

AI辅助个人笔记系统的设计与实现

周 莉¹, 李芳芳¹, 宋 菲¹, 何 怡¹, 颜 丽¹, 罗建军²

¹萍乡学院信息与计算机工程学院, 江西 萍乡

²广州掌动智能科技有限公司, 广东 广州

收稿日期: 2026年9月1日; 录用日期: 2026年10月8日; 发布日期: 2026年10月16日

摘 要

随着数字化学习与办公的日益普及, 个人笔记软件已成为用户整理知识、沉淀信息的重要工具。然而, 当笔记数量急剧增长时, 传统笔记软件偏重记录而缺乏智能辅助的短板日益凸显, 用户难以快速从海量内容中提取关键信息。针对该问题, 本文设计并实现了一款AI辅助的个人笔记管理系统, 旨在通过轻量级自然语言处理技术的本地化部署, 为用户提供更智能、更安全的笔记管理体验。系统不仅实现了层级文件夹、富文本编辑、版本控制、全文搜索、标签管理、附件上传与在线预览等基础功能, 还借助KeyBERT与ONNX Runtime在本地运行, 自动完成笔记关键词提取、摘要生成及智能标签推荐, 所有AI处理均在用户本地完成, 确保数据隐私无泄露。同时, 系统支持带密码与权限的笔记分享、评论与@通知、分享数据统计等协作模块, 兼顾个人效率与团队交流。后端采用Django与Django REST Framework, 通过JWT实现用户认证, 数据库选用PostgreSQL (开发阶段兼容SQLite); 前端基于Vue 3与TypeScript, 并借助Electron打包为跨平台桌面应用, 兼顾可扩展性与日常使用便捷性。整体而言, 本系统在满足笔记管理基本需求的基础上, 借助本地AI能力有效减轻了用户整理信息的负担, 提升了信息检索与回顾的效率, 为隐私优先的智能笔记工具设计提供了可行参考。

关键词

个人笔记管理, 人工智能辅助, 关键词提取, Django, 本地部署

AI-Assisted Personal Note Management System: Design and Implementation

Li Zhou¹, Fangfang Li¹, Fei Song¹, Yi He¹, Li Yan¹, Jianjun Luo²

¹School of Information and Computer Engineering, Pingxiang University, Pingxiang Jiangxi

²Guangzhou Zhangdong TurboMobile Technology Co., Ltd., Guangzhou Guangdong

Received: September 1, 2026; accepted: October 8, 2026; published: October 16, 2026

文章引用: 周莉, 李芳芳, 宋菲, 何怡, 颜丽, 罗建军. AI 辅助个人笔记系统的设计与实现[J]. 软件工程与应用, 2026, 15(5): 716-725. DOI: 10.12677/sea.2026.155067

Abstract

With the increasing prevalence of digital learning and office work, personal note-taking software has become an important tool for users to organize knowledge and store information. However, as the number of notes grows rapidly, traditional note-taking software—which emphasizes recording functions but lacks intelligent assistance—has become increasingly inadequate, making it difficult for users to quickly extract key information from massive amounts of content. To address this issue, this paper designs and implements an AI-assisted personal note management system that leverages the local deployment of lightweight natural language processing technologies to provide users with a more intelligent and more secure note management experience. The system not only implements basic functions such as hierarchical folders, rich-text editing, version control, full-text search, tag management, file attachment upload and online preview, but also leverages KeyBERT and ONNX Runtime to run locally, automatically performing keyword extraction, summarization, and intelligent tag recommendation for notes. All AI processing is completed on the user's local machine, ensuring that data privacy remains uncompromised. In addition, the system supports note sharing with password and permission control, comments with @mentions, and sharing statistics, addressing both personal productivity and team collaboration needs. The backend is built with Django and Django REST Framework, with JWT for user authentication, and PostgreSQL as the database (SQLite is supported during development). The frontend is developed with Vue 3 and TypeScript, and packaged as a cross-platform desktop application using Electron, balancing extensibility with ease of daily use. Overall, while meeting the basic requirements of note management, this system effectively reduces the burden of information organization for users through local AI capabilities, improves the efficiency of information retrieval and review, and provides a feasible reference for the design of privacy-first intelligent note-taking tools.

