知识中台项目需求规格说明书

[1 引言 2](#_Toc160801862)

[1.1 编写目的 2](#_Toc160801863)

[1.2 项目背景 2](#_Toc160801864)

[1.3 参考资料 2](#_Toc160801865)

[2 任务概述 3](#_Toc160801866)

[2.1 产品总体效果 3](#_Toc160801867)

[2.2 用户特征 3](#_Toc160801868)

[2.3 产品目标 3](#_Toc160801869)

[2.4 运行环境 3](#_Toc160801870)

[2.5 条件与限制 4](#_Toc160801871)

[3 需求详述 4](#_Toc160801872)

[3.1 功能性需求 4](#_Toc160801873)

[3.1.1 功能列表及优先级 4](#_Toc160801874)

[3.1.2 知识问答页面 5](#_Toc160801875)

[3.1.3 应用服务API 9](#_Toc160801876)

[3.1.4 插件管理 16](#_Toc160801877)

[3.1.5 函数管理 31](#_Toc160801878)

[3.2 非功能性需求 41](#_Toc160801879)

[3.2.1 性能需求 41](#_Toc160801880)

[3.2.2 适应性 42](#_Toc160801881)

[3.2.3 运行需求 42](#_Toc160801882)

[4 业务模型 42](#_Toc160801883)

[4.1 静态模型 42](#_Toc160801884)

[4.1.1 类图 42](#_Toc160801885)

[4.1.2 数据库图 43](#_Toc160801886)

[4.2 动态模型 43](#_Toc160801887)

[4.2.1 用例图 43](#_Toc160801888)

[4.2.2 业务流程图 43](#_Toc160801889)

[5 专业词汇表 44](#_Toc160801890)

[6 不确定的需求 46](#_Toc160801891)

[6.1 可使用性 46](#_Toc160801892)

[6.1.1 用户界面友好性 46](#_Toc160801893)

[6.1.2 响应时间 46](#_Toc160801894)

[6.2 安全保密 46](#_Toc160801895)

[6.2.1 数据加密 46](#_Toc160801896)

[6.2.2 用户身份验证 46](#_Toc160801897)

[6.3 可维护性 46](#_Toc160801898)

[6.3.1 代码注释与文档 46](#_Toc160801899)

[6.3.2 模块化设计 47](#_Toc160801900)

[6.4 可移植性 47](#_Toc160801901)

[6.4.1 跨平台兼容性 47](#_Toc160801902)

[6.4.2 外部服务集成 47](#_Toc160801903)

# 引言

## 编写目的

本需求规格说明书旨在为设计和实现基于大语言模型的智能问答系统提供清晰的指导规划：提供明确软件需求、安排项目规划与进度、组织开发与测试。该系统旨在从传统的顾问型问答系统转变为助理型问答系统，以更好的满足用户需求。本文档供项目经理、设计人员、开发人员和测试人员参考。

## 项目背景

1. 项目委托单位：众数科技股份有限公司
2. 项目开发单位：厦门大学
3. 主管部门：厦门大学
4. 软件系统介绍：本项目旨在通过引入大语言模型的先进技术，构建一个更智能、更全面的问答系统。该系统将利用大语言模型的Function Call功能，循环调用各种API、函数及插件，以提供更丰富、更准确的答案和解决方案。通过这种方式，系统将更好地适应用户的需求，实现顾问型问答系统向助理型问答系统的转变，提升用户体验和系统的实用性。

## 参考资料

1. “文心一言”官方文档
2. Spring Boot官方文档
3. Axios官方文档
4. Vue3官方文档
5. MySQL官方文档
6. Java编程思想（第四版）
7. 软件项目管理

# 任务概述

## 产品总体效果

智能问答系统旨在通过引入大语言模型的先进技术，实现从传统的顾问型问答系统向助理型问答系统的转变。系统将利用大语言模型的Function Call功能，循环调用各种API、函数及插件，以提供更丰富、更准确的答案和解决方案。通过这种方式，系统将更好地适应用户的需求，提升用户体验和系统的实用性。

