

NPULearn 软件需求说明书

(按 GB/T 8567-88 标准编制)

文档标识: NPU-SRS-001

版本: v1.0

编制日期: 2025 年 4 月 15 日

编制单位: NPULearn 开发团队

1. 文档修改记录

版本	修改日期	修改人	修改内容	审核人
v1.0	2025-04-15	NPULearn 团队	初始版本	孙家泽

目录

1. 文档修改记录	1
2. 引言	4
2.1. 编写目的	4
2.2. 背景	4
2.3. 定义	4
2.4. 参考资料	4
3. 任务概述	5
3.1. 目标	5
3.2. 用户的特点	5
3.2.1. 用户类型	5
3.2.2. 用户技能水平	5
3.3. 假定和约束	5
3.3.1. 假定	5
3.3.2. 约束	5
4. 需求规定	6
4.1. 对功能的规定	6
4.1.1. 功能划分	6
4.1.1.1. AI 智能对话模块	6
4.1.1.2. 富文本渲染模块	7
4.1.1.3. 智能工具协作模块	7
4.1.1.4. 文档处理模块	8
4.1.1.5. 用户管理模块	8
4.1.2. 功能组合	9
4.2. 对性能的规定	9
4.2.1. 精度	9
4.2.2. 时间特性要求	9
4.2.3. 灵活性	9
4.3. 对输入输出的规定	9
4.3.1. 输入格式	9
4.3.2. 输出格式	9
4.3.3. 数据格式	9
4.4. 对数据管理能力的规定	10
4.4.1. 数据量	10
4.4.2. 数据格式	10
4.4.3. 数据组织	10
4.5. 故障处理要求	10
4.5.1. 故障分类	10
4.5.2. 故障处理策略	10
4.5.3. 故障预防	10
4.6. 其他专门要求	10
4.6.1. 安全保密要求	10
4.6.2. 可靠性要求	11

4.6.3. 灵活性要求	11
4.7. 运行环境规定	11
4.7.1. 设备	11
4.7.2. 支持软件	11
4.7.3. 接口	11
4.7.4. 控制	11
5. 运行概念	12
6. 条件与限制	12
6.1. 条件	12
6.2. 限制	12
7. 验收标准	12
7.1. 功能测试标准	12
7.2. 性能测试标准	12
7.3. 兼容性测试标准	12
7.4. 安全测试标准	13

2. 引言

2.1. 编写目的

本文档按照 GB/T 8567-88《计算机软件需求说明书规范》编制，旨在全面、准确地描述 NPULearn 智能学习助手软件的功能需求、性能需求、设计约束和质量要求，为系统设计、编码实现、测试验证和项目管理提供依据。

本文档的预期读者包括：

- 项目开发团队成员
- 系统设计人员
- 测试工程师
- 项目管理人员
- 用户代表
- 维护人员

2.2. 背景

说明：

- a) 待开发软件系统的名称：NPULearn 智能学习助手
- b) 本项目的任务提出者、开发者、用户及实现该软件的计算中心或计算机网络：
 - 任务提出者：西北工业大学“AI 赋能·智启翱翔”主题竞赛
 - 开发者：NPULearn 开发团队
 - 用户：西北工业大学师生
 - 实现环境：跨平台桌面应用（Windows、macOS、Linux、Android）
- c) 该软件系统同其他系统或其他机构的基本的相互来往关系：
 - 与多种 AI 服务提供商的 API 接口（DeepSeek、Gemini）
 - 与 Wolfram Alpha 计算平台的集成
 - 与学术文献数据库的连接
 - 与校园网络环境的适配

2.3. 定义

本文档中使用的专门术语定义如下：

- **AI（人工智能）**：Artificial Intelligence，指通过机器学习等技术实现的智能系统
- **CoT（思维链）**：Chain of Thought，一种让 AI 模型展示推理过程的技术
- **API**：Application Programming Interface，应用程序编程接口
- **富文本**：包含格式信息的文本，支持字体、颜色、链接等样式
- **渲染**：将数据转换为可视化显示内容的过程
- **流式响应**：实时传输和显示数据的技术

