# 软件管理平台设计文档

## 架构设计

## 

### 1.1 系统架构概述

软件管理平台采用前后端分离的架构设计，基于大型语言模型实现需求分析与代码生成。整体架构分为前端展示层、后端服务层、AI模型层和数据存储层四个部分。

功能特点:

1. 系统架构大纲生成：基于通义千问模型，从需求文档生成系统架构大纲

2. 代码生成：基于DeepSeek模型，生成各模块代码实现

3. 数据库设计：自动生成MySQL数据库脚本

4. 测试用例生成：根据需求和代码生成测试用例

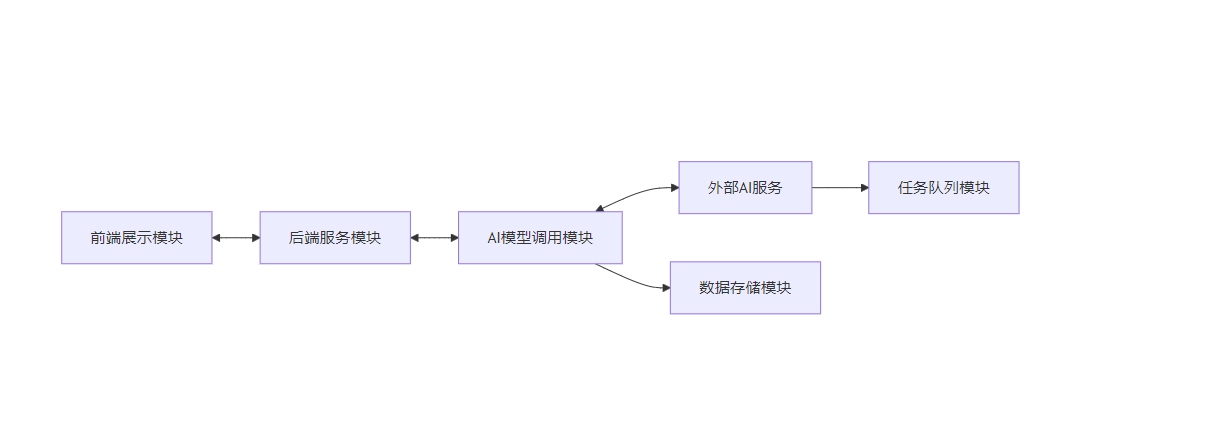
5. 部署指南生成：自动生成系统部署文档

### 1.2 模块组成及关系

#### 1.2.1 核心模块

|  |  |  |
| --- | --- | --- |
| 模块名称 | 职责描述 | 相关技术 |
| 前端展示模块 | 提供用户界面，展示生成的代码、架构和结果 | React, TypeScript |
| 后端服务模块 | 处理前端请求，协调AI模型调用 | Flask, Python |
| AI模型调用模块 | 封装对大模型API的调用逻辑 | Python |
| 数据存储模块 | 存储项目信息、生成历史和用户数据 | MySQL |
| 任务队列模块 | 管理异步任务和长时间运行的AI调用 | Redis, Celery |

#### 1.2.2 模块间关系



### 1.3 前后端与数据库关系

#### 1.3.1 前端组件

1. **项目管理界面**：管理所有软件项目
2. **需求编辑器**：输入和编辑需求文档
3. **代码展示器**：显示生成的代码，支持语法高亮
4. **架构可视化**：展示系统架构的图形化表示
5. **配置面板**：设置AI模型参数和项目配置

#### 1.3.2 后端服务

1. **认证服务**：处理用户认证和授权
2. **项目服务**：管理项目元数据和状态
3. **AI协调服务**：协调不同AI模型的调用和结果整合
4. **导出服务**：处理代码和文档的导出功能

#### 1.3.3 数据流向

1. 用户通过前端界面提交需求文档
2. 后端服务接收请求并创建异步任务
3. AI模型调用模块分别调用通义千问生成大纲和DeepSeek生成代码
4. 生成结果存储在数据库中
5. 前端通过轮询或WebSocket获取最新结果
6. 用户可查看、编辑或导出生成的内容

