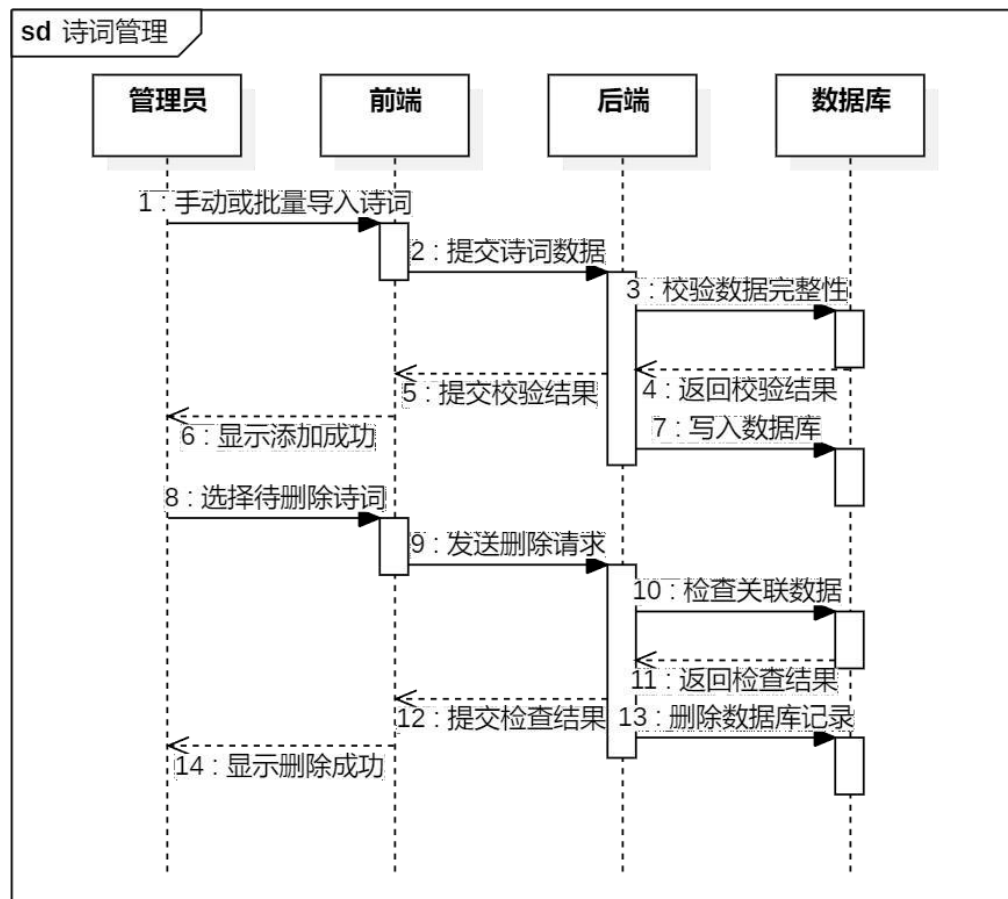
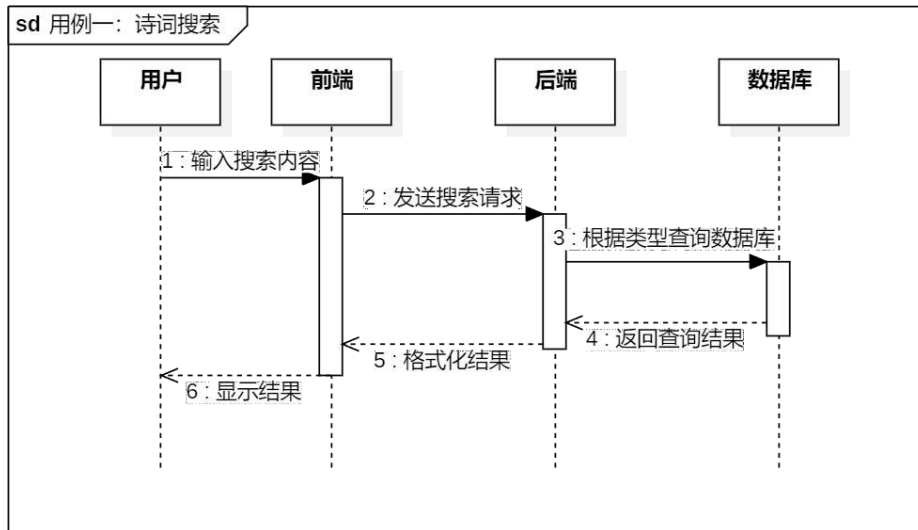
#### 9. 帖子管理 (PostManager) 类

PostManager的auditKeywordList动态加载最新敏感词库，每日凌晨通过HTTP接口从安全中心更新。方法deletePost(postId)执行物理删除前备份至HDFS集群，保留周期为30天。autoAuditContent()方法集成BERT模型识别语义违规（如“代写诗词”的隐晦表述），准确率较传统规则引擎提升40%。generateDailyReport()生成PDF格式的审核日报，包含违规类型分布图和TOP10高频敏感词统计，通过企业微信机器人推送至管理群组。批量删除时采用分页查询（每页500条）避免内存溢出。