## 用户特征

智能问答系统的用户主要分为两类：普通用户和管理员。

普通用户通过系统进行提问、获取答案，而管理员则具有对系统进行配置、管理用户权限以及监控系统运行状态的特殊权限

## 产品目标

1. 提高问答准确性： 通过大语言模型的功能，系统将以更智能的方式解析用户问题，提供更准确、全面的答案。
2. 提升用户体验： 助力用户在获取信息的过程中更加高效、便捷，通过友好的界面和智能化的交互方式提升用户满意度。
3. 系统可配置性： 管理员能够通过系统配置界面调整系统参数，灵活配置系统行为，以适应不同应用场景。

## 运行环境

操作系统：Microsoft Windows 2000 Advanced Server

支持环境：IIS 5.0

数 据 库：Microsoft SQL Server 2000

## 条件与限制

1. 语言模型训练时间： 训练大语言模型需要充足的时间和计算资源，系统上线前需要确保语言模型已经充分训练。
2. 第三方插件稳定性： 对于调用第三方API、函数及插件的稳定性和兼容性进行充分测试，确保系统的可靠性。
3. 系统安全性： 系统需要具备严格的安全机制，包括用户身份验证、数据加密等，以保障用户隐私和系统的稳定性。

# 需求详述

## 功能性需求

### 功能列表及优先级

|  |  |  |
| --- | --- | --- |
| 功能模块 | 具体功能点 | 优先级 |
| 知识问答 | 新建对话 | 高 |
| 编辑对话名 | 中 |
| 删除对话 | 中 |
| 展示名片 | 中 |
| 选择插件 | 高 |
| 展示推荐问题 | 中 |
| 输入问题并发送 | 高 |
| 查看用户协议 | 低 |
| 查看隐私声明 | 低 |
| 应用服务API管理 | 通过api名字查询api | 中 |
| 新增api | 高 |
| 新增api入参字段定义 | 高 |
| 新增api出参字段定义 | 高 |
| 查看api详情 | 中 |
| 编辑api | 高 |
| 编辑api入参字段 | 高 |
| 编辑api出参字段 | 高 |
| 删除api入参字段 | 低 |
| 删除api出参字段 | 低 |
| 删除api | 低 |
| 插件管理 | 创建插件 | 高 |
| 添加插件参数 | 高 |
| 编辑插件参数 | 高 |
| 删除编辑参数 | 中 |
| 添加用户参数 | 高 |
| 编辑用户参数 | 高 |
| 删除用户参数 | 中 |
| 函数管理 | 查询函数 | 中 |
| 新建函数 | 高 |
| 编辑函数 | 高 |
| 配置函数 | 高 |
| 删除函数 | 中 |
| 展示函数 | 低 |

### 知识问答页面

1. 新增对话

|  |  |
| --- | --- |
| 需求名称 | 新增对话 |
| 业务流程 | 用户新增对话 |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素： * 新增按钮 * 交互逻辑： * 用户点击新增按钮后，显示新的对话页面 |
| 数据说明 | * 对话记录名称   + 数据来源：给出默认名称 * 刷新条件：用户点击修改名称后 |
| 异常场景 | * 服务异常：用户网络连接不稳定，无法顺利访问系统 |

1. 管理对话记录

|  |  |
| --- | --- |
| 需求名称 | 管理对话记录 |
| 业务流程 | 用户修改对话记录名称和删除对话记录 |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素： * 修改按钮 * 删除按钮 * 交互逻辑： * 用户点击修改按钮后，显示对话记录名称输入框 * 用户点击删除按钮后，系统给出删除提示 |
| 数据说明 | * 对话记录名称   + 数据来源：根据对话 id查询 * 刷新条件：用户点击修改/删除按钮后 |
| 异常场景 | * 服务异常：用户网络连接不稳定，无法顺利访问系统 |

1. 对话交互

|  |  |
| --- | --- |
| 需求名称 | 对话交互 |
| 业务流程 | 用户与问答系统进行对话 |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素： * 发送按钮 * 输入框 * 插件选择按钮 * 交互逻辑： * 用户点击发送按钮后，系统对输入框内容进行合法性检查并给出提示 * 用户可以选择不同插件 * 用户可以在输入框输入对话内容 |
| 数据说明 | * 对话记录   + 数据来源：用户输入与模型回答 * 刷新条件：用户点击发送按钮后 |
| 异常场景 | * 服务异常：用户网络连接不稳定，无法顺利访问系统 |

