美食菜谱详细设计说明

Ver:1.0

**目 录**

[1 引言 3](#_Toc38125355)

[1.1 目标 3](#_Toc38125356)

[1.2 文档范围 3](#_Toc38125357)

[1.3 术语和缩略语 3](#_Toc38125358)

[1.4 参考资料 3](#_Toc38125359)

[1.5 系统目标和约束 3](#_Toc38125360)

[2 系统设计 4](#_Toc38125361)

[2.1 系统架构概述 4](#_Toc38125362)

[2.2 对象模型 4](#_Toc38125363)

[2.3 接口 4](#_Toc38125364)

[2.4 特性实现 4](#_Toc38125365)

[2.5 错误代码 20](#_Toc38125366)

[2.6 错误日志 20](#_Toc38125367)

[2.7 部署视图 20](#_Toc38125368)

[3 数据库设计 22](#_Toc38125369)

[3.1 逻辑模型 22](#_Toc38125370)

[3.2 物理模型 22](#_Toc38125371)

[4 质量及其他方面 22](#_Toc38125372)

[4.1 可维护性 22](#_Toc38125373)

[4.2 安全性 23](#_Toc38125374)

[4.3 可扩展性 23](#_Toc38125375)

[4.4 可靠性 23](#_Toc38125376)

[4.5 可用性 23](#_Toc38125377)

[4.6 性能设计 23](#_Toc38125378)

[5 附录 24](#_Toc38125379)

[5.1 附件 24](#_Toc38125380)

[5.2 修过记录 24](#_Toc38125381)

# 引言

## 目标

在完成软件开发前期的准备工作如项目需求等，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美食菜谱系统的功能分配，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