## 3.2 软件需求的分析模型

本部分描述系统的 7 个用例的设计。

### 3.2.1 “诗词搜索”的用例描述



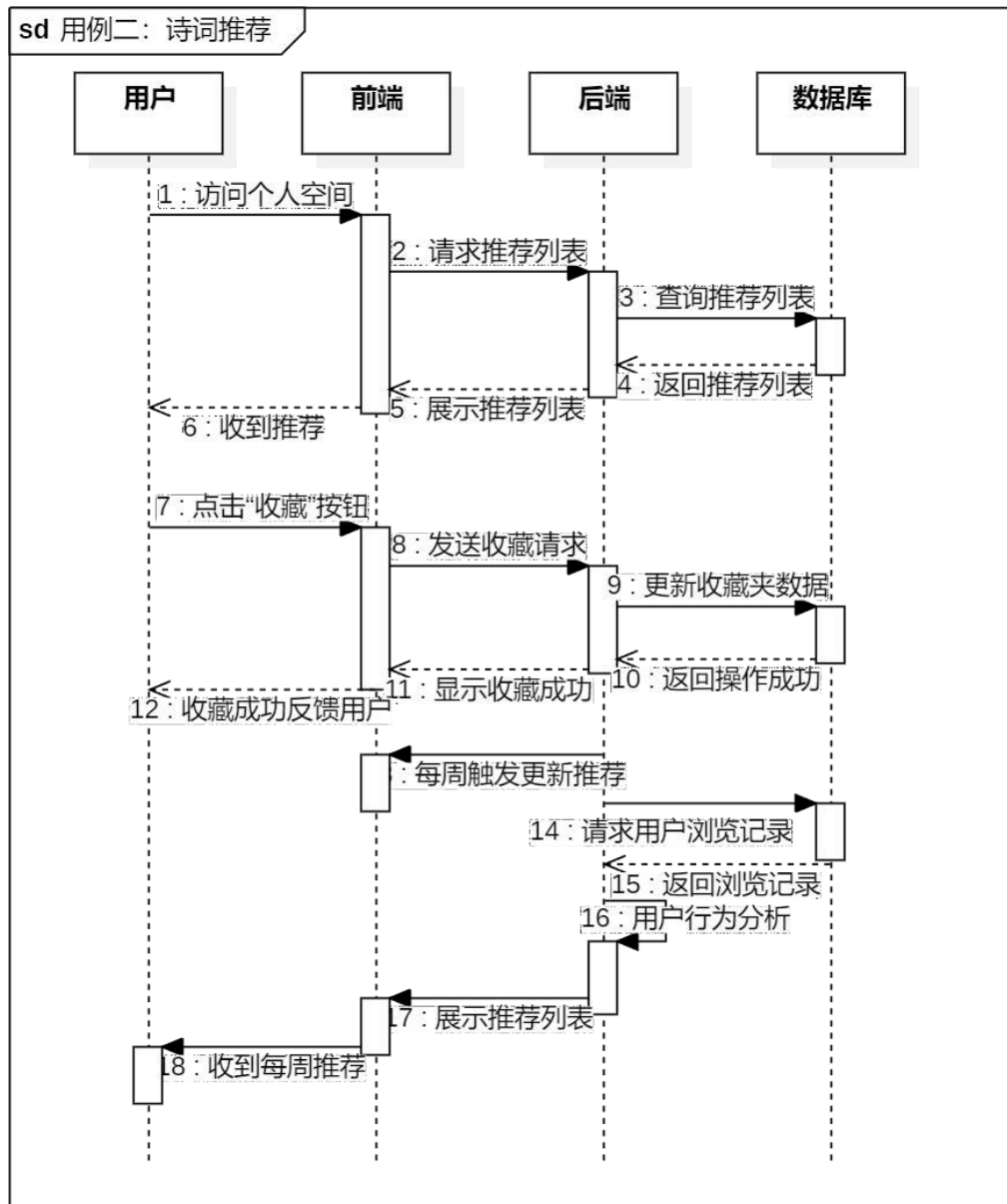
图：管理员诗词管理顺序图

用例名	诗词搜索
用例描述	用户通过作者、标题或诗句片段搜索目标诗词，系统返回匹配结果并展示。 管理员同步维护诗词库的完整性与准确性，

	支持添加和删除诗词。
参与者	用户、管理员、系统（前端界面、后端服务、数据库）
过程	<p>一、搜索诗词</p> <ol style="list-style-type: none"> <li>1. 用户输入搜索内容（作者名、标题或诗句片段）。</li> <li>2. 系统根据输入类型调用对应查询逻辑： <ul style="list-style-type: none"> <li>- 作者查询：从数据库检索该作者所有作品，返回列表。</li> <li>- 标题查询：精确匹配诗词标题，直接跳转至详情页。</li> <li>- 诗句查询：全文检索包含该句的诗词，返回完整内容及出处。</li> </ul> </li> <li>3. 系统将结果格式化后展示给用户。</li> </ol> <p>二、管理员管理诗词</p> <p>添加诗词：</p> <ol style="list-style-type: none"> <li>1. 管理员选择添加方式（手动录入或批量导入文件）。</li> <li>2. 系统校验必填字段（标题、作者、正文）。</li> <li>3. 数据合法时存入数据库，非法时提示错误。</li> </ol> <p>删除诗词：</p> <ol style="list-style-type: none"> <li>1. 管理员选择待删除诗词。</li> <li>2. 系统检查关联数据（如用户收藏记录）。</li> <li>3. 无关联数据时删除诗词，存在关联时提示需先解除依赖。</li> </ol>

