2014-04-21 更新格式

**5**

**2014**

**王 宾**

**2010114090**

**信息管理与信息系统**

**信息管理与工程**

**基于RoR的果园之家电子商务网站开发与设计**

2014-04-21 添加大纲

2014-04-21 修改数据库部分大纲结构，添加少量描述性文字

2014-04-26修改大纲结构，参照已有论文

2014-05-04 注意商品管理部分商品类别，供货管理，供应商管理，配送订单部分，找回密码

2014-05-07 注意dfd图

网站待完善部分:

购买确认和完成页

商品详情页

收藏页

admin

文档:

图。。。。。

找回密码，供货管理，留言评论

绪论一般分为背景与意义, 即对选题作一个简单的需求分析, 指出工作的价值所在。

系统分析与设计

需求分析(功能模块图，用例图，DFD图，）

数据库设计（ER模型设计 - 10， 数据库逻辑设计--表结构，）

软硬件环境设计（硬件，软件，环境，介绍）

网站具体实现（贴网页，文字）

开发环境

技术

技术

综合实现部分

系统测试

基于RoR的果园之家电子商务网站开发与设计

摘 要

随着电子商务的飞速发展, 越来越多的企业将这一交易模式作为其销售、管理甚至展现企业实力的重要手段。电子商务极大地推动了经济的快速发展，其应用和推广为企业和顾客的信息交流带来了极大的便利。电子商务在水果销售市场也将扮演越来越重要的角色。

本文通过对水果电商网站的功能分析，对于销售前台和管理后台的功能模块进行了详细设计，实现了前台用户查询、查看详细信息、购买、订单管理以及个人信息管理功能，后台管理员对商品、会员、订单等的管理功能。本设计使用MVC设计模式进行网站开发设计，使用RoR作为开发框架，编程语言主要使用Ruby和 JavaScript，使用Notepad++作为开发工具，后台数据库使用SQLite。

关键字：电子商务 水果 Ruby Rails MVC

E-commerce website FruitHome

Abstract

With the rapid development of E-Commerce, more and more companies are use this trading pattern as a sales and operation management, and even a way to show their strength. E-Commerce has promoted rapid development of electronic, and its application has brought great benefits on exchange of information for both enterprises and customers. E-commerce will play an more and more important role in fruit sales market.

This project achieves functions such as searching and showing buying fruits and managing orders for front-desk user, managing users` information, order information, fruit information and so on for administrator. This project uses RoR as the development framework, MVC design-pattern to design the website, almost Ruby and JavaScript as programming languages, and notepad++ as development tool, SQLite as background database.

**Key words:** E-Commerce fruit Ruby Rails MVC

目 录

[一、序 言 1](#_Toc387670308)

[（一）电子商务发展现状 1](#_Toc387670309)

[（二）水果电商网站概况 1](#_Toc387670310)

[（三）网上购物系统技术 2](#_Toc387670311)

[二、系统需求分析 2](#_Toc387670312)

[（一）系统功能需求 2](#_Toc387670313)

[（二）系统数据流图（DFD图） 5](#_Toc387670314)

[（三）业务流程图 3](#_Toc387670315)

[三、系统总体设计 9](#_Toc387670316)

[（一）系统开发环境设计 9](#_Toc387670317)

[（二）系统功能模块设计 12](#_Toc387670318)

[四、数据库设计 14](#_Toc387670319)

[（一）数据库概念模型设计 14](#_Toc387670320)

[（二）数据库物理模型设计 15](#_Toc387670321)

[五、系统实现设计 18](#_Toc387670322)

[（一）界面设计 18](#_Toc387670323)

[（二）前台界面实现 20](#_Toc387670324)

[（三）后台界面实现 24](#_Toc387670325)

[（四）实现难点 28](#_Toc387670326)

[六、总结 29](#_Toc387670327)

[资料来源和参考文献 31](#_Toc387670328)

一、序 言

（一）电子商务发展现状

随着计算机不断地深入各行业，互联网已经走进了千家万户，也渐渐进入了社会的各个领域之中，并在不同程度上改变了人们的生活习惯。越来越多的顾客选择在网站上购物，从家居家装到日用百货，从家电3C到锅碗瓢盆，从服装鞋包到图书资料，网上商城在逐渐地替代传统的商场店铺，电子商务将获得非常广阔的发展前景。所以企业必须认识到电子商务在市场竞争中对自身企业的影响以及电子商务本身的特点，并结合自身企业的实际特点，探索在电子商务背景下如何改进自身的发展模式， 从而推动企业的健康发展、持续发展。

电子商务是指在网络上以电子交易方式进行交易活动和相关服务的活动，是传统商业活动各环节的电子化、网络化。按照1997年在法国巴黎举行的世界电子商务会议对电子商务做的权威概念阐述：电子商务EC（Electronic Commerce）是指对整个贸易活动实现电子化。电子商务可以分为Business to Business（B2B），Business to Consumer（B2C），Consumer to Consumer（C2C），Business to Government（B2G），Consumer to Government（C2G）等多种模式。我国的电子商务始于1997年，从诞生之日起我国电子商务行业发展就非常迅猛，在2010年我国电子商务市场交易额已达4.5万亿元，到2011年交易总额再创新高，达到5.88万亿元，其发展势头真可谓历久弥新

此外，我国网名人数以及网购人数也在飞速增长，这预示着“互联网”已成为一种日益重要的经济载体，网购已成为现代人的一种生活方式，网购将逐渐成为主流的购物方式。

当前我国主流的电子商务网站有淘宝、京东等。淘宝网（[taobao.com](http://www.taobao.com/)）经过不断的发展壮大，目前已发展为我国最大的C2C电子商务网站，在B2C方面天猫商城（[tmall.com](http://www.tmall.com/)）的发展也成绩斐然，截止2013年3月31日的年度，淘宝网和天猫平台的交易额合计突破人民币10,000亿元。京东商城（[jd.com](http://www.jd.com/)）作为中国最大的自营式电商企业，其发展毫不逊色于淘宝，2013年营业额也已超过1000亿元。除此之外，我国还有一大批发展迅速的电子商务网站，例如当当网（[dangdang.com](http://www.dangdang.com/)）、拍拍网（[paipai.com](http://www.paipai.com/)）、聚美优品（[jumei.com](http://bj.jumei.com/)）、凡客诚品（[vancl.com](http://www.vancl.com/)）、苏宁易购（[suning.com](http://www.suning.com/)）等，其也拥有相当数量的客户群体。

（二）水果电商网站概况

我国是一个生产和消费水果的大国，传统的水果消费市场十分巨大，线上的水果市场却目前还处在发展阶段。目前已上线运营的水果电商有：天天果园、优果网（上海）、百森（广州、佛山）等，另外，一号店、京东商城等电商网站也开始兼营线上水果销售。可以想见，电子商务在水果消费市场中将会扮演越来越重要的角色。