## 参考资料

## 系统目标和约束

系统目标：项目需要完成发布帖子，添加我的喜欢，喜欢收藏，登录注册，购物，论坛发布和评论点赞功能。

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

## 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。

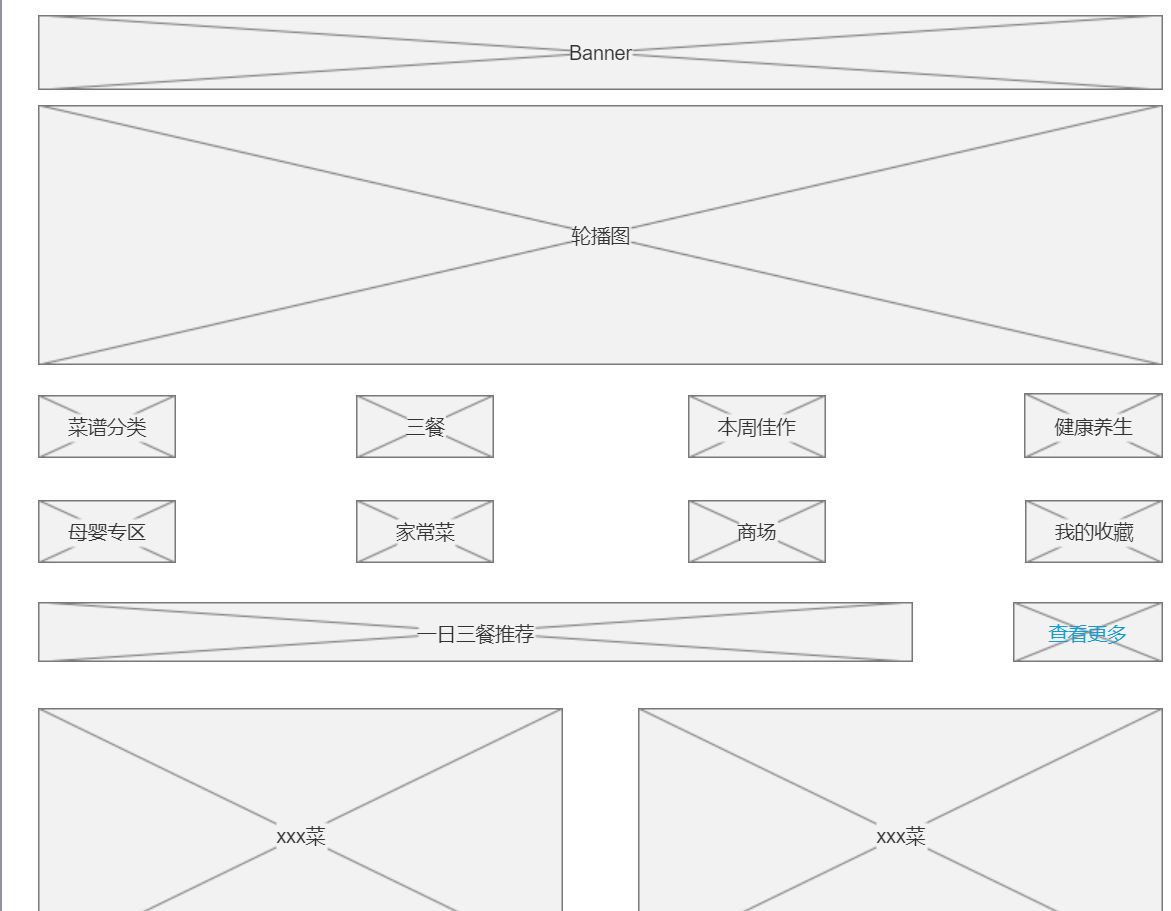