### 应用服务API

1. 创建应用服务API

|  |  |
| --- | --- |
| 需求名称 | 创建应用服务API |
| 业务流程 | 用户创建应用服务API |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素： * 新增按钮 * 创建应用服务API表单 * 交互逻辑： * 用户点击新增按钮后，显示创建应用服务API表单页面 * 用户向表单输入数据 * 用户点击确定按钮，系统对输入数据合法性进行验证并给出提示 * 用户点击取消按钮，关闭表单 |
| 数据说明 | * 表单数据   + 数据来源：用户输入 * 刷新条件：用户点击确定/取消按钮后 |
| 异常场景 | * 用户输入的问题存在错误：系统对表单数据进行合法性检验，并给出提示 * 服务异常：用户网络连接不稳定，无法顺利访问系统 |

1. 编辑应用服务API

|  |  |
| --- | --- |
| 需求名称 | 应用服务API编辑 |
| 业务流程 | 用户编辑应用服务API |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素：   + 编辑按钮   + 编辑应用服务API表单 * 交互逻辑：   + 用户点击编辑按钮后，显示创建应用服务API表单页面及数据   + 用户修改表单数据   + 用户点击确定按钮，系统对输入数据合法性进行验证并给出提示   + 用户点击取消按钮，关闭表单 |
| 数据说明 | * 表单原始数据 * 数据来源：根据应用服务API id返回 * 刷新条件：用户点击确认按钮后 * 表单修改数据 * 数据来源：用户输入 * 刷新条件：用户点击确认/取消按钮后 |
| 异常场景 | * 用户输入的问题存在错误：系统对表单数据进行合法性检验，并给出提示 * 服务异常：用户网络连接不稳定，无法顺利访问系统 |

1. 查看应用服务API详情

|  |  |
| --- | --- |
| 需求名称 | 查看应用服务API详情 |
| 业务流程 | 用户查看应用服务API详情 |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素： * 详情按钮 * 应用服务API表单详情 * 交互逻辑： * 用户点击详情按钮后，显示创建应用服务API表单页面及数据 * 用户点击确定/取消按钮，关闭表单 |
| 数据说明 | * 表单数据 * 数据来源：根据应用服务API id返回 * 刷新条件：用户修改数据并保存后 |
| 异常场景 | * 服务异常：用户网络连接不稳定，无法顺利访问系统 |

1. 删除应用服务API

|  |  |
| --- | --- |
| 需求名称 | 删除应用服务API |
| 业务流程 | 用户删除应用服务API详情 |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素： * 删除按钮 * 提示窗口 * 交互逻辑： * 用户点击删除按钮后，显示删除提示窗口 * 用户点击取消按钮，关闭提示 * 用户点击确定按钮，删除对应API |
| 数据说明 | * 删除数据 * 数据来源：根据应用服务API id返回 * 刷新条件：用户点击确定按钮后 |
| 异常场景 | * 服务异常：用户网络连接不稳定，无法顺利访问系统 * 不存在该API：系统无法删除该API，并提示相关情况 |

1. 查询应用服务API

|  |  |
| --- | --- |
| 需求名称 | 查询应用服务API |
| 业务流程 | 用户查询应用服务API |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素： * 查询按钮 * 查询输入框 * 交互逻辑： * 用户输入API名称 * 用户点击查询按钮，系统进行输入合法性检验并反馈给用户 * 页面显示查询结果 |
| 数据说明 | * API名称 * 数据来源：用户输入 * 刷新条件：用户点击搜索/清空按钮 * 查询结果 * 数据来源：根据API名称返回 * 刷新条件：用户点击搜索按钮且搜索框内容合法 |
| 异常场景 | * 不存在该API：系统无法查询该API，并提示相关情况 * 服务异常：用户网络连接不稳定，无法顺利访问系统。 |

### 插件管理

1. 创建插件

|  |  |
| --- | --- |
| 需求名称 | 创建插件 |
| 业务流程 | 点击侧边栏插件管理，再点击创建插件，按照填写规则填写内容 |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素 * 关闭表单按钮 * 确认提交表单按钮 * 新增插件标题 * 需要填写的插件数据 * 展示已配置的插件参数 * 展示已配置的用户参数 * 添加插件配置参数按钮 * 添加用户参数按钮 * 是否可用选择 * 是否公开选择 * 交互逻辑 * 点击关闭，回到插件管理页面 * 点击确认，提交表单，回到插件管理页面 * 点击插件参数配置板块添加参数按钮，弹出添加插件参数表单 * 点击用户参数板块添加参数按钮，弹出添加用户参数表单 * 点击是否可用，修改该插件的可用性 * 点击是否公开，修改该插件的可见性 |
| 数据说明 | * 已配置的插件参数 * 数据来源：根据插件名称查询查询属于该插件的插件参数 * 刷新条件：添加参数提交表单后或者再次点击添加插件时 * 已定义的用户参数 * 数据来源：根据插件名称查询查询属于该插件的用户参数 * 刷新条件：添加参数提交表单后或者再次点击添加插件时 |
| 异常场景 | * 网络异常 * 请求超时 |

