小智智能生活助理平台

系统设计说明书

小智智能生活助理平台 • 概要设计（GB/T 8567-2006 风格）

文档名称	智行小组_系统设计说明书
项目名称	小智智能生活助理平台
团队名称	智行小组
文档版本	V1.0
编制日期	2026-11-27
文档状态	评审稿

文档修订记录

版本	修订日期	修订人	修订说明
V1.0	2026-11-27	本人（智行小组）	完成概要设计初稿

目 录

文档修订记录 2

1 引言 4

1.1 编写目的 4

1.2 项目背景 4

1.3 术语与定义 4

1.4 参考资料 4

2 系统总体设计 5

2.1 设计目标与设计原则 5

2.2 技术选型 5

2.3 运行环境 5

2.4 系统总体结构 6

2.5 系统功能结构 6

3 详细设计 8

3.1 用例设计 8

3.2 业务流程设计 10

3.3 系统结构设计（类图） 12

3.4 关键流程时序设计 13

3.5 用户接口设计 14

3.6 外部接口设计 15

3.7 内部接口设计 15

4 出错处理设计 16

4.1 错误码约定 16

4.2 处理策略 16

5 安全保密设计 17

6 部署与维护设计 17

7 性能设计 18

8 数据设计概述 18

1 引言

1.1 编写目的

本文档依据《软件设计说明书》国标规范（GB/T 8567-2006）的框架编写，给出小智智能生活助理平台（以下简称“本系统”或“小智平台”）的概要设计，明确系统总体架构、功能模块划分、主要业务流程、UML 模型、接口设计与部署方案，作为后续详细设计、编码实现与测试验收的依据。本文档的读者为课程老师、助教、评审人员及项目组成员。

1.2 项目背景

AI 智能体（Agent）已成为人工智能应用创新的核心方向。本小组在第一次团队作业中完成了“小智”AI 生活助理智能体的原型验证：它既能与用户自然语言对话（会说），又能调用工具完成查天气、管理待办、生成出行计划、记笔记等实际操作（会做），并采用 MCP（Model Context Protocol）实现工具与智能体的解耦。

在此基础上，本系统将其扩展为一个完整的多用户 Web 平台：前端提供用户应用与管理后台，后端采用 FastAPI 服务化智能体能力，数据持久化到 MySQL，形成可演示、可扩展的智能体应用系统。

1.3 术语与定义

缩写 / 术语	全称	说明
AI	Artificial Intelligence	人工智能
LLM	Large Language Model	大语言模型，本系统使用 OpenAI 提供的模型（如 gpt-4o-mini）
MCP	Model Context Protocol	模型上下文协议，用于标准化地暴露与调用工具
API	Application Programming Interface	应用程序编程接口
UML	Unified Modeling Language	统一建模语言，本文档采用用例图、活动图、类图、时序图、协作图
ER	Entity-Relationship	实体关系模型
ORM	Object Relational Mapping	对象关系映射，本系统使用 SQLAlchemy
CRUD	Create/Read/Update/Delete	增删改查
RBAC	Role-Based Access Control	基于角色的访问控制
JWT	JSON Web Token	无状态身份认证令牌
SSE	Server-Sent Events	服务器推送事件，用于流式对话输出
SPA	Single Page Application	单页应用，本系统前端采用 Vue3 实现
DAU	Daily Active Users	日活跃用户数

1.4 参考资料

- GB/T 8567-2006《计算机软件文档编制规范》

- Model Context Protocol 官方规范（https://modelcontextprotocol.io）
- OpenAI API 文档（Chat Completions、Function Calling）
- Vue 3 官方文档（https://vuejs.org）
- FastAPI 官方文档（https://fastapi.tiangolo.com）
- MySQL 8.0 Reference Manual

2 系统总体设计

2.1 设计目标与设计原则

- 能用：核心价值是“能说会做”——自然语言交互 + 工具调用（MCP），第一次作业已验证的智能体能力全部保留；
- 分层：严格采用分层架构（接入层 / 表现层 / 应用服务层 / 智能体核心层 / 基础设施层），层与层之间单向依赖；
- 解耦：工具定义与工具使用解耦，工具既可进程内直调，也可经 MCP 服务器调用；
- 可扩展：新增一个工具只需在注册表追加一条配置，无需改动智能体核心代码；
- 可演示：无 API Key 时系统提供离线意图识别兜底，保证演示零成本；
- 安全：统一认证（JWT）、后台 RBAC、参数校验、敏感信息（密钥）不入库不入仓。

2.2 技术选型

层次	选型	说明
前端	Vue 3 + Element Plus + Vite	SPA 单页应用，前台与后台两套界面
后端	Python 3.12 + FastAPI	REST API + SSE 流式对话
智能体核心	自研 AgentCore（第一次作业成果）	Function Calling 循环 + 离线意图识别
工具协议	MCP（官方 SDK 2.x）	工具服务器与客户端桥，stdio 传输
大模型	OpenAI gpt-4o-mini（可配置）	仅用于在线模式
数据库	MySQL 8.0	业务数据持久化
缓存	Redis 7	登录态缓存、限流计数
反向代理	Nginx	静态资源与负载均衡

2.3 运行环境

类别	项目	要求
服务器硬件	CPU / 内存 / 磁盘	4 核 / 8GB / 100GB SSD 及以上
服务器软件	操作系统	Ubuntu 22.04 或 Windows Server 2019+
运行依赖	Python	3.10 及以上
运行依赖	数据库	MySQL 8.0 / Redis 7
客户端	浏览器	Chrome / Edge 等现代浏览器

