

**本科毕业论文**

题 目： 基于JavaEE的美食分享与推荐系统

专业班级： 计科1302

学 号： 2013001977

学生姓名： 林廷勇

指导教师： 马垚

计算机科学与技术学院

2017年05月28日

基于JavaEE实现的美食分享与推荐系统

摘 要

伴随互联网时代的飞速发展，人们的生活变得越来越方便，互联网时代，在家足不出户即可购物，伴随着手机时代普及，手机成了人们生活必备的一个产品，人们的生活处处离不开手机，手机也逐渐成为人们生活和娱乐的工具，网上订餐也逐渐成为部分上班族、学生党，甚至喜欢宅的懒人们的吃饭方式，吃成了每天生活不可缺少的部分，发现和分享身边所不知到的美食，成为本次设计的一个着重点，本着发现世界各地的美食，设计了一款基于JavaEE实现的美食与分享系统，主要技术是H5，采用前、后端分开的开发模式，前端做视图的渲染，后端做业务处理，前端和后端通过ajax进行通信和数据交换，后台使用springMVC+Hibernate作为开发框架，前端使用mui作为基础架构，采用模块化开发，迅速搭建整个应用。

关键词：JAVA；H5；WebAPP；SpringMvc；Hibernate

Food sharing and recommendation system

based on JavaEE

Abstract

With the rapid development of the Internet era, people's lives become more and more convenient, the Internet age, at home can not go home to shop, along with the popularity of the mobile phone era, the phone has become a must-life products, people's lives everywhere Can not be separated from the phone, the phone has gradually become a tool for people's lives and entertainment, online ordering has gradually become part of the office workers, student party, and even like the lazy people to eat the way, eat an indispensable part of daily life, Share the side of the food that we do not know, as a focus of this design, in the discovery of food around the world, designed a JavaEE based on the realization of food and sharing system, the main technology is H5, before and after the back Development mode, front-end view rendering, back-end business processing, front-end and back-end through ajax communication and data exchange, background use springMVC + Hibernate as a development framework, front use mui as the infrastructure, modular development, The whole application.

**Key words:** JAVA，H5，WebAPP，SpringMvc，Hibernate

目 录

[1 绪论 1](#_Toc6283)

[1.1课题来源 1](#_Toc14293)

[1.2国内现状 1](#_Toc10427)

[1.3研究意义 1](#_Toc15788)

[2.系统开发平台介绍 3](#_Toc3103)

[2.1 SpringMVC框架介绍 3](#_Toc27747)

[2.2 Hibernate框架介绍 5](#_Toc20077)

[2.3 Ajax技术介绍 6](#_Toc11963)

[2.4 MUI前端框架 8](#_Toc3520)

[2.5 HTML5介绍 9](#_Toc30549)

[2.6 jQuery介绍 10](#_Toc21338)

[2.7开发环境介绍 10](#_Toc12010)

[3.需求分析 11](#_Toc24461)

[3.1 系统定义 11](#_Toc3353)

[3.2系统用例分析 11](#_Toc27078)

[3.3系统功能模块分析 14](#_Toc30909)

[4 系统数据库设计 18](#_Toc7693)

[4.1数据分析 18](#_Toc13297)

[4.2数据字典 18](#_Toc11132)

[4.3数据访问对象及实体类设计 21](#_Toc31013)

[4.4实体-关系设计 24](#_Toc4987)

[4.5数据表结构 24](#_Toc6646)

[5.系统功能设计 30](#_Toc10176)

[5.1系统功能模块设计 30](#_Toc30116)

[5.2系统关键用例设计 35](#_Toc21857)

[6.系统编码实现 39](#_Toc3164)

[6.1系统整体框架搭建 39](#_Toc3460)

[6.2登录注册模块设计 42](#_Toc23985)

[6.3其他模块设计 46](#_Toc24681)

[7.系统测试 47](#_Toc29687)

[7.1 系统功能性及性能测试 47](#_Toc17887)

[7.2 系统评价 49](#_Toc19294)

[7.3 系统存在的问题 49](#_Toc9839)

[总结 51](#_Toc31486)

[致 谢 52](#_Toc4794)

[参考文献 53](#_Toc30520)

1 绪论

1.1课题来源

二十一世纪，随着互联网的逐渐发展，互联网产品日渐丰富，传统方式和产品慢慢向互联网方向演变和发展，随着社会的的进步与发展，人的生活水平在不断提高，消费升级也随之而来，这时，人们就更愿意花钱去体验一些更好的生活方式。

基于这样的互联网时代，我的毕设选择了美食分享和推荐系统这个选题，设计一款更接近人们生活的产品，作为一个中国人，由于消费理念的改变，随之而来的消费结构选择的升级，简单的说，就是在消费力不改变的前提下， 消费者会把钱花去干做什么，由于消费环境的不断变化，即使在同一个场景下，我们会做出的选择也会有不同，因为消费环境和社会流行文化，对消费带来的变化，在同一品类下，由于对品牌、品质的要求的变化造成的内容选择升级；

1.2国内现状

为了更好地收集美食相关的产品应用，在各大应用商店里的搜索框中输入了“美食”两字，一共出来了超过500款相关App应用。经过分析发现，现在的美食类应用大致分制作美食类和寻找美食类两大类。前者主要以美食天下、美食杰、网上厨房为代表；后者主要以美团大众点评、饿了么为代表。

目前生活美食服务类的行业巨头当属美团App与饿了么App，这两个平台不仅仅改变了现代人的生活习惯，也改变了大家的饮食习惯，在一定程度上，推动了中国大部分地区人们生活方式的演变。当然它们也是当年的外卖平台垄断大战中胜出的赢家，最后也成为了行业里面的美食外卖巨头。现在也有其美食相关的app出现，如百度的糯米、阿里的口碑；其他的类似的如零号线，外卖超人，订餐小秘书App和一些当地的外卖平台也会有一点份额，但是大局已定，这些平台也只能跟在美团和饿了么后面喝点汤。