1. 添加插件参数

|  |  |
| --- | --- |
| 需求名称 | 添加插件参数 |
| 业务流程 | 点击侧边栏插件管理，再点击创建插件，点击插件参数配置版块的添加参数按钮，进入添加插件参数配置 |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素 * 关闭表单按钮 * 确定提交表单按钮 * 参数名和参数值输入框 * 交互逻辑 * 点击关闭，回到新增插件页面 * 点击确定，提交表单，回到新增插件页面 |
| 数据说明 | * 插件参数 * 数据来源：用户填写 |
| 异常场景 | * 网络异常 * 请求超时 |

1. 编辑插件参数

|  |  |
| --- | --- |
| 需求名称 | 编辑插件参数 |
| 业务流程 | 点击侧边栏插件管理，再点击创建插件，点击插件参数配置版块的任意插件参数右侧的编辑按钮，进入编辑插件参数页面 |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素 * 关闭表单按钮 * 确定提交表单按钮 * 参数名和参数值输入框 * 原参数名和参数输入值 * 交互逻辑 * 点击关闭，回到新增插件页面 * 点击确定，提交表单，修改原插件参数，回到新增插件页面 |
| 数据说明 | * 插件参数 * 数据来源：用户改写 |
| 异常场景 | * 网络异常 * 请求超时 * 用户未修改任何参数 |

1. 删除插件参数

|  |  |
| --- | --- |
| 需求名称 | 删除插件参数 |
| 业务流程 | 点击侧边栏插件管理，再点击创建插件，点击插件参数配置版块的删除参数按钮，弹出是否确认删除框 |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素 * 关闭确认框按钮 * 确定删除按钮 * 交互逻辑 * 点击关闭，回到新增插件页面 * 点击确定，确认删除，回到新增插件页面 |
| 数据说明 | * 插件参数：由后端获取所有插件参数展示在前端页面 |
| 异常场景 | * 网络异常 * 请求超时 |

1. 添加用户参数

|  |  |
| --- | --- |
| 需求名称 | 添加用户参数 |
| 业务流程 | 点击侧边栏插件管理，再点击创建插件，点击用户参数定义版块的添加参数按钮，进入添加用户参数定义 |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素 * 关闭表单按钮 * 确定提交表单按钮 * 用户参数名和用户参数值输入框 * 交互逻辑 * 点击关闭，回到新增插件页面 * 点击确定，提交表单，回到新增插件页面 |
| 数据说明 | * 用户参数 * 数据来源：用户填写 |
| 异常场景 | * 网络异常 * 请求超时 |

1. 编辑用户参数

|  |  |
| --- | --- |
| 需求名称 | 编辑用户参数 |
| 业务流程 | 点击侧边栏插件管理，再点击创建插件，点击用户参数定义版块的编辑参数按钮，进入编辑用户参数定义 |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素 * 关闭表单按钮 * 确定提交表单按钮 * 参数名和参数值输入框 * 原参数名和参数输入值 * 交互逻辑 * 点击关闭，回到新增插件页面 * 点击确定，提交修改后的表单，回到新增插件页面 |
| 数据说明 | * 用户参数 * 数据来源：用户改写 |
| 异常场景 | * 网络异常 * 请求超时 * 用户未修改任何参数 |

1. 删除用户参数定义

|  |  |
| --- | --- |
| 需求名称 | 删除用户参数定义 |
| 业务流程 | 点击侧边栏插件管理，再点击创建插件，点击用户参数定义版块的删除参数按钮，弹出是否确认删除框 |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素 * 关闭确认框按钮 * 确定删除按钮 * 交互逻辑 * 点击关闭，回到新增插件页面 * 点击确定，确认删除，回到新增插件页面 |
| 数据说明 | * 用户参数定义：由后端获取所有用户参数定义展示在前端页面 |
| 异常场景 | * 网络异常 * 请求超时 |

