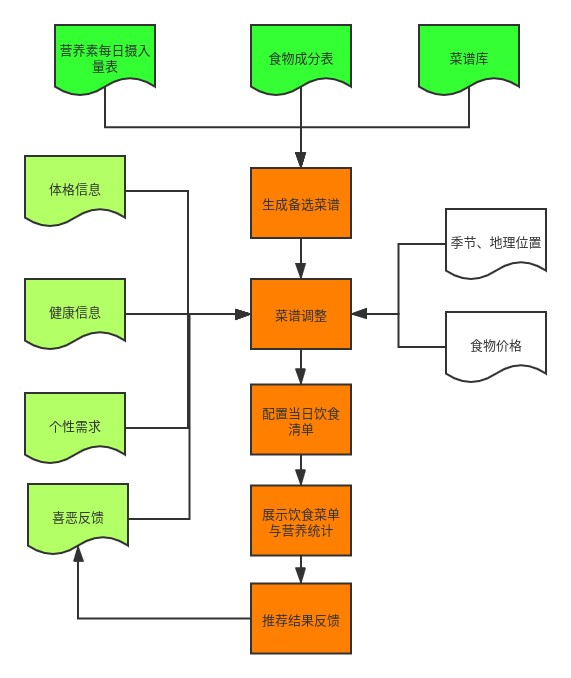
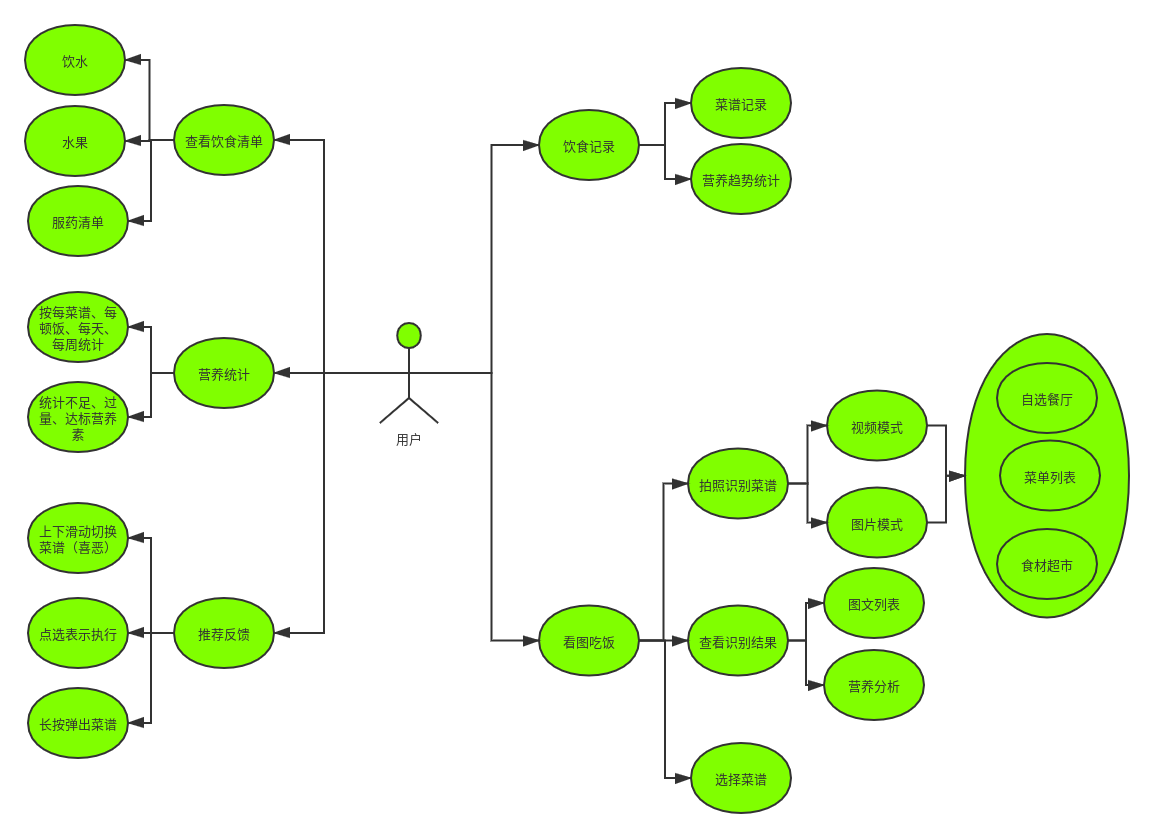
# 智能健康饮食