1.3研究意义

本次设计本质在于满足美食爱好者寻找美食、发现没事、分享美食、制作美食的需求，最终形成一个以美食推荐与分享为纽带的美食爱好者的分享交流兴趣圈。

在这个互联网信息高速发展时代，连我们的衣食住行都很难摆脱互联网的关注。网上预订客房、网上购片、网上预约旅游、网上订餐的出现便是有效的明证。

通过本次研究，了解美食行业在当前时代的发展趋势，了解用户的喜好，同时，当下美食应用存在的痛点与不足，为自己的本次设计提供方向，既要吸取当下美食应用的优点，也要补充它们存在的不足。

在本次研究过程中，通过调查，发现用户期待的美食应用应该是具备推荐美食和美食分享的，并且能从中学习一些美食的制作方法，提供相互交流的渠道。

2.系统开发平台介绍

2.1 SpringMVC框架介绍

SSpringMVC 是Spring官方提供的一个方便且易用的后端Java开发框架，它围绕DispatcherServlet设计，将所有的请求分派给指定的处理程序，它的处理程序和视图解析器是可以自定义配置的，并且具有区域设置、时区和主题解析等功能，默认的处理程序是基于@Controller注解配置的类作为控制类，相当于传统开发模式的servlet,处理器所对应的处理程序有@RequestMapping注解表明提供，该注解的value值指明处理程序接受的请求路径，method指明请求的方式，当然还有其他注解参数，如@PathVariable、@Autowired、@Service、@Repository、@Transactional等注解，这为SpringMVC提供了广泛而灵活的处理方法。

Spring的视图解析器的配置是非常灵活的，在同一个Srping应用中可以配置多个视图解析器，解决在在同一个应用程序中使用不同的视图，如常见的jsp和html，Controller通常负责指定请求的调用程序，将拥有所有的请求与相应放入一个Map映射中，具有数据和视图选择模型，当然也可以写入相应流中完成请求，视图名称的解析可以通过在spring的dispatcher配置文件xml中配置接收的文件扩展名或者在Accept标头内容类型协商，甚至可以自定义视图解析器ViewResolver，SpringMVC中的MVC模型中的M是一个Map接口，是一个可以基于模板渲染的视图提取技术，如JSP、Velocity和Freemarker,该模型Map可以简单的转换成适当的格式，如直接生成XML、JSON、Atom和许多其他类型的内容。

Spring的Web模块中包含了许多Web独特的支持，如角色分离，M-模型层，V-视图层，C-控制层，其中的角色都可以由专门的对象来实现，包括控制器，验证器，表单对象，DispatcherServlet模型对象，处理程序映射，视图解析器等。

SpringMVC具有很高的灵活性、适应性和非入侵性。常用@RequestHeader、@RequestParam、@PathVariable等注释定义需要的控制器方法，将处理程序接受的请求对象更明确，范围更小。

当用重用的业务代码时，不需要重复写相同的代码，只需使用现有的业务对象作为命令重新组织处理逻辑，即可实现新的业务处理，并不需要扩展框架的基类来达到业务处理目的。

SpringMVC具有可定制的验证和绑定方法，可制定不匹配作为应用程序级别的验证错误，本地化日期和数字绑定等。而不是只是用仅包含字符串的表单对象进行业务转换和手动解析参数。

SpringMVC还具有可定制的处理程序和视图解析，处理程序和视图解析的方式从基于简单的URL映射到复杂的专用视图解析策略，因此SpringMVC比其他WebMVC框架更具有灵活性。灵活的模型，让具有名称-值的模型传输Map支持与任何视图技术轻松的集成在一起。可定制的区域设置，时区和主题解析，无需额外的桥接技术，就可以支持带有或不带有Spring标签库的JSP，同时支持JSTL和Velocity。

Spring与其他Web MVC框架一样，以请求作为驱动，围绕中央的DispatcherServlet设计，将请求交给控制器，由控制器调用其他处理程序，并且Spring的IoC容器完全集成，实现bean的管理交给Spring的IoC容器进行管理，程序会在使用到该bean的时候主动注入，不需要开发者手动管理bean的整个生命明周期。

Spring Web MVC的请求处理工作流程DispatcherServlet如下图2.1\_1所示。DispatcherServlet作为“前端控制器”的设计模式。

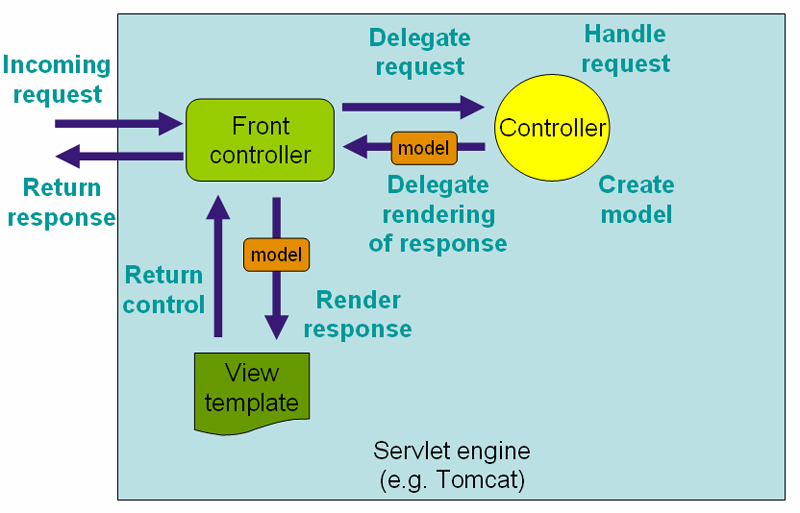


图2.1\_1Spring Web MVC 中的请求处理工作流程

2.1.1 实现控制器

控制器提供服务接口定义的应用程序行为进行访问。控制器解释用户输入并将其转换为由视图模型返回给用户。Spring以抽象的方式实现控制器，使您很容易的创建各种各样的控制器，达到自定义接口服务的目的。