### 函数管理

1. 查询函数

|  |  |
| --- | --- |
| 需求名称 | 查询函数 |
| 业务流程 | 点击侧边栏插件管理，再点击查询函数，由函数名获取函数 |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素 * 查询按钮 * 函数名称输入框 * 查询到的结果 * 交互逻辑 * 输入要查询的函数名字，点击查询，获取符合该函数名称的函数 |
| 数据说明 | * 函数 * 数据来源：由前端获取用户输入的函数名字传递到后端，通过函数名字查询出函数并展示函数详情 |
| 异常场景 | * 网络异常 * 请求超时 * 未找到符合该名字的函数 |

1. 创建函数

|  |  |
| --- | --- |
| 需求名称 | 创建函数 |
| 业务流程 | 点击侧边栏插件管理，再点击新建函数，按照填写规则填写内容 |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素 * 关闭表单按钮 * 确认提交表单按钮 * 新增函数标题 * 需要填写的函数数据 * 交互逻辑 * 点击关闭，回到插件管理页面 * 点击确认，提交表单，回到插件管理页面 |
| 数据说明 | * 函数参数 * 数据来源：用户输入 |
| 异常场景 | * 网络异常 * 请求超时 |

1. 编辑函数

|  |  |
| --- | --- |
| 需求名称 | 编辑函数 |
| 业务流程 | 点击侧边栏插件管理，再点击要编辑的函数右边的编辑按钮，进入编辑页面 |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素 * 关闭表单按钮 * 确认提交表单按钮 * 编辑函数标题 * 需要修改的函数数据 * 交互逻辑 * 点击关闭，回到插件管理页面 * 点击确认，提交表单，回到插件管理页面 * 点击输入框可修改函数原数据 |
| 数据说明 | * 函数参数 * 数据来源：用户改写 |
| 异常场景 | * 网络异常 * 请求超时 * 用户并未修改函数数据 * 用户修改的数据不合规 |

1. 配置函数

|  |  |
| --- | --- |
| 需求名称 | 配置函数 |
| 业务流程 | 点击侧边栏插件管理，再点击要配置的函数右边的配置按钮，进入配置页面 |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素 * 关闭表单按钮 * 确认提交表单按钮 * 配置函数标题 * 配置函数进度条 * 需要配置的函数的各类输入框 * 交互逻辑 * 点击关闭，回到插件管理页面 * 点击确认，提交表单，回到插件管理页面 * 点击输入框可配置函数参数 * 点击上一步或下一步修改配置进度 |
| 数据说明 | * 函数参数 * 数据来源：用户改写 |
| 异常场景 | * 网络异常 * 请求超时 * 用户并未配置函数数据 * 用户配置的数据不合规 |

1. 删除函数

|  |  |
| --- | --- |
| 需求名称 | 删除函数 |
| 业务流程 | 点击侧边栏插件管理，再点击要删除的函数右边的删除按钮，弹出删除确认框 |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素 * 取消删除按钮 * 确认删除按钮 * 交互逻辑 * 点击关闭，回到插件管理页面 * 点击确认，删除该函数 |
| 数据说明 | * 函数参数 * 数据来源：用户改写 |
| 异常场景 | * 网络异常 * 请求超时 |

1. 展示函数

|  |  |
| --- | --- |
| 需求名称 | 展示函数 |
| 业务流程 | 点击侧边栏插件管理，在右侧显示所有已添加的函数 |
| 产品原型图 |  |
| 界面交互说明 | * 页面展示要素 * 所有函数 * 交互逻辑 * 点击插件管理进入该页面 |
| 数据说明 | * 函数参数 * 数据来源：从后端获取所有函数，并展示函数部分详情 |
| 异常场景 | * 网络异常 * 请求超时 |

## 非功能性需求

### 性能需求

1. 响应时间： 系统应在用户提出问题后的数秒内生成准确的回答，响应时间不应超过3秒。
2. 更新处理时间： 当系统需要更新或重新加载模型、API、函数或插件时，更新处理时间不应超过10分钟，以确保系统能够及时响应用户需求。
3. 数据转换与传输时间： 数据转换与传输时间应尽可能短暂，确保系统能够高效地获取和处理数据。