## 接口

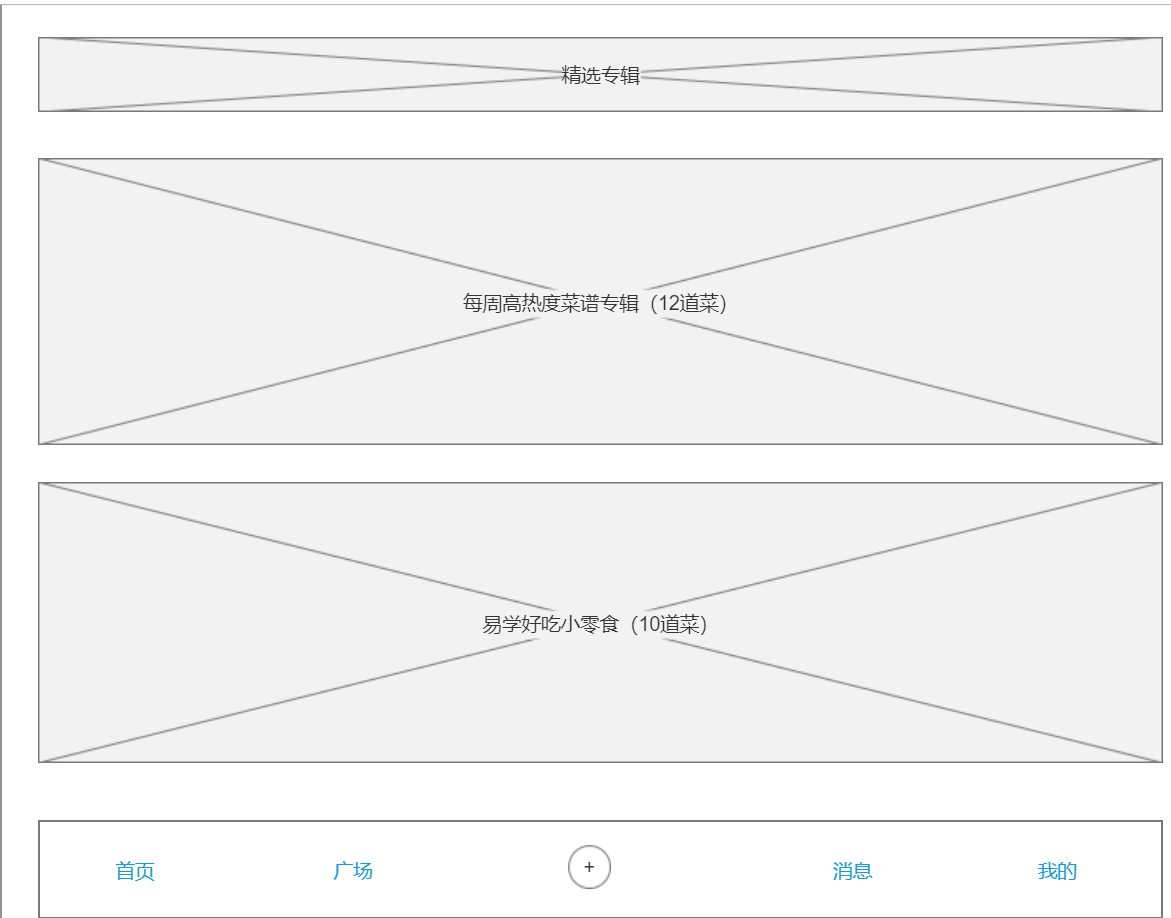
*[详细说明本系统内部每个接口的每个方法的定义。*

*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
2. 用户注册页面
3. 注册成功，可进入首页
4. 查看不同模块的内容，进入二级页面





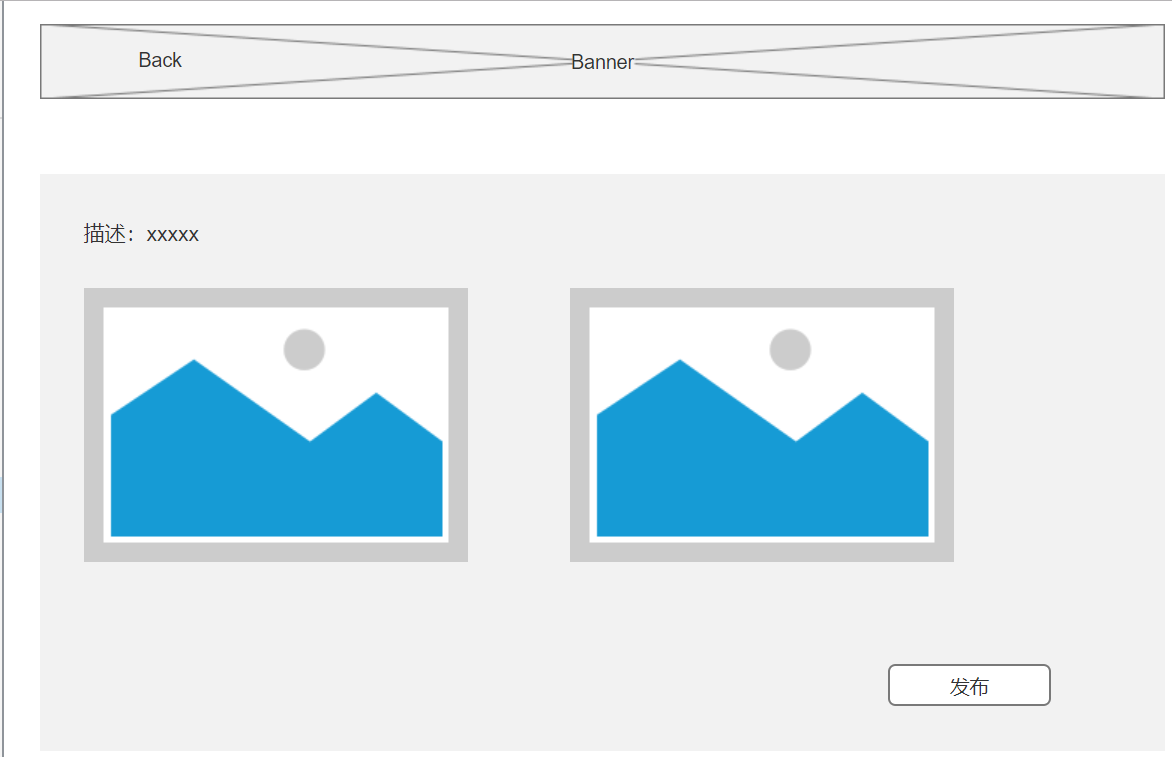
1. 用户切换到广场页面
2. 进入到广场页面
3. 用户可以浏览其他用户上传的美食视频或者图片