表：“诗词搜索”用例描述表

### 3.2.2 “诗词推荐”的用例描述



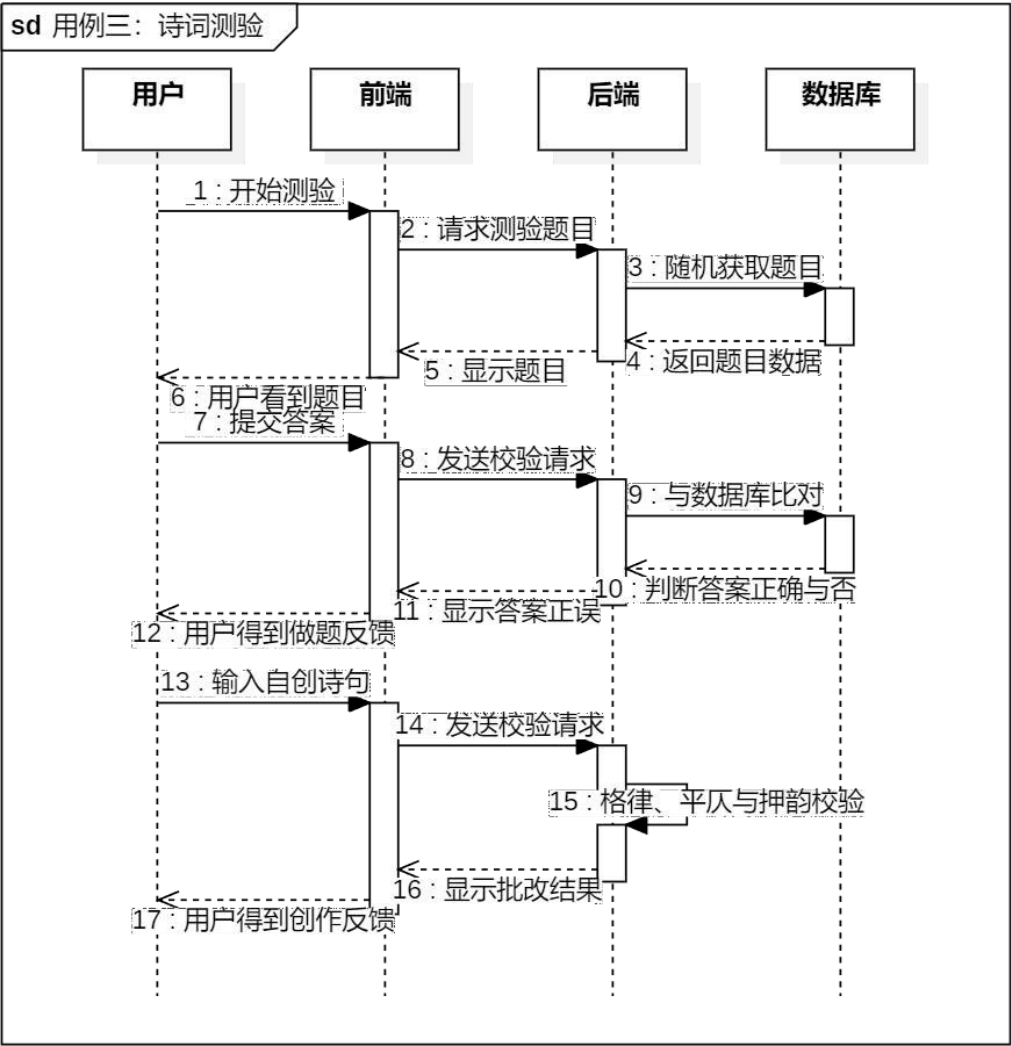
图：“诗词推荐”顺序图

用例名	诗词推荐
用例描述	系统根据用户行为生成个性化推荐列表，并支持用户收藏诗词。
参与者	用户、系统（后端推荐算法、数据库、前端界面）
过程	<p>1. 主动推荐流程：</p> <ul style="list-style-type: none"> <li>- 系统每周分析用户浏览记录，生成兴趣标签。</li> <li>- 根据标签从数据库筛选诗词，更新推荐列表（如“本周宋词精选”）。</li> </ul> <p>2. 用户交互流程：</p>

	<ul style="list-style-type: none"><li>- 用户访问个人空间时，系统展示推荐列表。</li><li>- 用户点击“收藏”按钮，系统将该诗词加入个人收藏夹。</li></ul>
--	---