## 2. 数据设计

### 2.1 数据库设计

### 核心表结构

系统主要包含以下核心数据表：

用户表(users)：存储用户基本信息，包括用户名、密码哈希、电子邮箱等字段，使用自增ID作为主键

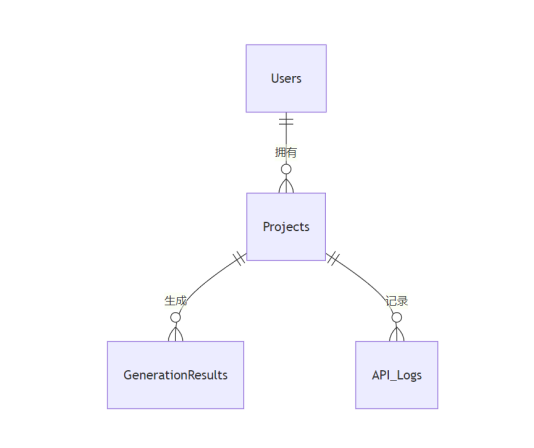
项目表(projects)：存储项目相关信息，包括项目名称、描述、需求文档、状态等字段，关联到用户表

生成结果表(generation\_results)：保存AI生成的内容，包括结果类型(大纲、代码、数据库脚本、测试用例、部署方案)、内容、使用的模型等字段，关联到项目表