Keywords

Personal Note Management, AI Assistance, Keyword Extraction, Django, Local Deployment

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 问题提出

目前,市面上的个人笔记管理软件已较为普及,但大多停留在“记录 + 存储”层面,智能化程度不高。用户需手动完成分类、标注和整理,笔记一多便杂乱无章,检索有效信息费时费力。部分软件虽引入 AI 功能,但多依赖云端,用户数据需上传至第三方,存在隐私安全隐患,且高级功能常需付费[1]。针对上述问题,本课题将 AI 能力下沉到本地,让用户无需联网、不上传数据,即可使用关键词提取和摘要生成等辅助功能,在享受智能便利的同时保障数据安全,使笔记成为真正可检索、可回顾的知识库。

2. 技术路线

AI 辅助个人笔记管理系统由前端桌面应用、后端 API 服务和数据库三部分组成。系统采用前后端分离架构,后端基于 Python 的 Django 及 Django REST Framework 构建,这套框架开发效率高,能快速将数据模型转换为 API 接口,路由管理也很方便[2]。前端采用 Vue 3 和 TypeScript,结合 ElementPlus 组件库和 Tiptap 富文本编辑器实现界面交互,组件丰富、上手快,并通过 Electron 打包为跨平台桌面客户端,

用户无需打开浏览器即可使用。AI 模块利用 KeyBERT 和 ONNX Runtime 在本地运行，实现笔记关键词提取和摘要生成，所有处理均不依赖云端，保障用户数据隐私，且响应速度快。文件处理方面集成 pdfjs-dist 和 mammoth，支持 PDF 与 Word 文档的在线预览及 AI 摘要，选用的都是成熟的轻量级方案，兼容性好。测试阶段结合白盒测试与黑盒测试，覆盖面全，并通过 JMeter 验证接口性能[3]。

3. 系统设计

3.1. 系统功能分析

AI 辅助个人笔记管理系统有笔记管理、组织结构管理、附件管理、分享协作、用户管理等功能。用户登录之后，可以创建层级文件夹和标签来分类整理笔记，使用富文本编辑器撰写笔记内容，上传图片、PDF 和 Word 等格式的附件并在线预览，AI 会自动提取笔记关键词并生成内容摘要，帮助用户快速了解笔记要点[4]。用户还可以将笔记生成分享链接给他人，支持设置访问密码和有效期，其他人在分享页面可以查看笔记内容并发表评论，评论时@他人会触发通知提醒。系统提供全文搜索功能，支持按关键词、文件夹、标签等条件筛选笔记，同时具备版本控制功能，用户可查看和恢复笔记的历史版本。用户也可在个人中心查看登录历史记录，并切换主题颜色(浅色/深色/跟随系统)和界面语言(中文/英文)[5]。具体的系统整体架构如图 1 所示。