表：“诗词推荐”用例分析表

3.2.3 “诗词测验”的用例描述

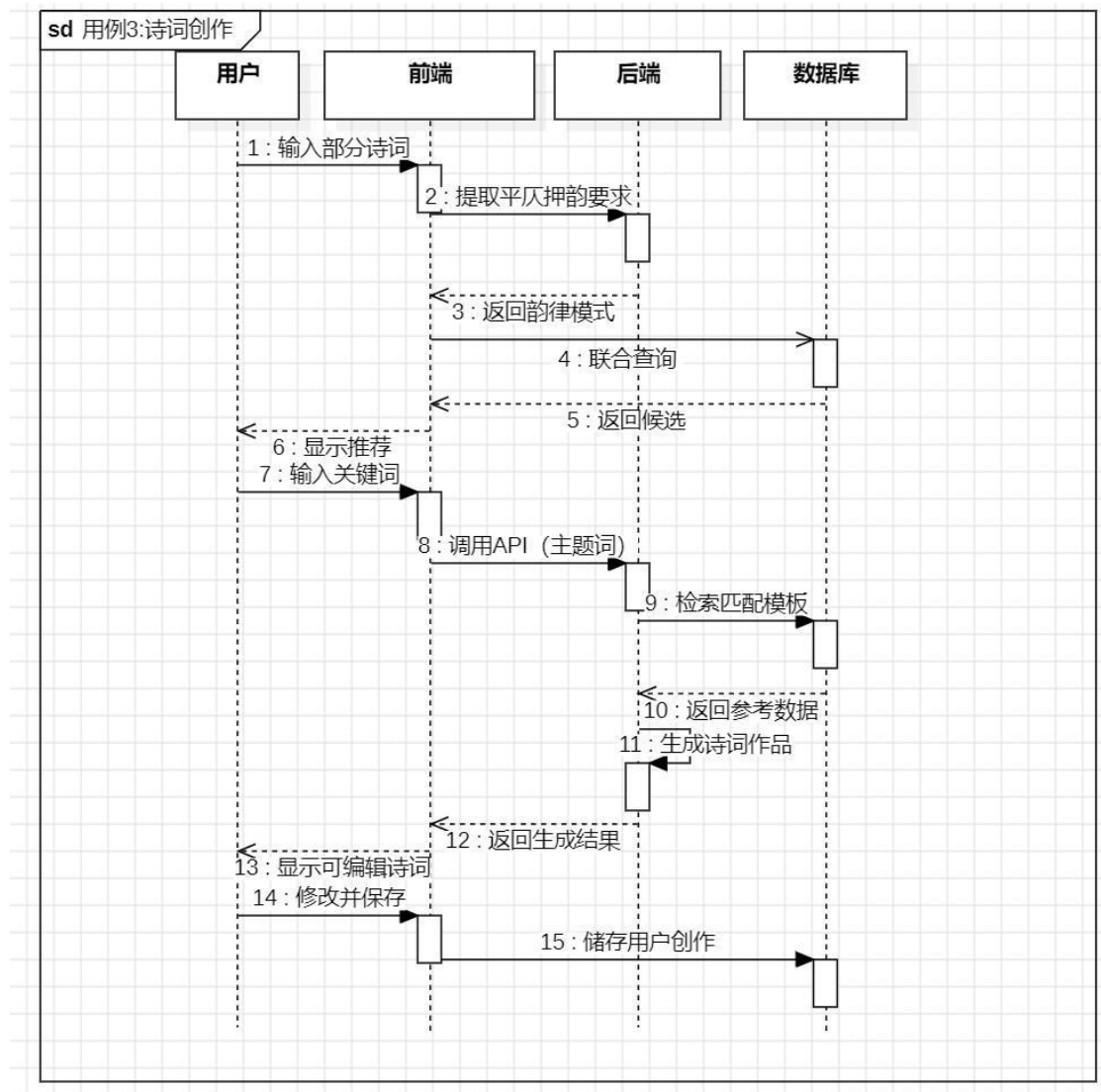


图：“诗词测验”顺序图

用例名	诗词测验
用例描述	用户参与诗词填空测验；或输入自创诗句，自动检测平仄与押韵规则，并标记错误位置。
参与者	用户、系统（前端界面，后端算法，数据库）
过程	1. 填空测验流程： <ul style="list-style-type: none"><li>- 系统随机从数据库选取诗句，隐藏部分文字（如“_前明月光”）。</li><li>- 用户填写答案后，系统比对正确答案并计</li></ul>

	<p>分。</p> <p>2. 格律校验流程：</p> <ul style="list-style-type: none"> <li>- 用户输入自创诗句，提交至系统。</li> <li>- 系统调用格律校验器，检测平仄、押韵规则。</li> <li>- 标记错误位置（如“第三句平仄不符”），返回校验结果。</li> </ul>
--	---