类别	项目	要求
外部服务	网络	可访问 OpenAI API 与 wttr.in（离线模式可不依赖）

2.4 系统总体结构

本系统自顶向下分为五层。上层只依赖紧邻的下层，跨层调用通过约定接口进行。

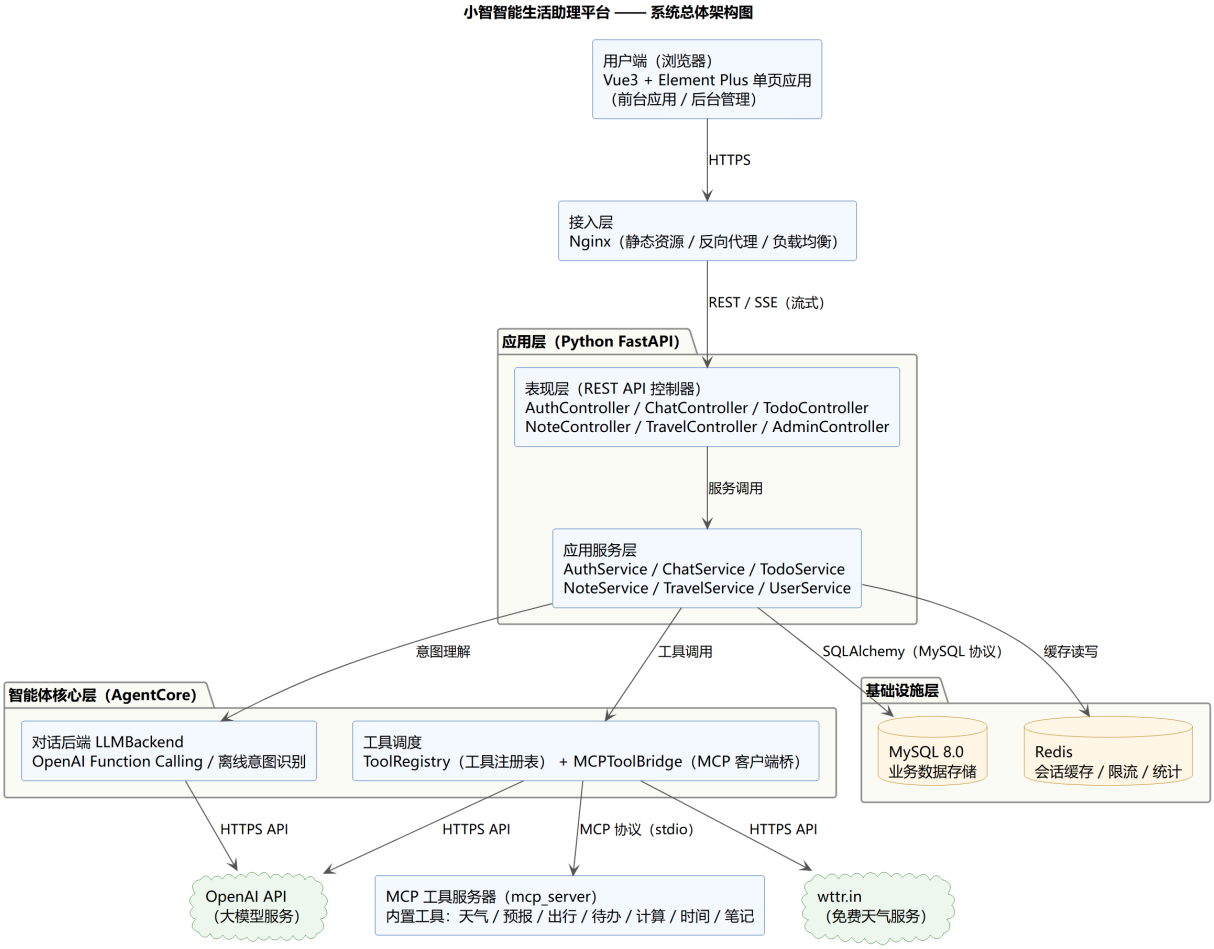


图 2-1 系统总体架构图

- 接入层（Nginx）：承担 HTTPS 终止、静态资源分发与反向代理，将 REST/SSE 请求转发到应用层；
- 表现层（REST API 控制器）：对前端暴露统一 API，负责参数校验、鉴权与响应组装；
- 应用服务层：实现业务用例（聊天、待办、笔记、出行、认证），编排智能体核心与数据访问；
- 智能体核心层（AgentCore）：意图理解（大模型 Function Calling / 离线识别）+ 工具调度（ToolRegistry + MCP 客户端桥），是系统“能说会做”的中枢；
- 基础设施层：MySQL 持久化业务数据，Redis 提供缓存与限流；外部依赖 OpenAI API（大模型）、wttr.in（天气数据）、MCP 工具服务器（工具执行）。

