# 智能生活助手应用 - 数据设计说明书

## 1. 引言

### 1.1 编写目的

本文档旨在详细说明智能生活助手应用的数据设计，包括数据结构、数据流、数据采集与处理方式等。本说明书为开发团队提供数据架构参考，确保数据设计满足应用需求，并为后续开发、测试和维护提供依据。

### 1.2 背景

**a) 待开发软件系统的名称：** 智能生活助手移动应用

**b) 项目背景：** 智能生活助手是一款集待办清单、智能搜索、健康监测和生活便签等功能于一体的移动应用，旨在帮助用户高效管理日常生活和工作。应用采用React Native开发前端，FastAPI构建后端，PostgreSQL作为数据库系统。

### 1.3 定义

* **系统管理员权限**：具有管理整个系统的权限，包括用户管理、系统配置等。
* **普通用户权限**：具有使用应用功能的基本权限，包括创建和管理自己的待办事项、便签等。
* **游客权限**：未登录用户，仅可访问应用的公开内容和功能。
* **敏感数据**：用户个人信息、健康数据等需要特殊保护的数据。
* **JWT令牌**：用于用户身份验证的JSON Web Token。

### 1.4 参考资料

* 《智能生活助手APP需求分析.md》
* 《智能生活助手APP设计文档.md》
* 《产品面向对象设计.md》
* 《CODE\_SPECIFICATION.md》
* FastAPI官方文档
* PostgreSQL官方文档
* React Native官方文档

## 2. 数据的逻辑描述

### 2.1 静态数据

以下是系统中的静态数据：

1. **系统配置数据**
   * API版本信息
   * 系统常量（如待办事项状态枚举：待办/进行中/已完成/待定）
   * 错误代码与消息映射
   * 页面配置信息
2. **用户界面静态资源**
   * 图标和图片资源
   * UI主题和样式配置
   * 国际化语言包
3. **业务规则数据**
   * 健康指标参考值（如每日步数目标范围）
   * 密码策略配置
   * 安全规则配置

### 2.2 动态输入数据

系统接收的主要动态输入数据：

1. **用户注册与认证数据**
   * 用户名、密码、手机号、邮箱等注册信息
   * 登录凭证
   * 验证码
   * 密码重置信息
2. **待办事项数据**
   * 待办事项标题和描述
   * 开始和结束时间
   * 提醒时间
   * 地点信息
   * 状态变更
3. **搜索数据**
   * 搜索关键词
   * 搜索过滤条件
   * 无痕搜索标志
4. **健康监测数据**
   * 步数记录（可能来自设备传感器）
   * 喝水提醒设置
   * 健康设置参数
5. **生活便签数据**
   * 便签标题和内容
   * 附加图片
   * 便签标签/分类

### 2.3 动态输出数据

系统生成的主要动态输出数据：

1. **用户信息输出**
   * 用户个人资料
   * 身份验证令牌
   * 登录状态信息
2. **待办事项列表与详情**
   * 用户待办事项列表
   * 待办事项详情
   * 待办事项统计信息（已完成数量、进行中数量等）
   * 日历视图数据
3. **搜索结果**
   * 搜索结果列表
   * 推荐内容
   * 搜索历史
4. **健康统计数据**
   * 步数统计报告（日报、周报、月报）
   * 喝水提醒通知
5. **便签内容**
   * 便签列表
   * 便签详情（含图文内容）
   * 便签搜索结果
6. **系统通知**
   * 待办提醒通知
   * 系统消息
   * 健康提示

### 2.4 内部生成数据

系统内部生成的数据：

1. **系统日志**
   * 用户活动日志
   * 错误日志
   * 安全审计日志
   * 性能监控数据
2. **计算生成的数据**
   * 待办事项完成率统计
   * 健康数据趋势分析
   * 用户行为分析
   * 搜索关键词热度
3. **缓存数据**
   * 用户会话缓存
   * 频繁查询数据缓存
   * 应用状态缓存
4. **临时文件**
   * 图片处理临时文件
   * 上传文件缓存