Spring 2.5中引入了基于注解的编程模型MVC控制器的，使用注解如@RequestMapping，@RequestParam，@ModelAttribute，等。此注解支持可用于Portlet MVC和Servlet MVC。以这种风格实现的控制器不必实现特定的接口或扩展特定的基类。此外，它们通常不直接依赖于Portlet API或Servlet，尽管您可以轻松配置对Portlet或Servlet设施的访问。

（一）使用@Controller定义控制器

@Controller注解指明特定类作为控制器。不需要引用Servlet API或扩展任何控制器基类。

@Controller注释作为注释类的原型，表示其作用。调度程序通过检测 @RequestMapping注释以获取映射方法

（二）使用@RequestMapping映射请求

SpringMVC使用@RequestMapping注释来将URL映射到特定的处理程序方法或整个类。通常，类级注释代表特定的请求路径（或路径模式）映射到表单控制器上，利用其他方法级或注释HTTP请求参数条件来缩小了特定HTTP请求方法（“GET”，“POST”等）的主映射。

2.2 Hibernate框架介绍

Hibernate ORM，简称Hibernate，是Java编程语言的对象关系映射工具，使用面向对象的方法关联关系数据库可能是麻烦且耗时的。由于数据在对象与关系数据库之间的表示方式有很大差异，从而导致开发成本明显更高。Hibernate是针对Java环境的对象-关系映射解决方案。术语“对象/关系映射”是指将关系数据模型表示与对象模型表示之间的数据映射的技术。是一个用于映射的面向对象的域模型的关系数据库的框架。Hibernate 通过用高级对象处理函数替换直接解决对象关系阻抗不匹配问题。主要功能是数据库表映射到Java类，以及从SQL数据类型映射到Java数据类型。Hibernate提供检索和数据查询、自动SQL调用的功能。缓解开发人员对结果集的对象转换和手动处理。

Java类到数据库表的映射是通过使用Java注释或配置XML文件实现的，它还可以使用XML文件或Java注释来维护数据库模式。当使用XML文件时，Hibernate可以为持久性类生成框架源代码。而使用注释时则以辅助的形式。

Hibernate支持自定义值类型的映射。这使得在将列映射到属性时，覆盖默认SQL类型、将单个属性映射到多个列、将Java 枚举映射到列等成为可能。

Hibernate提供HQL的SQL启发语言，允许针对Hibernate的数据对象编写类似SQL的查询语句。HQL（Hibernate Query Language）是面向对象的SQL版本。它生成独立于数据库的查询，因此不需要编写特定于数据库的查询。

Hibernate为普通Java对象（POJO）提供透明的持久性。无参构造函数是持久化类的唯一严格要求，但不一定是public。通常用Java集合类来存储数据对象的集合，例如Set和List接口的实现。可以将相关对象配置为将操作从一个级联到另一个。还可以在独立的Java应用程序和使用servlet，EJB会话bean和JBI服务组件的Java EE应用程序中使用。它也可以被包括在其他编程语言中。

在Hibernate 会话中，Hibernate的一个独立的对象持久机构就是一个实体，其可以被操纵独立于其他对象。相比之下，组件从属于一个实体，只能针对该实体进行操作。

org.hibernate.Session接口表示一个Hibernate会话，即对数据库实体执行操作的要点。主要活动包括管理持久性状态的对象，从数据库中获取持久化的事物和管理事务划分，一个会话旨在持续数据库上的逻辑事务。

Hibernate可以显着减少开发时间，否则在SQL和JDBC中手动处理数据。Hibernate的设计目标是通过消除对使用SQL和JDBC的手动，手工制作的数据处理的需要，缓解开发人员95％的常见数据持久性编程任务。然而，与许多其他持久性解决方案不同，Hibernate并不会隐藏SQL的强大功能，并保证您对关系技术和知识的投资与以往一样有效。

对于仅使用存储过程来实现数据库中的业务逻辑的以数据为中心的应用程序，Hibernate可能不是最佳解决方案，它在基于Java的中间层中面向对象的域模型和业务逻辑最为有用。但是，Hibernate当然可以帮助您删除或封装供应商特定的SQL代码，并简化表达式取结果集转换为对象的过程。

2.3 Ajax技术介绍

AJAX是由杰西·詹姆士·贾瑞特所提出的异步JavaScript与XML技术，一套综合了多项技术的浏览器端网页开发技术。

传统的web开发方式，是用户填写表单，当用户提交表单时就向服务器发送一个请求，服务器接收用户发送过来的请求，解析其传过来的表单，处理之后，给用户客户端返回一个新的网页。这种开发方式有很大一个缺点，就是浪费贷款，因为提交表单前后的两个页面有时内容大致相同，其中代码大致是相同的，但是由于每次客户点与服务器的沟通，都需要发送请求，这就导致响应往往会变得很慢，用户体验变得很差。

与传统的web开发方式不同的是，Ajax只向服务器发送和取回所需要的数据，在客户端采用Javascript接收来之服务器的响应，因此这大大减少了浏览器客户端与服务器交换的数据量，减少带看的占用，是服务响应时间更短，响应速度更快，同时，在利用ajax发送请求之前，可以先在客户端先处理数据之后再发送，这样大大减轻了服务器的负荷。

Ajax是一个有机的利用一系列前端技术相关的技术，类似LAMP或DHTML,Ajax传送的数据可以由JSON数据代替，使用Ajax最大的优点，能在不刷新整个页面的情况下更新网页的内容，这使得web应用源源不断的向服务器取数据，并且可以快速的响应用户的动作，避免发送一些不需要改变的信息。