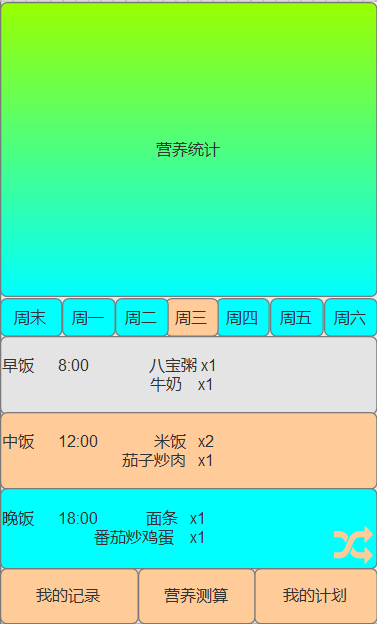
## 算法实现原理

* 基于《中国膳食营养素参考摄入量RNI》(https://www.59baike.com/a/42787-33)计算用户每日营养素需求量
  + 与用户年龄、体力活动目标相关
  + 根据用户健康状态、健康目标调整
  + [临床营养学](https://mp.weixin.qq.com/s?__biz=MzU5NTY4ODk3Ng==&mid=2247500920&idx=4&sn=276c518cd5b49e2f1adf9e55bd3ad160&chksm=fe6caf07c91b261156ed701d260295112082751e9be72fbafe16cdeeeb975251b1af03fa8c31&mpshare=1&scene=1&srcid=06306B2nbliOyJ5tnYVlsJJp&sharer_sharetime=1607605307248&sharer_shareid=0d85ec0b9d5bff73305fe979dc34a6d1&exportkey=A%2FiqrfzfS0UTD1Ew29qmo%2B4%3D&pass_ticket=LgihzCL4tjPIQy9sUVlDtuTGp7Hp2zUe1VIwOGU%2FGoZ6SY6quyL1anlv7S58R35m&wx_header=0#rd)
* 结合当日需求量，基于[食物成分表》(https://fq.chinafcd.org/)计算备选食物
* 结合备选食物与菜谱知识库计算当日推荐菜谱
  + 江浙菜谱、粤菜、徽菜
  + 杭帮菜菜谱：https://www.shipuxiu.com/zuofadaquan/861852/
  + 根据用户的喜爱调整推荐参数
  + 考虑当地容易买到、价格便宜
  + 一周保证25种以上食材
* 将最优推荐显示到app食品列表中
  + 每顿饭什么时候吃、吃什么、吃多少
  + 展示营养素摄入情况，并对比标准量



