农副产品交易平台

概

要

设

计

|  |  |
| --- | --- |
| **项目名称** | 农副产品交易平台 |
| **文档撰写** | 常冬梅 |
| **建立日期** | 2020年4月1日 |

**目录**

[1. 概述 3](#_Toc501005194)

[2. 系统架构图 3](#_Toc501005195)

[3. 系统逻辑架构图 4](#_Toc501005196)

[4. 物理拓扑图 4](#_Toc501005197)

[5. 部署结构图 5](#_Toc501005198)

[6. 关键技术 5](#_Toc501005199)

[6.1 文件上传 7](#_Toc501005200)

[6.2 定位 9](#_Toc501005201)

[6.3 分页与搜索 10](#_Toc501005202)

[6.4 Cookie 13](#_Toc501005203)

[7. 关键模块设计 15](#_Toc501005199)

[7.1 部分功能界面 15](#_Toc501005203)

[7.2 数据库设计 16](#_Toc501005203)

[7.3 关键功能模块 17](#_Toc501005203)

[7.4 业务逻辑 18](#_Toc501005203)

# 

# 概述

本文档用于说明农副产品交易平台项目的概要设计，旨在帮助开发人员了解系统的整体架构，关键技术和关键模块的设计，帮助开发人员了解系统的设计概况，为详细设计做铺垫。

# 系统架构图

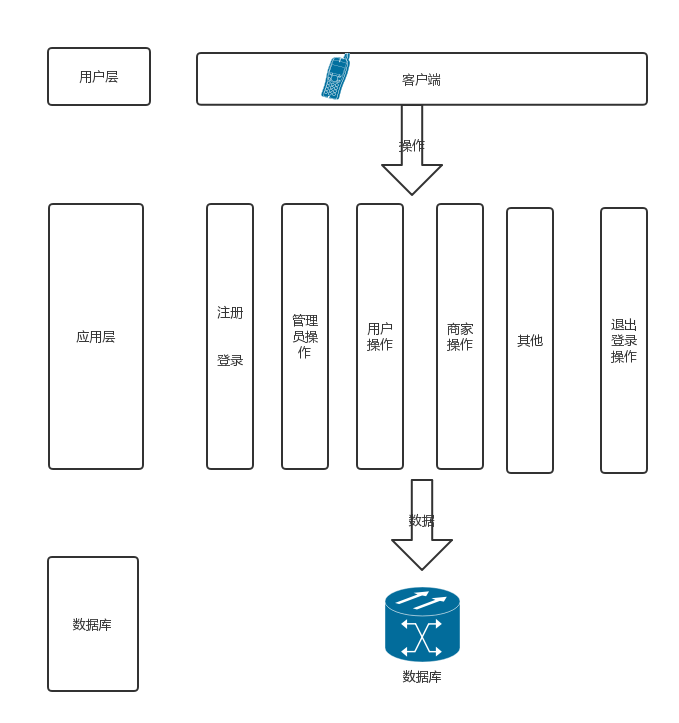


图2-1

# 

# 系统逻辑架构图

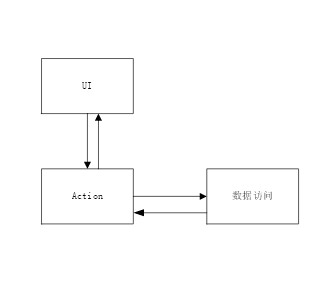


图3-1

# 物理拓扑图

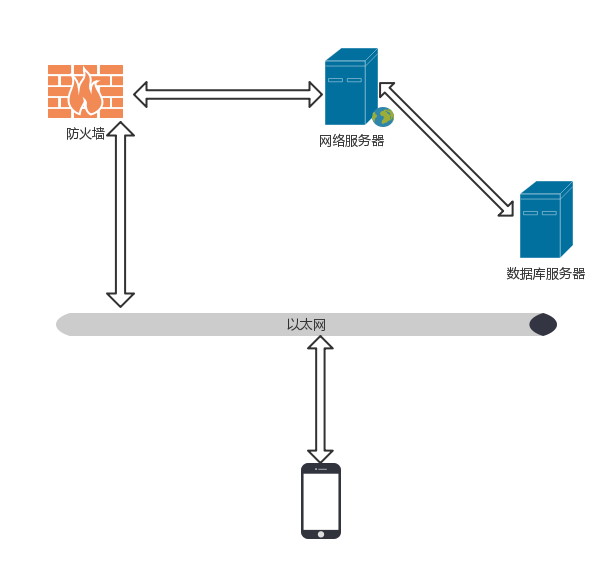


图4-1

# 部署结构图

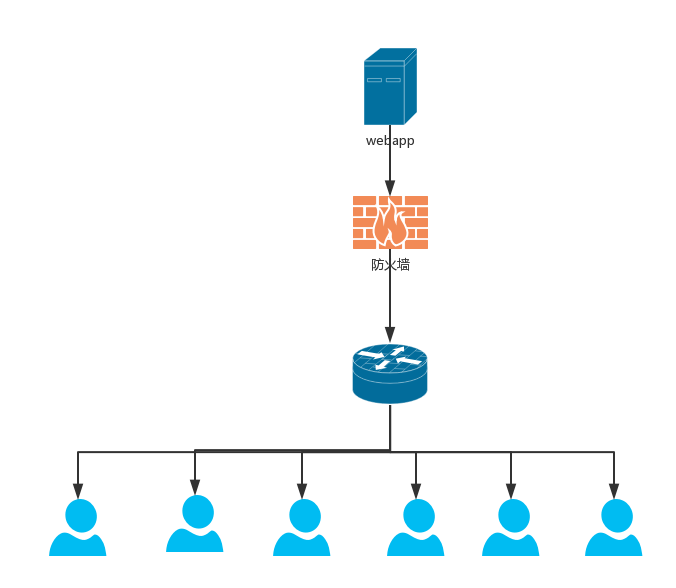


图5-1

# 关键技术

为了提高项目开发效率，项目采用了国内目前流行的框架SSM框架作为项目开发框架，这样就实现了前后端分离，提高了项目开发速度，缩短了开发周期。

SSM全名是Spring Springmvc Mybatis，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。SSM被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中 。