2.4. 参考资料

- GB/T 8567-88 计算机软件需求说明书规范
- GB/T 8566-88 计算机软件开发规范
- IEEE Std 830-1998 软件需求规格说明推荐实践
- “AI 赋能·智启翱翔”竞赛技术要求文档

3. 任务概述

3.1. 目标

NPULearn 是一个专为西北工业大学校园场景设计的基于人工智能的智能学习助手桌面应用。系统旨在通过集成多种 AI 模型和专业工具，为不同层次的学习者和科研人员创造个性化、高效且易用的智能学习环境，响应“AI 赋能·智启翱翔”主题要求。

3.2. 用户的特点

3.2.1. 用户类型

1. 本科生群体

- 特点：学习任务重、基础知识需要巩固、对新技术接受度高
- 需求：课程学习辅导、作业解答、考试复习指导

2. 研究生群体

- 特点：具备一定专业基础、需要深入研究、时间相对充裕
- 需求：学术研究支持、论文写作指导、实验数据分析

3. 教师群体

- 特点：专业知识深厚、教学经验丰富、对工具效率要求高
- 需求：教学内容准备、课程设计支持、学术研究辅助

4. 科研团队

- 特点：团队协作需求强、数据处理量大、对精确性要求高
- 需求：文献调研、数据分析、实验结果解读

3.2.2. 用户技能水平

- 计算机操作：中等到高级水平
- 专业知识：根据用户类型从入门到专家级别
- AI 工具使用：初级到中级水平

3.3. 假定和约束

3.3.1. 假定

1. 用户具备基本的计算机操作能力
2. 用户能够接入稳定的互联网连接
3. AI 服务提供商的 API 服务稳定可用
4. 用户愿意学习和使用新的 AI 辅助工具

3.3.2. 约束

1. 技术约束

- 必须基于 Tauri + Vue 3 + TypeScript 架构
- 后端核心逻辑使用 Rust 语言实现
- 需要支持跨平台部署
- 优先选择开源组件和框架

2. 法律约束

- 遵循各 AI 服务商的使用条款和限制

- 符合相关开源许可证要求
- 满足数据保护和隐私法规
- 确保知识产权合规使用

3. 资源约束

- 开发周期限制在 2 个月内
- 团队规模和技术能力限制
- API 调用成本需要控制在合理范围
- 服务器和基础设施成本限制

4. 需求规定

4.1. 对功能的规定

4.1.1. 功能划分

系统功能按照模块化设计原则划分为以下主要部分：

4.1.1.1. AI 智能对话模块

功能标识： F001

功能描述： 提供基于多种 AI 模型的智能对话功能，支持学术问答、知识查询、问题解答等核心交互。

输入：

- 用户文本输入
- 用户选择的 AI 模型
- 对话历史上下文
- 系统配置参数

处理：

- 请求预处理和安全检查
- AI 模型 API 调用
- 响应内容解析和格式化
- CoT 思维链处理（可选）
- 流式响应处理

输出：

- 格式化的 AI 响应内容
- 思维链推理过程（如启用）
- 相关建议和推荐
- 错误信息（如有）

性能要求：

- 响应延迟 < 3 秒（不含 AI 模型处理时间）
- 支持并发请求处理
- 99.5% 的可用性保证

4.1.1.2. 富文本渲染模块

功能标识: F002

功能描述: 提供强大的富文本内容渲染能力，支持 Markdown、数学公式、图表等多种格式的显示。

输入:

- Markdown 格式文本
- LaTeX 数学公式
- Mermaid/Pintora 图表描述
- 主题配置信息

处理:

- 内容格式识别和解析
- 异步渲染处理
- 缓存机制应用
- 主题适配处理

输出:

- 渲染后的 HTML 内容
- 样式化的数学公式
- 交互式图表
- 错误提示信息

性能要求:

- 渲染时间 < 2 秒
- 支持批量渲染
- 内存占用优化

4.1.1.3. 智能工具协作模块

功能标识: F003

功能描述: 集成专业工具，通过智能编排提供一站式问题解决方案。

输入:

- 用户查询请求
- 工具选择偏好
- 计算参数
- 代码内容

处理:

- 需求识别和工具选择
- Wolfram Alpha API 调用
- Python 代码安全执行
- 结果整合和优化

输出:

- 计算结果和可视化
- 代码执行结果

- 知识库查询结果
- 工具推荐建议

4.1.1.4. 文档处理模块

功能标识: F004

功能描述: 智能处理多种格式文档，提供文档分析、总结和问答功能。

输入:

- PDF/Word/文本文档
- 用户查询问题
- 分析配置参数

处理:

- 文档内容提取和解析
- 智能摘要生成
- 关键信息识别
- 问答匹配处理

输出:

- 文档摘要和要点
- 问题答案和引用
- 结构化分析结果
- 相关内容推荐

4.1.1.5. 用户管理模块

功能标识: F005

功能描述: 管理用户设置、历史记录、主题配置等个性化信息。

输入:

- 用户配置信息
- API 密钥设置
- 主题偏好选择
- 历史记录数据

处理:

- 安全的数据存储
- 配置验证和管理
- 历史记录智能分组
- 主题自动适配

输出:

- 个性化界面配置
- 安全的数据管理
- 智能历史分类
- 用户偏好应用

4.1.2. 功能组合

各功能模块通过标准化接口进行组合，形成完整的用户 workflow：

1. **学习辅导流程：** AI 对话 → 富文本渲染 → 工具协作 → 结果展示
2. **文档分析流程：** 文档上传 → 内容解析 → AI 分析 → 问答交互
3. **科研支持流程：** 问题输入 → 工具选择 → 计算处理 → 结果可视化

4.2. 对性能的规定

4.2.1. 精度

- 数学计算精度：支持双精度浮点运算
- 文本处理精度：99% 以上的字符识别准确率
- AI 响应准确性：依赖于所选 AI 模型的固有精度

4.2.2. 时间特性要求

- 系统启动时间：≤ 5 秒
- 用户界面响应时间：≤ 500 毫秒
- AI 对话响应时间：≤ 3 秒（不含 AI 模型处理时间）
- 文档渲染时间：≤ 2 秒
- 文件上传处理时间：≤ 10 秒（100MB 以内文件）

4.2.3. 灵活性

- 支持多种 AI 模型动态切换
- 支持插件化功能扩展
- 支持自定义主题和界面配置
- 支持多语言界面（中文、英文）

4.3. 对输入输出的规定

4.3.1. 输入格式

- 文本输入：UTF-8 编码，支持多语言
- 文档格式：PDF、DOCX、TXT、MD 等常见格式
- 图像格式：PNG、JPEG、GIF、SVG 等
- 代码格式：支持主流编程语言语法

4.3.2. 输出格式

- 文本输出：格式化 HTML、纯文本、Markdown
- 图表输出：SVG、PNG 格式
- 数据输出：JSON、CSV 格式
- 文档输出：PDF、HTML 格式

4.3.3. 数据格式

- 配置数据：JSON 格式存储
- 历史记录：SQLite 数据库
- 缓存数据：二进制格式
- 日志数据：结构化文本格式

4.4. 对数据管理能力的规定

4.4.1. 数据量

- 历史对话记录：支持 10,000 条以上
- 文档缓存：支持 1GB 本地存储
- 配置数据：支持复杂的层级配置结构
- 临时数据：支持高效的内存管理

4.4.2. 数据格式

- 统一使用 UTF-8 编码
- 结构化数据采用 JSON 格式
- 二进制数据采用标准压缩格式
- 数据库采用 SQLite 标准格式

4.4.3. 数据组织

- 按用户会话组织历史记录
- 按时间和主题分类数据
- 采用索引提高查询效率
- 实现数据的增量备份

4.5. 故障处理要求

4.5.1. 故障分类

1. **系统级故障**：应用程序崩溃、内存溢出等
2. **网络级故障**：API 调用失败、网络中断等
3. **数据级故障**：数据损坏、存储空间不足等
4. **用户级故障**：输入错误、操作异常等