1. 查看制作美食详细步骤
2. 用户点击每一个美食食谱
3. 列出美食的制作过程



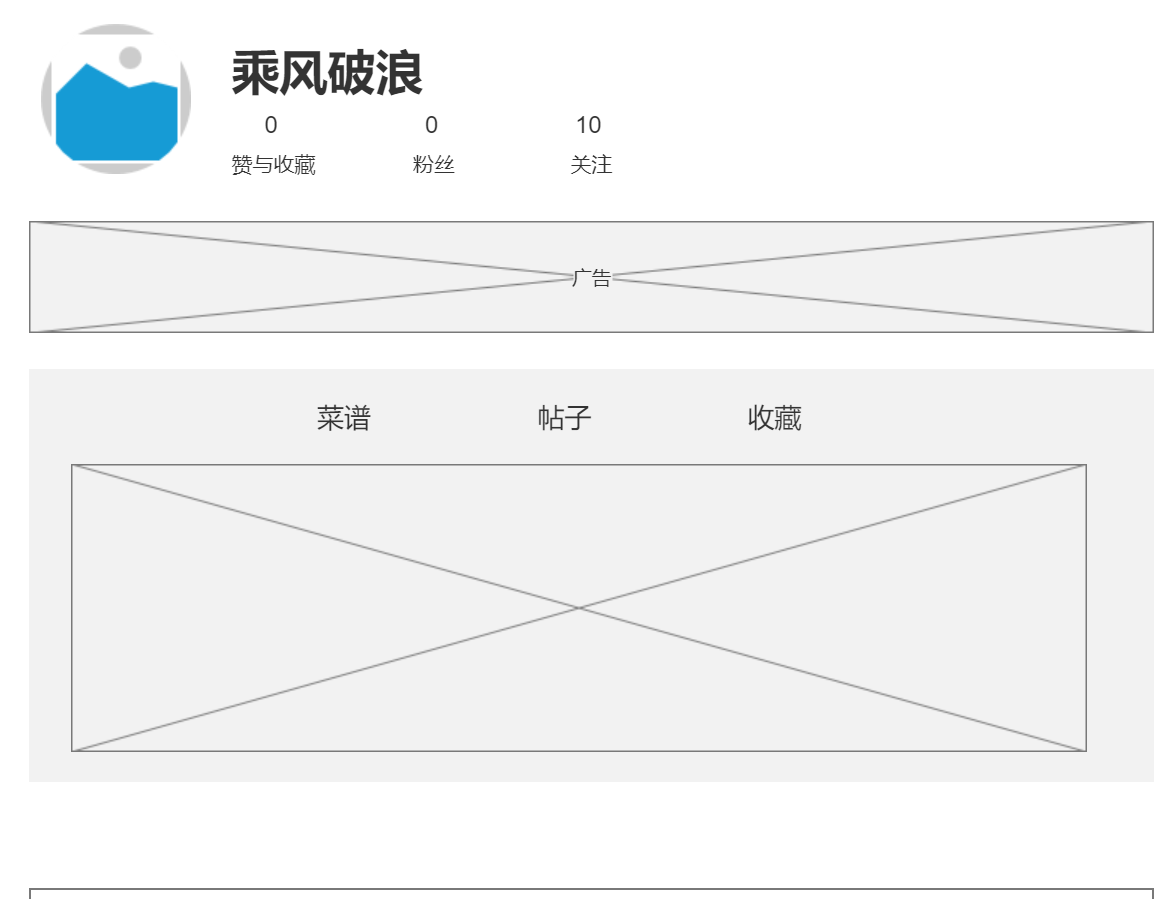
1. 上传自己的美食食谱
2. 进入到页面
3. 填写所需的信息和图片，点击发布