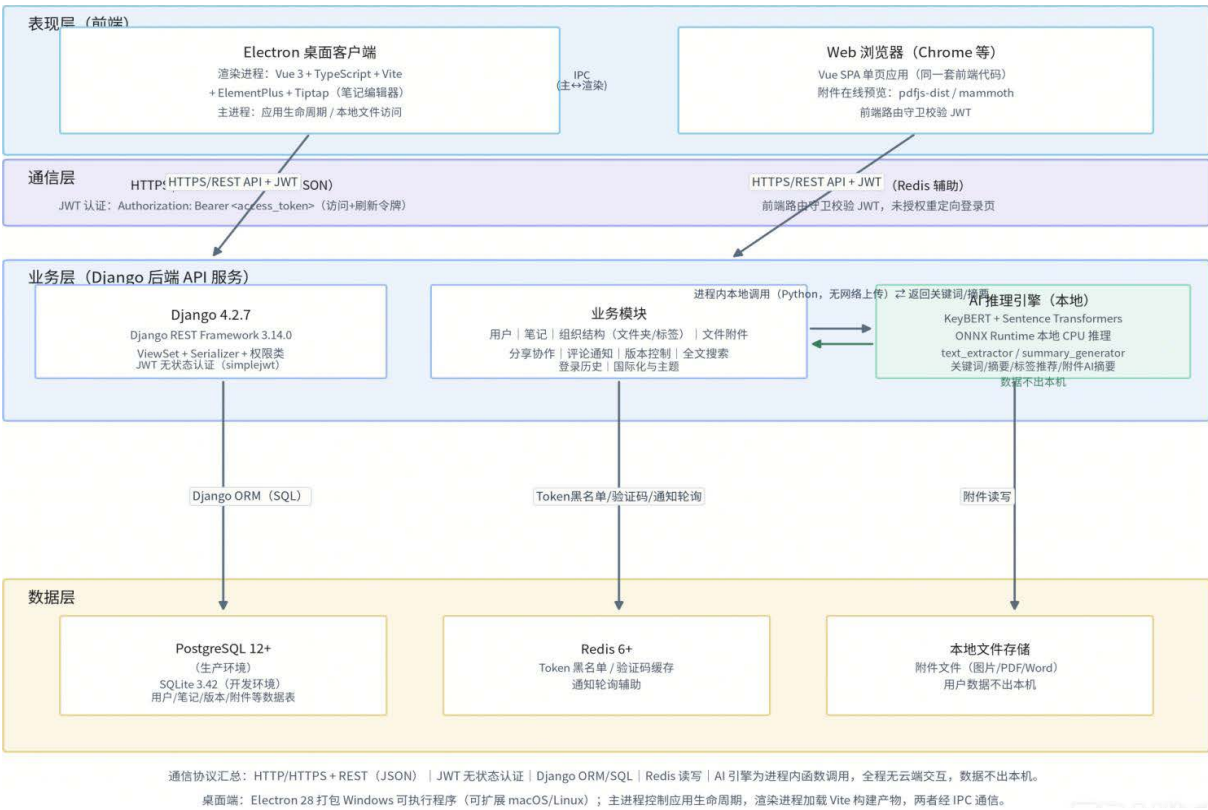


Figure 1. Overall system architecture diagram
图 1. 系统整体架构图

3.2. 数据库设计

根据实体设计，可以得到各个实体的 E-R 图。

3.2.1. 笔记实体

显示笔记的属性，E-R 图如图 2 所示。

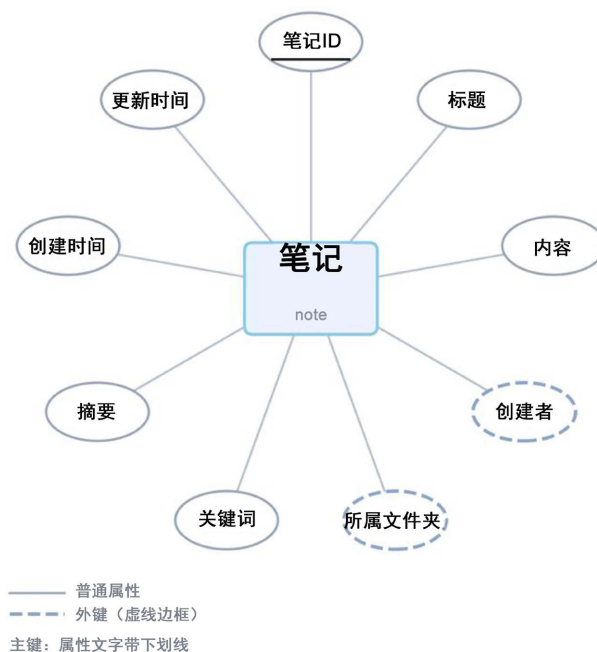


Figure 2. E-R diagram of note entity

图 2. 笔记实体 E-R 图

3.2.2. 用户实体

显示用户的属性，E-R 图如图 3 所示。

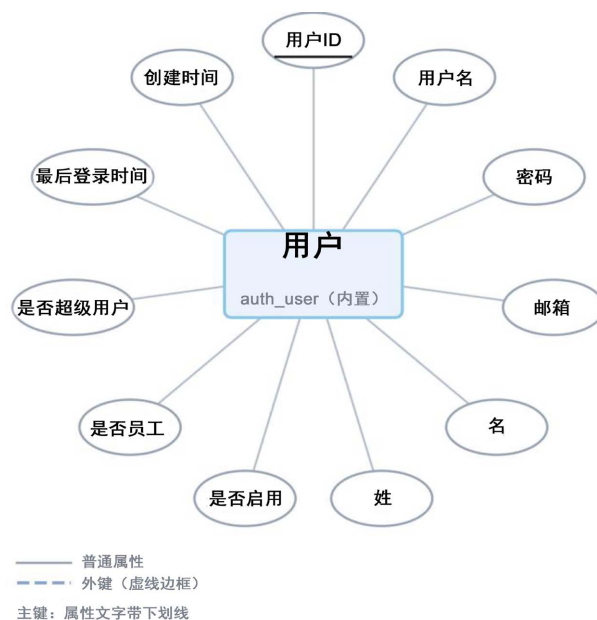


Figure 3. User entity E-R diagram

图 3. 用户实体 E-R 图

3.2.3. 系统 E-R 图

显示系统所有属性，E-R 图如图 4 所示。

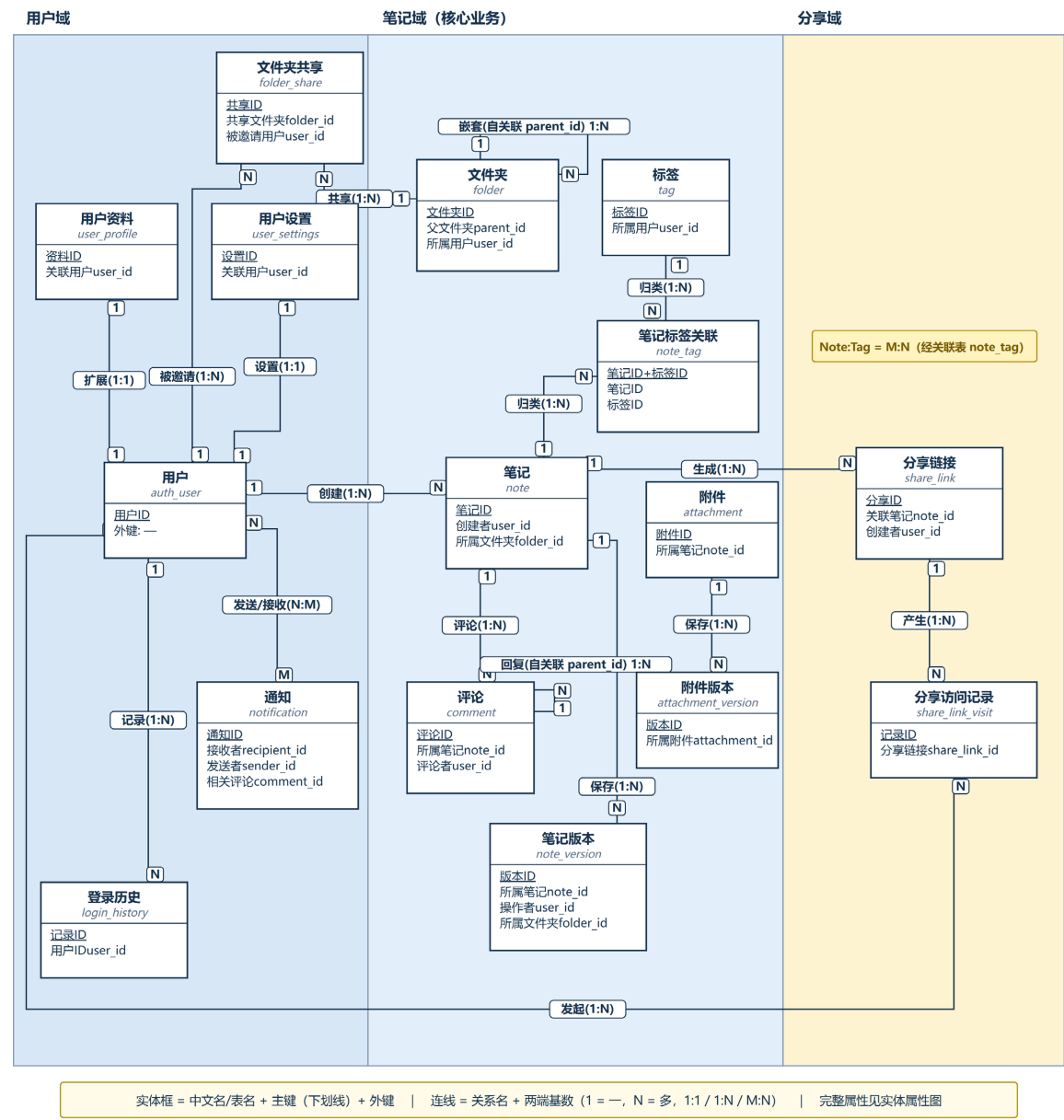


Figure 4. System E-R diagram
图 4. 系统 E-R

3.3. AI 模块设计

3.3.1. 摘要生成算法

摘要生成采用抽取式方法,通过三个维度的句子重要性评估实现:(1) 位置权重:基于文档结构先验,首句和末句通常承载主题引入与全文总结功能。系统对首句赋予 1.5 倍权重,第二句 1.3 倍,倒数第一句 1.5 倍,倒数第二句 1.15 倍,其余普通句子权重为 1.0。当文档总句子数不超过 3 句时,权重从首句至尾