### 适应性

1. 操作方式：系统应该提供多种操作方式，包括文本输入、图片输入等，以满足不同用户的需求。
2. 运行环境： 系统应能够适应不同的运行环境，包括各种操作系统（如Windows、MacOS等）和不同的硬件平台。
3. 与其他软件的接口： 系统应提供良好的接口和文档，以便与其他软件进行集成和交互，实现更丰富的功能和服务。
4. 开发计划变化： 当开发计划发生变化时，系统应具备灵活性和可扩展性，能够快速适应和调整，确保项目进度和质量不受影响。

### 运行需求

1. 稳定性： 系统应具有高度稳定性，能够在长时间运行中保持良好的性能和可靠性。
2. 安全性： 系统应具备安全机制，保护用户数据和隐私，防范各类网络攻击和恶意行为。
3. 易用性： 系统应具有良好的用户界面和交互设计，易于操作和理解，用户无需专业培训即可轻松使用。
4. 可维护性： 系统应易于维护和管理，具有清晰的代码结构和文档说明，方便开发人员进行代码修改和更新。
5. 可扩展性： 系统应具备良好的可扩展性，能够根据需求不断添加新功能和模块，以满足用户不断增长的需求。
6. 性能优化： 系统应定期进行性能优化和调整，以确保系统能够高效地运行并处理大量用户请求。