2.5 系统功能结构

系统功能分为前台应用（用户端）、后台管理（管理端）、智能体核心层与公共支撑四部分，思维导图如下。

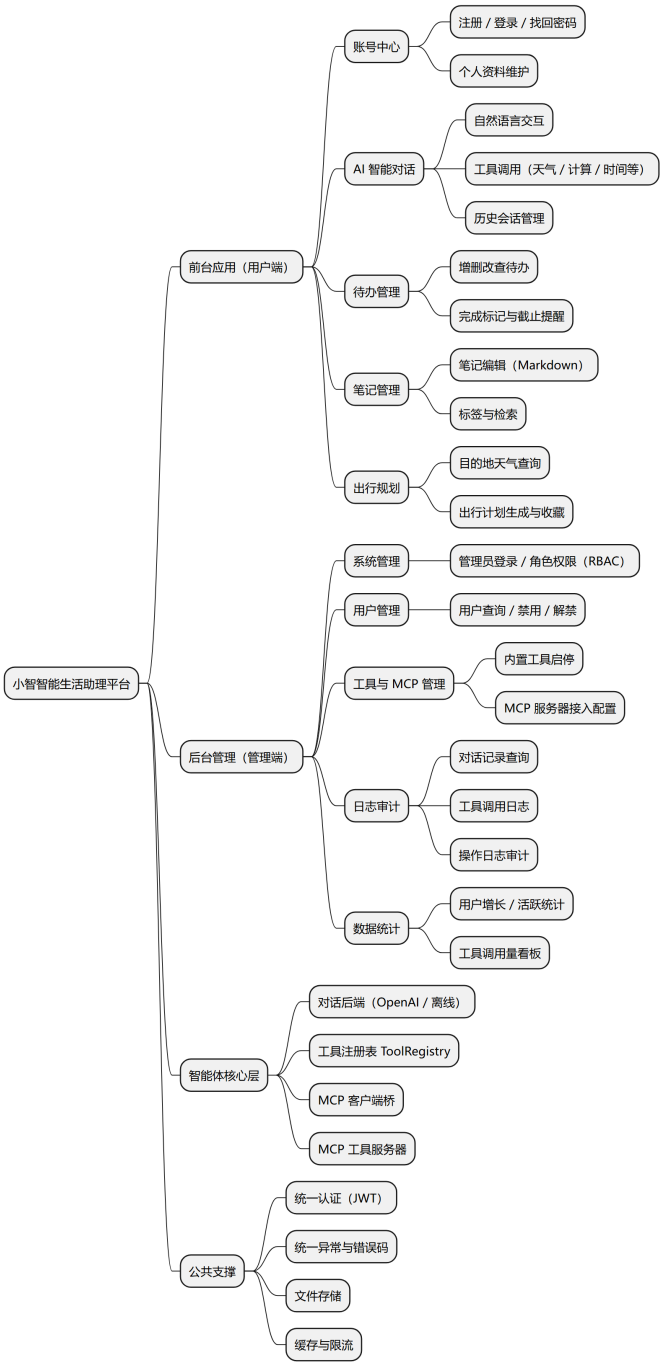


图 2-2 系统功能模块图（思维导图）

模块编号	模块名称	职责	所属子系统
M01	账号中心	注册、登录、找回密码、个人资料维护	前台应用
M02	AI 智能对话	自然语言交互、工具调用、历史会话管理	前台应用
M03	待办管理	待办增删改查、完成标记、截止提醒	前台应用
M04	笔记管理	Markdown 笔记编辑、标签与检索	前台应用
M05	出行规划	天气查询、出行计划生成与收藏	前台应用
M06	管理员登录	后台入口与 RBAC 角色校验	后台管理

模块编号	模块名称	职责	所属子系统
M07	用户管理	用户查询、启用/禁用、导出	后台管理
M08	工具与 MCP 管理	内置工具启停、MCP 服务器接入配置	后台管理
M09	日志审计	对话记录、工具调用日志、操作日志查询	后台管理
M10	数据统计	用户增长、活跃、工具调用量看板	后台管理
M11	智能体核心	意图理解、工具注册与调度、MCP 桥接	智能体核心层
M12	公共支撑	JWT 认证、错误码、缓存与限流、文件存储	公共支撑