### 3.2.4 “诗词创作”的用例描述



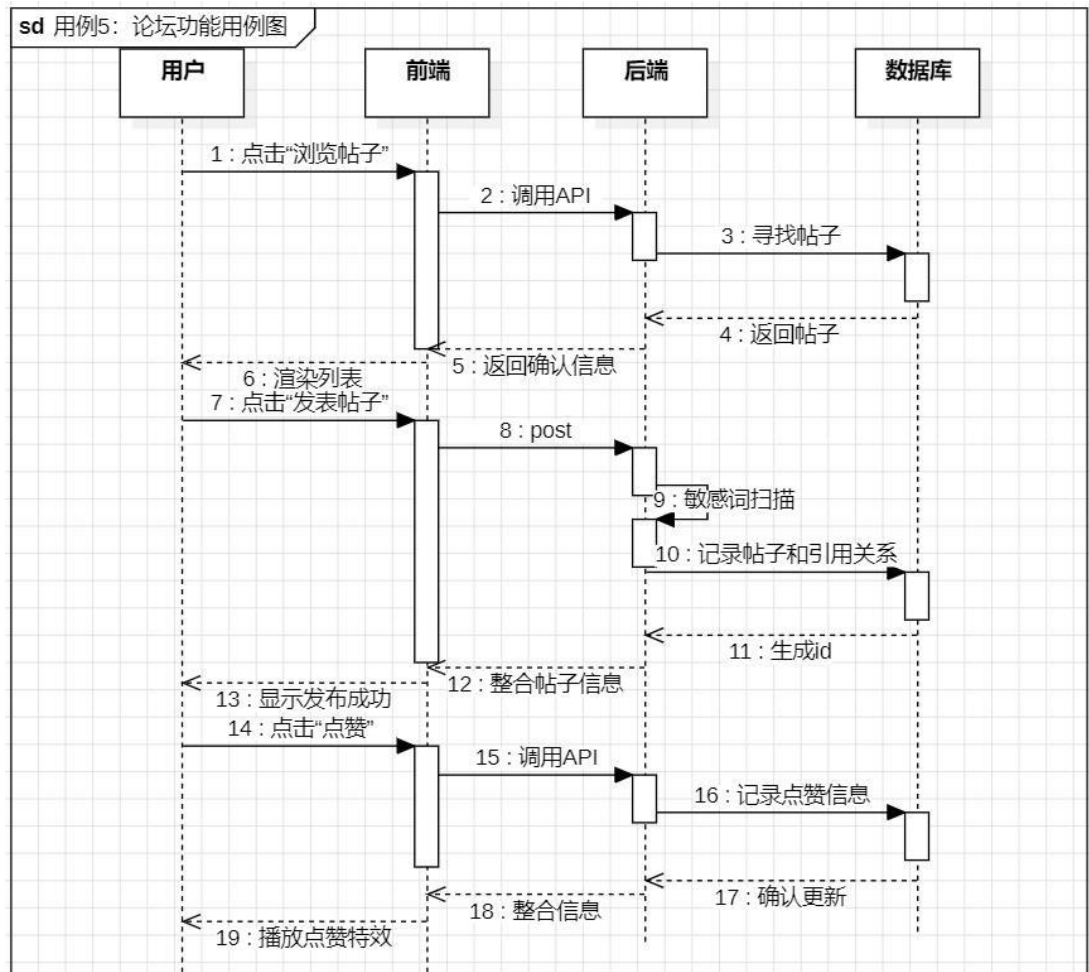
图：“诗词创作”顺序图

用例名	诗词创作
用例描述	支持用户通过部分诗词或者关键词自由创作诗词，并提供 AI 辅助工具，通过押韵需求，主题词等等为用户提供辅助；此外，还能支持用户多次删改。
参与者	用户、系统（前端界面、后端服务、数据库）
过程	创作诗词：：