Model（模型）是应用程序中用于处理应用程序数据逻辑的部分。通常模型对象负责在数据库中存取数据。

View（视图）是应用程序中处理数据显示的部分。通常视图是依据模型数据创建的。

Controller（控制器）是应用程序中处理用户交互的部分。通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。

**SSM使用优势：**

**耦合性低** 视图层和业务层分离，这样就允许更改视图层代码而不用重新编译模型和控制器代码，同样，一个应用的业务流程或者业务规则的改变只需要改动MVC的模型层即可。因为模型与控制器和视图相分离，所以很容易改变应用程序的数据层和业务规则。

**重用性高** 随着技术的不断进步，需要用越来越多的方式来访问应用程序。MVC模式允许使用各种不同样式的视图来访问同一个服务器端的代码，因为多个视图能共享一个模型，它包括任何WEB（HTTP）浏览器或者无线浏览器（wap）。

**生命周期成本低** MVC使开发和维护用户接口的技术含量降低。

**部署快** 使用MVC模式使开发时间得到相当大的缩减，它使程序员（Java开发人员）集中精力于业务逻辑，界面程序员（HTML和JSP开发人员）集中精力于表现形式上。

**可维护性高** 分离视图层和业务逻辑层也使得WEB应用更易于维护和修改。

## 文件上传

创建一个文件上传表单允许用户从表单上传文件是非常有用的。请看下面这个功上传文件的HTML表单：

注释：允许用户上传文件是一个巨大的安全风险。请仅仅允许可信的用户执行文件上传操作。

这是一种非常简单文件上传方式。基于安全方面的考虑，可以适当增加有关什么用户有权上传文件的限制。

## 定位

HTML5提供了地理位置定位功能（Geolocation API），能确定用户位置，我们可以借助HTML5的该特性开发基于地理位置信息的应用。我们可以借助百度、谷歌地图接口来获取用户准确的地理位置信息。通过地理定位，使得收货地址更准确、更方便。提高了用户的体验。

## 分页与搜索

所谓分页显示，也就是讲数据库中的结果集，一段一段显示出来怎么分段，当前在第几段（每页有几条，当前再第几页）。当商家、订单、商品等逐渐增多时、超过一定数量时，分页与搜索功能的愈发重要。分页使得界面不会过于冗长，影响整体美感；搜索让用户可以快速找到商品，节约时间。