3 详细设计

3.1 用例设计

系统角色包括：游客、注册用户、管理员三类人机角色，以及 OpenAI 大模型服务、wttr.in 天气服务、MCP 工具服务器三个外部系统角色。

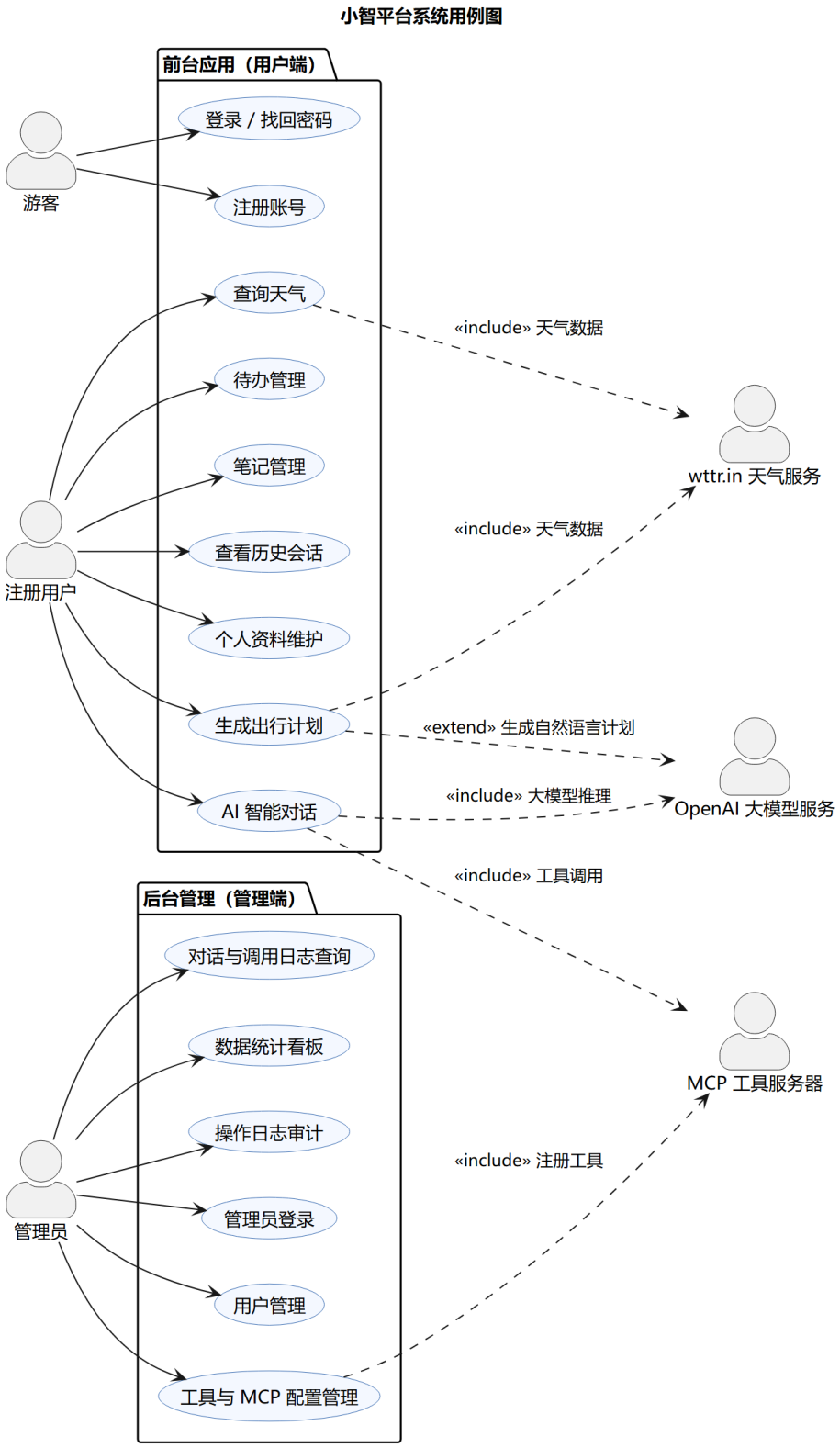


图 3-1 系统用例图

用例	参与者	前置条件	基本事件流
AI 智能对话	注册用户	已登录	输入自然语言 → 智能体理解意图 →（需要时）调用工具 → 整理自然语言回复 → 流式展示

用例	参与者	前置条件	基本事件流
查询天气	注册用户	已登录	在对话中说明城市 → 调用 get_weather 工具 → 返回天气摘要
待办管理	注册用户	已登录	新增/查看/完成待办，AI 对话亦可自动创建待办
工具配置管理	管理员	管理员登录	查看工具列表 → 启停工具 / 接入 MCP 服务器 → 保存配置
日志审计	管理员	管理员登录	按时间/用户/工具筛选 → 查看调用详情与结果

