# 教师AI智慧备课系统设计文档

## 1. 项目概述

随着教育信息化的发展，教师备课工作亟需智能化工具支持。本系统旨在为教师提供智能化的备课环境，结合人工智能技术辅助教师高效创建课程内容和练习题，同时为学生提供基于主流Java技术的开发实践平台。系统通过集成AI能力，实现智能化的题目生成，并提供教师和学生之间高效互动的功能。

### 项目目标：

提供课程和知识点管理平台；

集成AI能力辅助生成知识点相关的练习题；

为学生提供基于Spring生态的开发实践环境。

## 2. 系统架构设计

本系统采用**前后端分离**架构，前端使用 **Vue** 框架，后端使用 **Spring Boot**，**Spring MVC** 提供服务。系统通过 **MyBatis** 进行数据库交互，AI部分通过集成第三方AI平台（如OpenAI）生成题目。数据库使用 **MySQL** 存储所有数据。

### 2.1 技术栈

**前端**：Vue 3 + Element Plus

**后端**：Spring Boot + Spring MVC

**数据库**：MySQL 8.0 + MyBatis

**AI集成**：通过RESTful API与OpenAI或其他合规平台集成

**安全**：Spring Security实现基于RBAC的认证与授权

**开发环境**：IntelliJ IDEA、Maven、Git

### 2.2 系统架构图

系统架构包括以下主要层次：

**前端**：用户界面通过Vue与后端交互，展示课程管理、题目生成、学生练习等功能。

**后端**：使用Spring Boot和Spring MVC处理RESTful API请求，管理业务逻辑和数据持久化。

**数据库**：MySQL存储用户数据、课程、知识点、题目等信息。

**AI集成**：后端通过AI平台接口生成题目，并返回给前端进行预览和编辑。

## 3. 核心功能模块

### 3.1 用户角色与权限

系统设定不同的用户角色，每个角色拥有不同的权限，主要包括：

**教师**：课程管理、知识点维护、题目管理、AI辅助生成练习题。

**学生**：查看课程内容、按知识点做练习题。

**管理员**：用户管理。

### 3.2 核心功能模块

系统的核心功能模块如下：

**课程管理模块**：

功能：课程的创建、编辑、删除，课程分类管理。

操作：提供增、删、改、查等操作，支持课程的学科、年级分类。

**知识点管理模块**：

功能：维护课程相关的知识点，包括增、删、改、查以及导入导出功能。

操作：支持按照课程管理知识点信息，设定难易程度，支持批量导入导出。

**题目管理模块**：

功能：管理练习题（包括选择题、填空题、简答题等），支持题目与知识点的关联。

操作：支持题目增删改查，支持设置题目难度和类型。

**智能题目生成模块**：

功能：通过集成AI生成相关知识点的练习题，支持题目筛选、预览和编辑。

操作：用户可以设置生成题目的难度、类型等参数，并对生成结果进行编辑和筛选。

**学生练习模块**：

功能：学生根据知识点进行练习，并通过智能组卷功能完成自我测试。

操作：学生可以选择知识点进行练习，并且系统可以根据其学习情况智能生成试卷。

## 4. 前端设计

### 4.1 前端项目结构

前端使用 **Vue 3 + Element Plus** 技术栈开发。以下是项目的基本结构：

├── public

│ └── favicon.ico # 项目图标

├── src

│ ├── api # API 请求管理（admin.js、director.js、teacher.js）

│ ├── assets # 静态资源

│ ├── components # 可复用组件

│ ├── router # 路由配置

│ ├── store # 全局状态管理（Pinia/Vuex）

│ ├── styles # 样式管理（SCSS 变量、全局样式）

│ ├── views # 页面级组件（Admin、Director、Teacher）

│ └── App.vue # 根组件

└── package.json # 依赖包配置

### 4.2 核心页面

**管理员模块**：包括用户管理、角色与权限管理、通知管理等页面。

**系主任模块**：包括专业培养方案管理、课程支撑关系表管理、毕业要求达成情况评价等页面。

**教师模块**：包括教学大纲管理、成绩管理、课程目标达成评价、毕业要求达成情况评价等页面。

### 4.3 核心代码示例

// admin.js (管理员API)

import request from '@/utils/request'