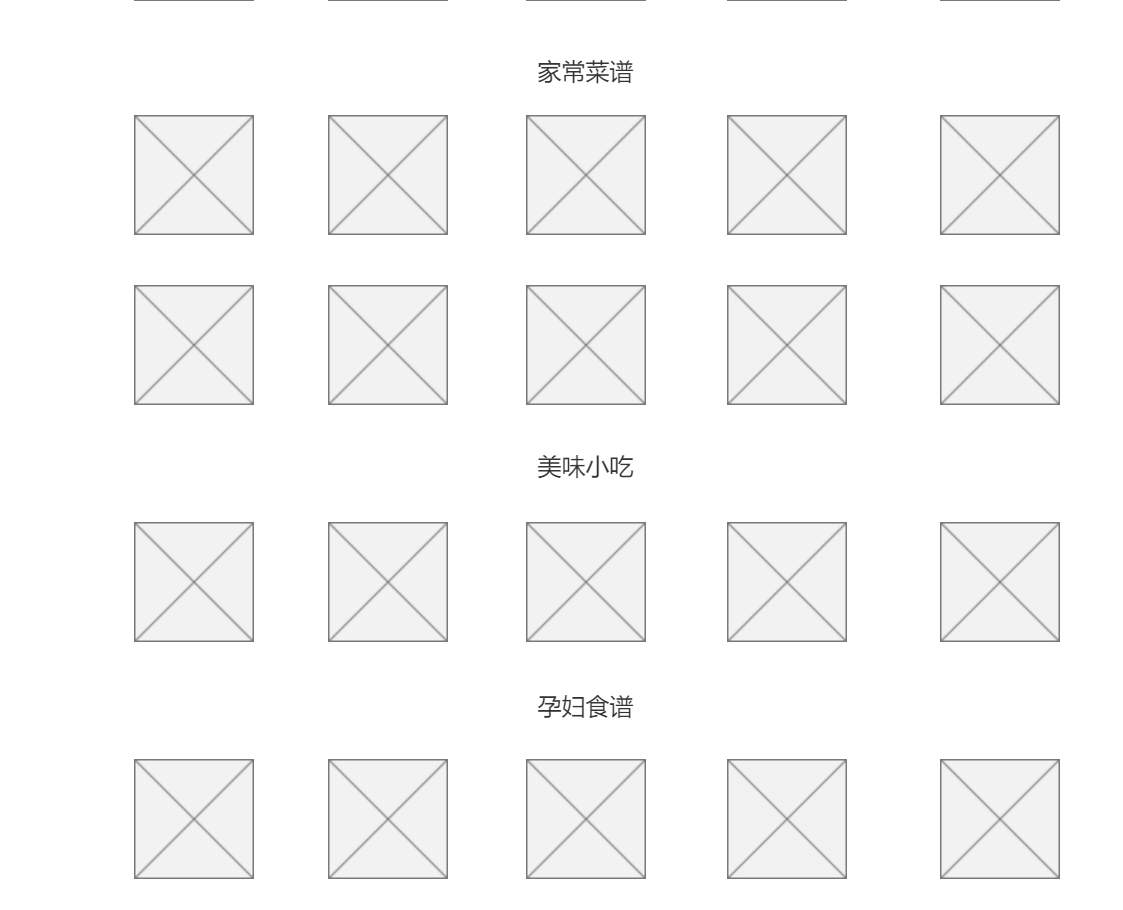
1. 查看消息
2. 点击消息
3. 进入消息页面