句依次递减。(2) 信号词加权：若句子以“总之”“综上所述”“综上所述”“总而言之”这四个特定总结信号词开头，则该句在位置权重基础上额外乘以加权系数 2.0，以增强总结句被选中的概率。

3.3.2. KeyBERT 底层语言模型选型

本系统底层采用 Sentence Transformers 框架提供的预训练语义模型对文本进行向量化编码，选型依据包括：(1) 多语言支持。提供多语言预训练版本，能有效支持中文文本的语义理解。(2) 轻量高效。模型参数量控制在百兆级别，可在普通 CPU 上完成推理，无需 GPU 加速。(3) 语义表征能力。在语义相似度基准测试中表现良好，能有效捕捉文本的语义信息。(4) 生态成熟。与 KeyBERT 集成方案成熟，开源社区活跃，便于后续维护与迭代。

3.3.3. ONNX Runtime 模型部署与优化

模型优化分三步实施：(1) ONNX 导出——利用 transformers.onnx 将 PyTorch 模型转换为 ONNX 格式，固定 max_length = 128，配置动态轴支持变长输入；(2) 推理会话配置——启用 CPUExecutionProvider 的 Intel MKL-DNN 加速，设置图优化等级为 ORT_ENABLE_ALL (包含算子融合、常量折叠等优化)，根据 CPU 核心数动态配置线程池；(3) 通过上述优化，模型在 CPU 上的推理效率得到明显提升，内存占用也有所降低，能够满足本地实时推理的需求。

4. 系统功能实现

4.1. 笔记管理与版本控制