Ajax是Javascript的方法，并不需要用户安装任何浏览器插件就可使用，但是各大浏览器的兼容性不同，使得在不同浏览器上使用Ajax与服务器交互时的方法存在差异，但是基本原理都相同，随着前端技术的不断成熟，一些简化的ajax使用方法或程序库也相继被设计出来，如jQuery就对Ajax进行了很好的封装，使开发者使用Ajax变得更加方便，并且具有很高的浏览器兼容性。

但是使用Ajax也有一个缺点，就是Ajax破换了浏览器的后退与收藏功能，当使用Ajax动态更新网页内容之后，用户就无法回到之前的页面状态，浏览器只会记下历史记录，并不会记录网页的动态改变，在Ajax应用中，用户是无法通过单击后退按钮或者返回按钮，取消前一次操作或者回到之前的状态。

但HTML5的出现，使得用户可以操作历史记录，解决了使用Ajax无法返回之前状态和无法回退的问题。

在使用Ajax开发时，需要考虑的一个问题就是网络延迟，因为往往在开发时，都是使用ajax的异步更新的方式，如果在用户发送请求时，没有给用户明确的操作提示，如正在请求数据，会让用户感觉不到自己已经发送了请求，会多次触发请求，造成想服务器发送多次相同的请求，通常在做Ajax操作时，使用一个提示向用户提示正在操作或正在读取数据，提高系统的友好性。XmlHttpRequest对象创建方法:

var xmlHttp;

if (typeof XMLHttpRequest != "undefined") {

xmlHttp = new XMLHttpRequest();

} else if (window.ActiveXObject) {

var aVersions = ["Msxml2.XMLHttp.5.0", "Msxml2.XMLHttp.4.0", "Msxml2.XMLHttp.3.0", "Msxml2.XMLHttp", "Microsoft.XMLHttp"];

for (var i = 0; i < aVersions.length; i++) {

try {

xmlHttp = new ActiveXObject(aVersions[i]);

break;

} catch (e) {}

}

}

Ajax是异步向服务器提交需求的通信方式。对服务器来说，其与传统的提交窗体并没有什么不同，正由于是以异步的方式提交请求，如果同时发送多个Ajax请求，将无法保证是哪一个请求先到达服务器并获得服务器的响应。这就会造成服务器应用程序的多线程（thread）或多进程（process）的竞争（racing）问题。所以开发者在开发时，必须处理好请求的先后次序，解决好竞争问题。

2.4 MUI前端框架

MUI是国内优秀的跨平台开发框架，在传统移动应用开发中，性能和体验一直是开发者最求的，但是两者是不可兼得的，原生Java语言开发的app，开发成本大，并且开发效率低，但采用HTML5开发的webApp在性能和体验上目前又达不到原生想要的效果，MUI则是专为解决HTML5开发遇到的切页白屏，转页动画难看，浮动元素会抖动，并且在低端手机上下拉不流畅、卡顿的现象而开发的一款跨平台的开发框架，使用HTML5技术为基础，混合的开发模式，将原生app和H5开发很好的结合在一起，取长补短，解决使用原生语言开发效率低下，开发成本高的问题。

MUI框架的首要目标是追求性能体验，并且不依赖第三方插件库，自己开发MUIjs库，整合了javascript和原生语言的通信机制，让javascript可以调用原生样式，从而达到原生的体验。MUI同时还追求原生感觉，所以整个UI控件模仿了IOS，通知还加入了android的一些常用UI控件。

在传统的H5实现下拉刷新的开发方式上，采用的是DIV模拟下拉刷新，这种方式，在低端手机上会显得卡顿，为了解决这个问题，MUI采用双webview来解决DIV拖动不流畅的问题，拖动时，是拖动一整个webview,回弹使用原生的动画实现，这使得下拉卡顿的问题在低端手机上得到很好的解决。

2.5 HTML5介绍

HTML5 是一种标记语言，用于构建和呈现内容的万维网。它是HTML标准的第五个和当前版本。它由世界万维网联盟（W3C）在2014年10月发布，旨在通过支持最新的多媒体来改善语言，同时保持人类易于阅读，并通过电脑和设备（如网络）一贯理解浏览器，解析器等。HTML5不仅包含HTML 4，还包含XHTML 1和DOM Level 2 HTML。HTML5包括详细的处理模型，以鼓励更多的可互操作的实现; 它的延伸、改进和合理化可用于文档标记，并介绍了标记和应用程序编程接口复杂（API）的Web应用程序。由于同样的原因，HTML5也是跨平台移动应用程序的候选者，因为它包括设计在低功耗设备的功能。

包括许多新的语法特征。本地地包括和处理多媒体和图形内容，新的<video>，<audio>并且<canvas> 元件为可伸缩矢量图形（加入和支持SVG）含量和MATHML为数学公式。以丰富的语义的文件，新的页面结构元素，如内容<main>，<section>，<article>，<header>，<footer>，<aside>，<nav>和<figure>，被添加。引入了新的属性，重新定义或标准化了其他元素。

HTML5无法在网页内提供动画。动画HTML元素需要额外的JavaScript或CSS3功能。

HTML5新功能：

1）HTML5的DOCTYPE声明更简单：

<!DOCTYPE html>

2）字符编码（charset）声明也很简单：

<meta charset="UTF-8">

HTML5新的元素：

1）新的语义元素， 如<header>，<footer>，<article>和<section>。

2）表单元素的新属性，如数字，日期，时间，日历和范围。

3）新的图形元素：<svg>和<canvas>。

4）新的多媒体元素：<audio>和<video>。

2.6 jQuery介绍

传统模式WEB前端的开发是先由UI设计师（User Interface Designer）使用Photoshop、Adobe illustrator等图像编辑软件将设计出的网页效果制作出图片格式的效果图，再由前端程序员按照设计出的图编写HTML、CSS、JavaScript代码。这种方式的优点显而易见，它比较灵活。但缺点也很突出，它开发的效率不高。因为在有些情况下，开发人员人手不足，这时，同一个程序员既需要实现服务器端的业务逻辑，又需要编写前端页面的样式，对于代码质量的提高有很大难度。