制约电子商务在水果消费市场发展的难点主要是：水果属于生鲜农产品，保鲜难度大，保鲜时间短，对物流配送的要求也高；另外水果属于季节性产品，需要相对较高商品信息更新频率。

从整体上来看，水果的电子商务市场还远未成熟，但其具有极大的发展潜力。随着保鲜技术和物流技术的不断提高和完善，预期水果消费市场的电商化的步伐将会迅速加快。

（三）网上购物系统技术

电子商务网站一般分为两个部分：网站前台和后台，网站前、后台又各自包含多个功能模块。前台系统包含浏览商品、下订单、维护个人信息等模块；后台系统是管理员的管理功能模块，包含会员管理、商品管理、订单管理等模块。

电商网站前台系统是向用户直接展示商品的途径，也是企业展示企业形象的窗口，所以一个网站的前台的设计好坏，将在很大程度上影响用户对网站的喜爱程度。因此，在这个层面上来说，前台系统比后台系统更为重要。所以，对电子商务网站前台系统的最基本要求就是必须保证信息和服务质量的高水准，并在此基础上简化用户的购物流程，提高用户的控制程度。

本次开发将使用MVC框架，即Model-View-Controller模型-视图-控制器框架。MVC是一种软件设计模式，通过将业务逻辑和数据显式分离的方法来组织代码，它将业务逻辑聚集到一个部件里面，在界面和用户围绕数据的交互能被改进和个性化定制的同时而不需要重新编写业务逻辑。MVC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。MVC设计模式具有耦合性低、重用性高、生命周期成本低、部署快、可维护性高有利软件工程化管理等特点。

二、系统需求分析

（一）系统功能需求

根据对一些电子商务网站的考察和研究，本次开发将电商网站分为前台展示系统和后台管理系统两大部分。

1. 前台功能模块

前台功能模块包括个人中心模块、商品展示模块、购物车模块以及注册、登录模块。

用户可在个人中心模块对个人信息进行管理，也查看历史订单等信息。商品展示模块主要用于搜索、查询商品，展示商品的详细信息，包括与该商品相关的历史订单等。用户选择好商品后，可选择需要的数量加入购物车；用户收到商品后，可更新订单的状态，确认完成订单。其中未注册会员的用户（游客）只能进行搜索商品和查看商品详情信息，注册并登陆后方可使用全部功能。

表2.1 网上水果销售系统前台功能

|  |  |
| --- | --- |
| 1.1个人中心 | 注册、登录；个人信息更新；  修改密码；修改安全问题；  订单查询和状态更新； |
| 1.2商品展示 | 商品查询、搜索；商品信息信息查看； |
| 1.3物车 | 加入购物车；生成订单； |

2. 后台功能模块

后台管理模块主要提供对前台信息的添加、修改、删除等管理操作，主要包括会员管理模块，商品管理模块，订单管理模块，其中商品管理还包含商品类别管理和产地管理等子模块。管理员可在商品管理模块对商品信息进行维护，包括对商品进行分类等，也可发现存货较低的商品并进行补货操作。

表2.2 网上水果销售系统后台功能

|  |  |
| --- | --- |
| 2.1会员管理 | 查看会员信息；修改会员状态； |
| 2.2商品管理 | 添加商品；更新商品信息；删除商品； |
| 2.2.1商品类别管理 | 添加商品类别；更新商品类别信息；删除商品类别； |
| 2.2.1产地管理 | 添加商品产地；更新商品产地信息；删除商品产地 |
| 2.3订单管理 | 查看订单；处理订单； |

（二）业务流程图

1. 前台业务流程图

系统前台的目标用户为网站注册会员和未注册的潜在顾客。顾客购物流程如下：

1). 顾客登录系统，查看商品信息并选购；

2). 顾客将选中的商品加入购物车中，完成后点击“去结算”按钮进入订单生成页面，检查并输入订单详细后，如收货人、收货地址、邮政编码等信息，并再次确认订单中商品的信息；

3). 顾客确认完成订单后，点击“提交订单”按钮，提交并生成新订单，新订单状态为“已确认”；

4). 管理员审核并确认订单后，更新订单为“已发货”状态，并且管理员及时联系快递进行发货处理；

5). 管理员确认送货完成后，更新订单中状态为“已签收”；

6). 顾客确认货物送达并评价后，可完成此订单。

此外，管理员也可根据用户的请求将订单退回为购物车状态，或者关闭此订单，关闭订单时需要在备注中指明原因。前台用户业务流程图如下图：



图2.11 网上水果销售系统前台业务流程图

2. 后台业务流程图

系统前台的目标用户为网站管理员。管理员登录后台系统后可对管理员、商品、会员以及订单信息进行管理。

在管理员信息管理模块，管理员可对自身信息进行查看、更新等操作，管理员修改自己的登录密码，但需要提供正确原始密码。

在商品管理模块，管理员可对商品进行添加、查询、删除、更新等操作。此外，管理员还可以手工设置热销商品，给商品设置类别、产地等。商品的类别和产地管理模块，其功能和操作流程均类似商品管理模块，需要注意的是商品类别和产地分为大小类或父子类，父类选项置空则为大类。

在会员管理模块，管理员可查看所有用户信息，如有必要可对用户进行状态处理，屏蔽用户登录。管理员也可以查看注册会员的详细信息，以及删除会员。

在订单管理模块，管理员可以并处理订单，即更新订单状态，也可以根据用户的请求修改订单详细信息，如收货地址修改、价格更高等。

后台业务流程图如下：



图2.12 网上水果销售系统后台业务流程图

（三）系统数据流图（DFD图）

1. 顶层数据流图（DFD0）

网上水果销售系统包含2个实体：会员，管理员。等层数据流图如下图：



图2.1 网上水果销售系统顶层数据流图

2. 前台系统数据流图（DFD1）

会员用户在前台系统主要进行购物操作，也可对个人信息进行维护更新。其中购物操作包括查看商品信息，加入购物车，生成订单等步骤。所涉及的数据库表有：会员信息表、订单信息表、商品信息表、购物车信息表。数据流图如下图：



图2.2 用户前台模块 一层数据流图

2.1 个人中心数管理据流图

在会员个人信息维护模块，用户可进行注册/登录操作，登录可以维护用户的个人账户信息，包括用户名、密码、密码安全问题等信息，也可查询用户相关的历史订单、管理订单状态等。用户可以在输入当前正确密码之后可以修改密码。所涉及的数据库表有：会员信息表、订单信息表。数据流图如下图：