# 业务模型

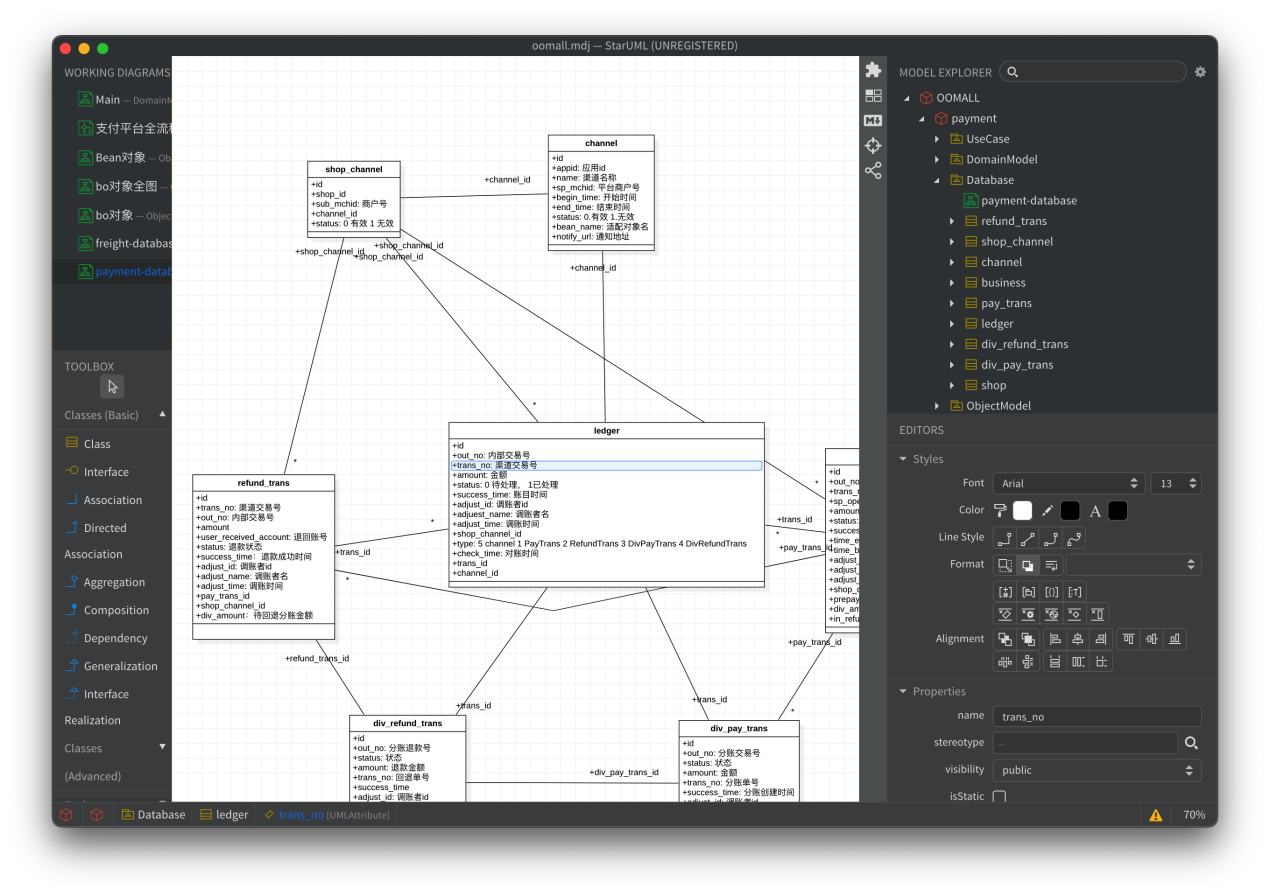
## 静态模型

### 类图

[类图画法](https://blog.csdn.net/weixin_42445727/article/details/125239807?ops_request_misc=&request_id=&biz_id=102&utm_term=类图&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-125239807.nonecase&spm=1018.2226.3001.4187)