为了解决这些问题，业界顺势而产生了许多前端的框架。对于一般项目经常使用的组件，许多前端框架都已经实现了一套或多套样式，开发人员只需要简单地引入这些组件即可实现许多丰富的功能。这样使程序员无需花大量精力编写前端代码，而更多的将精力集中于业务逻辑的实现。

jQuery是轻量级的JavaScript库，创始人和技术领袖是John Resig，吸引了世界各地的许多高端人才加入其团队。jQuery也是一个既快速又简洁的JavaScript库，他使得用户更方便的处理HTML DOM文档、事件监听和处理、以及实现动画效果等，对常用方法的封装，如常用的Ajax，使得开发者开发更加方便。

jQuery兼容许多浏览器，写得更少,做得更多(write less,do more)成为其的核心理念。

开源免费，使用MIT许可协议。使开发更加便捷，例如选择DOM元素、操作文档对象、事件处理、制作动画效果、使用Ajax以及其他功能。除此以外，jQuery可以让开发人员自由编写插件。开发人员可以利用其模块化方式开发出功能强大无比的静态网页或动态网页。

2.7开发环境介绍

表2.7\_1系统开发环境配置表

|  |  |
| --- | --- |
| 操作系统版本 | Windows 10 64-bit |
| 集成开发环境 | Eclipse + sublime |
| 服务器 | Tomcat 8 |
| 数据库 | MySQL 5.6 |
| J2EE | JDK8 |
| Spring Framework | Spring 4.3.4 |
| SpringMVC Framework | Spring 4.3.4 |
| Hibernate Framework | Hibernate 5.2.9 |

3.需求分析

3.1 系统定义

满足美食爱好者发现美食、寻找美食、制作美食、分享美食、推荐美食的需求，最终形成一个以美食爱好者分享与交流的兴趣圈。

系统采用b/s架构，以Html5形式向App提供服务。以美食推荐和社交关系为纽带，为用户提供优质的本地美食信息服务。说得简单点就是，美食推荐类应用—专注于本地精品美食的发现与推荐，让其他用户能方便快捷地找到更可信任、更可靠、更感兴趣的美食信息。

本系统本着移动优先的设计理念，因为手机成为现在人们生活的一个必备品，当你在等车，坐地铁、或是闲时无聊的时候，都会看看手机，打发时间，随着手机网名在逐渐的增多，中国手机网名占到了60%以上，并且选择适配移动端，更能适合用户的使用习惯，所以H5加响应适配的开发方案是我开发本系统的首选技术和方案，并且用户不需要额外安装客户端，随便在一个浏览器或者在微信浏览器中就能轻松打开，跨平台、开发成本小、开发效率高一直受到很多开发者的青睐。

3.2系统用例分析

3.2.1 用户注册登录用例分析与描述

首先是注册，用户账号必须使用手机号注册，密码是自定义的，注册时，先查询该手机号是否已经注册，如果注册了，则返回该手机号已注册，如果没有注册，则将注册信息插入数据库，包括手机号和密码，这里的存入数据库的密码是经过前端MD5加密后的32位字符串，加密过程在前端完成，传到后台进行保存。

注册成功之后，自动切换到登录面板，用刚刚注册的手机号和密码进行登录，如果账号和密码都正确的化，则返回登录成功，跳到首页，如果账号和密码错误，则提示登录失败。

这里的注册和登录都是通过ajax与后台交互，注册时，js获取输入的手机号和密码，发送到后台，后台接收请求，解析参数，然后根据手机号去数据库中查询，返回响应的结果。

当用户登录成功后，后台返回用户的手机号和id,前端将用户的手机号和id存在sessionStorage中，用于应用的全局使用，比如购买美食时，需要获取用户的登录信息和手机号，查看订单、我的推荐和我的收藏时，都需要用户的id作为关联标志。

把用户的登录信息存在sessionStorage中，当用户退出应用或者关闭浏览器之后，这些信息就会自动的销毁掉，当下次进入应用时，需要重新登录验证用户。

3.2.2首页用例分析与描述

首页由banner幻灯和美食列表组成，美食列表分类有：美食、食材、食谱、其他四个分类。

美食列表主要是一些可以购买的美食，用户可以查看美食的详情，如果有喜欢的美食，可以点击下单购买。

食谱和食材主要是介绍内的文章，有介绍食材的营养成分，食材营养搭配，食谱中有美食制作的图文教程，和生活小tip。

在初始化首页时，先去请求banner列表，包括banner的图片地址和链接地址，当banner请求成功之后，就去请求美食列表信息，这里的banner请求和美食列表的请求时串行的，因为如果设计是并行话，两个请求同时到达后台，两个请求对数据库数据查询操作时，先完成的请求会把数据库连接关闭，导致剩下的另一个请求操作失败，包Resultset is closed 异常。

3.2.3 推荐页和社区页用例分析与描述

推荐页的功能主要是推介最近上传的美食，有标明美食的名称、地址、和美食相关的图片，有美食、食材、食谱、美食教程等类容，点击相关类表可以查看详情。

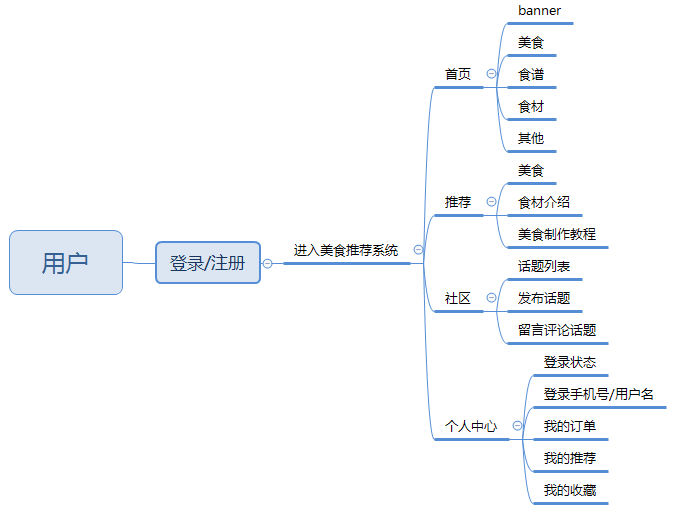
社区主要用于用户讨论有关美食相关的话题，可以是吃过哪家美食之后的心得，可以推荐给其他网友，或者是某种美食做法的讨论，更或者是有关美食其他话题，其他用户参与话题的方式是可以参与评论留言。