先看效果图：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 用户姓名 | 年龄 | 性别 | 电话 | 地址 |
| 1 | 张三 | 23 | 男 | 15755333300 | 文津西路8号安徽信息工程学院 |
| 2 | 李四 | 21 | 女 | 15755333321 | 阜阳柴集 |
| 3 | 小李 | 20 | 女 | 15755333378 | 阜阳新村 |
| 4 | 小张 | 25 | 男 | 15755333360 | 合肥 |
| 当前1/10页 共4条 首页 上一页 下一页 尾页 | | | | | |

以下是分页的实现内容：

//分页的实现原理

//1.获取数据表中总记录数

//每页显示条数

//总共有几页

Cookie

服务器端为了保存某些数据，或实现某些必要的功能，当用户访问服务器时，从服务器回传到客户端的一个或多个数据，这些数据因设置的保存时间不同，故保存在浏览器内存中或写入用户电脑或手机的内存当中，当下次用户再次访问服务器端时，则带着这些文件去与服务器端进行联系，这些数据或写入内存当中的数据文件就是cookie，本项目中主要采用cookie保存用户名、商家名称和管理员名称。

以下为一个cookie简单例子，用于保存用户名（商家名、管理员名都类似）：

public static void addCookie(String telephone,HttpServletResponse response, HttpServletRequest request) throws UnsupportedEncodingException

{

if(StringUtils.isNotBlank(telephone)) {

//创建cookie

Cookie username = new Cookie("telephone", URLEncoder.encode(telephone, "utf-8"));

//设置cookie路径

username.setPath(request.getContextPath()+"/");

//设置cookie保存的时间 单位：秒

username.setMaxAge(7\*24\*60\*60);

//将cookie添加到响应

response.addCookie(username);

}

}

然后在登录方法中登录成功的时候调用一下就可以了:

result.setSuccess(true);

result.setMsg("登录成功");

session.setAttribute("user", user);

//用户首页

addCookie(telephone, response, request);

return ajaxJsonSuccessMessage("登录成功",response);

# 关键设计模块

## 部分功能界面

1. 首页
2. 我的订单界面
3. 我的购物车界面
4. 管理员界面
5. 用户个人信息界面

## 数据库设计

农副产品交易平台涉及到关键的表有管理员表，商家表，普通用户表，商品表、订单表、购物车表、评价表等。

以下是关键表的设计：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 大小 | 允许为空 | 注释 |
| userID | int | 255 | Not null | ID |
| userName | varchar | 255 | Not null | 用户名 |
| userPhone | int | 255 | Not null | 电话 |
| userAddress | varchar | 255 | Not null | 收货地址 |

表1 普通用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 大小 | 允许为空 | 注释 |
| productId | int | 11 | NOT NULL | 食品编号 |
| userId | int | 11 | NOT NULL | 用户编号 |
| quantity | int | 11 | NOT NULL | 订购数量 |
| totalPrice | decimal | 10,2 | NOT NULL | 总价 |

## 表2 购物车表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 大小 | 允许为空 | 注释 |
| orderId | int | 11 | 自增编号 | 订单编号 |
| userId | int | 11 | NOT NULL | 用户编号 |
| productId | int | 11 | NOT NULL | 食品编号 |
| orderTime | time |  | NOT NULL | 下单时间 |
| totalPrice | decimal | 255 | NOT NULL | 总价 |

表3 订单表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 大小 | 允许为空 | 注释 |
| commentId | int | 255 | 自增编号 | 评论编号 |
| productId | int | 255 | NOT NULL | 食品编号 |
| content | varchar | 255 | NOT NULL | 评论内容 |
| commentTime | time |  | NOT NULL | 评论时间 |

表4评论表

## 关键功能模块

普通用户功能模块：可以查看，修改个人基本信息和登录密码、可以查询商家基本信息，查询店铺信息。把喜欢的东西添加购物车进行下单。

管理员模块：管理员可以查看商家信息，添加，删除商家信息。

商家模块：商家可以对自己的店铺进行管理，包括店铺的基本信息，商家能够对店铺内的商品进行增删改查操作。

## 业务逻辑

1. 用户：可以通过农副产品交易平台将喜欢的物品添加购物车，然后用户可以对购物车中的物品进行增加、删除等操作。

2. 订单：包括订单编号，用户编号，商品编号，下单时间，订单总价。订单和用户属于一对多的关系。

3. 商家：可以查看订单和用户的部分信息，用户购买物品后，商家就可以看到用户下单的时间，订单详情，用户信息。同时商家可以对订单进行删除、修改和查看操作。商家和订单是一对多的关系。