用户登录后进入笔记列表，左侧为文件夹树和标签列表，右侧以卡片形式展示笔记。点击“新建笔记”或现有笔记进入编辑器。编辑器集成 Tiptap，提供丰富的文本格式化工具。系统每 30 秒检测内容变化并自动保存至后端。版本控制采用快照策略，每次保存时完整记录笔记的 HTML 内容、版本号、变更摘要及操作者，用户可在版本历史中一键回滚。

笔记编辑页面如图 5 所示。



Figure 5. Note editing home screen
图 5. 笔记编辑首页

4.2. 附件管理与 AI 摘要

用户可在笔记编辑页面上传附件。系统根据文件类型进行差异化处理：图片支持预览轮播；PDF 通过 pdfjs-dist 在浏览器端渲染；Word 文档通过 mammoth 转换为 HTML 展示。每篇笔记的“AI 摘要”生成过程如下：用户点击“生成摘要”按钮后，前端请求后端 API，后端调用本地 AI 模块，采用基于位置权重(首句/尾句加权)与总结信号词检测的规则方法筛选核心句子，并利用 KeyBERT 提取关键词，最终将结构化结果(关键词列表 + 摘要文本)保存至数据库并返回前端展示。输出结果如图 6 所示。



Figure 6. List results
图 6. 列表结果

4.3. 笔记分享模块

用户可在笔记详情页点击“分享”按钮创建分享链接。分享支持公开和私密两种模式：公开分享无需密码即可访问；私密分享需设置 4 至 20 位访问密码。过期时间可选永久、5 分钟、1 小时、1 天、7 天、30 天或自定义精确到分钟。生成链接后弹出对话框，可一键复制链接，底部提供统计按钮查看访问数据。笔记分享页面如图 7 所示。