图2.3 个人信息维护 二层数据流图

2.2 购物管理数据流图

在会员购物管理模块中，会员用户可以根据关键字或分类搜索查看商品，浏览商品详细信息。用户选定商品加入购物车，核对商品信息无误后即可生成订单。其中未注册会员的用户（游客）只能进行搜索商品和查看商品详情信息，注册并登陆方可使用全部功能。所涉及的数据库表有：商品信息表、订单信息表、购物车信息表。数据流图如下图：



图2.4 会员购物管理 二层数据流图

3. 后台系统数据流图（DFD1）

后台系统包含五个子模块：会员管理、订单管理、商品管理，所涉及的数据库表有：会员信息表、订单信息表、商品信息表。数据流图如下图：



图2.5 后台管理员模块 一层数据流图

3.1 会员管理数据流图

在会员管理模块，管理员可以对所有会员的信息进行查改，包括查看会员信息和修改会员的注册状态。此模块相关的数据表为会员信息表。数据流图如下图：



图2.6 会员管理模块 二层数据流图

3.2 订单管理数据流图

在订单管理模块中，管理员可以查看订单信息和处理订单，包括修改订单状态等操作。管理员确认订单后，应及时联系快递公司进行发货。此模块相关的数据表为订单信息表。数据流图如下图：



图2.7 订单管理模块 二层数据流图

3.3 商品管理数据流图

在商品管理模块中，管理员可以对商品信息、商品类别信息、商品产地信息进行增删改查操作。此模块相关的数据表有商品信息表、类别信息表、产地信息表。商品类别和产地的管理数据流图与商品管理数据流图类似，商品管理数据流图如下图：



图2.8 商品管理模块 二层数据流图

三、系统总体设计

（一）系统开发环境设计

系统开发整体方案包括：

* 开发环境：windows 7，WEBrick 1.3.1，Chrome 34.0.1847.131 m
* 数据库：SQLite，Navicat for SQLite
* 开发工具：Notepad++ v6.5.5
* 开发语言：Ruby，JavaScript

本系统使用RoR框架（Ruby on Rails）作为开发平台，基于MVC框架进行开发，主要开发语言为Ruby；使用的数据库是SQLite，并将Navicat for SQLite作为数据数据库管理工具；主要开发工具为Notepad++ v6.5.5，同时使用FastStone Capture、Chrome 34.0.1847.131 m等软件作为网页设计辅助工具。

1. Ruby

Ruby，在20世纪90年代中期由日本人松本行弘设计并开发，其初衷是为了提高编程效率，这也就意味着Ruby的特点将优化软件开发的方式。Ruby是一种面向对象、命令式、函数式、动态的通用编程语言，其灵感与特性来自于 Perl、Smalltalk、Eiffel、Ada 以及 Lisp 语言，并且兼具脚本语言强大的字符串处理能力、正则表达式和解释型语言的动态性和快速性。

Ruby的特点：

* 任何东西都有值：Ruby中的所有表达式，乃至一个语句，都会有值；
* 动态载入：Ruby是可以在运行时候重定义自己，类也可以在运行时继承或取消继承，Ruby甚至支持对运算符等内置方法进行重载；
* 反射机制（内省）：Ruby程序可以在运行时自行追踪程序的运作，程序在运行过程中可以修改它自身的某些方面，或根据运行时的需求创建一个全新的对象；
* 元编程：结合动态载入和反射机制，使得Ruby程序可以根据程序员提供的信息自行生成、修改类型或对象，从而提高编程效率，使程序员可以更专心处理程序的运作，而不必累于编写重复的代码；RoR大量使用了这一特点；

2. RoR框架

RoR是Ruby on Rails的缩写，包括两部分内容：Ruby语言和Rails框架，于2004年7月由丹麦人David Heinemeier Hansson首次提出。RoR是一个用Ruby语言并严格严格按照MVC结构开发的一个开源Web应用框架，它是一个简洁且更高效的Web开发框架。RoR给程序开发人员提供了强大的框架支持，使实际的应用开发时的代码更少，使开发速度风快，使用最少的配置。RoR的数千个程序已经在线上得到应用，目前使用RoR的网站有Basecamp，Twitter，GitHub，Groupon，Shopify，Yellow Pages等。



图3.1 用户、浏览器和MVC结构图示

1). Rails框架的设计原则[5]

a). DRY（Don't Repeat Yourself）：“不做重复的事”

这一原则如果正确实施的话，将会减少项目中大量的重复性工。项目中的任何一种重复性工作都会导致更新、维护的困难，也会使得代码前后不一致。在RoR,你可以看到这一原则在几乎所有的组件中应用，从插件形式的可重用组件，到数据库表的映射方式

b). CoC（Convention Over Configuration）：“惯例优于设置”

RoR的原则是只提供与通常的应用程序设置不同的方面的信息。例如由RoR提供的ORM（对象关系映射）框架：除非你为一个ORM对象指定一个不同的名称，否则该对象使用的表名就是它所映射的表名称。一般情况下，架构的改变意味着在配置文件要有相应的变化，但在RoR中，模式的改变并不意味着对象也需要改变，除非表本身的名称更改。

3. WEBrick

WEBrick是一个提供简单HTTP网络服务器服务的Ruby库，其作为RoR中的web服务器，可用在开发环境中调试程序，也可用在生产环境中运行程序，现在是Ruby标准库的一部分。在web应用系统的根目录下执行rails server命令可启动WEBrick。WEBric截图如下：

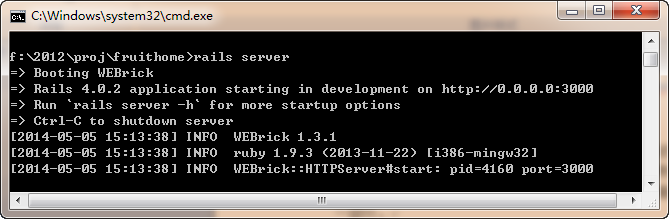


图3.2 运行WEBrick

4. SQLite

SQLite是一款轻型的遵守ACID的关系型数据库管理系统，诞生于2000年。它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，它占用资源非常的低，在嵌入式设备中，可能只需要几百K的内存就够了。它支持Windows/Linux/Unix等主流的操作系统，比起Mysql、PostgreSQL这两款开源世界著名的数据库管理系统来讲，它的处理速度比他们都快。

SQLite的整个数据库，包括定义、表、索引和数据本身，都存储在一个单一的文件中，这一简洁的设计通过在开始一个事务时锁定整个数据文件来完成，并且使用SQLite是不需要安装和配置的，可使用Navicat for SQLite作为数据库管理工具。另外它是完全开源的，可以将其用于任何用途，包括出售。

5. Notepad++

Notepad++是一款Windows环境下免费开源的纯文本编辑器，由台湾人侯今吾基于同是开放源代码的Scintilla文本编辑组件独力研发，整个项目托管于SourceForge.net之上，此软件曾两度获得SourceForge社区选择奖——最佳开发工具。