3.2 业务流程设计

图 3-2 给出 AI 智能对话的核心业务流程：系统在“意图理解 → 工具调用 → 结果回填 → 再理解”之间循环，最多 5 轮，最终生成自然语言回复。

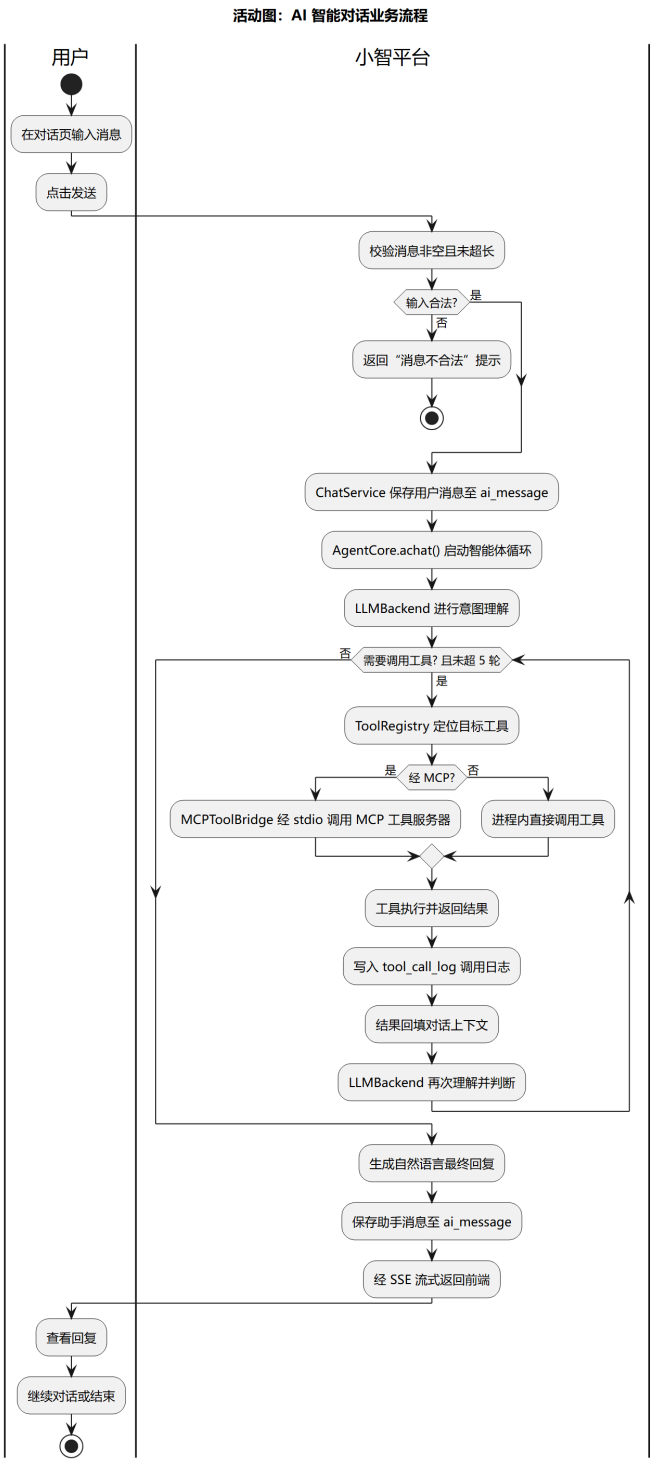


图 3-2 活动图：AI 智能对话业务流程

图 3-3 给出用户待办管理的业务流程，覆盖新增、完成标记、编辑/删除三个分支。

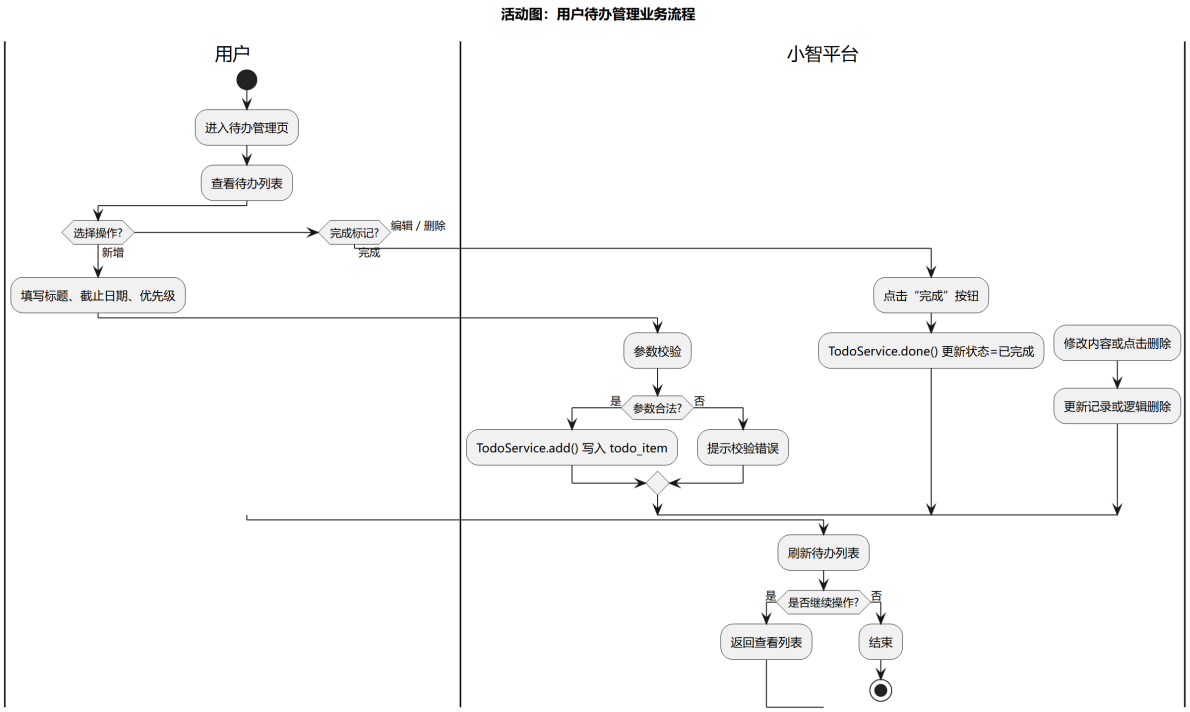


图 3-3 活动图：待办管理业务流程

3.3 系统结构设计（类图）

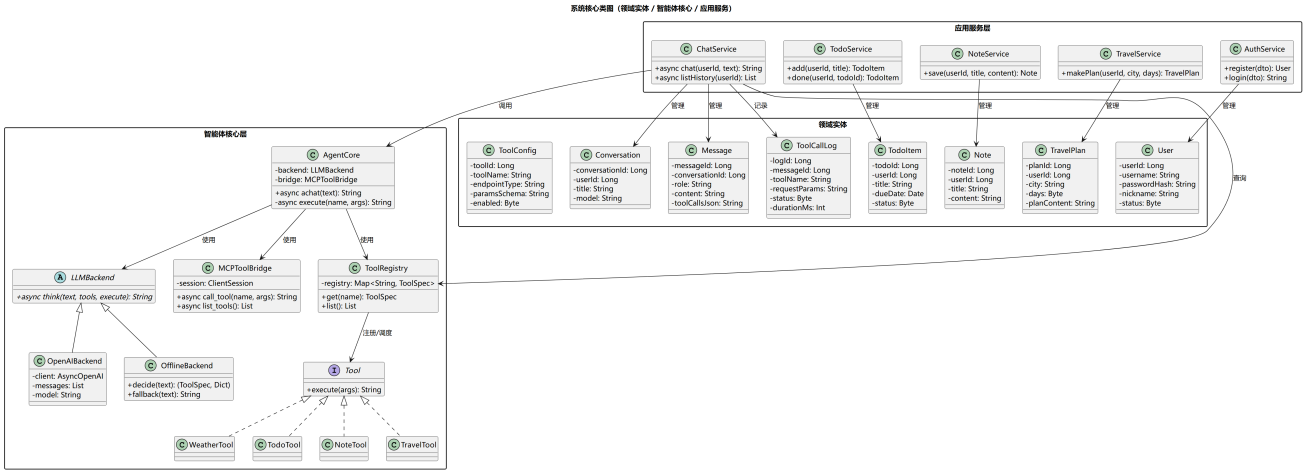


图 3-4 系统核心类图

类	层次	职责说明
AgentCore	智能体核心层	智能体主循环：意图理解 → 工具调度 → 结果回填 → 最终回复
LLMBackend（抽象）	智能体核心层	对话后端接口，think() 返回最终回复
OpenAIBackend	智能体核心层	基于 Chat Completions + Function Calling 的在线实现
OfflineBackend	智能体核心层	关键词 + 正则的离线意图识别，无 Key 兜底
ToolRegistry	智能体核心层	工具注册表：按名查找、列表、参数 Schema 维护
MCPToolBridge	智能体核心层	MCP stdio 客户端，代理工具调用到 MCP 服务器

类	层次	职责说明
Tool（接口）	智能体核心层	工具统一接口 execute(args)
WeatherTool 等	智能体核心层	内置工具实现（天气/待办/笔记/出行）
ChatService 等	应用服务层	业务用例编排，管理领域实体与调用智能体核心
User 等实体	领域层	业务数据模型，与数据库表一一映射

3.4 关键流程时序设计

图 3-5 描述一次“查天气”对话的完整时序：模型返回工具调用指令后，AgentCore 经 MCP 客户端桥把调用转发给 MCP 工具服务器（stdio 子进程），结果回填后模型给出最终回复。

