# AI-EduMate 智能教育助手软件需求分析

## 1. 引言

本文档旨在详尽阐述 AI-EduMate 智能教育助手的软件需求。AI-EduMate 是一款面向学生群体和科研工作者的智能学习效率工具，深度融合人工智能技术，致力于提供“计划-学习-整理-复习”全流程闭环支持。本系统通过实现智能化、个性化、无缝化的“三化一体”学习模式，旨在让 AI 主动理解学习需求，而非被动响应指令，并基于认知特征（如记忆曲线、注意力模式）动态调整策略，为用户个性化定制学习计划，打通“输入-处理-输出”全流程，从而消除多工具切换成本，全面提升学习效率和体验。

## 2. 软件需求分析

当代学生面临诸多学习困境，例如在多任务管理方面存在困难，平均需同时处理 3.2 个课程项目，繁杂的课程让学生难以合理安排时间。同时学生存在笔记碎片化问题，据统计有57%的学生面临笔记检索的难题，部分学习任务还存在跨语言学习障碍等问题。这些痛点反映出当前学生在学习过程中缺乏有效的工具和方法来应对复杂的学习任务和管理学习资料。

近年来，各国政府纷纷出台政策，大力推动教育与科技的深度融合。我国出台的《教育信息化 2.0 行动计划》，旨在推动人工智能、大数据等新兴技术在教育领域的广泛应用，构建智能化的教育环境，这为 AI 学习助手软件这类教育科技产品的研发与推广提供了良好的政策支持与发展机遇。在政策的引导下，学校和教育机构对数字化、智能化学习工具的采购意愿和投入显著增加，为产品进入市场打开了广阔的通道。综上所述，这款软件有充足的市场空间。

近些年随着 AI 技术的发展，GPT-4 等大模型的成熟度不断提升，为智能学习工具提供了强大的语言处理和理解能力。比方说，OCR 识别准确率达到 99%，使得手写文字等信息的数字化转换更加准确高效，这使得我们可以在软件中加入识别并整理手写笔记的功能。同时，跨平台开发框架的普及则降低了软件开发的成本和难度，便于实现多平台的应用。

市场方面，现有学习工具功能较为单一，无法满足用户的全面需求，并且缺乏整合 AI 能力的全流程学习解决方案。例如番茄 todo 软件的功能较为单一，侧重于时间管理，没有

涉及到多模态知识管理、AI 根据认知特征制定个性化学习计划等功能；ANKI 软件缺乏 AI 智能化日程编排，无法用 AI 来解析自然语言指令生成日程；印象笔记软件缺乏智能化日程编排，无法通过学习进度动态追踪，根据完成情况推荐优化方案。

## 3. 项目目标

AI-EduMate 项目的主要目标是构建一个高效、智能、用户友好的学习辅助平台，具体目标包括：

- **提升学习效率与质量**：通过 AI 驱动的智能问答、笔记整理和日程编排，显著减少用户在学习和资料管理上的时间消耗，提高学习成果。
- **实现个性化学习路径**：根据用户的学习进度、习惯和认知特点，动态调整学习计划和推荐优化方案，提供定制化的学习支持。
- **优化知识管理流程**：支持多模态笔记（语音转文字、手写识别）和关键词提取，构建个人知识图谱，解决笔记碎片化和检索难题。
- **增强沉浸式学习体验**：提供多语言翻译、屏幕取词和学术术语库联动功能，消除跨语言学习障碍，营造无缝的学习环境。
- **确保系统稳定性与可靠性**：构建一个高可用、安全且易于维护的系统，保障用户数据的完整性和服务的连续性。

## 4. 系统架构概述

AI-EduMate 系统采用 **MCP（Model-Controller-Persistence）架构** 进行设计与实现。这种架构模式旨在提供清晰的职责分离、高内聚和低耦合的系统结构，从而提高代码的可读性、可维护性、可扩展性，并促进团队协作开发。

**Model（模型层）**：此层负责定义核心业务逻辑、数据结构以及业务规则。它独立于用户界面和数据存储机制，专注于处理应用程序的核心功能，例如用户认证逻辑、消息处理、会话管理以及学习工具（课程表、预约行程、笔记）的数据模型和业务规则。

**Controller（控制层）**：此层作为用户请求的入口，负责接收并解析来自前端的请求。它协调模型层和持久层的工作，调用相应的业务逻辑，并准备数据以供前端（视图层）进

行展示。例如，处理用户登录、消息发送、新会话创建、以及课程表/预约行程/笔记的增删改查请求。

**Persistence（持久层）**：此层专注于数据存储和检索操作。它负责与底层数据库（如MySQL）进行交互，执行数据的持久化、查询、更新和删除。通过抽象数据访问逻辑，使得模型层无需关心具体的数据存储细节，提高了系统的灵活性和可移植性。

## 5. 功能需求

AI-EduMate 的核心功能需求如下：

### 5.1 用户管理模块

（1）**用户注册**：用户应能通过提供用户名、密码等信息完成新用户注册。

（2）**用户登录**：用户应能通过已注册的用户名和密码登录系统，系统应支持基于 JWT（JSON Web Token）的身份认证机制以确保会话安全。

（3）**用户信息更新**：用户应能查看并修改个人信息（如用户名、邮箱）。

（4）**用户注销**：用户应能安全地注销当前登录会话。

（5）**登录状态持久化**：系统应能通过本地存储（如 LocalStorage）持久化用户登录状态，实现自动登录或会话恢复。

### 5.2 智能问答模块（AI 聊天）

（1）**消息发送**：用户应能在聊天界面输入文本消息并发送给 AI 助手。

（2）**AI 回复接收**：AI 助手应能接收用户消息，并通过 DeepSeek 模型生成并返回智能回复。

（3）**Markdown 渲染**：AI 回复内容应支持 Markdown 格式的渲染，包括标题、段落、列表、代码块（带语法高亮和复制功能）、表格、引用和图片。