### 数据库图

示例



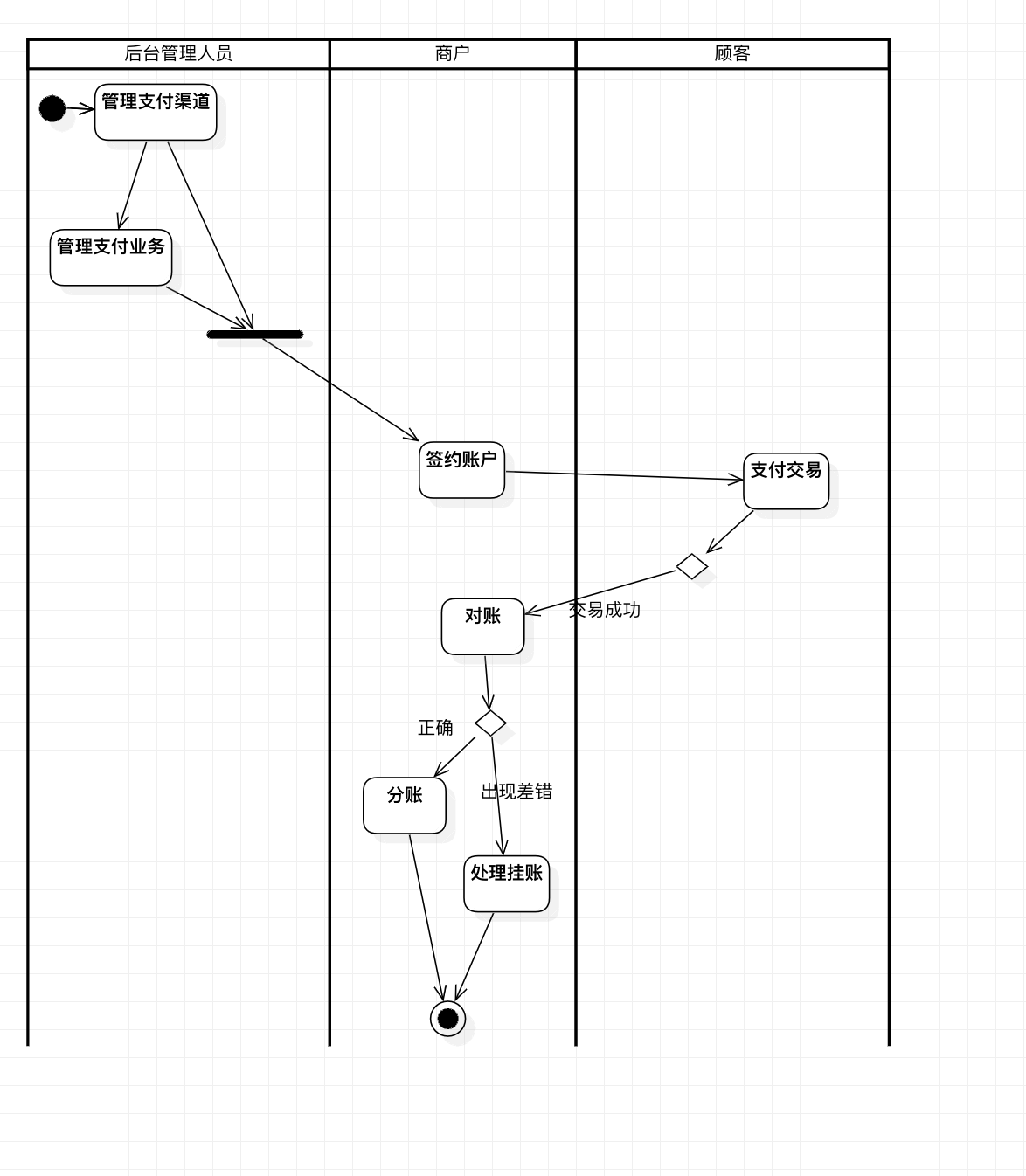
## 动态模型

### 用例图

分为用例总图 和各模块用例图

### 业务流程图

示例



# 专业词汇表

|  |  |
| --- | --- |
| 专业词汇 | 解释 |
| FC | 指定义一些function（需要指定函数名，函数用途的描述，参数名，参数描述），传给LLM，当用户输入一个问题时，LLM通过文本分析是否需要调用某一个function，如果需要调用，那么LLM返回一个json，json包括需要调用的function名，需要输入到function的参数名，以及参数值。function call做了两件事情：1.判断是否要调用某个预定义的函数。2.如果要调用，从用户输入的文本里提取出函数所需要的函数值 |
| ChatGPT | ChatGPT是由OpenAI开发的语言生成模型，是GPT（Generative Pre-trained Transformer）系列的一部分。它专注于自然语言对话生成，具有强大的文本生成和理解能力 |
| 文心一言 | 文心一言（英文名：ERNIE Bot）是[百度](file:////item/百度/6699%3ffromModule=lemma_inlink)全新一代知识增强[大语言模型](file:////item/大语言模型/62884793%3ffromModule=lemma_inlink)，[文心大模型](file:////item/文心大模型/61806144%3ffromModule=lemma_inlink)家族的新成员，能够与人对话互动、回答问题、协助创作，高效便捷地帮助人们获取[信息](file:////item/信息/111163%3ffromModule=lemma_inlink)、[知识](file:////item/知识/74245%3ffromModule=lemma_inlink)和[灵感](file:////item/灵感/18583%3ffromModule=lemma_inlink)。文心一言从数万亿数据和数千亿知识中融合学习，得到预训练大模型，在此基础上采用有监督精调、人类反馈强化学习、提示等技术，具备知识增强、检索增强和对话增强的技术优势。 |
| CthatGLM3 | ChatGLM3是智谱AI与清华大学KEG实验室联合发布的[新一代](https://so.csdn.net/so/search?q=%E6%96%B0%E4%B8%80%E4%BB%A3&spm=1001.2101.3001.7020)对话预训练模型。 |
| API | API是应用程序接口的缩写，是一组定义软件组件如何互相交互的协议。API定义了不同软件组件之间的通信规范，使它们能够有效地集成和协作 |
| 应用服务API | 一组定义，允许大型模型与外部应用程序或服务进行交互的接口。这些接口定义了模型与应用服务之间的通信协议和数据格式。通过调用应用服务API，大型模型可以获取外部服务提供的功能或信息，实现更丰富、全面的功能。这种通信通常涉及请求和响应，模型向应用服务发出请求，服务执行相应的操作并返回结果 |
| 插件 | 能完成除了对话以外的扩展任务，如文生图 |
| 函数 | 模型能够调用各种函数以完成特定任务，如查询实时天气 |

# 不确定的需求

## 可使用性

### 用户界面友好性

系统应具有直观且易用的用户界面，以确保用户能够轻松理解和操作系统的各项功能。

### 响应时间

系统应具备快速的响应时间，确保用户在进行查询和交互时能够获得即时的反馈，提升用户体验。

## 安全保密

### 数据加密

系统应实施强大的数据加密机制，确保用户的敏感信息在传输和存储过程中得到保护。

### 用户身份验证

系统应采用可靠的用户身份验证机制，防止未经授权的访问，保障系统的安全性。

## 可维护性

### 代码注释与文档

系统代码应具备清晰的注释和文档，以便未来的开发人员能够轻松理解和维护系统。

### 模块化设计

系统应采用模块化的设计，使得各个功能模块相对独立，便于单独更新和维护。

## 可移植性

### 跨平台兼容性

系统应在不同操作系统和环境中具备良好的兼容性，以确保在各种场景下能够稳定运行。

### 外部服务集成

系统需要考虑外部服务API的变化，确保与这些服务的集成是灵活、可维护的，以适应未来可能的变化。