### 2.5 数据约定

1. **命名约定**
   * 数据库表名：使用小写蛇形命名法，复数形式
     + 用户相关：users, user\_profiles, user\_settings, user\_devices
     + 待办相关：todo\_items, todo\_categories, todo\_tags
     + 搜索相关：search\_records, search\_keywords, recommended\_contents
     + 健康相关：step\_records, water\_reminders, health\_settings
     + 便签相关：notes, note\_images, note\_tags
     + 通知相关：notifications, notification\_settings
     + 系统相关：system\_logs, error\_logs, audit\_logs
   * 列名：使用小写蛇形命名法
     + 通用字段：id, created\_at, updated\_at, deleted\_at, is\_active
     + 用户相关：username, email, phone\_number, password\_hash, last\_login\_at
     + 待办相关：title, time\_begin, time\_end, time\_notify, where\_to\_go, state, priority
     + 便签相关：content, image\_path, is\_pinned, color\_code
   * 主键：统一使用id作为主键名，类型为字符串(UUID)
   * 外键：使用<关联表单数>\_id形式（如：user\_id, note\_id, todo\_id）
2. **数据格式约定**
   * 日期时间：使用ISO 8601标准（YYYY-MM-DDTHH:MM:SSZ）
     + 所有时间戳存储为UTC时间，显示时转换为用户时区
     + 使用PostgreSQL的timestamptz类型存储时间
   * 货币：保留两位小数，使用decimal(10,2)类型
   * 布尔值：使用true/false而非1/0，对应PostgreSQL的boolean类型
   * 字符编码：统一使用UTF-8
   * 枚举值：使用PostgreSQL的枚举类型，如待办状态(todo\_state\_enum)定义为：'doing', 'todo', 'done', 'pending'
3. **安全约定**
   * 密码存储：使用bcrypt算法哈希处理，不存储明文密码
   * 敏感数据：加密存储，访问时进行权限验证
     + 手机号/邮箱：部分脱敏显示（如1378888）
     + 健康数据：应用层加密存储，用户个人密钥管理
   * API访问：使用JWT令牌验证身份，token过期时间设置为8天
   * 日志记录：敏感操作记录到audit\_logs表，包含user\_id, action, resource\_type, resource\_id, ip\_address
4. **数据完整性约定**
   * 必填字段：明确标识，服务端强制验证
   * 数据验证：使用Pydantic模型进行数据验证
   * 级联删除：关联数据的删除策略明确定义
     + 用户删除：相关数据标记为已删除而非物理删除
     + 便签图片：当便签删除时，相关图片一并删除
     + 待办事项：删除时，相关标签关联记录一并删除

## 3. 数据的采集

### 3.1 要求和范围

1. **数据采集原则**
   * 最小必要原则：只采集必要的数据
   * 用户知情同意：明确告知用户数据采集用途
   * 安全加密：敏感数据传输过程加密
2. **采集范围**
   * 用户基本信息：注册信息、个人设置
   * 用户行为数据：应用使用记录、功能偏好
   * 内容数据：用户创建的待办事项、便签等
   * 健康数据：经用户授权的步数等健康指标
3. **采集限制**
   * 无痕模式：特定场景下不记录用户行为
   * 健康数据采集频率限制，避免过度耗电
   * 存储大小限制（如便签图片大小限制）

### 3.2 输入的承担者

1. **最终用户**
   * 直接输入：通过应用界面手动输入数据
   * 授权导入：授权应用访问设备数据（如健康数据）
2. **系统管理员**
   * 系统配置数据维护
   * 用户数据管理
3. **外部系统**
   * 设备传感器：提供步数等健康数据
   * 移动操作系统：提供设备信息、通知权限等
   * 第三方API：比如高德地图，提供地图、附近商户等外部数据
4. **自动采集系统**
   * 用户行为跟踪模块：记录用户使用模式
   * 性能监控模块：采集应用性能数据

### 3.3 预期处理

1. **数据验证与清洗**
   * 输入验证：格式、范围、类型验证
   * 数据净化：去除不必要字符、标准化格式
   * 数据去重：避免重复记录
2. **数据转换**
   * 坐标转换：地理位置标准化
   * 时区处理：统一存储UTC时间，显示时转换为用户本地时区
   * 单位转换：适应不同区域的计量单位
3. **数据存储**
   * 关系型数据：存储到PostgreSQL数据库
   * 文件存储：图片等媒体文件存储到文件系统或云存储
   * 缓存数据：存储到内存缓存或Redis
4. **数据分析与聚合**
   * 统计分析：生成用户行为统计报告
   * 趋势分析：分析健康数据趋势
   * 推荐算法：基于用户行为生成推荐内容

### 3.4 影响

1. **系统性能影响**
   * 数据量增长对存储系统的压力
   * 查询复杂度对响应时间的影响
   * 数据备份与恢复策略的影响
2. **用户体验影响**
   * 数据输入便捷性对用户满意度的影响
   * 数据处理速度对应用流畅度的影响
   * 隐私保护措施对用户信任度的影响
3. **运营成本影响**
   * 数据存储成本
   * 数据传输带宽成本
   * 数据维护与支持成本
4. **合规性影响**
   * 符合数据保护法规（如GDPR）的要求
   * 用户数据权利（访问、删除、导出等）的实现
   * 数据保留策略的合规性