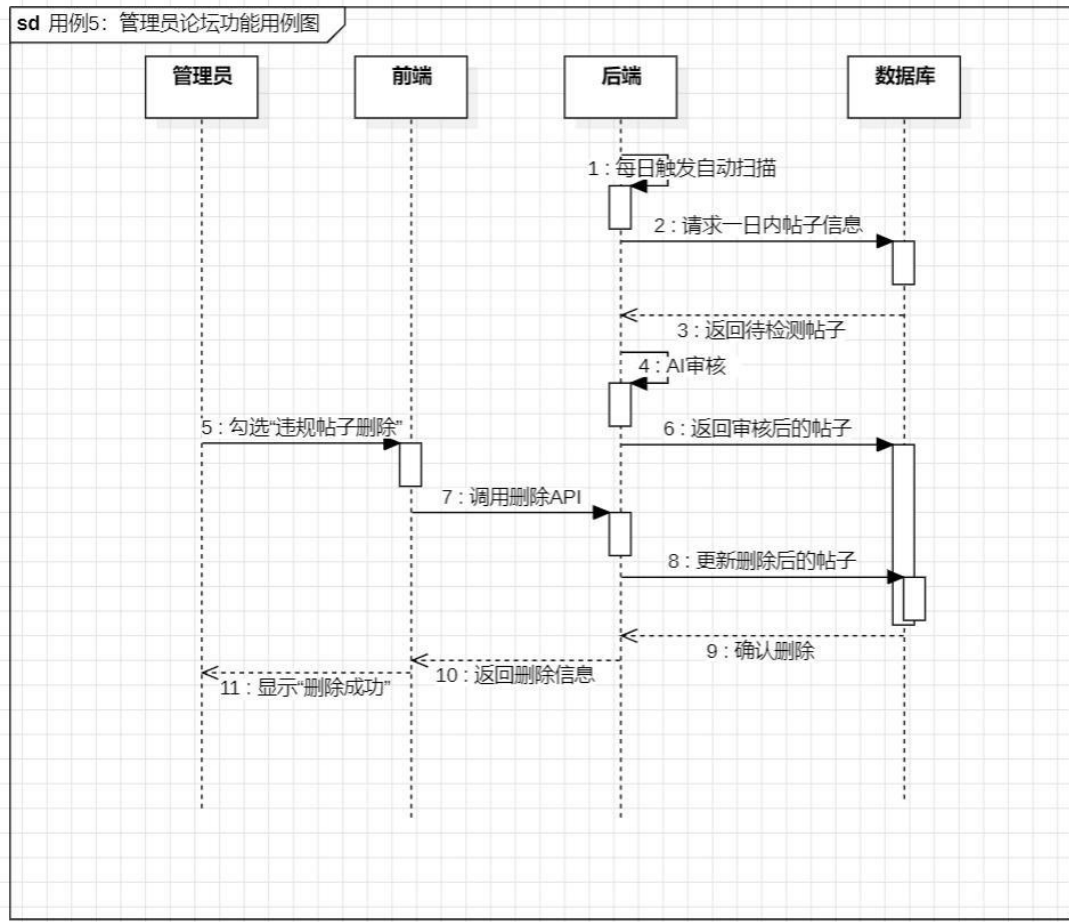
	<ol style="list-style-type: none"> <li>1. 用户输入关键词或者半句诗，和押韵句式的要求</li> <li>2. AI 向数据库请求，数据库返回候选诗词或者尤 AI 自行生成诗词内容</li> <li>3. 后端将数据整合后通过前端返回到用户界面呈现供用户选择</li> <li>4. 选择完成后储存用户创作传输到数据库</li> </ol>
--	--

表：“诗词创作”用例描述表

## 3.2.5 “论坛功能”的用例描述



图：“论坛功能”顺序图 1



图：“论坛功能”顺序图 2

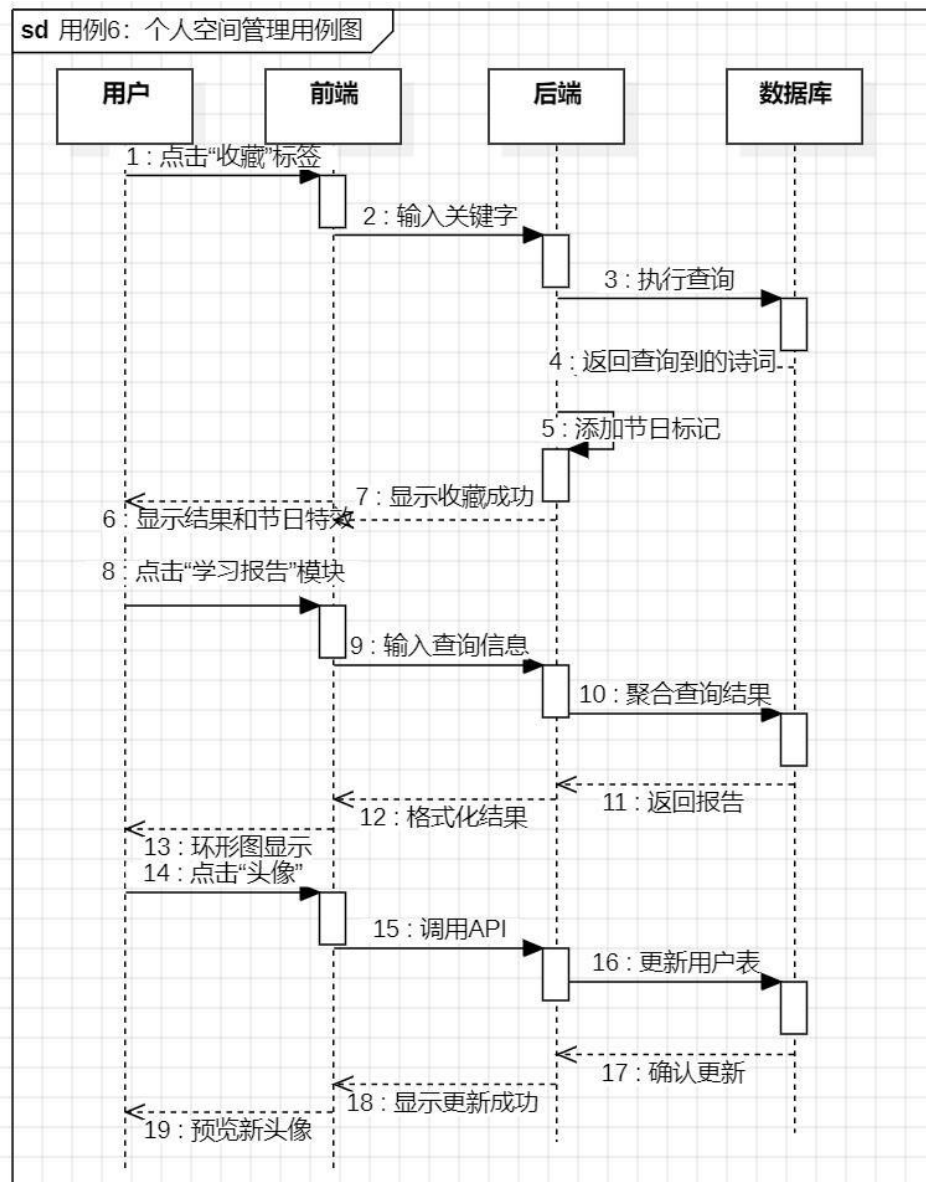
用例名	论坛功能
用例描述	<p>用户有浏览帖子，发表帖子，删除帖子，点赞帖子等等用户论坛功能。</p> <p>管理员同步有删除帖子和审核敏感内容的论坛管理功能。</p>
参与者	用户、管理员、系统（前端界面、后端服务、数据库）
过程	<p>一、用户论坛功能</p> <ol style="list-style-type: none"> <li>1. 浏览帖子：按时间或热度排序查看其他用户发布的帖子（如“《将进酒》赏析”）。</li> <li>2. 发表帖子：用户撰写标题与正文（支持插入诗词引用），点击发布后展示在论坛列表中。</li> <li>3. 删除帖子：用户可删除自己发布的帖子，操作后帖子从公共列表移除。</li> <li>4. 点赞帖子：用户点击“点赞”按钮，帖子热度值增加，并显示在热门板块。</li> </ol> <p>二、管理员管理帖子</p> <ol style="list-style-type: none"> <li>1. 删除帖子：强制删除含敏感信息或</li> </ol>



