农选设计规格说明

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农选系统的功能分配，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。

## 背景

·APP的名称：农选

·本项目的任务提出者：农选全体人员

·开发项目组名称：XHS-001

·APP的用户对象：

（1）希望买到物美价廉的当地特色农产品的消费者；

（2）拥有优秀当地特色农产品寻求更广阔的消费渠道的中小农户商家

·APP的基本概念：

结合React/React Native，NodeJS，Express，MySQL等技术搭建APP前端后台连接数据库，实现各参与方的协同交流、信息共享；

为千万中小农户连接消费者，直供正宗的物美价廉的当地商品，并且农户不在愁销路。

## 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。

## 定义

名称：详细设计说明书

MySQL：当下最流行的关系数据库管理系统

Express：Express是一个保持最小规模的灵活的Node.js Web应用程序开发框架，为Web和移动应用程序提供一组强大的功能。

NodeJS：Node.js是一个基于Chrome V8引擎的JavaScript运行环境，轻量且高效。

React/React-Native：React-Native完美兼容使用Objective-C、Java或是Swift编写的组件

## 术语和缩略语

## 参考资料

《问题描述》、《产品愿景和商业机会》、《用户分析》、《技术分析》、《资源需求估计》、《风险分析》《产品构思》、《项目范围说明书》、《界面原型》

## 系统目标和约束

系统目标：项目需要完成登陆注册，首页，产品分类，购物车，我的，我要开店和店铺管理等基本的功能。

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

### 注册模块

#### 描述

用户使用APP的前提条件，如果该用户第一次下载此APP，则需要使用手机号接收验证码的形式注册账号，如果该用户已有账号，则可直接登录

#### 功能

用于数据库存储用户信息，实现相应操作

### 登录模块

#### 描述

用户使用APP的前提条件，该用户有账号时，可通过此模块让用户直接登录，有账号密码登录和手机号快速登录两种功能供用户选择。

#### 功能

对进入APP的用户进行信息的排查

### 商品模块

#### 描述

显示商品分类信息，用户可通过不同分类，寻找自己心仪的商品。

#### 功能

显示所有商品分类信息，以及相应的详情页

### 买家模块

#### 描述

用户可对商品进行浏览、购买、收藏，对个人信息进行设置，对订单进行查询

#### 功能

加入购物车功能，查询订单功能，个人中心设置功能

### 卖家模块

#### 描述

可开店、发布商品、对商品进行管理

#### 功能

上传身份证明功能、身份确定功能、信息提交功能、发布上线功能、店铺管理功能

## 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。

# 数据存储设计描述

## 结构化数据存储描述

用于存储来自客户端的业务数据，并满足对所存储数据的安全、检索、统计等需求。

结构化数据常使用数据库来存储，数据库目前主要分为以下两大类：

a.关系型数据库（SQL）：建立在关系模型基础上的数据库，借助于集合代数等数学概念和方法来处理数据库中的数据。现实世界中的各种实体以及实体之间的各种联系均用关系模型来表示，可以用SQL语句方便的在一个表以及多个表之间做非常复杂的数据查询；关系型数据库支持复杂的事务管理，对于一致性要求较高的操作，常借助于数据库的事务管理保证程序操作的一致性，常见的有：SQLServer（付费）、Oracle（付费）、MySQL、DB2（付费）等；

b.非关系型数据库（NOSQL）：其数据存储可以不固定的表格模式，也经常会避免使用SQL的JOIN操作，一般有水平可扩展性的特征，由于减少了对数据依赖关系的管理，对数据管理的重点放在了存储和检索上，因此相对SQL数据库，NOSQL数据库具备相抵较高的性能以及横向扩展能力，常用的有MongoDB、ElasticSearch等。

根据业务场景的需要，BIM数据中心的建设选用主流的关系型数据库和非关系型数据库进行搭建。

## 非结构化数据存储描述

用于存储文件、图片、视频等非结构化数据，并满足用户对这类数据的检索，存取，由于非结构化数据存储要解决的关键问题是：

单个数据集，也既文件比较大；

对单个文件的写入读取，依赖于硬盘的读写速度，单个硬盘的读写是有极限的；

## 安全性设计

对用户密码、信息和商家信息进行加密

设计数据完整性检验条件

# 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]*

# 尚未解决的问题

（1）订单支付问题：如何与支付平台取得连接；以及如何给用户提供多种支付方式；

（2）发送验证码问题：如何给买家的手机号发送验证码。