概要设计说明书

1. 总体设计
   1. 需求规定
      1. 功能需求

* 小麦-产地-病害数据采集与预处理
* 知识提取
* 属性查找
* 自然语言查找（知识查找）
* 可视化展示与交互
  + 1. 性能需求

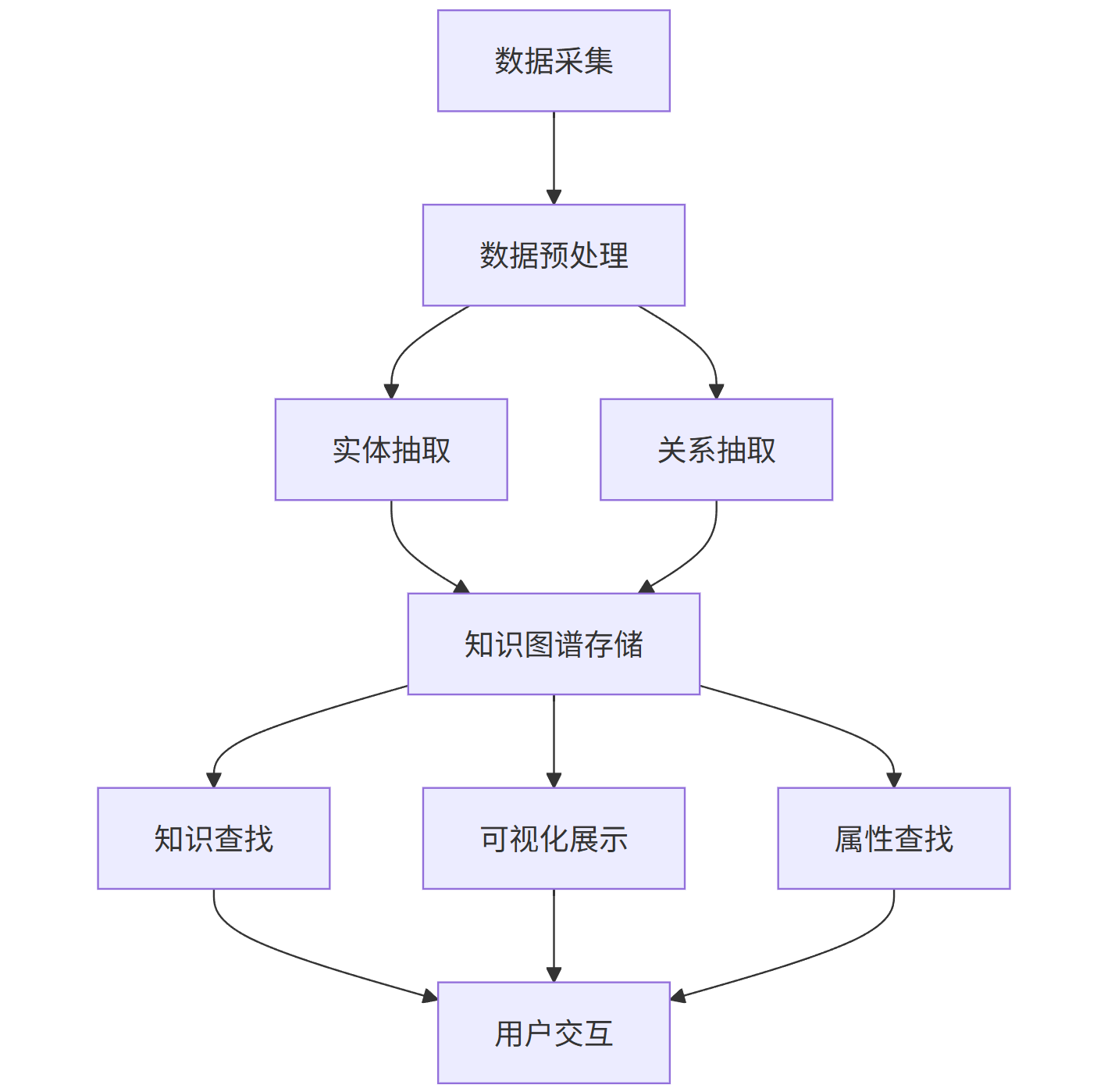
低延迟、定期健康性检查

* 1. 运行环境

|  |  |
| --- | --- |
| 类别 | 要求 |
| 系统环境 | Windows/macOS/Linus  Python3.x |
| 依赖 | Flask==2.3以上  neo4j==5.x  tencentcloud-sdk-python==3.0.1000  py2neo==2021.2.3以上  pandas==2.0.3以上 |