// 用户管理

export function importUsers(formData) {

return request({

url: '/admin/user/import',

method: 'POST',

data: formData

})

}

// 获取用户列表

export function fetchUserList(params) {

return request({

url: '/admin/user/list',

method: 'GET',

params

})

}

// director.js (系主任API)

import request from '@/utils/request'

// 导入专业培养方案

export function importProgram(formData) {

return request({

url: '/director/program/import',

method: 'POST',

data: formData

})

}

// teacher.js (教师API)

import request from '@/utils/request'

// 导入教学大纲

export function importSyllabus(data) {

return request({

url: '/teacher/syllabus/import',

method: 'POST',

data

})

}

## 5. 后端设计

### 5.1 后端架构

后端采用 **Spring Boot + Spring MVC** 结构，使用 **MyBatis** 进行数据库访问，系统的主要功能由不同的Controller层处理，业务逻辑由Service层实现，数据持久化操作由Repository层完成。

### 5.2 核心API设计

#### 管理员模块 API 示例：

POST /admin/user: 新增用户

GET /admin/user/list: 获取用户列表

PUT /admin/user/status: 启用/禁用用户

#### 系主任模块 API 示例：

POST /director/program/import: 导入专业培养方案

GET /director/program: 获取专业培养方案

#### 教师模块 API 示例：

POST /teacher/syllabus/import: 导入教学大纲

GET /teacher/grades: 获取学生成绩

### 5.3 安全设计

系统使用 **Spring Security** 实现认证和授权，采用 **BCrypt** 对用户密码进行加密。接口通过JWT令牌进行身份验证，确保数据传输的安全性。

## 6. 数据库设计

### 6.1 数据库表设计

**用户表** (users): 存储用户信息，如用户名、密码、角色等。

**课程表** (courses): 存储课程的基本信息，如课程名、描述等。

**知识点表** (knowledge\_points): 存储与课程相关的知识点。

**题目表** (questions): 存储各类题目信息，如题目类型、难度等。

### 6.2 示例数据库表结构

CREATE TABLE users (

id INT PRIMARY KEY AUTO\_INCREMENT,

username VARCHAR(50) NOT NULL,

password VARCHAR(255) NOT NULL,

role VARCHAR(20) NOT NULL

);

CREATE TABLE courses (

id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100) NOT NULL,

description TEXT,

teacher\_id INT,

FOREIGN KEY (teacher\_id) REFERENCES users(id)

);

CREATE TABLE knowledge\_points (

id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100) NOT NULL,

course\_id INT,

difficulty VARCHAR(20),

FOREIGN KEY (course\_id) REFERENCES courses(id)

);

CREATE TABLE questions (

id INT PRIMARY KEY AUTO\_INCREMENT,

content TEXT NOT NULL,

type VARCHAR(20) NOT NULL,

difficulty VARCHAR(20),

knowledge\_point\_id INT,

FOREIGN KEY (knowledge\_point\_id) REFERENCES knowledge\_points(id)

);

## 7. 非功能性需求

### 7.1 性能需求

页面响应时间应小于 2 秒。

系统应支持至少 100 名教师并发使用。

### 7.2 可维护性

代码注释率应大于 30%。

单元测试覆盖率应大于 70%。

### 7.3 安全需求

**认证与授权**：基于 Spring Security 实现 RBAC 模型。

**数据安全**：敏感数据加密存储，防止 SQL 注入、XSS、CSRF 攻击。

## 8. 总结

本设计文档详细描述了 **教师AI智慧备课系统** 的整体架构、功能模块、接口设计、数据库设计以及非功能性需求等内容。通过结合 Vue、Spring Boot、MyBatis 等技术，系统不仅实现了智能化的题目生成，还提供了高效的教师备课和学生练习平台。系统的设计考虑了可扩展性、安全性和高性能，为后续的功能扩展和系统优化奠定了坚实基础。