## 功能用例

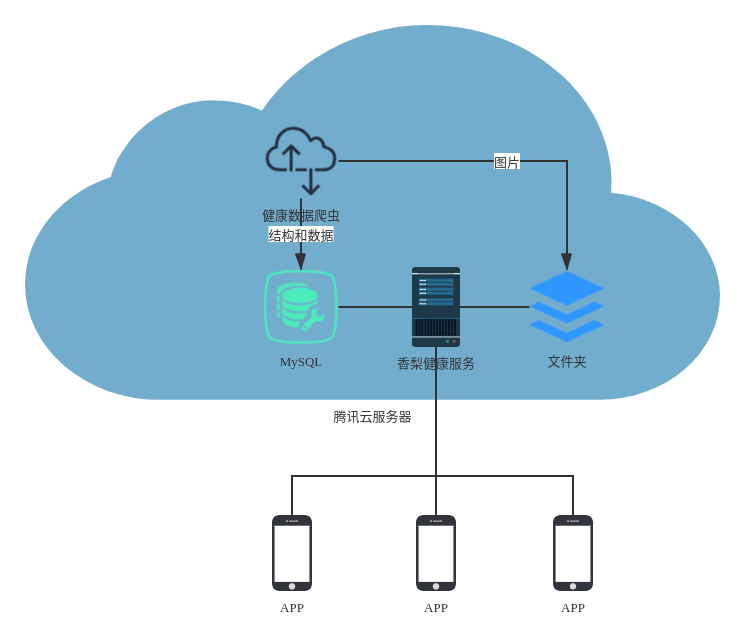




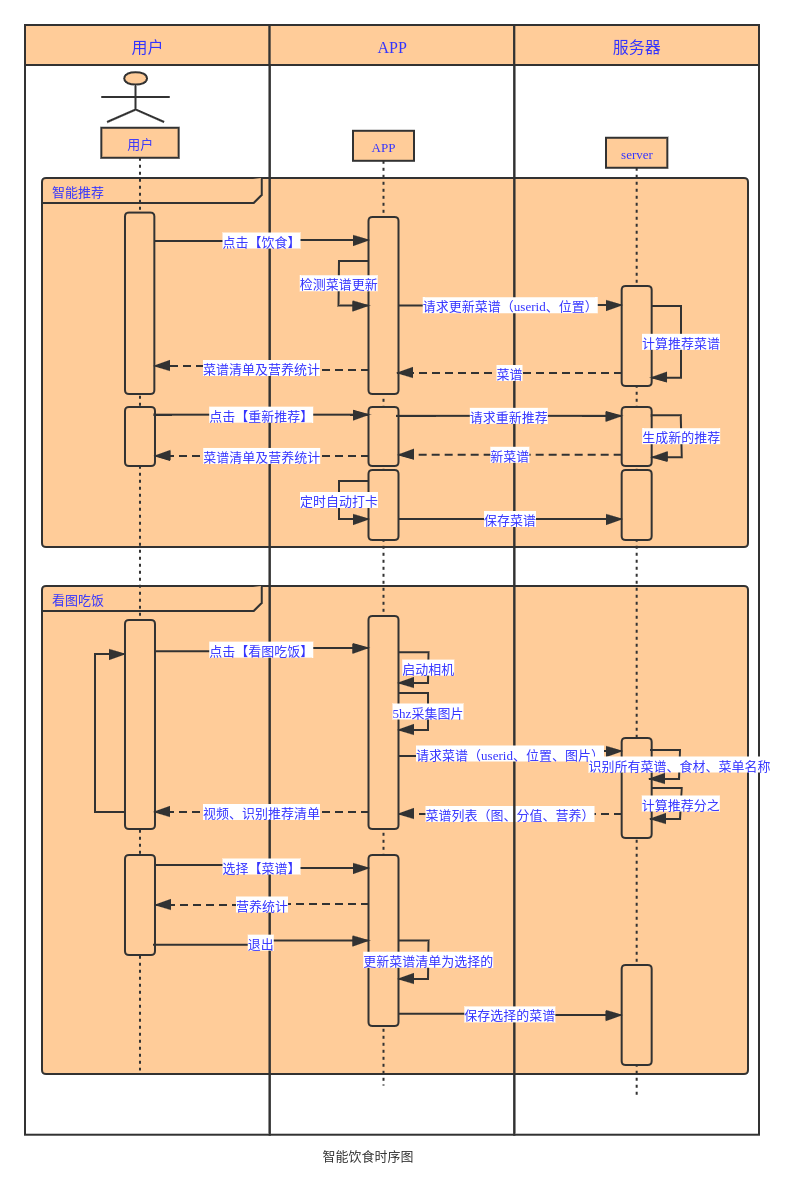
说明：

* 基于用户的视角，需要操作的功能
* 要求：展示简介任性、内涵底蕴深
* 补充：需要展示总能量摄入情况（碳水化合物+脂肪+糖）
* 上图是功能用例，下图是体现功能用例的APP设计样例
* 看图吃饭功能本版本不实现，作为第二代产品；用户通过相机拍照或者打开视频，APP通过机器视觉识别图中菜品、菜单、食材，并结合营养推荐标准等信息推荐处营养健康的菜谱列表，并显示选中菜谱的营养分析；图像分析过程中记录用户地址和识别出的所有菜谱，以丰富菜谱库，增加智能推荐能力