它堪比Windows自带Notepad（记事本）加强版，除了基本的纯文本编辑之外，它十分适合当作大部分主流语言的代码编辑器。Notepad++支持有语法高亮度显示，语法折叠等功能，并且支持宏以及扩充基本功能的各类插件，通过插件的辅助，Notepad++可以成为一个堪比IDE的开发工具。Notepad++截图如下：

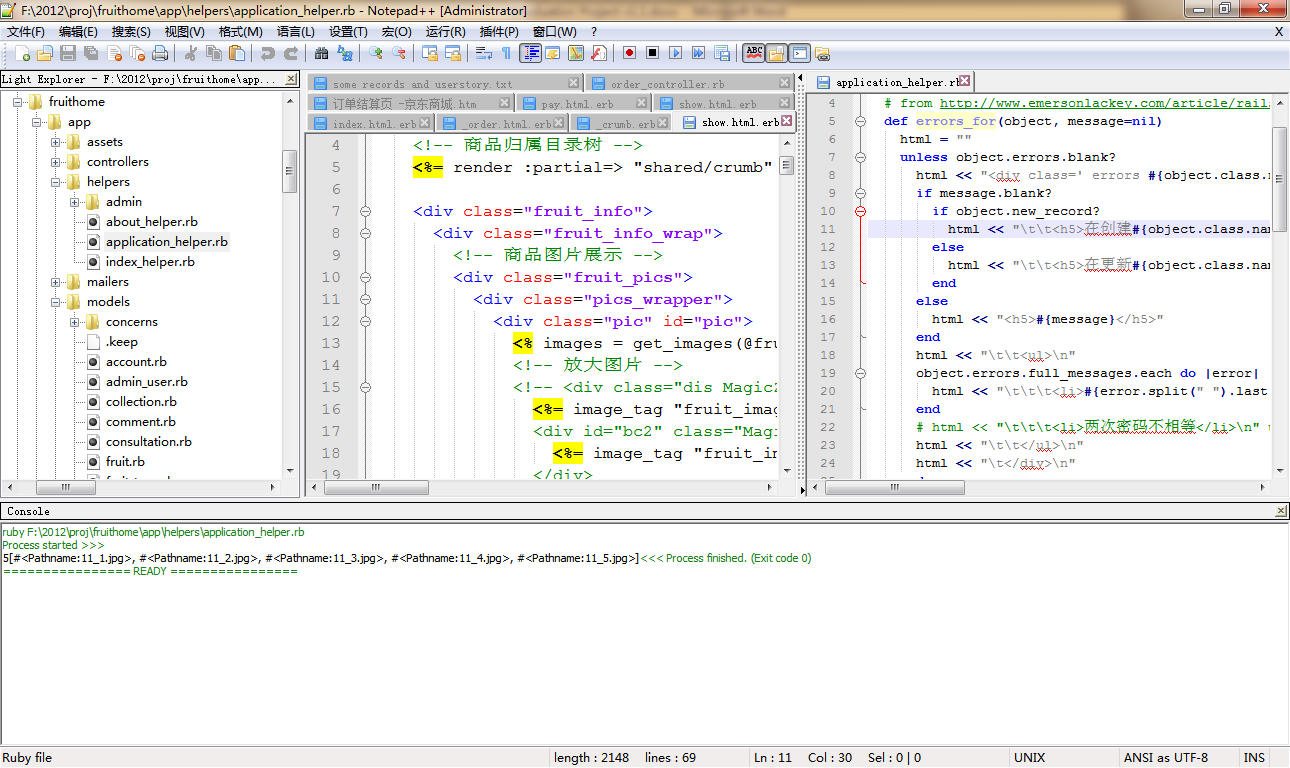


图3.3 Notepad++截面图

（二）系统功能模块设计

网上水果销售系统的功能模块图如下：



图3.4 功能模块图

1. 前台功能

系统前台的目标用户是网站注册会员和未注册的潜在顾客，系统的前台功能模块如下图：

表3.1 系统前台功能明细表

|  |  |
| --- | --- |
| 1.1个人信息管理 | 用户注册、登录 |
| 个人信息更新 |
| 修改密码 |
| 修改安全问题 |
| 历史订单查询 |
| 1.2购物管理 | 商品查询（通过商品名称、分类、等相关信息进行模糊查询） |
| 商品详细信息查看 |
| 生成订单 |
| 1.3订单管理 | 查看订单 |
| 修改订单状态 |

2. 后台功能

系统前台的目标用户是网站管理员，系统的后台功能模块如下图：

表3.2 系统后台功能明细表

|  |  |
| --- | --- |
| 2.1会员管理 | 查看会员信息 |
| 修改会员状态 |
| 删除会员 |
| 2.2订单管理 | 查看订单 |
| 处理订单 |
| 2.3商品管理 | 添加商品 |
| 查看商品信息 |
| 更新商品信息 |
| 删除商品 |
| 2.3.1商品类别管理 | 添加商品类别 |
| 查看商品类别信息 |
| 更新商品类别信息； |
| 删除商品类别 |
| 2.3.1商品产地管理 | 添加商品产地 |
| 查看商品产地信息 |
| 更新商品产地信息； |
| 删除商品产地 |

四、数据库设计

（一）数据库概念模型设计

1. 实体

根据前述需求分析和总体设计，网上水果销售系统的数据库实体应包括会员、商品、商品类别、产地、订单、订单明细、收藏、管理员、留言、评论、购物车、供应商、供应明细信息等。

表4.1 实体表

|  |  |
| --- | --- |
| **实体名** | **说明** |
| user | 用户，注册会员 |
| fruit | 水果（商品均为水果，故而使用水果作为实体名） |
| fruit\_type\_big | 水果大类 |
| fruit\_type\_small | 水果小类 |
| origin\_place | 产地 |
| order | 订单 |
| order\_item | 订单详情 |
| order\_status | 订单状态 |
| collection | 收藏 |
| account | 账户 |
| mail\_address | 邮递地址 |
| admin\_user | 管理员用户 |

2. 实体关系图

根据实体及其属性和数据结构，并根据其关系得出数据库概念模型实体关系图（E-R图）；



图 4.1 数据库概念模型ER图

（二）数据库物理模型设计

1. 数据库表（实体）

在数据库概念模型中，可以发现实体cart和order、cart\_item和order\_item内容相似，所以在实体order的status属性增加0值，以表示cart状态；fruit\_type\_big和fruit\_type\_small表的内容页相似，并且大类条目并不多，所以将其融合为fruit\_types，增加super\_id字段表示外键大类。