3.2.4个人中心用例分析与描述

在个人中心，用户可以看到自己登录的手机号，可以自己设置头像，如果用户没有设置头像，则会显示默认的头像，有登录和退出的入口，另外在个人中心面板上，用户可以看到自己之前购买的订单，订单的状态，以及自己推荐的文章和收藏的话题。

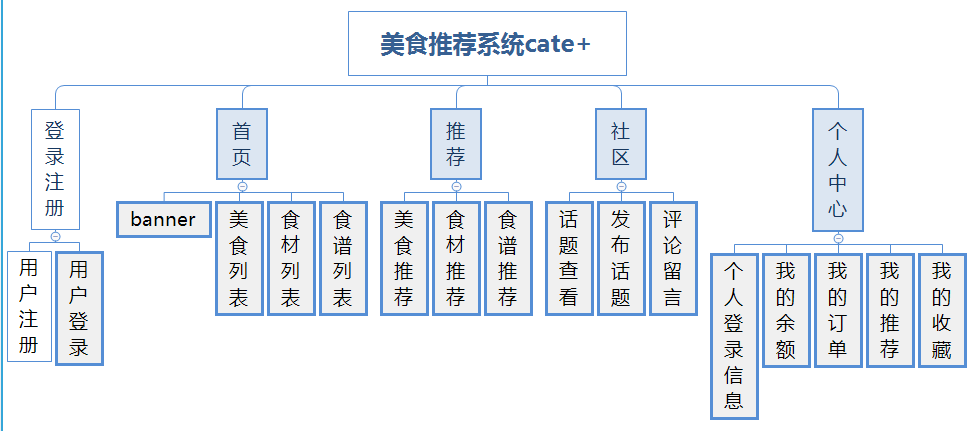
点击点单，可以查看订单的详情，点击我的推荐和我的收藏可以查看相关文章类表详情。

3.2.5 系统用例图



3.2\_1系统用例图

3.3系统功能模块分析

3.3.1 系统功能模块设计

3.3\_1系统功能模块框图

3.3.2登录注册模块分析

注册面板元素由三个输入框加一个注册按钮组成，第一个为电话号码输入框，第二个为密码输入框，第三个为确认密码输入框，点击注册时，首先检查用户输入的手机号的格式是否是正确的手机号格式，这里用到的是正则表达式匹配，密码输入要报两次输入的密码输入一致才能通过，否则注册不通过。

登录面板元素由账号输入框和密码输入框组成，用户输入密码之后，点击登录，用户的手机号和密码将通过ajax发送到后端进行验证，如果用户合法，则登录成功，否则返回失败。



3.3\_2登录注册界面截图

3.3.3首页模块分析

首页主要由banner和美食列表组成，美食列表由美食、食材、食谱和其他等4个选项卡组成，其中美食可以查看详情，点击购买下单。下单模块中可以由选择收货地址、选择支付方式、选择用餐人数和口味偏好等功能模块组成。

点击食材列表可以查看食材详情，点击食谱列表可以常看食谱详情，内容包含食材介绍，美食制作图文教程



3.3\_3首页界面截图

3.3.4推荐模块分析

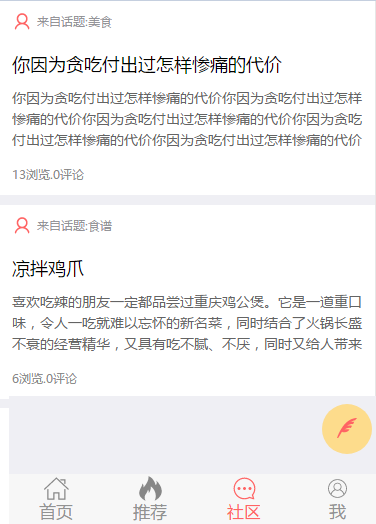
推荐模块主要推荐最近推荐的美食、食材、还有食谱，点及推荐的美食，可以查看美食的详情，包括店名，地址等信息，用户可以推荐食材和美食的做法，需要写成文档发布，其他用户可以查看别人的推荐，并且可以收藏专题。



3.3\_4推荐界面截图

3.3.5社区模块分析

在社区模块中，用户可以查看别人发布的话题，并且可以以评论留言的方式参与到比人发布的话题中讨论，当然自己也可以发布一个话题，社区的内容是和美食相关，比如吃过的美食感悟分享，到某地旅行中吃过的美食，以及想知道某种美食有什么营养成分等，当然也不限于其他美食的话题，主要功能是一美食为纽带，建立一个美食社交圈，提供大家相互交流学习的入口。



3.3\_5社区界面截图

3.3.6个人中心模块分析

个人中心面板主要有用户头像，用户登录账号（也就登录的手机号）登录退出按钮，在个人中心，可以查看余额信息，订单信息，我的推荐列表，我的收藏列表等。

进入我的订单面板，可以查看最进下过的订单，包括订单的状态，购买美食的名字，下单的时间，花费的金额等信息。

进入我的推荐和我的收藏，可以查看自己的发布的推荐信息列表和收藏列表。



3.3\_6个人中心界面截图

4 系统数据库设计

4.1数据分析

用户信息，主要记录用户的账号/手机号,密码，默认地址,用户名，代金券，性别，默认地址id，头像链接地址，余额，积分等信息，当用户没有登录时，可以进入系统，可以查看美食、食材、食谱等列表及详情，推荐及推荐详情，社区及社区详情等，但是当用要购买美食，或者在发布推荐信息、社区上发布话题是都需要用户的登录信息。