表格 1

* 1. 基本结构



* 1. 功能需求和程序的关系

|  |  |
| --- | --- |
| 功能 | 关键技术 |
| 数据采集与预处理 | Requests库、Pandas库 |
| 实体提取与关系提取 | 模式匹配 |
| 数据存储 | Neo4j图数据库、Py2neo库 |
| 属性查找 | 模式匹配 |
| 知识查找 | hunyuan-turbos-latest对话模型 |
| 数据可视化与人机交互 | Flask |

表格 2

* 1. 尚未解决的问题

功能拓展：暂未提供决策支持功能，功能开发中

1. 接口设计
   1. 外部接口

|  |  |  |
| --- | --- | --- |
| 接口名称 | 来源 | 功能 |
| ChatCompletions | hunyuan-turbos-latest | 处理自然语言到 Cypher 查询的转换以及 Cypher 查询结果到自然语言回答的转换。调用时使用了SECRET\_ID和SECRET\_KEY进行身份验证。 |

表格 3

* 1. 内部接口
     1. 用户认证相关

|  |  |
| --- | --- |
| 接口名称 | 功能 |
| POST /api/login | 用户登录接口，接收用户名和密码，验证成功后创建会话 |
| POST /api/register | 用户注册接口，接收用户名、邮箱和密码，创建新用户 |
| POST /api/logout | 用户登出接口，清除当前会话 |
| GET /api/user/info | 获取当前登录用户信息 |

表格 4

* + 1. 图谱数据相关

|  |  |
| --- | --- |
| 接口名称 | 功能 |
| GET /api/graph-data | 获取初始图谱中节点和关系信息 |
| POST /api/search | 根据实体类型、属性名和属性值搜索实体及其相关节点和关系 |
| GET /api/node/<int:node\_id> | 获取指定 ID 节点的详细信息，包括标签和属性 |
| GET /api/view/<view\_type> | 获取特定视图的数据，支持 "wheat-region"（小麦 - 地区视图）、"wheat-disease"（小麦 - 病害视图）等 |

表格 5

* + 1. 知识查询相关

|  |  |
| --- | --- |
| 接口名称 | 功能 |
| POST /api/question | 智能问答接口，接收用户问题，生成 Cypher 查询、执行查询并返回自然语言回答 |

表格 6

* + 1. 健康检查相关

|  |  |
| --- | --- |
| 接口名称 | 功能 |
| GET /api/health | 健康检查端点，返回应用和 Neo4j 数据库的连接状态 |

表格 7

* + 1. 管理员功能相关

|  |  |
| --- | --- |
| 接口名称 | 功能 |
| GET /api/admin/users | 获取所有用户信息（仅管理员可访问） |
| GET /api/admin/activities | 获取用户行为记录（仅管理员可访问），支持按用户 ID 筛选和限制数量 |

表格 8

* 1. 内部模块接口

|  |  |
| --- | --- |
| 模块 | 功能 |
| app.py | 提供Neo4jConnection类，实现与数据库交互的各种方法 |
| ai\_q\_and\_a.py | 实现自然语言与Cypher转换等方法，实现知识智能问答功能 |