1.用户表（会员）

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 约束 | 字段含义 |
| id | integer | PK | 自动递增编号 |
| username | varchar |  | 用户名（大小写英文字母） |
| real\_name | varchar |  | 真实姓名 |
| sex | varchar |  | 性别 |
| birthdate | varchar |  | 出生日期 |
| e\_mail | varchar |  | 邮箱 |
| password\_text | varchar |  | 明文密码 |
| password\_digest | varchar |  | 加密密码 |
| password\_comfirmation | varchar |  | 密码确认（大小写英文和数字） |
| status | integer |  | 用户状态（为0则用户被屏蔽） |
| question | varchar |  | 找回密码问题 |
| answer | varchar |  | 找回密码答案 |
| phone\_number | varchar |  | 手机号码 |
| shopping\_exp | varchar |  | 经验值 |
| member\_type | varchar |  | 会员类别 |
| city | varchar |  | 城市 |
| post\_code | varchar |  | 邮政编码 |
| advance\_payment | varchar |  | 支付方式 |
| latest\_log\_date | varchar |  | 最后登录日期 |
| created\_at | datetime |  | 创建日期 |
| updated\_at | datetime |  | 更新日期 |

2.水果类别

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 约束 | 字段含义 |
| id | integer | PK | 自动递增编号 |
| name | varchar |  | 类别名称 |
| description | varchar |  | 类别描述 |
| super\_id | integer | FK | 父类别 |
| created\_at | datetime |  | 创建日期 |
| updated\_at | datetime |  | 更新日期 |

3.水果

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 约束 | 字段含义 |
| id | integer | PK | 自动递增编号 |
| name | varchar |  | 水果名称 |
| introduction | varchar |  | 水果描述 |
| brief\_introduction | varchar |  | 水果简介 |
| fruit\_type\_id | integer | FK | 水果类别 |
| origin\_place\_id | integer | FK | 产地 |
| inventory | integer |  | 库存 |
| unit | varchar |  | 计价单位 |
| price\_present | float |  | 当前价格 |
| price\_history | float |  | 历史价格 |
| price\_other | float |  | 其他价格 |
| discount | float |  | 优惠 |
| is\_hot | boolean |  | 是否热销 |
| created\_at | datetime |  | 创建日期 |
| updated\_at | datetime |  | 更新日期 |

4.订单

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 约束 | 字段含义 |
| id | integer | PK | 自动递增编号 |
| user\_id | integer | FK | 所属用户 |
| consignee\_name | varchar |  | 收货人姓名 |
| consignee\_phone | varchar |  | 收货人电话 |
| consignee\_address | varchar |  | 收货人地址 |
| remarks | varchar |  | 备注 |
| status | integer |  | 订单状态 |
| created\_at | datetime |  | 创建日期 |
| updated\_at | datetime |  | 更新日期 |

5.订单详情

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 约束 | 字段含义 |
| id | integer | PK | 自动递增编号 |
| order\_id | integer | FK | 所属订单 |
| fruit\_id | integer | FK | 包含水果 |
| quantity | integer |  | 水果数量 |
| price | floar |  | 水果价格 |
| unit | varchar |  | 计量单位 |
| created\_at | datetime |  | 创建日期 |
| updated\_at | datetime |  | 更新日期 |

6.邮递地址

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 约束 | 字段含义 |
| id | integer | PK | 自动递增编号 |
| user\_id | integer | FK | 所属用户 |
| address | varchar |  | 具体地址 |
| created\_at | datetime |  | 创建日期 |
| updated\_at | datetime |  | 更新日期 |

7.管理员用户

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 约束 | 字段含义 |
| id | integer | PK | 自动递增编号 |
| username | varchar |  | 用户名 |
| password | varchar |  | 密码 |
| admin\_type | varchar |  | 管理员类型 |
| e\_mail | varchar |  | 邮箱 |
| created\_at | datetime |  | 创建日期 |
| updated\_at | datetime |  | 更新日期 |

8.原产地

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 约束 | 字段含义 |
| id | integer | PK | 自动递增编号 |
| name | varchar |  | 产地名 |
| description | varchar |  | 产地描述 |
| super\_id | integer | FK | 产地归属 |
| created\_at | datetime |  | 创建日期 |
| updated\_at | datetime |  | 更新日期 |

9.收藏

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 约束 | 字段含义 |
| id | integer | PK | 自动递增编号 |
| user\_id | integer | FK | 所属用户 |
| fruit\_id | integer | FK | 收藏水果 |
| created\_at | datetime |  | 创建日期 |
| updated\_at | datetime |  | 更新日期 |

2. 数据库结构（关系）

在Rails框架中，一个实体对应一个相应的Model类，实体间的关系在相关的Model类中实现，通过Model的父类Active Record来实现与数据库之间的数据传输。



图4.2.1 数据库物理模型ER图

五、系统设计与实现

（一）界面设计

通过对一些电商网站的研究和参考，本次独立自主设计开发一个网上水果销售系统，锻炼在电子商务网站设计与制作方面的能力，同时探讨电子商务模式为果蔬行业带来的机遇和挑战。

电商网站应具有合理的版面设计、清晰的导航和有层次感的页面，同时简洁明快，力求呈现出商品本身的特性。本网站的设计参考了天天果园等网站，对水果之家系统进行用户和管理员的界面设计。天天果园的网站截图如下：



图5.1 天天果园首页

1.前台系统界面

网上水果销售网站前台系统的界面背景使用纯白色，网站颜色以淡绿色为主色调，辅以灰白色表格用以显示订单列表等信息。

前台系统分为注册/登录、商品浏览、个人信息浏览和网站其他信息浏览。

1.1注册/登录界面，用户通过界面顶部的导航条进行注册和登录连接可进入相应的操作界面。

1.2商品浏览界面，网站顶部是一些常用的导航功能条，这些导航条会一直存在，以方便用户使用。网站的用户导航条位于网站上方，包含登录、注册、个人中心、我的收藏夹、所有商品、客户中心等导航信息。用户导航条下方是网站LOGO、商品搜索栏和购物车按钮。再下方就是商品导航条，根据商品的分类、产地、事宜人群等条目进行导航，在大类条目上点击可显示小类的条目信息。

首页主要显示一个商品轮换展示的大图，和根据热门类别显示的一个商品。商品的具体信息查看页面除了展示商品的详细信息和展示图片之外，还将显示该商品相关的评价、留言和购买记录。