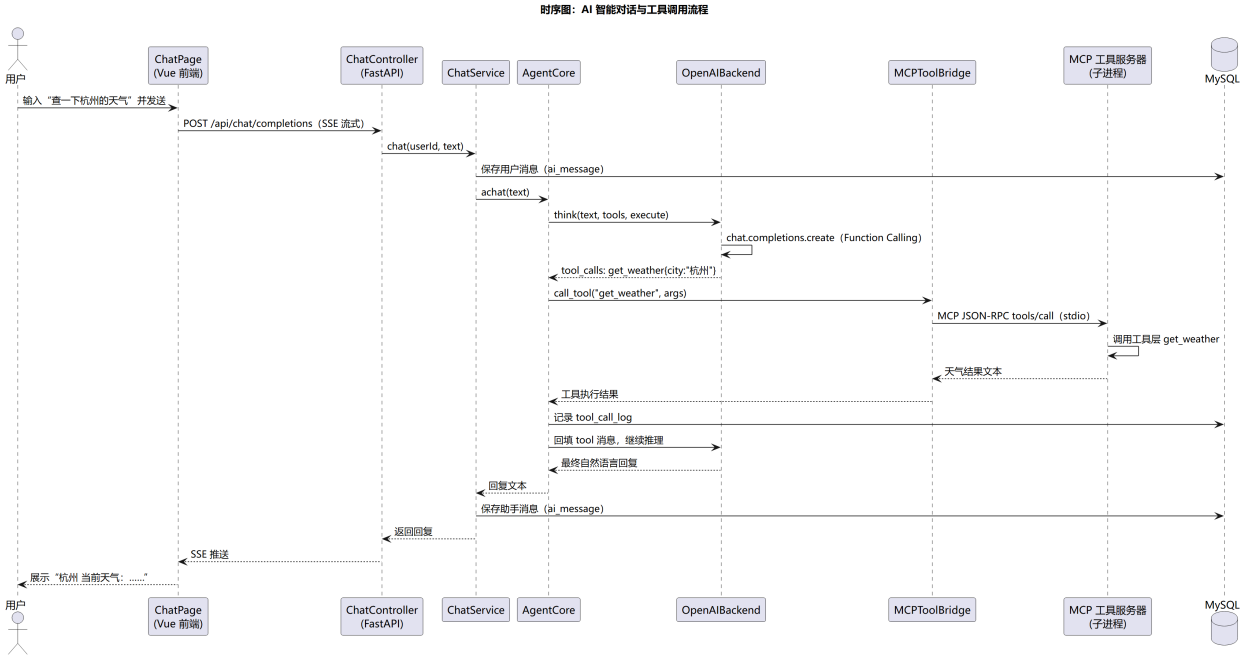


图 3-5 时序图：AI 智能对话与工具调用

图 3-6 描述登录认证流程：bcrypt 校验密码后签发 JWT 并写入 Redis 登录态缓存。

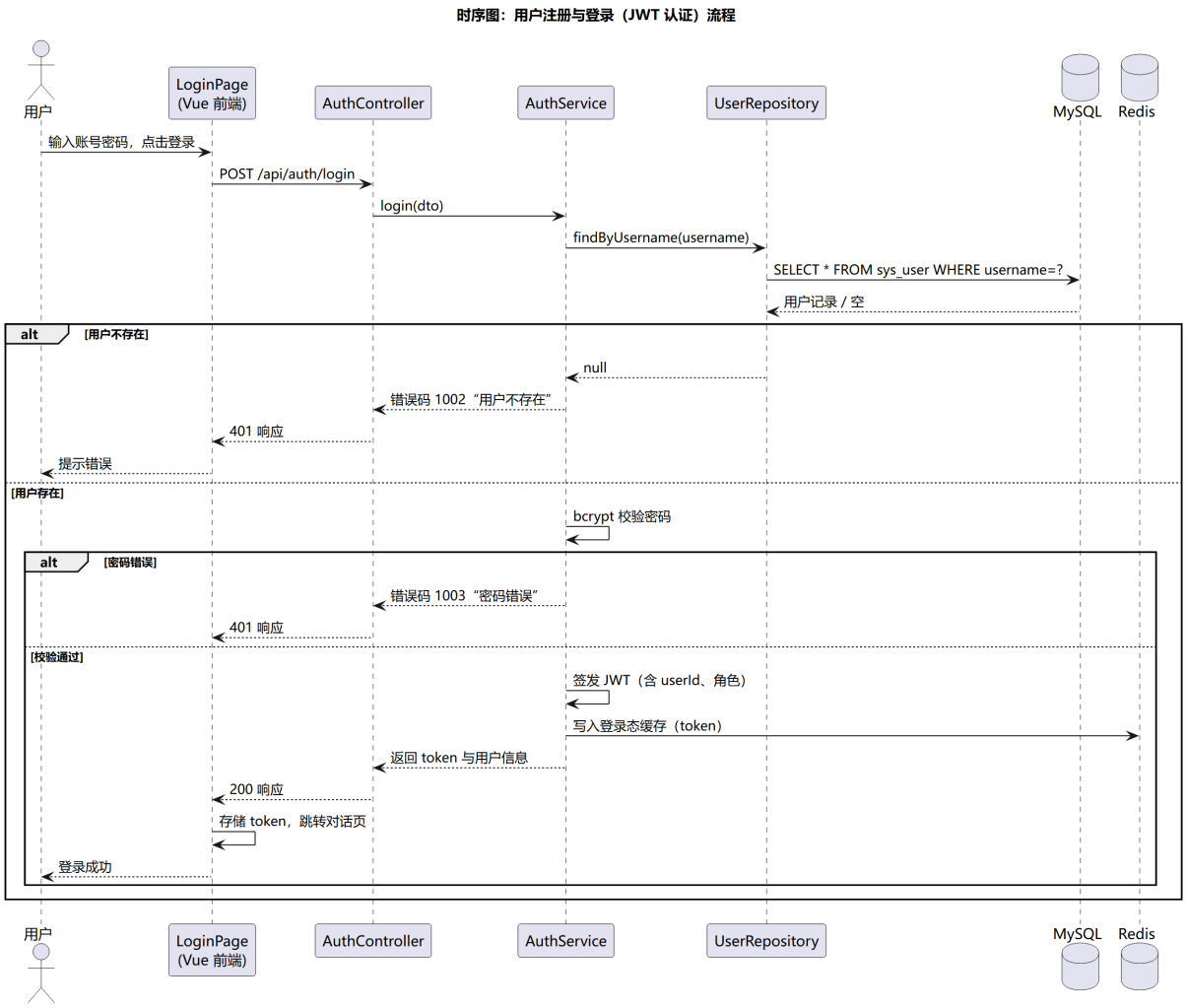


图 3-6 时序图：登录认证（JWT）

图 3-7 用协作图（通信图）视角展示对话场景中各对象间的消息传递顺序，与图 3-5 对应。

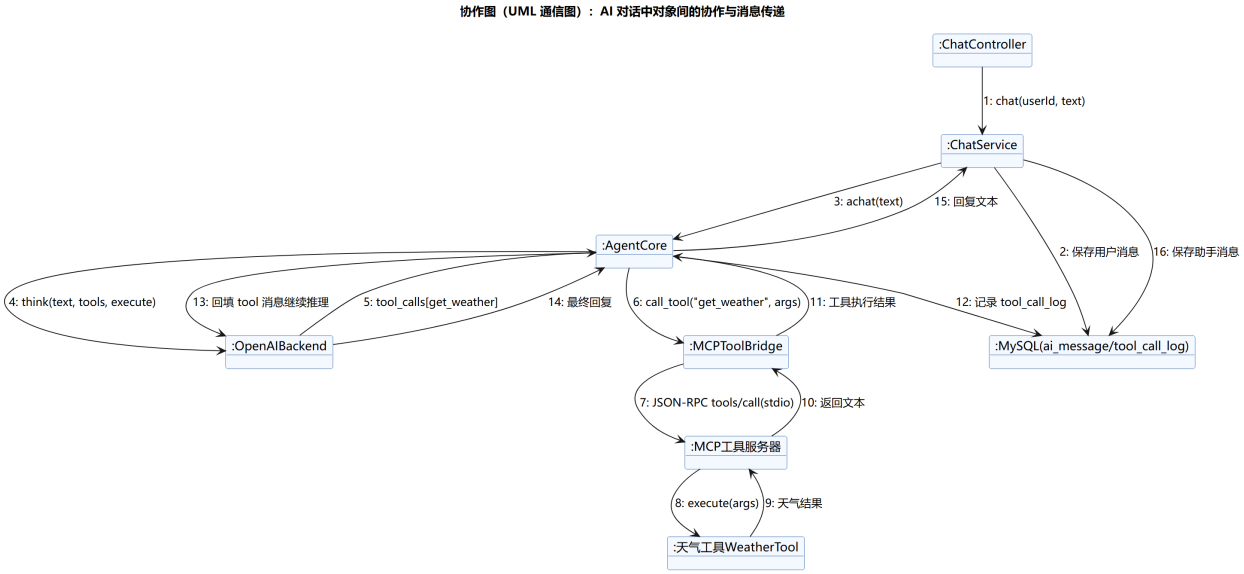


图 3-7 协作图：AI 对话对象协作

3.5 用户接口设计

原型采用高保真可点击 HTML 交付（等价于 Axure/墨刀导出的交互原型），在线预览地址见博客文末。前台页面清单：

页面编号	页面名称	主要元素与交互
F-01	登录 / 注册	账号密码登录、注册切换、忘记密码入口、记住我
F-02	AI 智能对话	历史会话列表、聊天气泡、工具调用卡片、快捷指令、新会话
F-03	待办管理	统计卡片、待办清单、勾选完成、筛选、新增待办
F-04	笔记管理	笔记列表、标签、Markdown 编辑器、查看/编辑切换
F-05	出行规划	参数表单、计划生成、历史计划列表
F-06	个人中心	资料维护、数据统计、偏好设置、退出登录

后台管理页面清单：

页面编号	页面名称	主要元素与交互
B-01	管理员登录	独立入口、RBAC 角色校验
B-02	数据看板	核心指标卡片、调用趋势柱状图、工具占比、系统状态
B-03	用户管理	用户列表、搜索筛选、详情、启用/禁用、导出 CSV
B-04	工具与 MCP 管理	内置工具列表与启停、MCP 服务器接入配置、连接测试
B-05	日志审计	工具调用日志筛选、请求/响应详情、导出

3.6 外部接口设计

接口	协议	方向	调用约定
OpenAI Chat Completions	HTTPS	系统 → OpenAI	携带 tools（JSON Schema）与 messages，tool_choice=auto；超时 60s，重试 2 次
wttr.in 天气接口	HTTPS	系统 → wttr.in	format=j1，无鉴权，超时 15s；失败时返回友好提示而非崩溃
MCP 工具服务器	stdio / JSON-RPC	AgentCore → 子进程	initialize → tools/list → tools/call；心跳缺失时自动重启子进程

3.7 内部接口设计

表现层对应用服务层通过进程内方法调用；对前端统一暴露 REST API，主要接口清单如下（均需 JWT 鉴权，标注“公开”的除外）。

