闲逛Recycle设计规格说明

Ver:1.0

**目 录**

[1 引言 3](#_Toc301301027)

[1.1 目标 3](#_Toc301301028)

[1.2 文档范围 3](#_Toc301301029)

[1.3 术语和缩略语 3](#_Toc301301030)

[1.4 参考资料 3](#_Toc301301031)

[1.5 系统目标和约束 3](#_Toc301301032)

[2 系统设计 4](#_Toc301301033)

[2.1 系统架构概述 4](#_Toc301301034)

[2.2 对象模型 4](#_Toc301301035)

[2.3 接口 4](#_Toc301301036)

[2.4 特性实现 4](#_Toc301301037)

[2.5 错误代码 9](#_Toc301301038)

[2.6 错误日志 9](#_Toc301301039)

[2.7 部署视图 9](#_Toc301301040)

[3 数据库设计 11](#_Toc301301041)

[3.1 逻辑模型 11](#_Toc301301042)

[3.2 物理模型 11](#_Toc301301043)

[4 质量及其他方面 12](#_Toc301301044)

[4.1 可维护性 12](#_Toc301301045)

[4.2 安全性 12](#_Toc301301046)

[4.3 可扩展性 12](#_Toc301301047)

[4.4 可靠性 12](#_Toc301301048)

[4.5 可用性 12](#_Toc301301049)

[4.6 性能设计 13](#_Toc301301050)

[5 附录 13](#_Toc301301051)

[5.1 附件 13](#_Toc301301052)

[5.2 修过记录 13](#_Toc301301053)

# 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闲逛Recycle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

部分内容参考了《闲逛Recycle需求确认书》

## 系统目标和约束

系统目标：项目需要完成网上身份认证，卖闲置，消息回复，购物车，我的设置等基本的商品管理。

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。

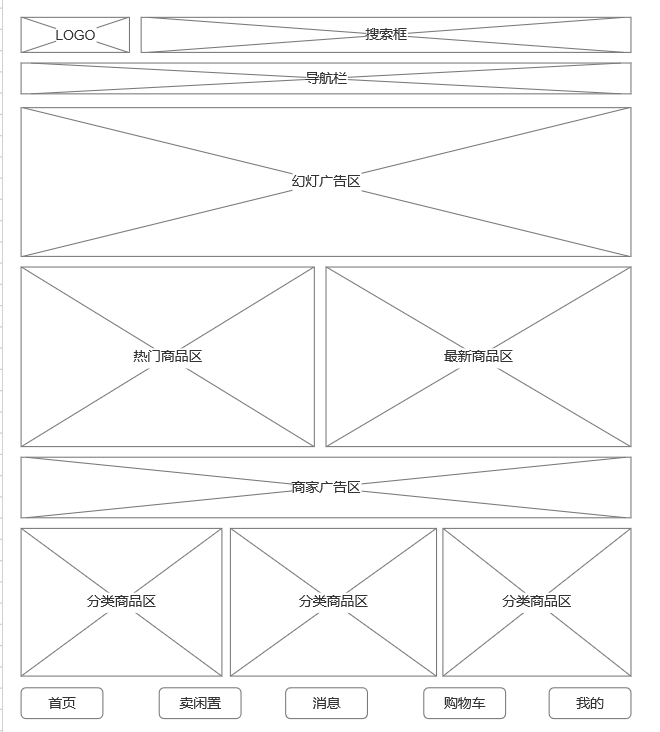
## 接口

*[详细说明本系统内部每个接口的每个方法的定义。*

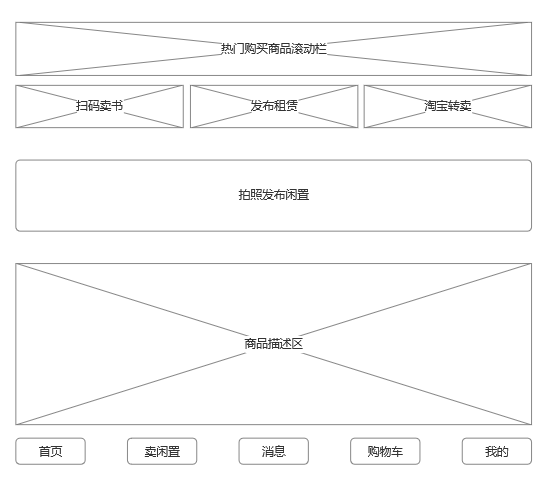
*注意：如果本系统需要和其他系统交互，则应该将交互的接口协议单独成册，而不是写入本节中。]*