1.3个人账户信息浏览页面，上述的导航功能条仍位于页面上方，对个人账户信息查看、维护的链接网页左侧的菜单栏里。

2.后台系统界面

后台系统界面的主色调使用黑灰色，背景使用白色，以求简洁大方。

后台系统界面分为登录界面和管理模块界面。

2.1管理员登录界面，用户通过界面顶部的导航条“网站导航”下的“我是管理员”按钮进行登陆。

2.2在管理员管理界面的左侧显示管理员的操作菜单，管理员可以点击相应的菜单选项进行管理操作。

（二）前台界面实现

1. 首页

网站首页包含各种导航条、搜索框、购物车链接，以及一些热门的商品。用户可通过导航条进入相关的页面、进行商品查询，购物车功能模块需要用户登录后方可进入。首页如下图：

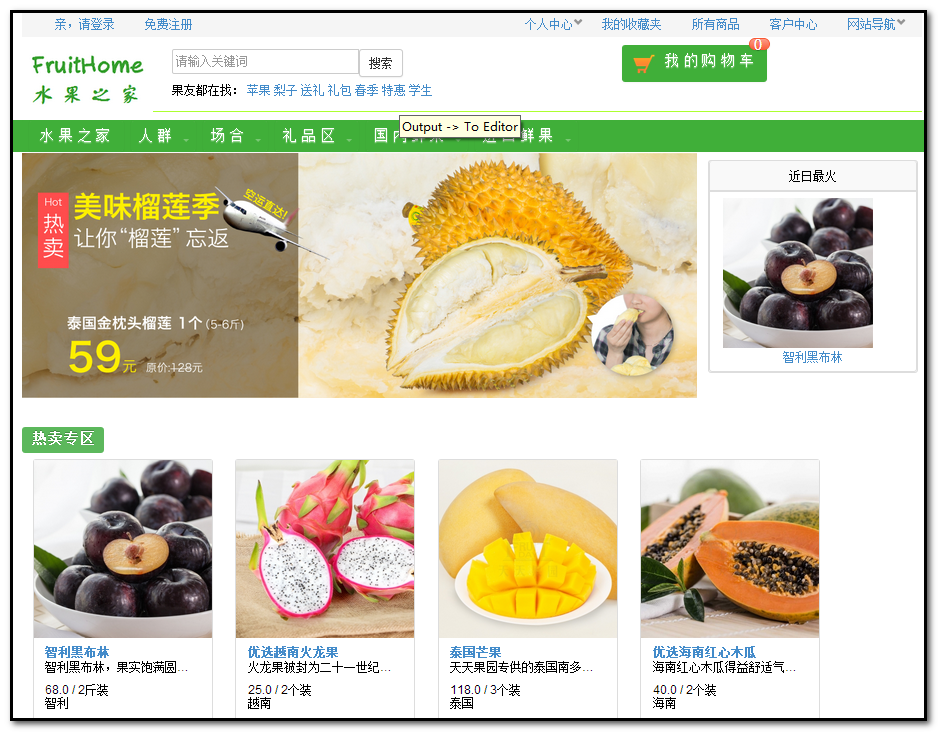


图5.2 网站首页

2. 注册/登录

用户在购买商品之前需要注册成为会用用户，并登录系统之后方可进行购物等，注册和登录界面如下图：



图5.3注册页面

图5.4 登录页面

3. 修改会员个人资料

用户登录系统之后，可对个人账户信息进行更新维护，个人中心如下图：



图5.5 个人信息管理页面

用户登录系统之后，可以更新自己的个人信息，如账户信息、收货地址信息、修改密码等，界面如下图：



图5.6 编辑个人信息



图5.7 收货地址管理

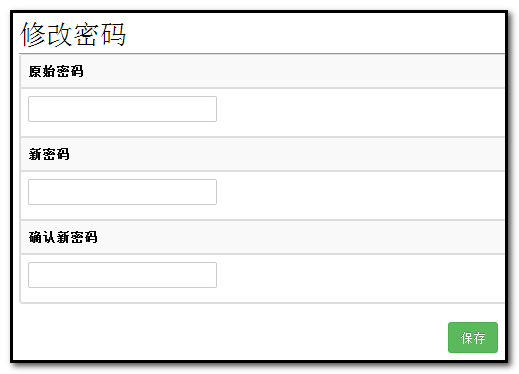


图5.8 修改密码

4. 商品查询

用户可以在网站上查看所有商品的信息，可以根据商品的分类、产地、适宜人群场合等信息进行分类查询，如图5.9；也可根据关键字进行查询，如图5.10。在查询结果页面点击相应的商品链接，可进入查看商品详细信息，如图5.11。在商品详情页面，用户可选择需要购买的数量，点击“加入购物车”按钮，将该商品加入购物车。如网站数据库中没有响应的商品信息，则会显示没有找到与此相关的商品，如图5.12 。



图5.9 分类查询

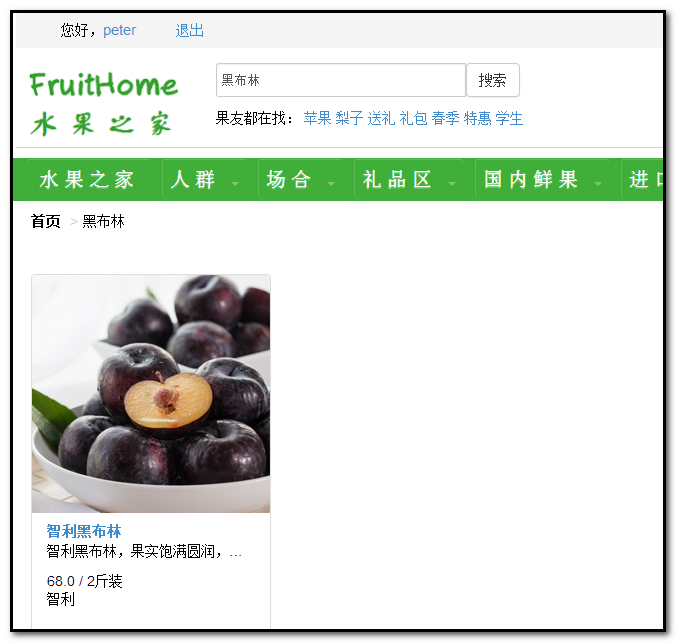


图5.10 关键字查询



图5.11 商品详情页

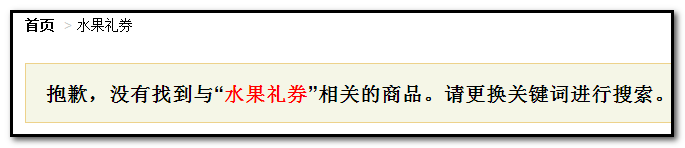


图5.12 查询失败页面

5. 购物车实现

注册用户在将商品加入购物车之后，可点击页面右上方的“我的购物车”按钮进入购物车页面，未登录的用户需要登录方可进入，购物车界面如图5.14。用户在此界面可选择收藏或者删除购物车中商品，也可点击“继续购物”按钮回到网站首页，或者点击“去结算”按钮进行订单生产操作。