分组	方法与路径	说明
认证（公开）	POST /api/auth/register	注册账号
认证（公开）	POST /api/auth/login	登录并返回 JWT
用户	GET/PUT /api/user/profile	查询 / 修改个人资料
对话	POST /api/chat/completions	发起对话（SSE 流式返回）
对话	GET /api/conversations	历史会话列表

分组	方法与路径	说明
对话	GET /api/conversations/{id}/messages	会话消息列表
对话	DELETE /api/conversations/{id}	删除会话
待办	GET/POST /api/todos	查询 / 新增待办
待办	PUT/DELETE /api/todos/{id}	修改 / 删除待办
待办	POST /api/todos/{id}/done	完成标记
笔记	GET/POST /api/notes	查询 / 新建笔记
笔记	GET/PUT/DELETE /api/notes/{id}	读取 / 修改 / 删除笔记
出行	POST /api/travel/plan	生成出行计划
出行	GET /api/travel/plans	历史计划列表
工具	GET /api/tools	内置工具列表（含参数 Schema）
管理后台	GET /api/admin/users	用户列表（分页/筛选）
管理后台	PUT /api/admin/users/{id}/status	启用 / 禁用用户
管理后台	GET/PUT /api/admin/tools	工具配置查询与启停
管理后台	GET /api/admin/logs/tool-calls	工具调用日志查询
管理后台	GET /api/admin/stats/dashboard	看板统计

4 出错处理设计

4.1 错误码约定

统一响应结构为 {code, message, data}，code=0 表示成功，其余为错误码。

错误码	含义	处理方式
1001	参数缺失或格式非法	前端提示具体字段
1002	用户名或密码错误	提示重新输入，5 次失败触发限流
1003	账号被禁用	提示联系管理员
2001	JWT 缺失或过期	跳转登录页，刷新令牌
3001	模型服务不可用（OpenAI）	自动降级到离线后端并提示
3002	天气服务不可用	返回友好文案，不影响会话
3003	MCP 工具服务器异常	重启子进程，重试 1 次，仍失败则降级为直调
4001	数据库操作失败	回滚事务，记录日志，返回通用错误

