

# 系统架构设计

## 一、系统概述

AI-EduMate 是一个集成先进 AI 技术的智能学习系统，包含客户端和服务端两部分。客户端负责与用户交互，服务端处理业务逻辑、数据存储和 AI 相关计算。

## 二、整体架构

### 2.1 架构图

系统采用前后端分离架构，整体架构如下：



### 2.2 分层架构

客户端：

- 表现层：负责用户界面展示和交互，使用 Thymeleaf 模板引擎渲染页面，集成前端框架展示系统功能。
- 控制层：处理用户请求，如注册、获取课程表、预约行程等，通过 HTTP 协议与服务端通信。
- 工具层：包含 JWT 工具，用于用户认证和授权。

服务端：

- 控制层：接收客户端请求，调用服务层处理业务逻辑。
- 服务层：实现具体业务逻辑，如课程管理、预约管理等。
- 数据访问层：使用 MyBatis 进行数据库操作，负责数据的增删改查。
- 工具层：集成 AI 工具和课程管理工具，提供 AI 支持和课程管理功能。

数据库：

使用 MySQL 数据库存储系统数据，其中包含了 `tbl_course_schedules`、`tbl_appointments`、`tbl_notes` 等表。

### 三、关键组件

#### 3.1 客户端组件

- `McpClientApplication`: Spring Boot 启动类，初始化客户端应用。
- `WebConfig`: 配置 Web 拦截器，使用 JWT 进行请求认证。
- `RegisterController`: 处理用户注册请求。
- `ScheduleController`: 处理获取课程表数据请求。
- `AppointmentController`: 处理获取预约行程请求。

#### 3.2 服务端组件

- `McpServerApplication`: Spring Boot 启动类，初始化服务端应用。
- `CourseSchedulesTool`: 课程管理工具，提供课程的插入、删除和查询功能。
- `MyBatis Generator`: 自动生成实体类、映射文件和 DAO 层代码。

#### 3.3 数据库组件

- MySQL: 存储系统的业务数据。
- MyBatis: 实现对象关系映射，简化数据库操作。

### 四、关键设计决策

#### 4.1 前后端分离

采用前后端分离架构，客户端和服务端通过 HTTP 协议通信，提高系统的可维护性和可扩展性。客户端专注于用户界面和交互，服务端专注于业务逻辑和数据处理。

#### 4.2 JWT 认证

使用 JWT 进行用户认证和授权，客户端在登录后获取 JWT 令牌，后续请求

携带该令牌进行身份验证。JWT 令牌无状态，适合分布式系统。

#### 4.3 MyBatis Generator

使用 MyBatis Generator 自动生成实体类、映射文件和 DAO 层代码，提高开发效率，减少手动编写重复代码的工作量。

#### 4.4 AI 集成

集成 DeepSeek 大模型，为系统提供智能学习支持，实现智能对话、内容生成等功能。采用混合推理模式，云端大模型与端侧轻量化模型协同计算，确保无网络时仍可使用核心功能。后续计划逐步集成更多先进 AI 模型，如 GPT - 4o、CLIP、Whisper 等，以提供更全方位的智能学习支持。

#### 4.5 日志管理

使用 Logback 进行日志管理，将日志输出到控制台和文件，设置日志文件的保留天数和最大文件大小，方便系统监控和问题排查。

### 五、技术栈

#### 5.1 客户端

- Spring Boot：快速搭建 Web 应用。
- Thymeleaf：模板引擎，用于渲染 HTML 页面。
- WebSocket：实现实时通信。
- JWT：用户认证和授权。

#### 5.2 服务端

- Spring Boot：快速搭建 Web 应用。
- Spring WebFlux：响应式编程框架，提高系统性能。
- MyBatis：对象关系映射框架。
- MySQL：关系型数据库。