图5.13 购物车按钮



图5.14 购物车页面

6. 订单管理

用户在购物车界面选择“去结算”后可显示订单的详细信息，用户确认后可点击“提交订单”按钮生成此订单，如图5.15和图5.16。订单提交成功后， 用户可选择继续购物或者进入管理界面，进行订单管理。在订单管理界面，用户可查看所有订单的详细信息，对订单进行状态确认和取消操作，订单管理界面如图5.18，订单详细信息如图5.19。



图5.15 订单确认页面

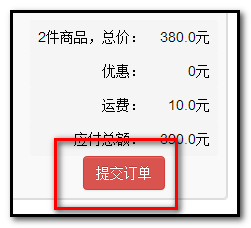


图5.16 “提交订单”按钮

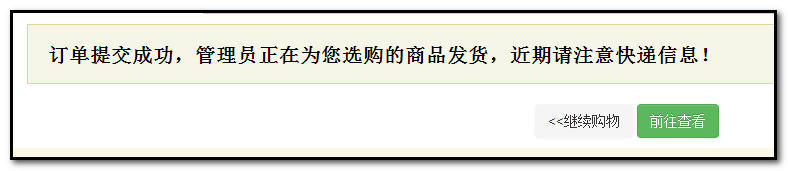


图5.17 订单提交成功页面

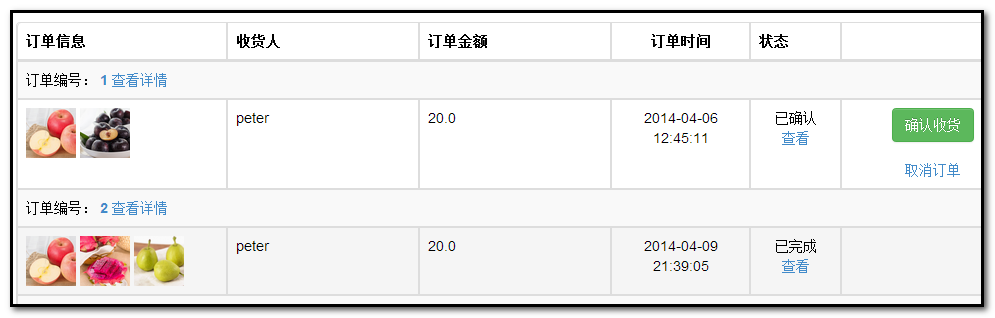


图5.18 订单管理界面



图5.19 订单详细信息

（三）后台界面实现

1.管理员登录

管理员用户可通过网站首页的“我是管理员”按钮或者直接输入网址http://localhost:3000/admin/login进行登陆操作。



图5.20 管理员登录链接



图5.21 管理员登录界面



图5.22 管理员首页

2.管理员维护个人账户

管理员登录后可对个人的信息进行编辑更新，如图5.23，也可对自己的密码进行更改，如图5.24。



图5.23 管理员个人信息管理



图5.24 修改密码

3.商品管理

管理员在商品管理模块，可对商品进行查询、更新、删除、增加等操作，如图5.25。也可对商品类别、产地进行相应的查询、更新、删除、增加等操作，如图5.26和图5.27。



5.25 商品管理模块



图5.25-1 新增商品

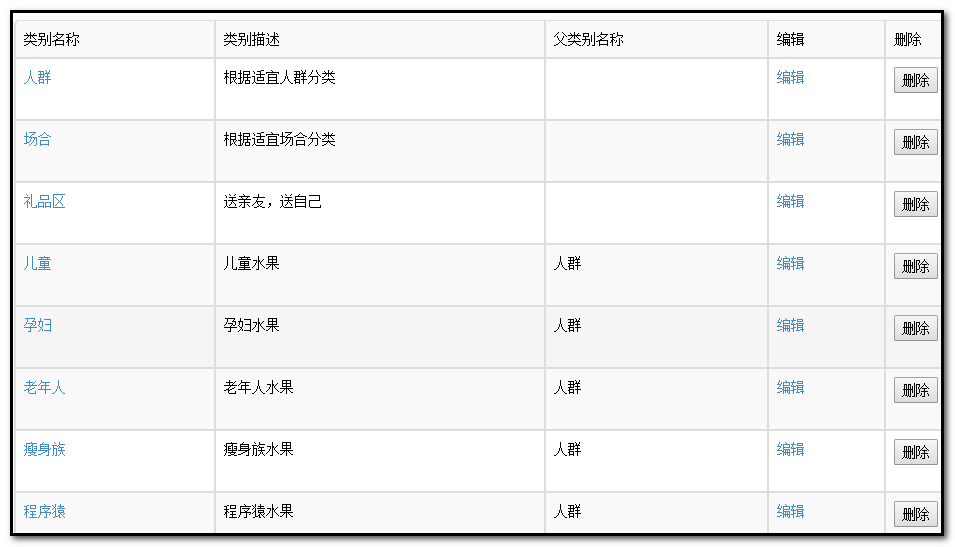


图5.26 商品类别管理

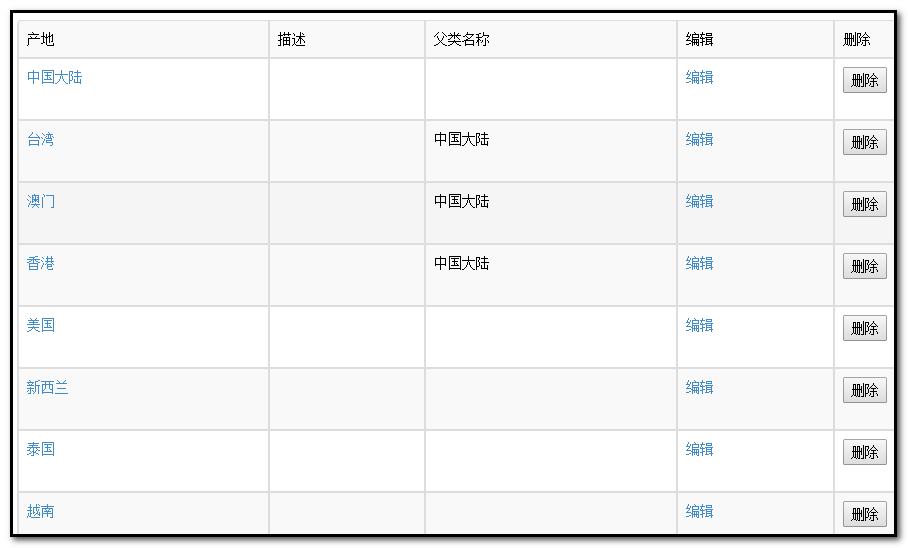


图5.27 商品产地管理

4.订单管理

管理员在订单管理模块可查看所有订单，并可根据实际情况对订单进行状态修改操作，如图5.26和图5.27。已完成的订单会全部显示在“历史订单页面中”，如图



图5.26 待处理订单



图5.27 订单操作界面

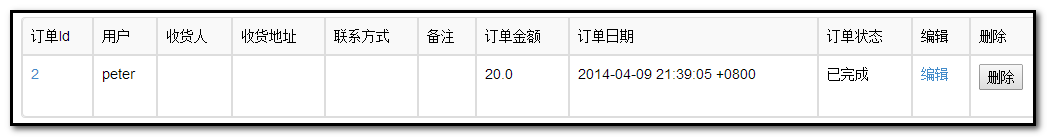


图5.28 历史订单界面

（四）实现难点

1. 收集验证错误信息

在Rails早期版本中使用ActionRecordHelper类error\_messages\_for方法来收集数据库验证产生的错误信息，以及其他的一些错误信息。但在Rails3及其之后的版本中，由于Rails的“nothing in Rails Core should dictate the look and feel of an app”的指导方针而从Rails核心中移除了这个方法。所以在参考相关资料后，本人编写了一个模仿error\_messages\_for的方法，并通过制定其显示的css加强了它的显示效果，如图5.29