## 网络拓扑



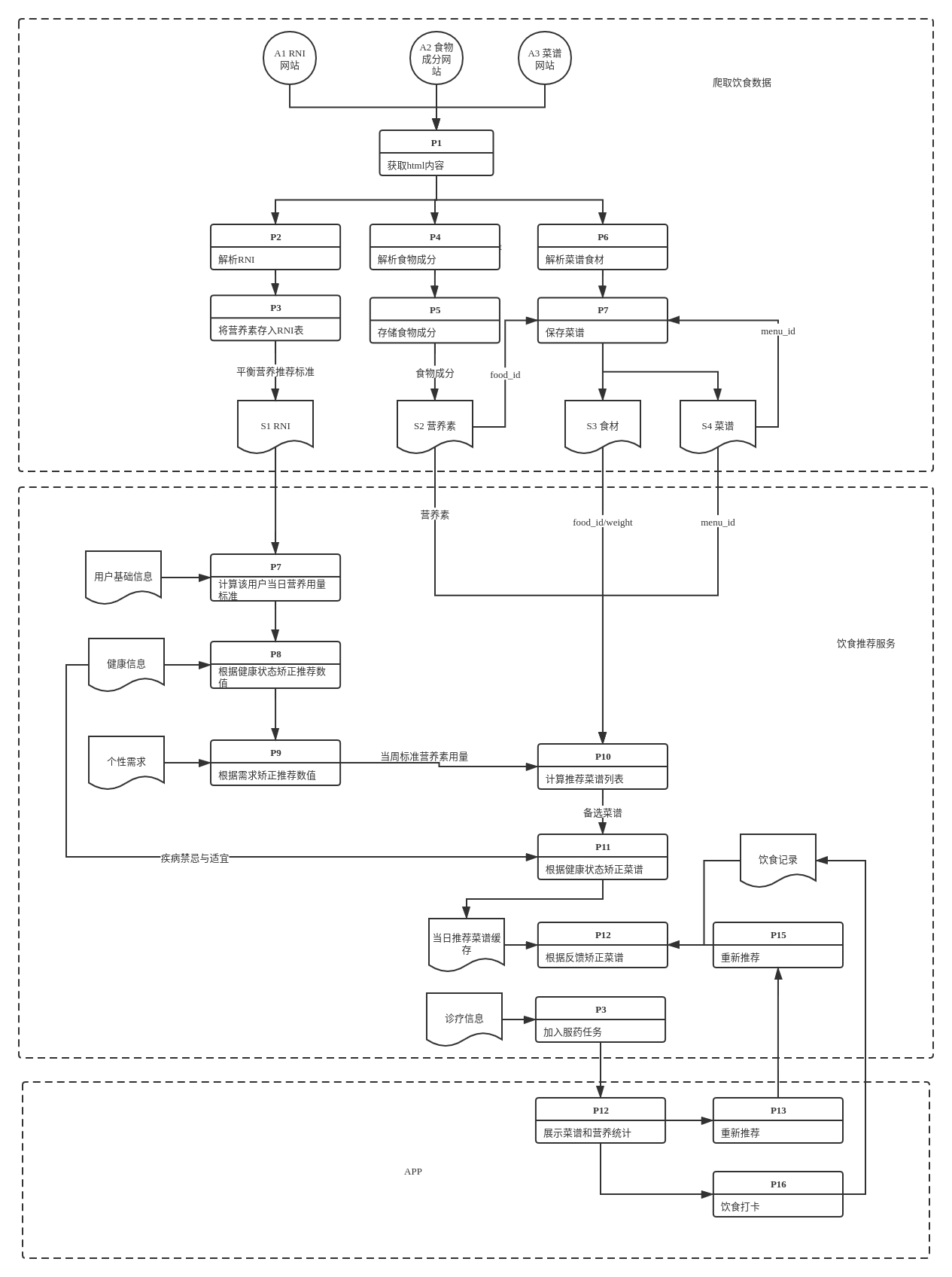
## 交互流程图



说明：

* 当用户进入APP饮食界面时，触发菜谱更新请求；APP获取用户位置和userID向服务器请求菜谱；
* 服务器会根据《智能健康饮食》中提到的算法计算菜谱清单，并返回结果；
* 用户可能不满意推荐结果，允许重置推荐结果；
* 看图吃饭是基于场景的智能推荐，用户打开相机采集成品菜、菜单或者食材，APP将图文数据上传到服务器进行识别，服务器根据识别结果及用户健康数据返回识别及推荐结果；