	<p>广告的帖子（如“招聘兼职”类违规内容）。</p> <p>2. 审核敏感内容：通过关键词过滤机制（如屏蔽政治敏感词），自动标记待审核帖子。</p>
--	---

表：“论坛功能”用例描述表

## 3.2.6 “个人空间”管理



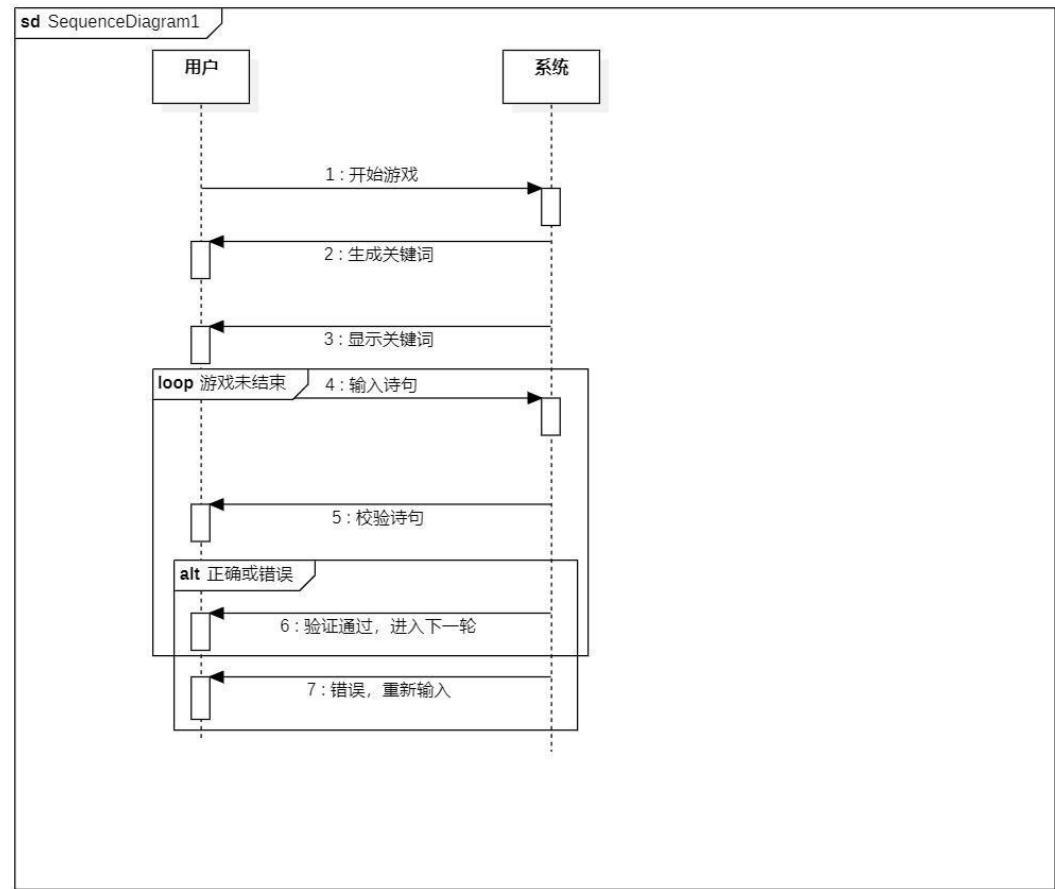
图：“个人空间管理”顺序图

用例名	个人空间管理
用例描述	用户可以查看和管理个人数据，例如个人的收藏，累计学习的诗词，查看个人信息等等
参与者	用户、系统（前端界面、后端服务、数据库）
过程	用户点击相应功能按钮，前端发送访问请求到后端，后端调用用户信息数据库内容：