代码如下：

def errors\_for(object, message=nil)

html = ""

unless object.errors.blank?

html << "<div class=' errors #{object.class.name.humanize.downcase}\_errors'>\n"

if message.blank?

if object.new\_record?

html << "\t\t<h5>在创建#{object.class.name.humanize.downcase}时出错</h5>\n"

else

html << "\t\t<h5>在更新#{object.class.name.humanize.downcase}时出错</h5>\n"

end

else

html << "<h5>#{message}</h5>"

end

html << "\t\t<ul>\n"

object.errors.full\_messages.each do |error|

html << "\t\t\t<li>#{error.split(" ").last}</li>\n"

end

html << "\t\t</ul>\n"

html << "\t</div>\n"

end

html

end



图5.29 校验显示效果

2. 搜索结果的分页功能

系统通过调用will\_paginate插件实现了搜索结果的上下页翻页功能，控制器代码如下：

def search

if params[:col] and params[:val]

return @fruits\_search = search\_by\_col(params[:col],params[:val])

end

@key\_word, type\_names = params[:key\_word], []

FruitType.all.each {|ft| type\_names << ft.name}

if type\_names.index @key\_word

@fruits\_search = FruitType.find\_by\_name(@key\_word).fruits.paginate(page: params[:page], per\_page: 16)

else

@fruits\_search = Fruit.where("name like ? or brief\_introduction like ?", "%#{@key\_word}%", "%#{@key\_word}%").paginate(page: params[:page], per\_page: 16)

end

end

def search\_by\_col(col,val)

search\_hash = {col => val}

Fruit.where(search\_hash).paginate(page: params[:page], per\_page: 16)

end

页面显示代码如下：

<ul class="pagination" style="clear:both;float:right;">

<li class=<%= "disabled" unless @fruits\_search.previous\_page %>>

<%= link\_to "上页", :page => @fruits\_search.previous\_page, key\_word: @key\_word %>

</li>

<% (1..@fruits\_search.total\_pages.to\_i).each do |i| %>

<li class="<%= "active" if i==@fruits\_search.current\_page %>">

<%= link\_to i, :page => i, key\_word: @key\_word %>

</li>

<% end %>

<li class=<%= "disabled" unless @fruits\_search.next\_page %>>

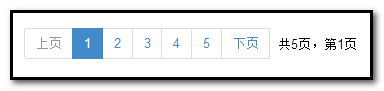
<%= link\_to "下页", :page => @fruits\_search.next\_page, key\_word: @key\_word %>

</li>

<li style="padding:9px;">共<%= @fruits\_search.total\_pages %>页，第<%= @fruits\_search.current\_page %>页</li>

</ul>

具体实现效果如下图：



5.30 分页效果

六、总结

通过对水果电商网站系统的设计，对系统的设计有了更深入的了解。本设计的系统开发任务包括电商网站系统的前台显示和后台管理等功能模块，在系统开发环境设计、系统模块设计、数据库设计、界面设计等方面有了缠足的进步。其次，通过此次系统的设计和开发，我也体会到了团队的重要性。毕业设计毕竟只是一个人在独立进行设计和开发，在很多方面都会有考虑不周到、不完善的地方，只有团队上的互补互助才能弥补。最后，我也认识到了个人知识储备的不足，毕竟很多知识都是两三年前学习的，并且此次开发使用的又是新学习RoR框架，短暂的接触和学习并不能够完全满足实际开发的需要。所以，我们不但要时刻回顾已经掌握的知识，温故而知新，在新知识面前也要迎难而上，不断的用知识来充实自己，增强自己。

以下是对于此次大学生职业规划系统的优点和局限性的总结：

优点：

1.系统数据库驱动的，会员用户和管理员可对数据库信息进行实时操作，使系统更加简洁明了，操作更加简单；

2.系统前台使用了Bootstrap的web前端开发工具包，系统界面效果比较清爽；

3.系统导航比较明确和清晰，在系统界面的顶部显示了所有的导航，优化了用户的体验

4.系统实现了密码加密和强制登陆功能，例如，必须登陆管理员账户才能进入管理员模块界面，前台用户也必须登录后才能对用户信息进行修改更新，保障了系统的安全性，同时也允许了未注册的用户对系统的部分功能进行访问。

局限性：

1.由于系统开发经验的不足，导致系统的设计与开发有些脱节，市场开发在前、设计在后，实际上做了很多的无用功和重复工作；

2.界面设计还有提升空间。在字体、图片的应用，显示内容的排版上还需要深入的考察和研究；

3.对编程语言还不够熟练，时不时需要查询文档，降低了开发效率。

虽然系统还存在很多不足，但是本人在系统的设计和开发过程中，学习并掌握了很多的知识和机能，积累了不少系统设计和开发的经验。同时我要感谢我的导师张雪凤老师，张老师在系统设计之初给我提了非常多宝贵的意见，特别是在数据库的设计上面给了我极大的帮助；在开发阶段，张老师也一直在为我提出不合适的地方和改进意见。真是张老师的耐心细致教导，才使得本次开发如此顺利。

资料来源和参考文献

1. 郝晓玲，韩冬梅，韩松乔编著，信息系统开发，清华大学出版社，2012
2. C.J.Data著，An Introduction to Database Systems，Addison Wesley/Pearson，第八版，2006
3. Paolo Perrotta著，Metaprogramming Ruby，Pragmatic Bookshelf，第四版，2012
4. Sam Ruby，Dave Thomas，David Heinemeier Hansson著，Web开发敏捷之道， 机械工业出版社，第四版，2012
5. A P Rajshekhar著，Building Dynamic Web 2.0 Websites with Ruby on Rails，Packt Publishing，2008
6. Christian Hellsten，Jarkko Laine著，《Ruby on Rails电子商务实战》，人民邮电出版，2008