# 引言（潘保恒）

## 目标

在完成软件开发前期的准备工作如项目需求等，结合《资源需求估计》《产品愿景》等，，项目小组提出了这份软件设计说明书。

此概要设计说明书对AI智能推荐食谱系统的功能分配，模块划分，程序的总体结构，输入输出和接口设计，运行设计，数据结构设计及出错设计等方面作了全面的概括性的说明，为软件详细设计奠定了基础，同时作为系统分析员工作的阶段性总结和程序员进行开发及未来测试的重要文档资料。

## 文档范围

本文档包括系统的架构设计、数据库设计、安全性、可靠性、可用性等方面的规划和设计。

## 术语和缩略语

Intelligent recommendation recipes

## 参考资料

部分内容参考了《AI智能推荐食谱-资源需求估计》《AI智能推荐食谱-产品愿景》等

## 系统目标和约束

系统目标：项目需要完成注册登入、身体条件咨询、食谱推荐、食物热量指标、点赞评论收藏等功能

系统的约束：因为时间有限，该系统只实现最基本的功能。

# 系统设计（王晶）

## 系统架构概述

该系统的架构采用两层架构的模式：展示层（视图+业务逻辑）和数据访问层。采用面向对象的设计方法。

## 对象模型

该系统采用面向对象的设计方法。通过调整对象这个类的属性来调整对象的模型

## 接口

*[详细说明本系统内部每个接口的每个方法的定义。*

*注意：如果本系统需要和其他系统交互，则应该将交互的接口协议单独成册，而不是写入本节中。]*

## 特性实现

1. 用户注册注册
   1. 用户注册页面
   2. 注册成功，可选择自己想要达到的目标



1. 用户登记自己的相关信息
   1. 登记APP个人页面
   2. 登记成功，可选择进入个人消息添加详细信息



1. 用户上传自己做的食谱
   1. 制作菜品视频
   2. 上传到自己的动态
2. 用户浏览别人上传的食谱
   1. 浏览视频
   2. 可以选择收藏视频，关注视频发布者



1. 系统推荐食谱
   1. 用户填完自己的信息后，可进入推荐页面
   2. 每日三餐都有多种推荐的食谱，和热量搭配
   3. 用户可以选择其中的一种进行烹饪



1. 热量测定
   1. 用户选择自己想吃的食谱进行烹饪后
   2. 拍照上传，系统会告诉用户根据制定的目标所吃的量的多少



1. 打卡签到
   1. 用户每餐吃完后，可以进行打卡签到
   2. 达到一定的天数或者连续多少天，可得到一定的奖励
   3. 用户可以从日历上得知自己的目标



1. VIP系统
   1. 用户可选择购买VIP服务
   2. VIP服务可以根据用户的口味进行个性化推荐
   3. VIP服务可以扩展收藏视频的数量和关注者的数量
   4. VIP服务可以每星期拿到点单优惠卷
   5. VIP用户可以免费配送拿到测量仪器，更精准的了解自己吃的热量多少

## 错误代码

*[详细列出错误处理机制所能处理的所有错误代码及其含义]*

## 错误日志

*[说明能够被处理和记录的错误类型和记录方式。]*

## 部署视图

1. 硬件环境：
   1. 需要互联网
   2. 至少需要一台服务器
2. 软件：
   1. 需要安卓和ios的操作系统。且与数据库进行连接，使用SQL SERVER可以换成EXPRESS版。
   2. 需要Apache2.2作为WEB服务器。
   3. 需要安装PHP解释器
3. 配置：
   1. 因为PHP5.3以上的版本已经不支持mssql扩展，所以需要自行下载微软提供的The SQL Server Driver for PHP(SQLSRV20.EXE)
   2. 下载后解压缩，将所有的.dll文件拷贝到php安装目录的ext文件夹中，然后打开php.ini，添加以下语句打开php\_sqlsrv和php\_pdo\_sqlsrv扩展：