表格 9

1. 运行设计
   1. 启动流程

检查依赖->启动数据库->启动web服务->打印启动日志

* 1. 核心业务运行流程

用户发送问题->API接受请求->调用相关模块->执行查询->返回并处理结果->保存用户行为

* 1. 退出流程
     1. 正常退出

接收Ctrl+C信号->停止 Web 服务->关闭数据库连接->释放资源->打印退出日志

* + 1. 异常退出

数据库连接中断->触发告警->终止服务->打印错误日志

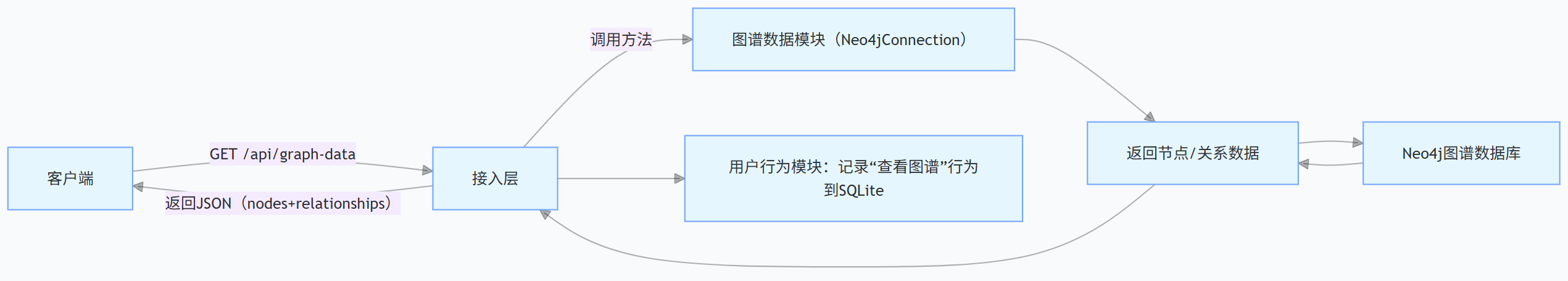
1. 系统数据结构设计
   1. 逻辑设计
      1. 模块划分

|  |  |  |
| --- | --- | --- |
| 模块类型 | 包含模块 | 核心职责 |
| 接入层 | Flask Web 服务 | 提供 HTTP API 接口，接收客户端请求（如用户登录、图谱查询、智能问答），返回响应。 |
| 认证授权层 | 用户认证模块 | 处理用户注册/登录/登出，管理会话，实现管理员权限控制。 |
| 业务逻辑层 | 1. 图谱数据模块 2. 智能问答模块、 3. 用户行为模块 | 1. 封装 Neo4j 数据库交互，提供图谱数据查询/搜索/视图切换功能； 2. 实现 “自然语言->Cypher->查询结果->自然语言” 的问答全流程； 3. 记录用户操作（登录、搜索、问答）到 SQLite 数据库。 |
| 数据层 | 1. Neo4j 图谱数据库 2. SQLite 用户数据库 | 1. 存储小麦知识图谱数据（实体、关系、属性）； 2. 存储用户信息（账号、密码哈希）和行为记录。 |
| 外部交互 | 腾讯云混元 AP | 调用大模型接口，完成自然语言与 Cypher 的双向转换。 |

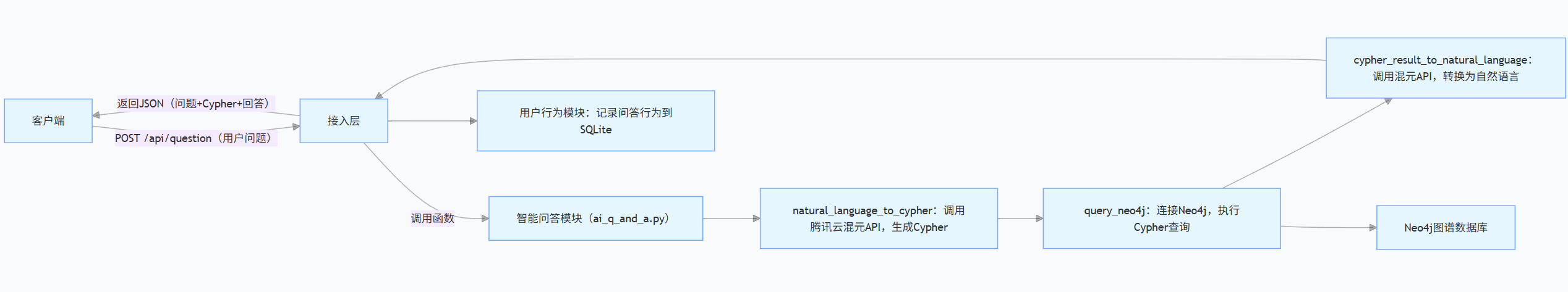
表格 10

* + 1. 数据流转逻辑

1. 属性查询场景



1. 知识查询场景



* 1. 物理设计
     1. 数据存储设计

|  |  |
| --- | --- |
| 存储方式 | 功能 |
| Neo4j存储 | 保存实体与关系 |
| SQLite存储 | 保存用户数据 |

表格 11