点击统计按钮弹出统计弹窗，显示总访问次数、访问者设备/浏览器/操作系统分布及占比，下方列出最近访问记录(含时间、IP、设备类型、浏览器、操作系统)，时间以“刚刚”“N 分钟前”等格式显示。访问次数统计页面如图 8 所示。

在分享页面中，已登录用户可以在笔记下方发表评论，输入框支持富文本编辑。评论发布后会显示在评论区，支持嵌套回复功能。分享笔记评论页面如图 9 所示。

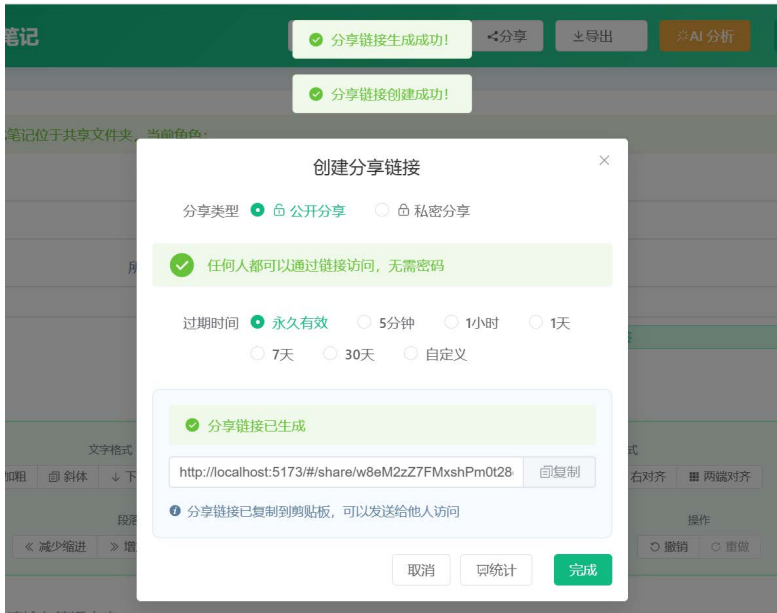


Figure 7. Note sharing page
图 7. 笔记分享页面



Figure 8. Visit count statistics page
图 8. 访问次数统计页面

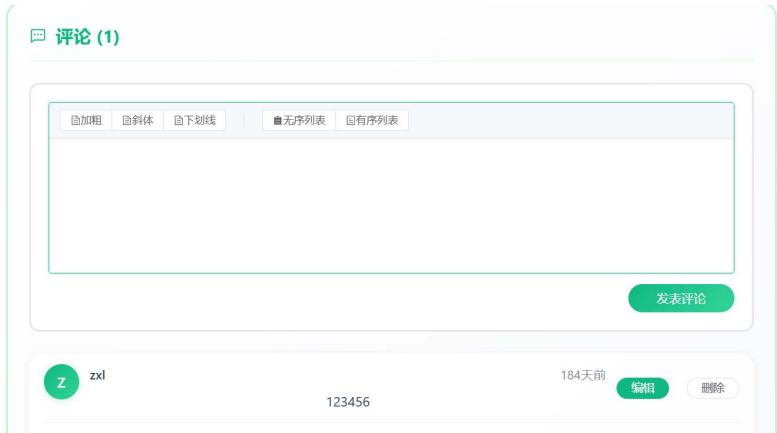


Figure 9. Note sharing and comment page
图 9. 分享笔记评论页面

5. 系统测试与性能评估

5.1. 系统功能与性能测试

采用白盒与黑盒测试相结合的策略进行验证。

功能测试覆盖用户认证、笔记 CRUD、版本控制、全文搜索、附件预览、AI 摘要、分享链接、评论通知等核心模块。白盒测试(4 个笔记模型单元测试)与黑盒测试(登录、编辑、分享、通知等关键功能)全部通过。

性能测试使用 JMeter 对登录接口压测(5 并发用户, 各 3 次请求), 结果如下表 1:

Table 1. Test results
表 1. 测试结果

指标	结果
平均响应时间	257 ms
第 95 百分位响应时间	293 ms
错误率	0%

结论: 系统响应迅速、运行稳定, 满足使用需求。

5.2. AI 功能性能评估

AI 功能在用户本地 CPU 运行, 数据不上传云端。

摘要生成: 基于位置权重(首/尾句 1.5 倍) + 信号词加权(“总之”“综上所述”等 2.0 倍)。

关键词提取: KeyBERT + Sentence Transformers。

处理耗时: ≤ 2 秒(秒级反馈)。

6. 结论

本文设计并实现的 AI 辅助个人笔记管理系统, 通过将 KeyBERT 与 ONNX Runtime 本地化部署, 在保障用户数据隐私的前提下, 提供了关键词提取、摘要生成和智能标签推荐等辅助功能, 有效缓解了传统笔记软件在海量信息面前检索与回顾效率低下的问题。系统同时具备了层级文件夹、富文本编辑、版本控制、全文搜索、附件预览以及带权限控制的分享协作等完备的基础功能, 前后端分离架构与 Electron 打包方案兼顾了可扩展性与跨平台桌面体验。测试结果表明, 系统功能完整、运行稳定, AI 处理可在秒级内完成反馈, 验证了本地轻量级 AI 方案在个人笔记管理场景中的可行性与实用价值。

然而, 当前工作仍存在若干局限性, 有待后续深入研究: (1) 本地模型性能上限——受限于端侧 CPU 推理能力, 当前采用的 Sentence Transformers 模型参数量较小, 语义表征深度有限, 对于长文档或专业领域术语的语义理解精度低于云端大模型, 抽取式摘要也容易出现句子衔接生硬、要点覆盖不全的问题; (2) AI 功能广度不足——系统目前仅支持关键词提取和抽取式摘要, 尚未实现基于用户行为的智能笔记推荐、知识图谱构建、语义搜索或对话式问答等更高级的辅助能力; (3) 协作与多端支持欠缺——分享协作目前仅支持评论与通知, 缺乏实时协同编辑、版本差异对比等团队功能, 且应用仅限于桌面端, 未覆盖移动端或 Web 端实时同步场景; (4) 个性化适配能力弱——AI 模块采用固定规则与通用模型, 未针对用户个人的笔记习惯、知识领域或写作风格进行自适应优化。

基于上述局限性, 未来研究方向可从以下几方面展开: (1) 探索更轻量且高性能的生成式摘要模型(如

DistilBART、TinyLLaMA 等), 结合模型量化与蒸馏技术, 在本地资源约束下提升摘要的连贯性与信息密度; (2) 引入增量学习或用户反馈微调机制, 使 AI 模块能根据用户的标注与使用习惯持续优化提取效果, 实现个性化知识辅助; (3) 扩展多模态处理能力, 支持手写笔记 OCR 识别、图片内文字提取与摘要、音视频转写等场景, 丰富笔记输入形式; (4) 基于图数据库构建笔记间的知识关联网络, 实现智能问答与知识路径推荐, 帮助用户发现隐性知识连接; (5) 完善协作生态, 增加实时协同编辑、变更冲突合并与评论 @消息的桌面推送, 并开发移动端适配版本, 实现全平台无缝同步。通过这些改进, 有望使本地 AI 笔记系统从“辅助整理”向“主动知识服务”演进, 更好地支撑用户在数智时代下的个人知识管理需求。

基金项目

本文由 2026 年江西省大学生创新创业训练计划项目智领思记——AI 辅助个人笔记系统(项目编号: S202610895049X)资助。

参考文献

- [1] 王竹立, 吴彦茹. 数智时代的知识管理: 知识不确定性的挑战及应对策略[J]. 现代远程教育研究, 2024, 36(1): 21-28.
- [2] 夏天生, 苏俏媚, 郭嘉跃. 基于国内网页端云笔记产品的可用性研究[J]. 包装工程, 2025, 46(20): 445-455.
- [3] 刘峻腾, 傅冠斌, 叶辉. 基于手写字体识别技术的笔记 APP 设计功能与策略探究[J]. 设计, 2024, 37(23): 140-144.
- [4] 杨佳豪. 基于深度学习的笔记推荐系统研究与实现[D]: [硕士学位论文]. 江西: 南昌大学, 2024.
- [5] 张标汉, 林宏华. 采用微服务的云笔记系统设计与实现[J]. 三明学院学报, 2024, 41(3): 67-80.