4.2 处理策略

- 所有异常经统一异常处理器捕获，转换为上述错误码响应，不向用户暴露堆栈信息；
- 写操作采用事务，失败自动回滚；外部服务调用设置超时与重试上限，避免雪崩；

- 工具调用失败时，把失败原因作为 tool 消息回填给模型，由模型向用户解释或降级处理。

5 安全保密设计

- 认证：用户与管理员统一 JWT 无状态认证，密码使用 bcrypt 单向哈希存储；
- 授权：后台按 RBAC 校验角色（超级管理员 / 运营），普通用户令牌无法访问管理接口；
- 传输：全站 HTTPS，Nginx 终止 TLS；
- 密钥管理：OpenAI API Key、数据库口令只存于服务器环境变量 / 配置中心，禁止写入代码库与数据库；
- 输入安全：所有入参白名单校验，ORM 参数化查询防 SQL 注入，前端输出转义防 XSS；
- 审计：工具调用、后台操作全量记录日志（tool_call_log / sys_operation_log），敏感操作留痕。

6 部署与维护设计

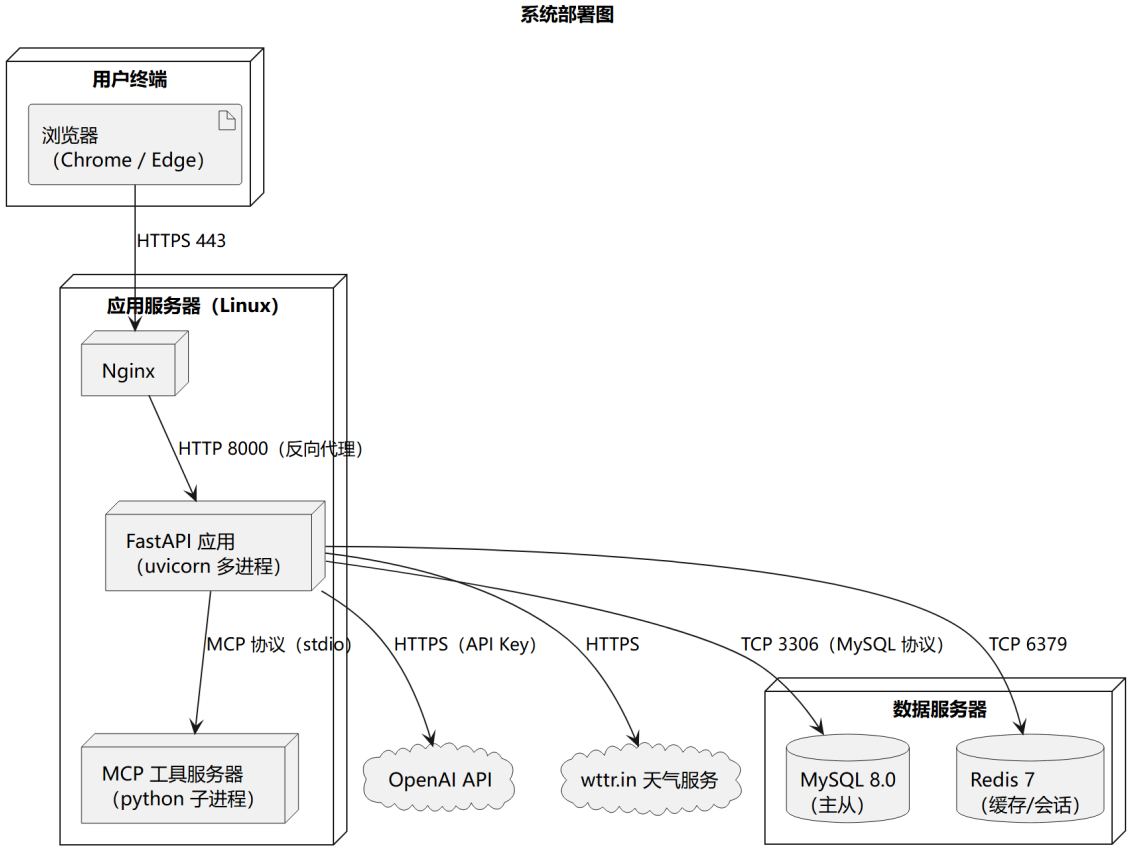


图 6-1 系统部署图

组件	部署位置	端口 / 说明
Nginx	应用服务器	443（HTTPS）→ 转发 8000
FastAPI（uvicorn）	应用服务器	8000，多进程
MCP 工具服务器	应用服务器	stdio 子进程，随主进程拉起

组件	部署位置	端口 / 说明
MySQL 8.0	数据服务器	3306，主从 + 每日全量备份
Redis 7	数据服务器	6379，缓存与会话
OpenAI / wttr.in	外部云服务	HTTPS 出网

维护要点：日志按天滚动归档；配置文件与代码分离；发布采用“先测试环境验证，再滚动更新”的方式。

7 性能设计

指标	目标值	保障手段
对话首字响应（在线）	≤ 3s	SSE 流式输出，首 token 即推送
工具调用（MCP 直调）	≤ 500ms	本地子进程 stdio，无网络往返
普通接口响应	P95 ≤ 300ms	索引优化 + Redis 缓存热点数据
并发对话	≥ 100 并发	uvicorn 多进程 + 无状态会话 + 限流
数据量	支持 10 万用户 / 500 万消息	归档策略 + 分页查询 + 分区表预留

8 数据设计概述

数据库采用 MySQL 8.0，库名 xzsl_db，包含用户、会话、消息、待办、笔记、出行计划、工具配置、工具调用日志、管理员、操作日志共 10 张表，关系模型与 ER 图详见《智行小组_数据库设计说明书》。本系统的数据访问通过 SQLAlchemy ORM 完成，实体类与数据表一一映射。