（4）**正在输入指示**：当 AI 正在生成回复时，系统应显示“正在输入”指示器，提升用户体验；回复完成后自动隐藏。

（5）**多轮对话**：系统应支持多轮对话，保持会话上下文，确保 AI 回复的连贯性。

（6）**新建对话**：用户应能主动创建新的对话，开始全新的 AI 会话。

（7）**对话历史管理**：系统应保存用户的对话历史，并在侧边栏清晰展示，用户可以切换查看不同的历史对话。对话历史应按时间顺序排列，并显示标题和日期。

（8）**自动生成会话标题**：系统应能根据对话的首条消息或核心内容自动生成会话标题。

（9）**对话历史数量限制**：对话历史列表应限制显示数量，例如最多显示最近 10 个对话，以保持界面的简洁性。

### 5.3 学习工具集成模块

（1）**课程表管理**：用户应能通过独立的界面查看、添加、编辑和删除个人课程表条目。

（2）**预约行程管理**：用户应能通过独立的界面管理和查看预约行程，包括添加、编辑和删除预约。

（3）**笔记本功能**：用户应能通过独立的界面进行笔记记录、编辑、分类和检索。

（4）**模块刷新**：课程表、预约行程和笔记本模块的界面应支持刷新功能，以确保数据最新。

### 5.4 用户界面与交互模块

（1）**响应式布局**：系统界面应具备完全响应式设计，能够自适应不同尺寸的设备（桌面、平板、移动设备）和屏幕方向。

（2）**侧边栏折叠/展开**：侧边栏应支持折叠和展开功能，以优化不同屏幕尺寸下的空间利用。

（3）**移动端菜单切换**：在移动设备上，侧边栏应默认为隐藏状态，并通过一个可切换的菜单按钮进行显示/隐藏。

（4）**用户信息展示**：侧边栏应清晰展示当前登录用户的头像（基于用户名首字母生成）、用户名和邮箱。

（5）**菜单导航**：侧边栏应集成登录、注销、课程表、预约行程和笔记本等核心功能的快捷导航入口。

（6）**模态对话框关闭**：所有模态对话框（登录、课程表、预约行程、笔记本）应支持点击对话框外部区域关闭。

## 6. 可维护性需求

- （1）**代码规范**：所有代码（包括后端 Java、前端 HTML/CSS/JS）应遵循统一的编码规范，具备良好的可读性、一致性和充分的注释。
- （2）**模块化设计**：系统应采用高度模块化的设计，各功能模块（如用户管理、聊天、课程表）应独立且职责明确，降低模块间的耦合度，方便独立开发、测试、部署和替换。
- （3）**日志记录**：系统应具备完善的日志记录机制，记录关键操作、异常信息和性能指标，方便故障排查、性能监控和审计。
- （4）**错误处理**：系统应具备健壮的错误处理机制，能够优雅地捕获和处理各类异常情况，并向用户提供清晰、友好的错误提示，避免系统崩溃。
- （5）**文档与测试**：核心模块应提供详细的设计文档和 API 文档。关键业务逻辑应覆盖单元测试和集成测试，确保代码质量和功能正确性。
- （6）**易于部署**：系统应提供清晰的部署指南和脚本，支持快速、自动化部署。

## 7. 技术实现栈

AI-EduMate 项目采用以下技术栈进行实现：

**后端开发：**

1) **语言与框架**：使用 Java 语言，基于 Spring Boot 框架进行后端服务开发。Spring Boot 提供快速的应用开发能力、强大的生态系统和便捷的部署方式，非常适合构建 RESTful API。
2) **AI 集成**：通过 API 调用集成 DeepSeek 大模型，用于实现智能问答、内容生成等核心 AI 功能。

**前端开发：**

1) **基础技术：**采用标准的 **HTML、CSS 和 JavaScript** 构建用户界面。HTML 负责页面结构，CSS 负责样式和布局，JavaScript 负责交互逻辑和动态内容。
2) **库/框架：**使用 SockJS 和 Stomp.js 实现 WebSocket 通信，Marked.js 用于 Markdown 解析，Highlight.js 用于代码高亮。

**数据库：**

1) **类型与产品：**使用 **MySQL** 作为关系型数据库，用于持久化存储用户数据、对话历史、课程表、预约行程和笔记等所有结构化数据。

**架构模式：**

1) **MCP 架构：**整个系统严格遵循 MCP 架构模式，确保了后端逻辑的清晰分层和前端与后端之间的有效分离，从而提升了项目的可维护性和可扩展性。

## 8. 安全性需求

（1）**数据加密：**所有敏感数据（如用户密码、API 密钥）在传输过程中应使用 HTTPS 加密，并在存储时进行哈希加盐处理。

（2）**身份认证与授权：**所有 API 接口均需进行严格的身份认证（通过 JWT）和权限控制，确保只有授权用户才能访问相应资源。

（3）**输入验证与过滤：**对所有用户输入进行严格的后端验证和过滤，有效防范 SQL 注入、跨站脚本（XSS）攻击等常见 Web 安全漏洞。

（4）**会话管理：**采用安全的会话管理机制，防止会话劫持、会话固定等攻击。

（5）**错误信息处理：**系统不应在前端或日志中暴露敏感的错误信息或系统内部细节。

（6）**访问控制：**实施基于角色的访问控制（RBAC），确保不同用户角色拥有恰当的权限。