## 数据流程

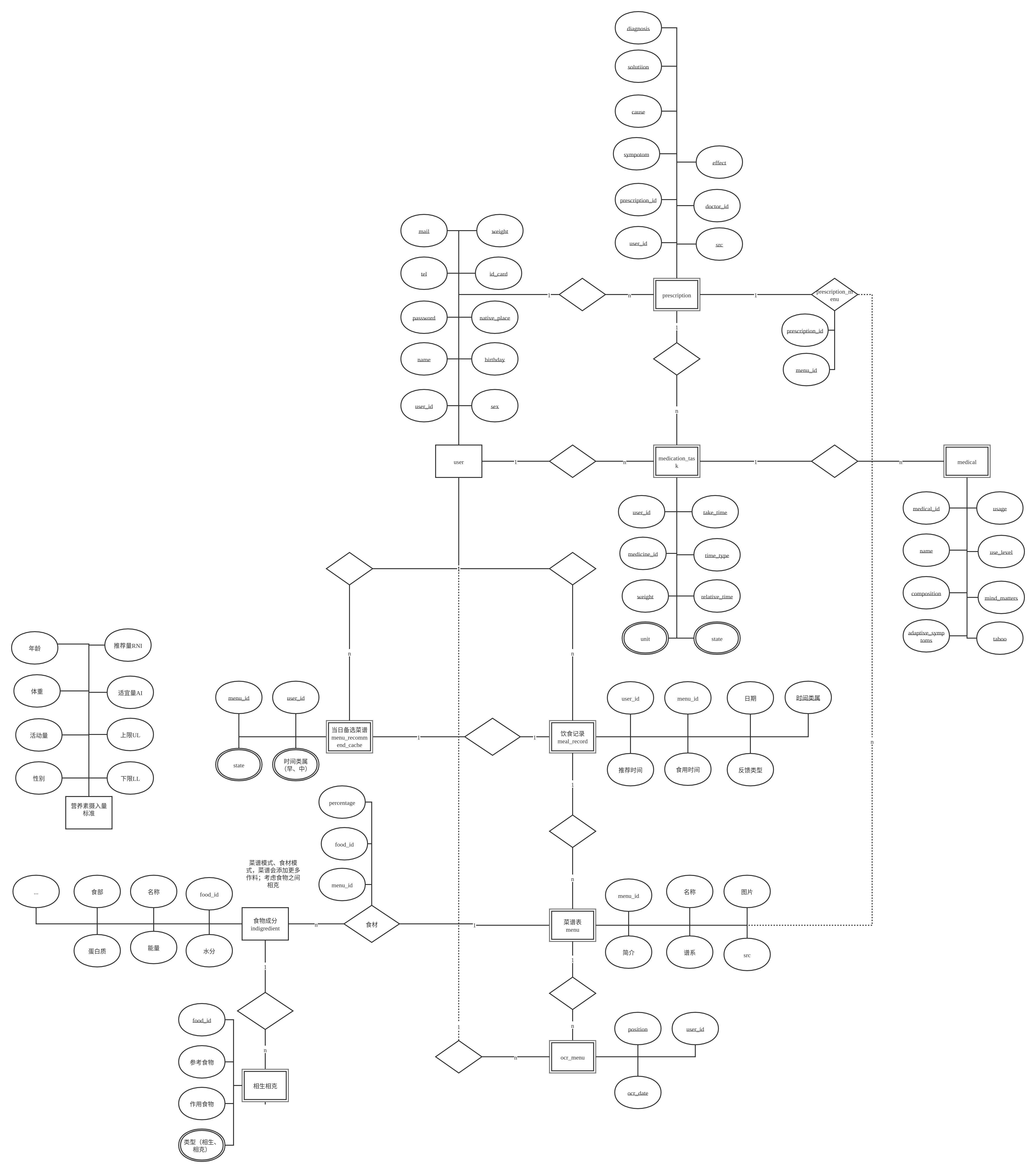


详细数据流程： https://www.processon.com/view/link/5fd3635163768906e6d70570

数据库表设计之ER模型： https://www.processon.com/view/link/5fd3639e5653bb06f3378d7c

服务器与APP接口： https://www.easyapi.com/api/?documentId=28162&code=XiangLiHealthyLife

## ER模型



ER说明：

* 矩形代表实体表、椭圆代表实体属性、菱形代表关系表，关系表之间的1：n表示一对多关系
* 主要分为营养素标准量表、食物成分表、菜谱表、饮食 记录表
* 字段名、字段大小和主键关系由数据库脚本设计时定义

## 数据库表结构

* menu*recommend*cache（菜谱缓存表）

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 字段名 | 字段类型 | 说明 |
| 1 | user\_id | int | 用户id，很多用户都会缓存到这张表 |
| 2 | munu\_id | int | 菜谱id，这个id可以查到包含的食材 |
| 3 | weight | float | 分量：g |
| 4 | recommend\_time | datetime | 推荐食用时间 |
| 5 | flag | int | 0 备用、1 推荐、2 不喜欢 |