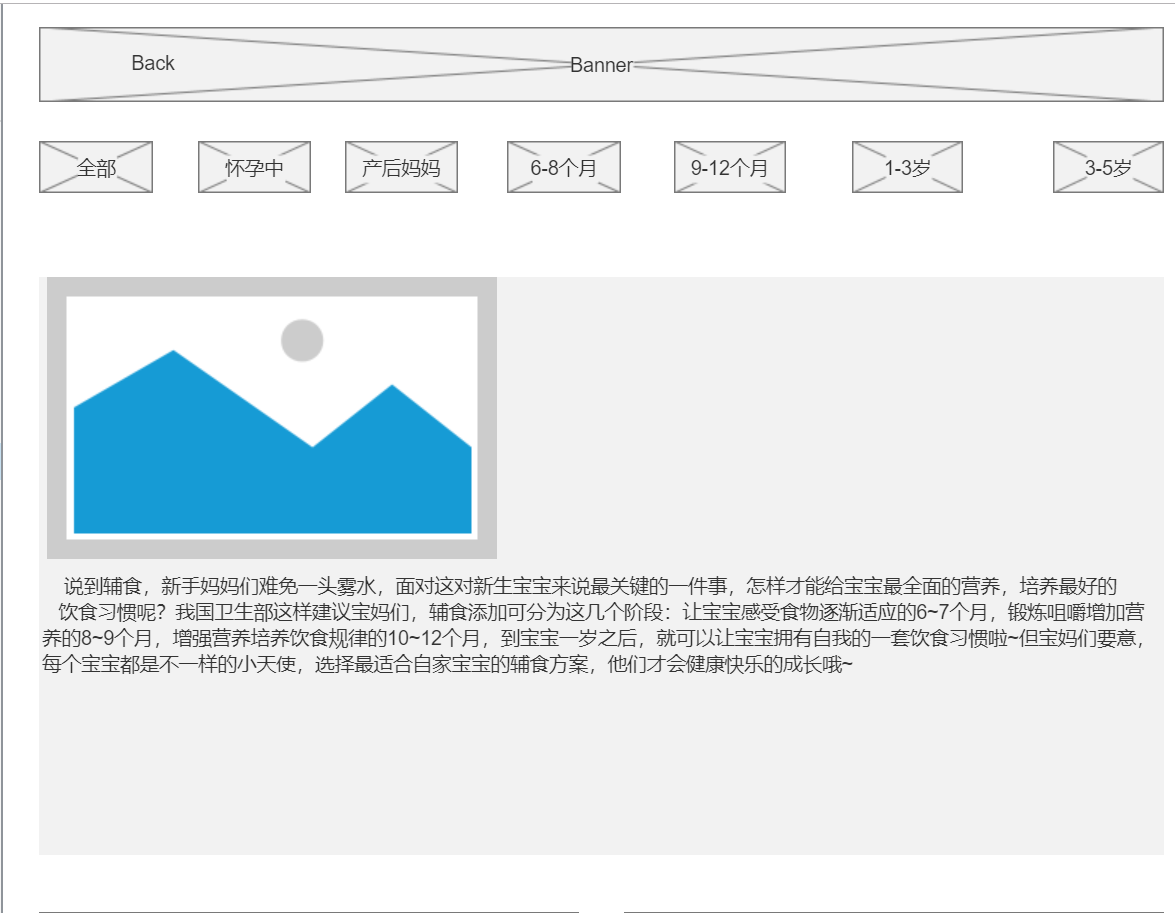
1. 查看我的信息或者修改个人信息
2. 进入页面



