做啥吃设计规格说明

Ver:1.0

**目 录**

[1 引言 3](#_Toc38272223)

[1.1 目标 3](#_Toc38272224)

[1.2 文档范围 3](#_Toc38272225)

[1.3 术语和缩略语 3](#_Toc38272226)

[1.4 参考资料 3](#_Toc38272227)

[1.5 系统目标和约束 3](#_Toc38272228)

[2 系统设计 4](#_Toc38272229)

[2.1 系统架构概述 4](#_Toc38272230)

[2.2 对象模型 4](#_Toc38272231)

[2.3 接口 4](#_Toc38272232)

[2.4 特性实现 4](#_Toc38272233)

[2.5 错误代码 10](#_Toc38272234)

[2.6 错误日志 10](#_Toc38272235)

[2.7 部署视图 11](#_Toc38272236)

[3 数据库设计 12](#_Toc38272237)

[3.1 逻辑模型 12](#_Toc38272238)

[3.2 物理模型 12](#_Toc38272239)

[4 质量及其他方面 13](#_Toc38272240)

[4.1 可维护性 13](#_Toc38272241)

[4.2 安全性 13](#_Toc38272242)

[4.3 可扩展性 13](#_Toc38272243)

[4.4 可靠性 13](#_Toc38272244)

[4.5 可用性 14](#_Toc38272245)

[4.6 性能设计 14](#_Toc38272246)

[5 附录 14](#_Toc38272247)

[5.1 附件 14](#_Toc38272248)

[5.2 修过记录 14](#_Toc38272249)

# 引言

## 目标

在完成软件开发前期的准备工作如项目需求等，结合《需求确认书》，，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做啥吃的功能分配，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

内容搜索，视频教学，食材购买，动态发布

## 参考资料

部分内容参考了《做啥吃需求确认书》、《技术分析》

## 系统目标和约束

系统目标：项目需要完成视频教学，食物浏览，食材购买，作品发布，用户关注，搜索对应的食物等基本功能。

系统的约束：因为现有技术水平和时间限制，该app只实现最基本的功能。

# 系统设计

## 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。采用面过程的设计方法。

## 对象模型

该系统采用非面向对象的设计方法。

## 接口

*[详细说明本系统内部每个接口的每个方法的定义。*

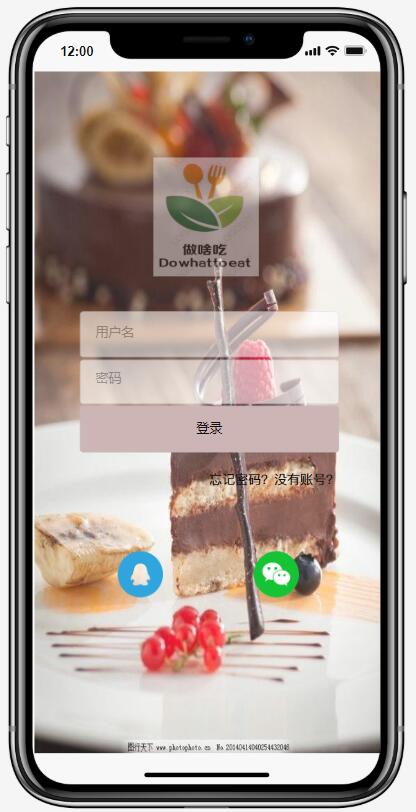
*注意：如果本系统需要和其他系统交互，则应该将交互的接口协议单独成册，而不是写入本节中。]*

## 特性实现

1. 用户注册
   1. 用户注册页面
   2. 注册成功，进入登录页面



1. 登录
   1. 登陆页面
   2. 登记成功，用户选择功能





1. 用户搜索食物
   1. 用户输入要搜索的名字
   2. App为其找到对应的食物



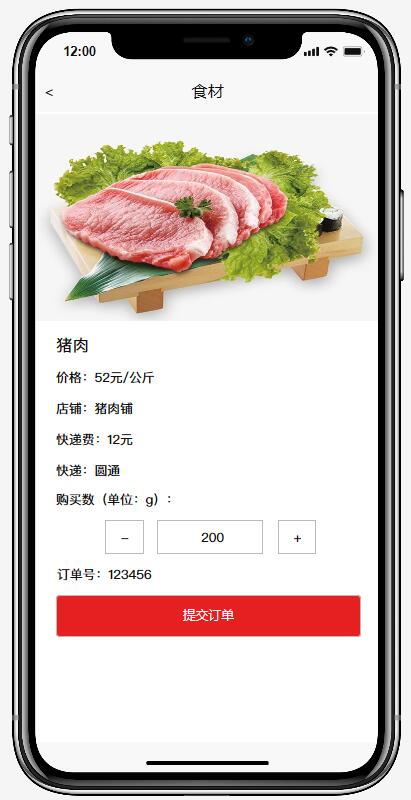
1. 视频学习
   1. 用户找到想学的食物做法
   2. 进入学习



1. 动态发布
   1. 进入我的页面
   2. 选择动态发布



1. 食材购买
   1. 用户找到想学的食物
   2. 进入详情即可购买所缺的食材



## 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

网络不通畅，导致内容无法检索

网络不通畅，导致视频无法播放，动态无法发布

服务器压力过大，系统崩溃

由于某些问题，无法购买食材

## 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. 需要安卓系统或苹果系统，以及其他主流手机操作系统。
   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. 安装任一主流操作系统

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。

# 数据库设计

## 逻辑模型



## 物理模型

1. 为每个表初始设置300M的存储空间，以10%的大小扩展。
2. 为上传的图片预留空间。

# 质量及其他方面

## 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
4. 将功能模块化，以提高可维护性。

## 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
4. 使用验证码，保证操作安全

## 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。
3. 丰富数据库数据，提高用户的用户享受

## 可靠性

1. 数据库按时备份，有日志记录
2. 提高数据库的安全性，设置访问权限
3. 定期进行数据库的维护

## 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]*

# 附录

## 附件

*[列出支撑本文档的附属文档的名字。如EA文件、绘制系统架构(如有)的viso文件，powerdesigner(如有)文件等]*

## 修过记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 修改人 | 修改时间 | 修改内容 | 核准 |
| 1.0 |  |  |  |  |