说明：

1. 食用菜谱缓存表是为了应对重置推荐的情况，重置推荐仅需要从缓存表中选择食物，节约计算时间；

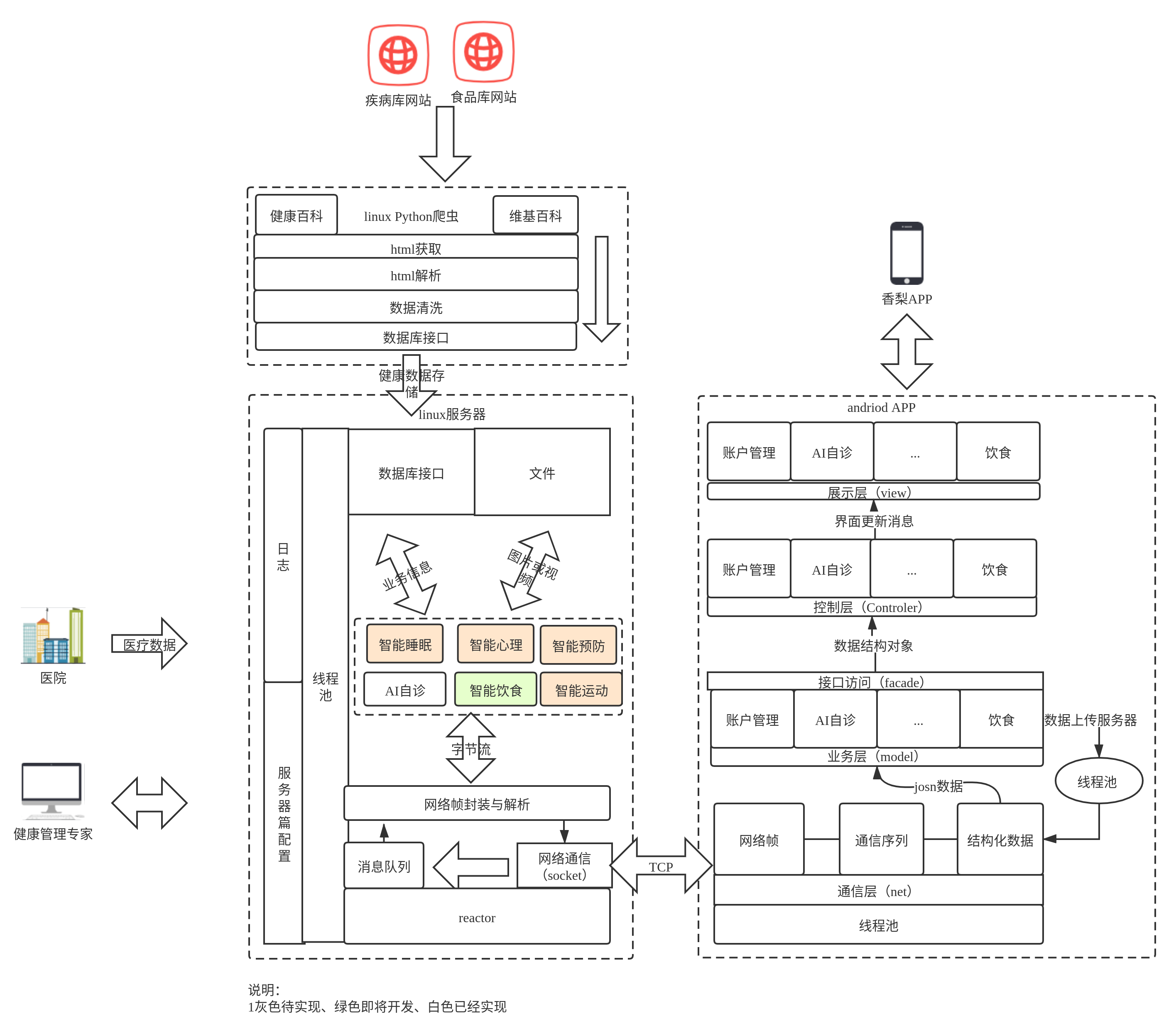
* meals\_record(饮食记录表)

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 字段名 | 字段类型 | 说明 |
| 1 | user\_id | int |  |
| 2 | memu\_id | int |  |
| 3 | weight | float |  |
| 4 | recommend\_time | datatime |  |
| 5 | meal\_time | datatime | 食用时间，可能与推荐时间不同 |

* medication\_task

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 字段名 | 字段类型 | 说明 |
| 1 | user\_id | int | 用户id |
| 2 | medicine\_id | int | 药物id |
| 3 | weight | float | 分量 |
| 4 | unit | varchar(8) | 分量单位，克、ml、包、颗粒、片 |
| 5 | take\_time | datetime | 建议服用时间 |
| 6 | time\_type | int | 0 任意时间 1饭前 2饭后 |
| 7 | relative\_time | datatime | 相对于饭前或者饭后的时间值 |
| 8 | state | int | 0 待服用 1 已服用 2 过期 |

## 模块架构



说明：

* 爬虫从网页获取营养推荐量表、食物成分表、菜谱，存入数据库
* 地理位置、用户个性需求、用户健康信息由用户录入并上传到服务器，存入数据库
* 服务器综合食物信息和用户信息，根据一套算法精准推荐营养平衡、符合用户口味的菜谱清单
* 推荐结果封装成json格式，再封装成tcp字节流传输到手机APP端
* APP端解析tcp字节流，再转化成json，再转化成java对象，根据设计展示到用户界面
* 用户反馈的信息从APP回传到服务器