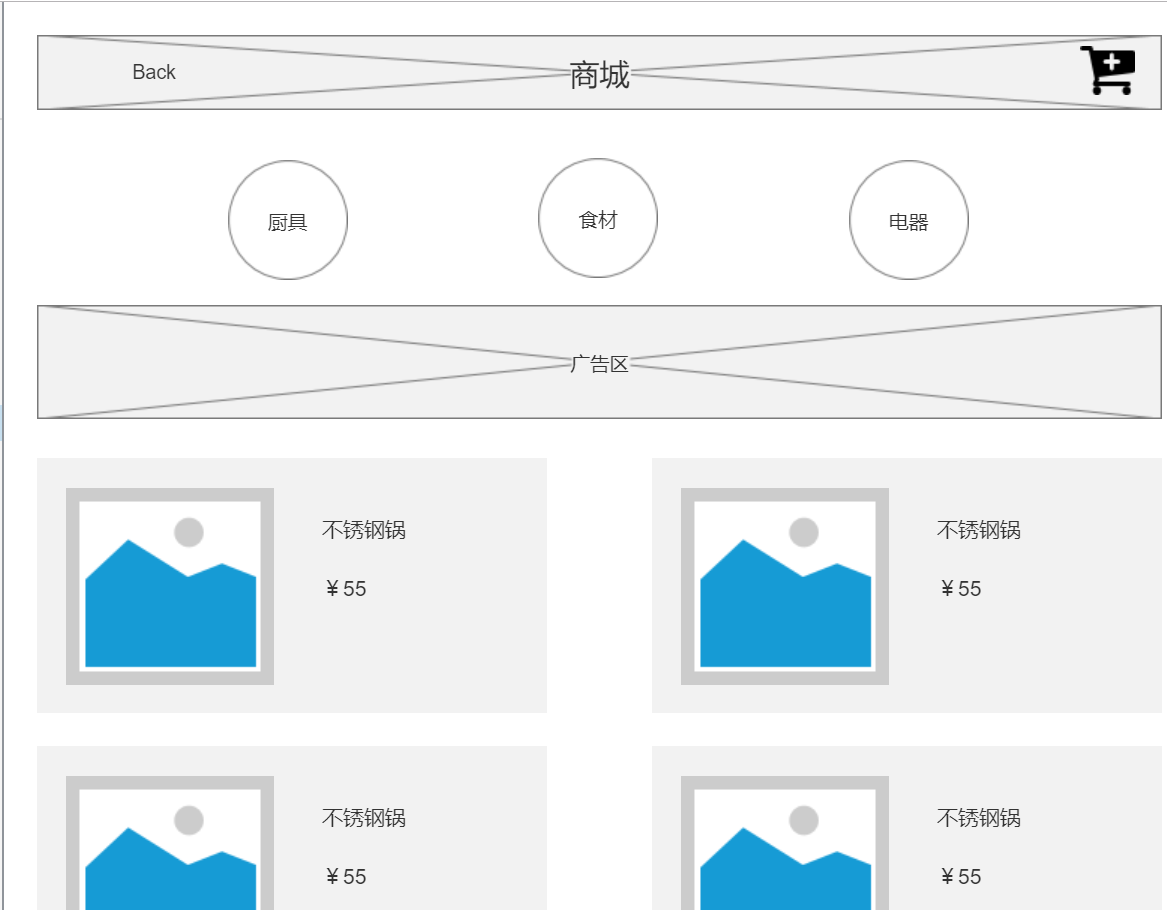
1. 查看菜谱分类
2. 进入首页
3. 点击菜谱分类
4. 进入菜谱分类页面，选择合适的分类