	<p>--查看收藏的诗词：按朝代、作者或收藏时间分类展示已收藏的诗词。</p> <p>--查看累计学习的诗词：统计用户学习过的诗词总数，并展示最近学习记录（如“近 7 天学习了 15 首唐诗”）。</p> <p>--查看个人信息：显示用户昵称、注册时间、创作作品数等基本信息，支持修改头像和简介。</p> <p>后端返回操作结果，前端格式化信息之后返回给用户界面。</p>
--	--

表：“个人空间管理”用例描述表

3.2.7 飞花令用例描述



图表 1：“飞花令”顺序图

用例名	飞花令游戏
用例描述	用户参与“飞花令”游戏，系统随机指定关键字（如“花”“月”），用户需在规定时间内接龙包含该字的诗句。系统校验用户输入的诗句是否符合要求，并可根据关键词自动生成符合规则的诗句。
参与者	普通用户

	系统
过程	<div>1.用户进入飞花令游戏界面。</div> <div>2.系统随机选择一个关键词（如“花”）。</div> <div>3.用户输入一行包含该关键词的诗句。</div> <div>4.系统校验诗句： 判断是否符合格律要求； 判断是否为真实存在的诗句； 判断是否重复使用过该诗句。</div> <div>5.如果用户输入正确，进入下一轮，并记录该诗句。</div> <div>6.如果用户输入错误： 提示错误原因（如“未包含关键词”“诗句不存在”“与上一轮诗句重复”）。</div> <div>允许用户重新输入，或结束游戏。</div> <div>7.在规定轮数内（如 10 轮），用户与系统或其他玩家轮流接龙。</div> <div>8.轮数达到上限或用户失败后，游戏结束，并显示本轮对战结果。</div>

图表 2：“飞花令”用例描述表

## 4 其它软件需求描述

### 4.1 性能要求

- 页面加载速度应在几秒钟内完成，用户能够快速访问诗词推荐、飞花令、论坛等功能模块，减少等待时间。
- AI 辅助功能响应时间应控制在秒级以内，用户在进行诗词搜索、测验等操作时能够快速获得反馈，提升交互体验。
- 交流论坛、飞花令、诗词测验等沉浸式功能应具备流畅的性能表现，确保切换平滑、无卡顿。
- 系统内存占用应合理控制，尤其在加载大量诗词资源或进行 AI 分析时，不应给设备造成负担。
- 应适配不同设备（如 PC、平板、手机）和复杂网络环境，确保平台整体运行稳定，不出现崩溃或严重延迟。

### 4.2 设计约束

- 用户界面设计应遵循简洁、传统文化氛围与现代美学结合的风格，便于用户快速上手并沉浸体验。
- 功能设计应支持模块化扩展，包括诗词查询、创作、学习、互动、社区等核心板块。
- 数据结构设计需支持标签分类、朝代/诗人索引、多语言注解（如原文、译文、注释）的灵活展示。
- 系统应支持 AI 组件的嵌入与扩展，如语义分析、情感识别、风格模仿、图文结合等功能模块的对接。

### 4.3 界面要求

- 界面应适配不同屏幕尺寸与分辨率，确保用户在桌面与移动端均可获得良好视觉与交互体验。
- 整体界面应保持诗意、雅致的视觉风格，结合古典元素与现代 UI 布局，营造沉浸式文化体验。
- 各功能入口应清晰可见，操作流程直观，用户可轻松完成诗词阅读、搜索、创作、分享等操作。
- 支持个性化设置，如字体风格切换、推荐内容定制、交互语言配置等，以提升用户自由度与满意度。

### 4.4 进度要求

- 项目开发应在 **2025 年 5 月中旬前** 完成主要功能开发与系统测试，达到可交付使用的状态。
- 开发期间应定期进行功能演示（如诗词搜索、飞花令、社区模块），同步收集用户反馈并进行及时调整，确保开发进度与预期目标一致。

### 4.5 交付要求

交付内容：

1. 完整可运行的云舟词渡 Web 端软件。

2. 软件设计规格说明书的电子版文档。
3. 使用说明书的电子文档。

## 4.6 验收要求

- 系统整体运行稳定，无严重 Bug，具备完善的异常处理机制，用户误操作时系统可给予友好提示且不崩溃。
- 对于不同层次用户（如初学者、教师、创作者），平台均能提供清晰指引并支持其顺利完成诗词学习与创作任务。
- 所有模块在交付前完成优化与调试，杜绝遗留的调试信息、测试代码或影响用户体验的未成功能。