API调用日志表(api\_logs)：记录API调用情况，包括API类型、请求内容、响应内容、状态码、错误信息等字段，用于监控和问题排查

所有表均采用InnoDB引擎，使用utf8mb4字符集，并包含创建时间和更新时间字段用于记录数据的变更历史。各表之间通过外键关系建立关联，保证数据完整性。

### 2.2 数据模型关系

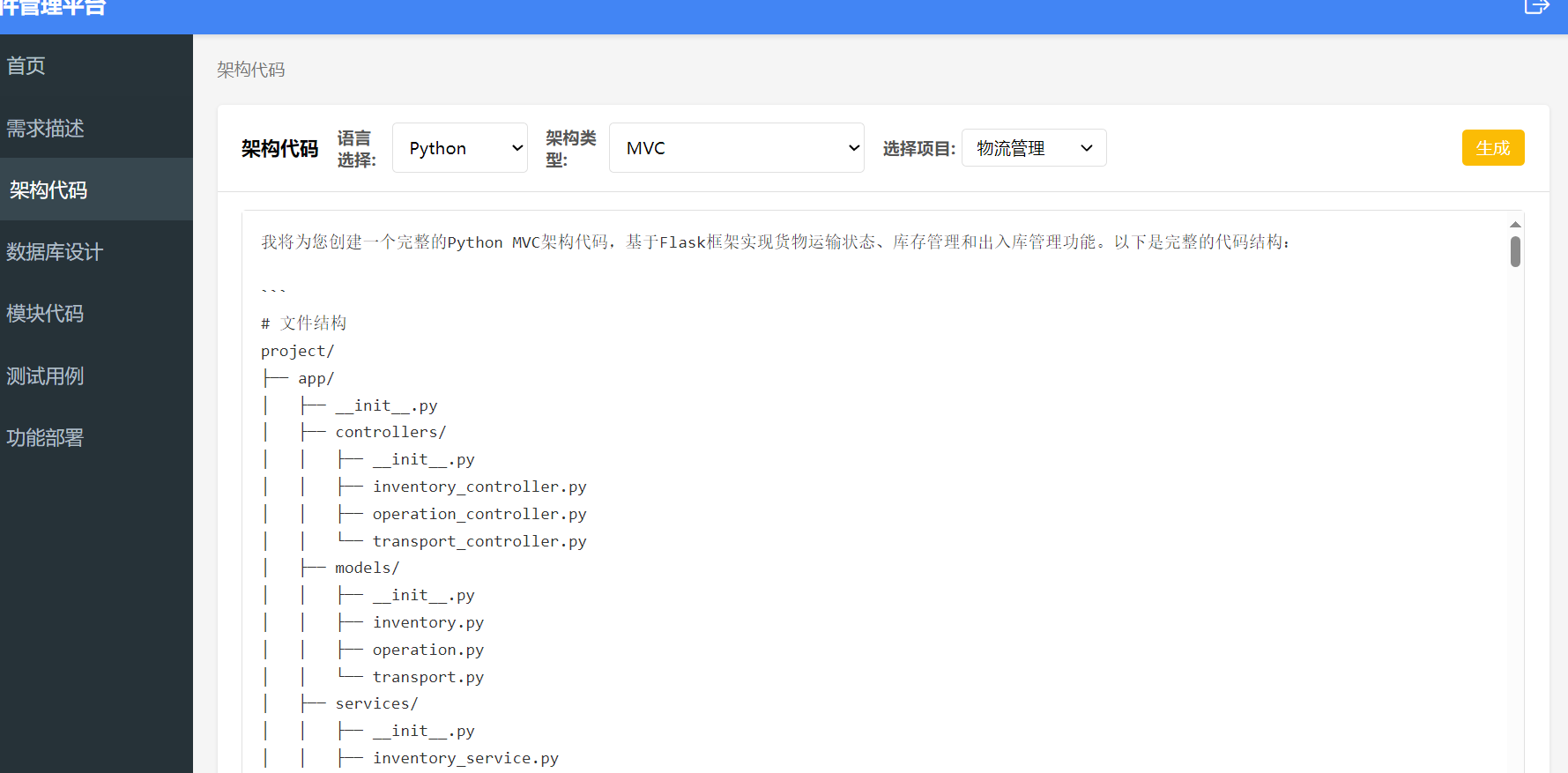


### 2.3 缓存设计

1. 使用Redis缓存频繁访问的数据
2. 缓存AI模型生成的大纲和结果，设置合理TTL
3. 利用缓存减少重复的API调用

## 3. AI代码生成功能

### 3.1 架构代码生成



功能描述: 根据需求文档生成系统架构和代码框架

使用流程:

1. 在项目详情页面选择"生成架构代码"

2. 选择目标语言和架构类型（MVC、微服务等）

3. 点击"开始生成"按钮

4. 系统先调用通义千问生成架构大纲并在控制台显示

5. 然后调用DeepSeek生成详细架构代码

6. 显示生成结果

特性:

支持多种编程语言（Python、Java、JavaScript等）

支持多种架构模式（MVC、微服务等）

生成结果可直接导出或在线编辑

### 3.2 数据库设计生成



功能描述: 根据需求和架构生成数据库设计和SQL脚本

使用流程:

1. 选择"生成数据库设计"功能

2. 选择数据库类型（MySQL、PostgreSQL、MongoDB等）

3. 点击"开始生成"按钮

4. 通义千问生成数据库设计大纲并显示在控制台

5. DeepSeek生成详细的数据库脚本

6. 显示生成结果

功能特点:

生成标准的数据库DDL脚本

包含表结构、字段定义、索引和约束

支持直接导出SQL脚本

### 3.3 模块代码生成



功能描述: 生成特定功能模块的代码实现

使用流程:

1. 选择"生成模块代码"功能

2. 选择目标模块（用户管理、权限控制、数据访问等）

3. 指定技术栈和依赖库

4. 点击"开始生成"按钮

5. 通义千问生成模块设计大纲并显示在控制台

6. DeepSeek生成详细模块代码

7. 显示生成结果

支持模块类型:

用户认证模块

数据访问层

REST API接口

业务逻辑层

### 3.4 测试用例生成



功能描述: 根据代码自动生成单元测试和集成测试用例

使用流程:

1. 选择"生成测试用例"功能

2. 上传或选择已生成的代码

3. 选择测试框架

4. 点击"开始生成"按钮

5. 通义千问生成测试大纲并显示在控制台

6. DeepSeek生成详细测试用例代码

7. 显示生成结果

测试类型:

单元测试

集成测试

接口测试

性能测试基准

### 3.5 部署方案生成



功能描述: 生成适合项目的部署配置和脚本

使用流程:

1. 选择"生成部署方案"功能

2. 选择目标部署环境

3. 指定基础设施需求

4. 点击"开始生成"按钮

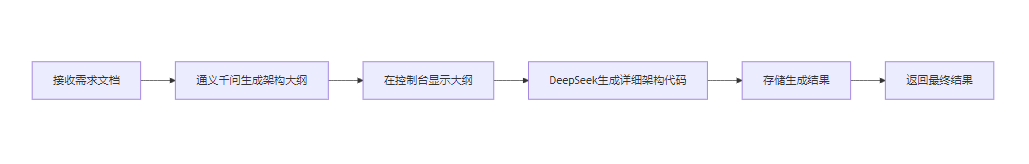
5. 通义千问生成部署大纲并显示在控制台

6. DeepSeek生成详细部署脚本和配置文件

7. 显示生成结果

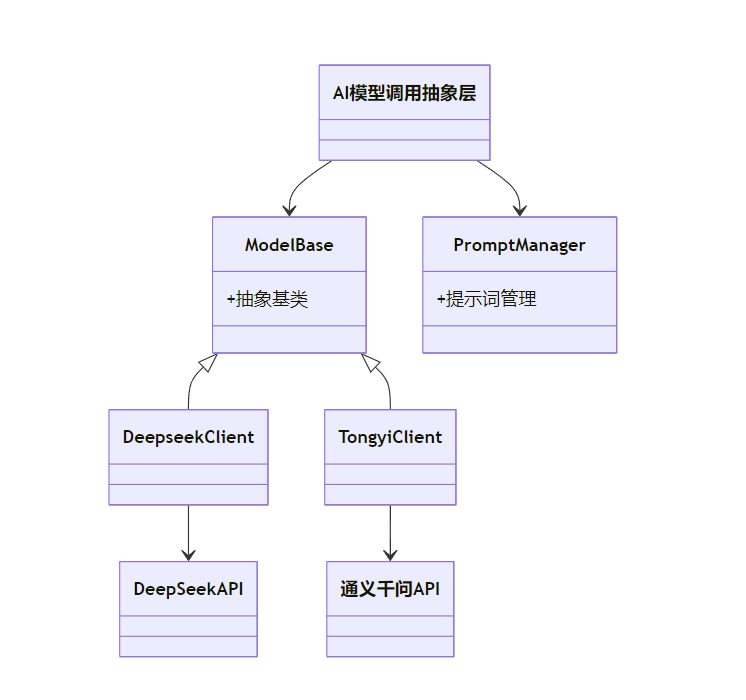
## 4. 详细设计

### 4.1 架构代码生成流程



## 5. 大模型调用设计

### 5.1 大模型集成架构



### 5.2 大模型功能分配

|  |  |  |  |
| --- | --- | --- | --- |
| 功能 | 大纲生成模型 | 内容生成模型 | 备注 |
| 架构代码生成 | 通义千问 | DeepSeek | 先生成大纲到控制台，再生成代码 |
| 数据库设计 | 通义千问 | DeepSeek | 先生成数据库大纲，再生成SQL脚本 |
| 模块代码生成 | 通义千问 | DeepSeek | 先生成模块设计大纲，再生成模块代码 |
| 测试用例生成 | 通义千问 | DeepSeek | 先生成测试大纲，再生成测试用例 |
| 部署步骤生成 | 通义千问 | DeepSeek | 先生成部署大纲，再生成部署步骤 |

### 5.3 提示词设计

#### 5.3.1 架构大纲生成提示词

请根据以下需求生成{language}语言的{architecture\_type}架构大纲: 需求描述: {requirement} 请生成详细的架构大纲，包括系统结构、核心模块、接口定义等，以便后续进行代码生成。

#### 5.3.2 数据库设计提示词

请分析以下需求，生成数据库设计大纲。大纲应包括： 1. 所需的数据库表 2. 每个表的主要字段 3. 表之间的关系 4. 索引设计建议 请输出JSON格式，确保格式正确可解析。 需求描述: {requirement}

## 6. 技术栈选择

### 6.1 前端技术栈

1. **框架**：React + TypeScript
2. **UI组件库**：Ant Design
3. **状态管理**：Redux Toolkit
4. **代码编辑器**：Monaco Editor
5. **HTTP客户端**：Axios

### 6.2 后端技术栈

1. **框架**：Flask/FastAPI
2. **ORM**：SQLAlchemy
3. **任务队列**：Celery
4. **认证**：JWT