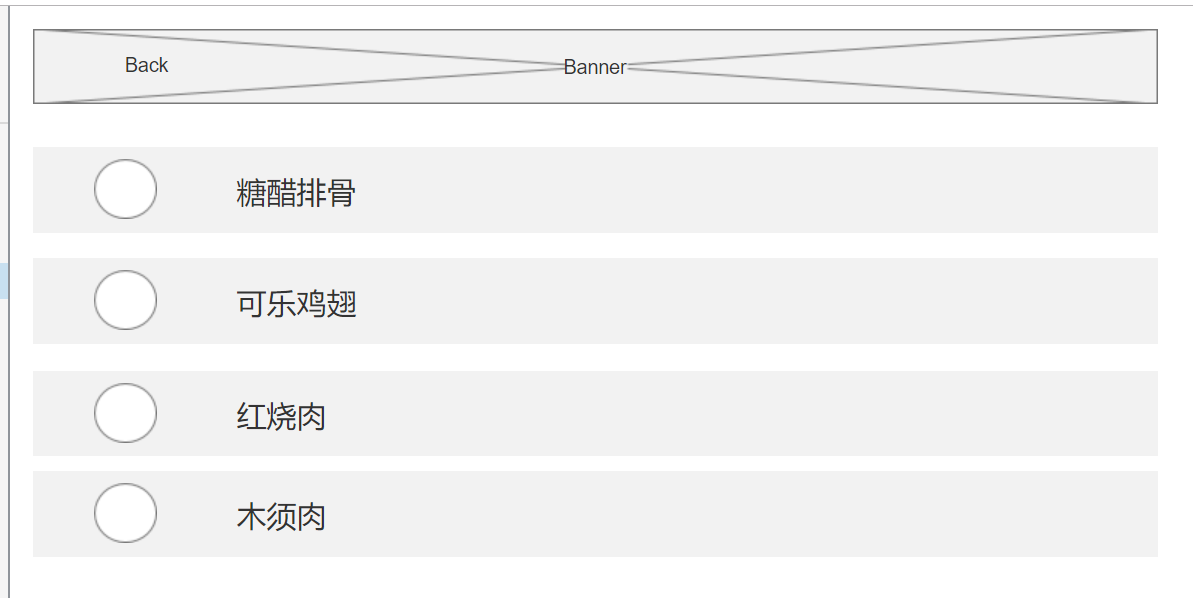
1. 关于孕妇的食谱
2. 进入首页
3. 点击母婴专区
4. 查看相关内容



1. 选购厨具
2. 进入首页
3. 点击商场
4. 选购心爱的物品



1. 查看自己感兴趣的食谱
2. 进入首页
3. 点击我的收藏
4. 查看用户收藏过得食谱



1. 指定合适的健康食谱
2. 进入首页
3. 点击健康养生
4. 进入页面，选择合适的信息，点击开始指定健康餐按钮



## 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。]*

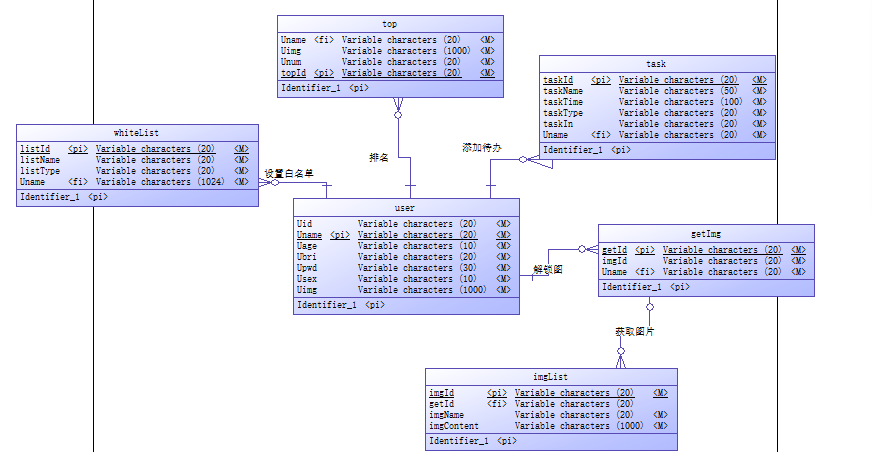
## 部署视图

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Windows Server 2003或其他Server版的操作系统以安装SQL SERVER2005企业版。或者其他操作系统，但是使用SQL SERVER就不能使用企业版了，可以换成EXPRESS版。
   2. 需要阿里云作为WEB服务器。
   3. 需要安装vscode
3. 配置：

配置vscode以及node环境

# 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

## 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]*

# 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 | 李沛伦 | 2020-4-18 |  |  |
| 1.0 | 韩雅宁 | 2020-4-18 |  |  |