在首页面板当中，记录的信息主要有头部的banner和食物列表相关的信息，所以数据库中，单独给banner和食物列表建一个数据表来存储数据，banner表主要banner图片的链接地址，这里只存图片上传的相对路径，在本系统中，将图片上传到tomcat的webapps目录下非项目目录（cate目录）中，这样做为了更好的移植性，并且eclipase重新发布目录后不会覆盖原有的图片，banner表中还保存banner跳转的链接地址，在美食列表中主要记录美食的所属分类、评分、更新时间、食物名、价格、店名、美食所在地区的省市区及详细地址、浏览量、优惠价格、满减优惠价格等信息，如果食物列表的食物类型是食材或者食谱，则会有关于食材或食谱的内容介绍。

在订单包中，重要记录用户每次下单的信息，包括订单id，所订的食物，价格、下单日期、食物单价、数量、总价、总花费、用餐人数、口味偏好等信息。用户可以在个人中心查看自己最进下过的单的信息。

地址列表主要用来存储用户的购买美食之后填写的收货地址，包括省市区、详细街道机街道门牌地址、收货人姓名、收货人手机号码等。

收藏列表和推荐列表，主要记录用户收藏的美食或者社区讨论话题等内容。

4.2数据字典

根据数据分析和数据流图写出相应的数据字典。

4.2.1 基本数据项

|  |
| --- |
| 名称：用户表  简述：用户信息，系统的用户信息，  类型：自定义类型  属性：id(主键，整型)，电话（字符串），密码（字符串），用户名（字符串），代金券（浮点型），性别（整型，1表示男，0表示女），默认地址id（整型，对应一条地址类表的id），余额（浮点型），头像（字符串），积分（整型） |

|  |
| --- |
| 名称：食物  简述：食物列表，食物的分类有美食，食材，食谱或者其他  类型：自定义类型  属性：食物id（整型），分类（字符串，包括meishi、shicai、shipu、other）,评分（浮点型），更新时间（DATETIME类型），经度（浮点型），纬度（浮点型），省市区（字符串），详细地址（字符串），浏览量（整型），图片地址（字符串），食物名称（字符串），价格（浮点型），满减优惠价格（浮点型），优惠价格（浮点型），其他优惠说明（字符串），食物内容介绍（字符串），店名（字符串） |

|  |
| --- |
| 名称：地址  简述：地址信息，收货人地址  类型：自定义类型  属性：地址id(整型)，关联的用户id(整型)，省市区（字符串），详细地址（字符串），收款人姓名（字符串），收款人性别（整型：1表示男，0表示女），收款人电话（字符串） |

|  |
| --- |
| 名称：订单  简述：用户订单信息  类型：自定义类型  属性：id(主键)，订单id(字符串，由java的uuid生成的随机字符串)，关联的食物id（整型），关联的用户id（整型），代金券（浮点型），店铺名（字符串），食物名称（字符串），食物价格（浮点型），食物花费（浮点型），购买数量（整型），包装费（浮点型），运送费（浮点型），满减优惠价格（浮点型），总花费（浮点型，不算代金券抵扣），待支付（浮点型），用餐人数（整型），口味偏好备注（字符串），订单状态（整型：0表示未支付，1表示已支付，-1表示取消订单），下单日期（DATATIEM类型），收货地址（字符串），支付方式（字符串） |

|  |
| --- |
| 名称：banner  简述：首页banner幻灯表  类型：自定义类型  属性：banner的id（整型，主键），排序编号（整型），图片地址（字符串），跳转链接地址（字符串），是否上架标志（整型：1表示上架，0表示下架），上架时间（DATATIME） |

|  |
| --- |
| 名称：收藏  简述：收藏信息，包括收藏的美食信息和收藏的社区话题信息  类型：自定义类型  属性：收藏id（主键），关联的用户id（整型），关联的食物id（整型），关联的推荐id（整型），关联的社区id（整型），是否收藏标志（整型：1表示收藏，0表示取消收藏） |

|  |
| --- |
| 名称：评论  简述：评论留言信息  类型：自定义类型  属性：评论id（整型，主键），关联的用户id（整型），关联的话题id（整型），评论时间（DATATIME）,评论内容（TEXT类型） |

|  |
| --- |
| 名称：社区  社区：社区话题信息  类型：自定义类型  属性：话题id（整型，主键），关联的用户id（整型），话题的分类（字符串），话题标题（字符串），话题内容（TEXT）,话题的摘要（字符串），话题浏览量（整型），话题发布时间（DATETIME）,是否参数（整型：1表示删除，0表示未删除） |

4.3数据访问对象及实体类设计

4.3\_1 系统数据访问对象设计表集合

|  |
| --- |
| UserDao |
| add(User): boolean  queryByid(int): User  queryByPhone(String): User  updateById(int, User): boolean  deleteById(int): boolean |

|  |
| --- |
| FoodDao |
| add(Food): boolean  getImg(int): String  queryByClassify(String):List<Food>  queryById(int): Food  queryRecomment()：List<food>  updateById(int, Food): boolean  deleteById(int): boolean |