[PHP\_PDO\_SQLSRV]

extension=php\_pdo\_sqlsrv\_53\_ts\_vc6.dll

[PHP\_SQLSRV]

extension=php\_sqlsrv\_53\_ts\_vc6.dll

* 1. 安装Microsoft SQL Server 2008 R2 Native Client，可以去http://msdn.microsoft.com/en-us/library/cc296170(SQL.90).aspx下载安装，因为微软的这个扩展包需要它的支持。（注SQL SERVER 2005的客户端也不行，必须要2008的）

1. 部署：
   1. 配置PHP.ini,关闭错误提示。
   2. 配置PHP连接MS SQL的文件，修改IP地址、数据库的用户名和密码。
   3. 将程序包放在Apache安装目录中得htdocs目录下。

# 3数据库设计（安博洋）

|  |
| --- |
| notice |
| nid Integer  from Integer  to Integer  title Variable characters (50)  content Variable characters (1024)  time Date & Time |
|  |

## 3.1逻辑模型

|  |
| --- |
| admin |
| admin\_id <pi> Integer <M>  loginname Variable characters (20) <M>  password Characters (32) <M> |
| Identifier\_1 <pi> |

sendby

send

Relationship\_1

sendto

|  |
| --- |
| user |
| user\_id <pi> Integer <M>  login\_name Variable characters (50) <M>  password Characters (32) <M>  sex Characters (4) <M>  age Integer <M>  job Variable characters (50) <M>  interest Variable characters (50) <M>  telephone Variable characters (12) <M>  address Variable characters (50) <M>  sickness Variable characters (50) <M> |
| Indentifier\_1 <pi> |

|  |
| --- |
| restaurant |
| rid <pi> Integer <M>  manager Variable characters (32) <M>  name Variable characters (50) <M>  districtid Integer  address Variable characters (200) <M>  business\_hours Variable characters (50) <M>  telephone Variable characters (12) <M>  image Variable characters (20) <M>. |
| Identifier\_1 <pi> |

|  |
| --- |
| vip |
| vip\_id <pi> Integer <M>  rank Integer <M>  ideal\_weight Characters (8) <M>  plan\_time Variable characters (20) <M> |
| Indentifier\_1 <pi> |

|  |
| --- |
| takeout\_orders |
| takeout\_id <pi> Integer <M>  user\_id Integer  rid Integer  food Variable characters (50)  time Date & Time  price Float |
| Indentifier\_1 <pi> |

|  |
| --- |
| district |
| district\_id Integer  districtname Variable characters (5) <M> |
|  |

Relationship\_2

receive

include belongto

Relationship\_3

has

Relationship\_4

belongto

belongto

has Relationship\_5

has Relationship\_6

belongto

## 3.2物理模型

1、为每个表初始设置400M的存储空间，以10%的大小扩展。

2、为上传的图片预留空间。

# 4质量及其他方面（潘保恒）

## 可维护性

1. 通过程序注释等方式增加代码的可读性和可维护性。
2. 将数据访问层分离，做成一个个函数，由其它层调用，以增加代码的可维护性。
3. 数据库有日志记录，系统一旦出现故障有恢复到故障之前的信息和数据的能力

## 安全性

1. 密码使用md5加密
2. 对用户的输入进行验证
3. 对用户的输入的特殊字符进行转义，防止sql注入攻击

## 可扩展性

1. 可以通过增加硬件资源的方式提高系统的响应速度。
2. 可以通过修改代码，扩充系统的功能。

## 可靠性

1. 数据库按时备份，有日志记录

## 可用性

1、做好充分的前期工作，系统一旦投入使用，尽量减少宕机的次数和时间

## 性能设计

*[系统性能通常使用事务处理能力或资源利用率来度量。确定系统需求的方法：*

*识别约束：包括预算、时间、基础结构、可选的开发工具和技术；*

*确定功能特性：要符合使用场景和用例*

*确定负载：使用本系统的客户数量]*