## 特性实现

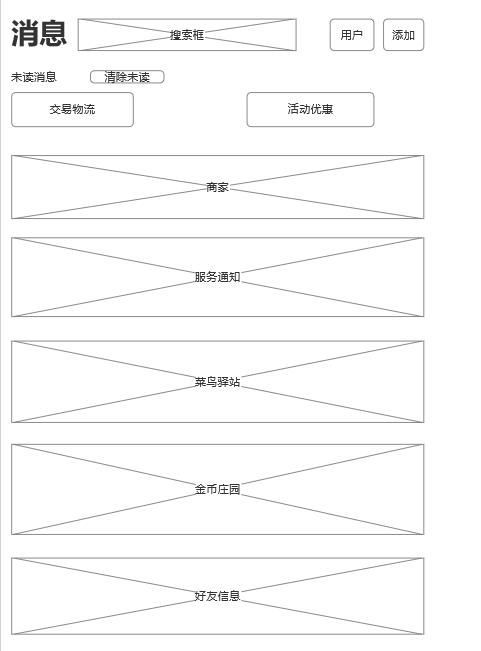
1. 首页
   1. 用户打开闲逛首页的页面
   2. 用户填入自己的个人信息，注册账号
   3. 注册成功，用户选择自己感兴趣的模块进行浏览



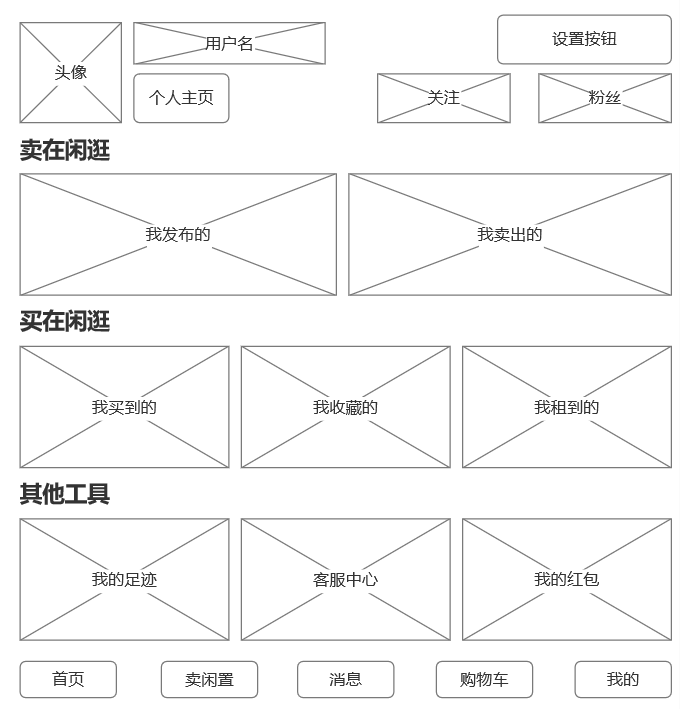
1. 卖家登记自己的商品信息
   1. 选择要出手的商品种类
   2. 拍照发布以及填写描述商品详情
   3. 热卖商品滚动点击进入查看推荐的热卖商品



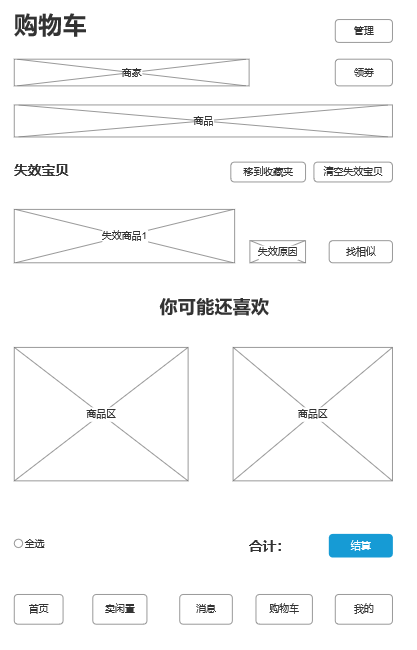
1. 消息处理
   1. 未读已读消息提醒
   2. 常用的合作绑定订阅号如菜鸟驿站、金币庄园的推送
   3. 优惠活动的提醒，金币的收取
   4. 交易物流查看订单以及运输状态
   5. 快捷搜索消息以及消息内容，快速清除未读功能



1. 用户通过‘我的’查看自己的足迹
2. 用户的个性化设计，比如头像，个人主页等
3. 查看买卖记录搜索订单， 有疑问可以详询客服中心
4. “我”关注的店铺或者其他好友以及我的粉丝数
5. 点击足迹查找曾经浏览过的商品
6. 点击红包查看可使用的红包



1. 用户查看购物车的信息
   1. 查看有意向的商家、商品
   2. 查看失效宝贝以及失效原因，再次寻找相似宝贝以及清空或者加入收藏夹
   3. 猜你喜欢功能，帮助用户发现可能感兴趣的商品
   4. 管理所选商品，对勾选的商品进行结算或删除



## 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 2008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。

# 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 | 蒋晓伟 | 2020-4-20 |  |  |