4.5.2. 故障处理策略

- 自动重试机制：网络请求失败时自动重试最多 3 次
- 优雅降级：部分功能不可用时保证核心功能正常
- 错误恢复：提供数据恢复和系统重置功能
- 用户提示：清晰的错误信息和解决建议

4.5.3. 故障预防

- 输入验证：对所有用户输入进行安全检查
- 资源监控：实时监控内存和存储空间使用
- 异常捕获：完善的异常处理机制
- 日志记录：详细的操作和错误日志

4.6. 其他专门要求

4.6.1. 安全保密要求

- API 密钥采用 AES-256 加密存储
- 用户数据本地化存储，不上传云端
- 网络通信采用 HTTPS 加密
- 定期进行安全漏洞扫描和修复

4.6.2. 可靠性要求

- 系统可用性 $\geq 99.5\%$
- 平均故障恢复时间 < 1 分钟
- 数据一致性和完整性保证
- 具备自动故障检测和报告能力

4.6.3. 灵活性要求

- 模块化架构支持功能扩展
- 支持新 AI 模型的快速接入
- 支持第三方工具和服务集成
- 支持个性化配置和定制

4.7. 运行环境规定

4.7.1. 设备

硬件要求:

- CPU: x86_64 或 ARM64 架构
- 内存: 最低 4GB RAM, 推荐 8GB RAM
- 存储: 1GB 可用磁盘空间
- 网络: 稳定的互联网连接

软件要求:

- 操作系统: Windows 10/11、macOS 10.15+、Linux 主流发行版、Android 8.0+
- 运行时环境: 系统自带, 无需额外安装
- 网络协议: 支持 HTTP/HTTPS 协议

4.7.2. 支持软件

- 系统浏览器引擎 (WebKit/Chromium)
- 系统网络栈
- 系统文件管理器
- 系统通知服务

4.7.3. 接口

- AI 服务商 API 接口 (DeepSeek、Gemini)
- Wolfram Alpha API 接口
- 系统文件系统接口
- 系统网络接口
- 系统 UI 框架接口

4.7.4. 控制

- 用户通过图形界面进行操作控制
- 系统配置通过设置界面管理
- 服务状态通过系统托盘监控
- 数据管理通过内置工具控制

5. 运行概念

系统采用事件驱动的架构模式，主要运行流程如下：

1. **系统初始化**：加载配置、初始化模块、检查网络连接
2. **用户交互**：接收用户输入、处理交互事件、更新界面状态
3. **AI 服务调用**：构建请求、调用 API、处理响应、展示结果
4. **数据管理**：自动保存、定期清理、备份恢复、同步更新
5. **系统维护**：性能监控、错误处理、日志记录、资源管理

6. 条件与限制

6.1. 条件

- 需要稳定的互联网连接访问 AI 服务
- 需要有效的 AI 服务商 API 密钥
- 需要符合最低硬件配置要求的设备
- 需要用户具备基本的计算机操作能力

6.2. 限制

- 受 AI 服务商 API 调用频率限制
- 受网络带宽和延迟影响
- 受本地存储空间限制
- 受操作系统兼容性限制
- 受开源许可证使用条款限制

7. 验收标准

7.1. 功能测试标准

- 所有核心功能模块正常运行
- AI 模型切换和调用成功率 $\geq 95\%$
- 文档处理和渲染准确率 $\geq 98\%$
- 用户设置保存和恢复成功率 = 100%

7.2. 性能测试标准

- 满足所有性能指标要求
- 压力测试下系统稳定运行
- 内存泄漏测试通过
- 并发用户测试达标

7.3. 兼容性测试标准

- 在所有目标操作系统上正常运行
- 不同硬件配置下功能完整
- 多分辨率屏幕正确显示
- 无障碍功能测试通过

7. 4. 安全测试标准

- 数据加密功能验证通过
- 网络安全测试无严重漏洞
- 输入验证机制有效
- 隐私保护措施完善