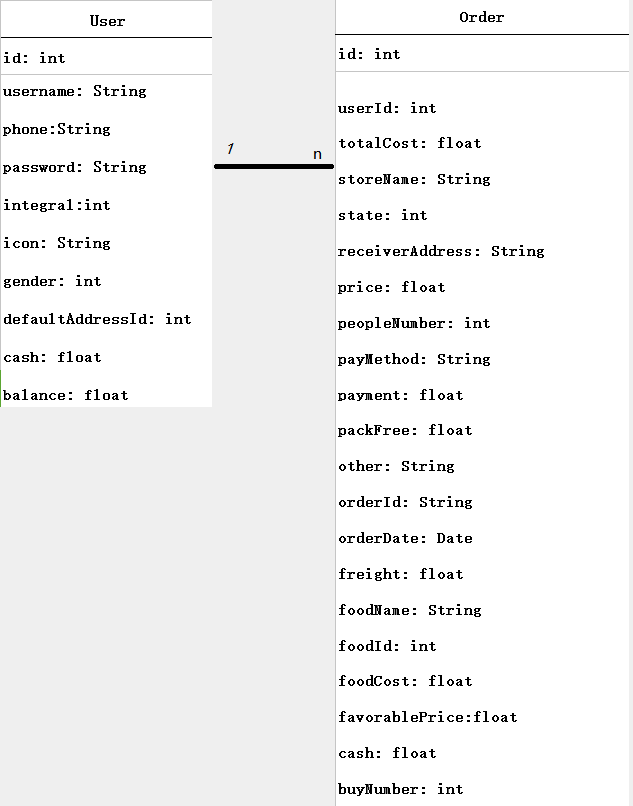
|  |
| --- |
| OrderDao |
| add(Order): boolean  queryById(int): Order  queryByUserId(int): List<Order>  updataOrderState(int): boolean  update(Order): boolean |

|  |
| --- |
| AddressDao |
| add(Address): boolean  queryAll(int):List<Address>  update(Address): boolean  deleteById(int): boolean |

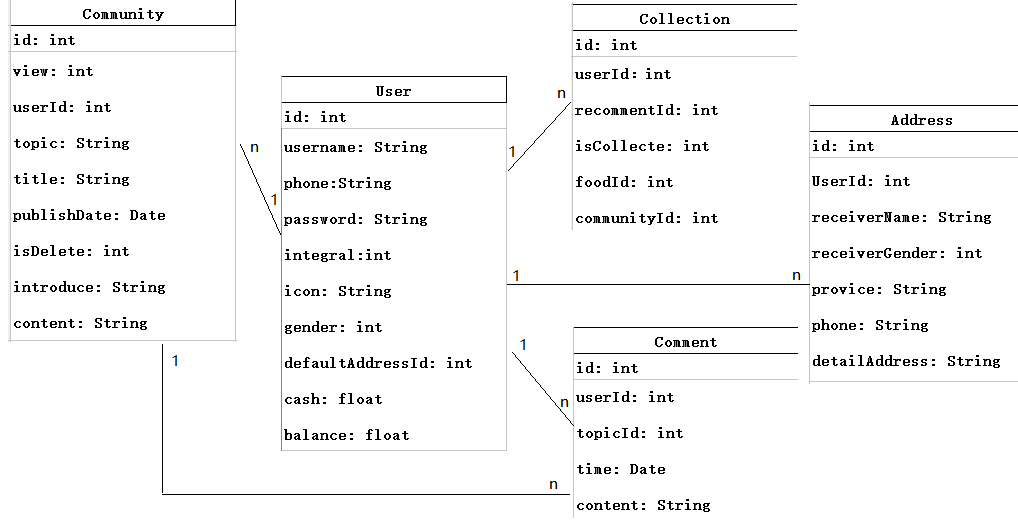
|  |
| --- |
| BannerDao |
| add(Banner): boolean  query(): List<Banner>  deleteById(int): boolean  update(Banner): boolean |

|  |
| --- |
| CommentDao |
| add(Comment): boolean  countTopicNumber(int): int  deleteById(int): boolean |

|  |
| --- |
| CommunityDao |
| addTopicView(int): boolean  queryAll(): List<Community>  queryById(int): Community |

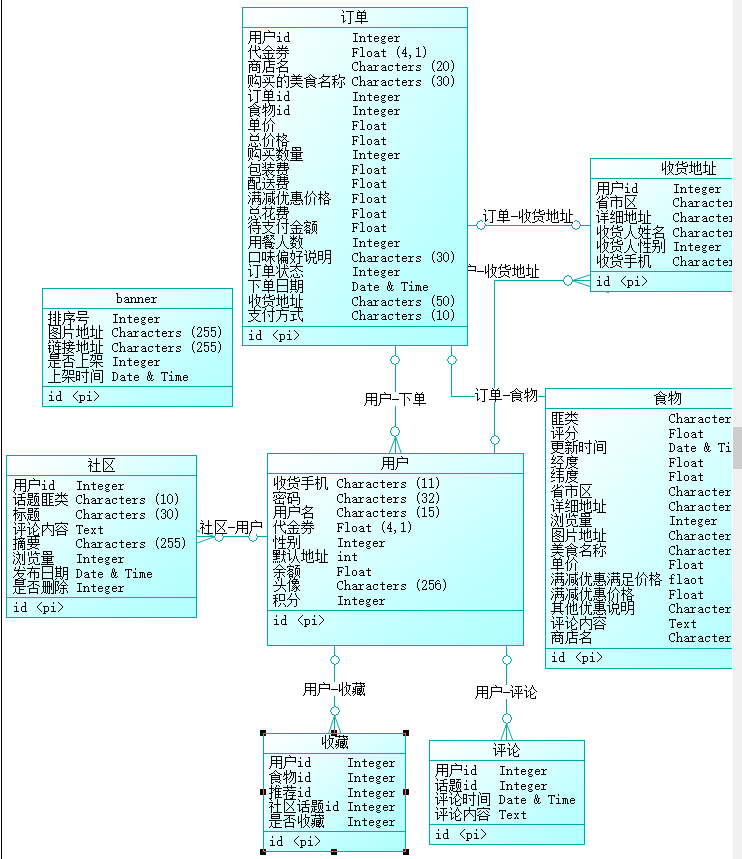


4.3\_2系统用户-订单实体类图



4.3\_3系统用户-社区-收藏-评论-地址实体类图

4.4实体-关系设计



4.4\_1实体关系设计截图

4.5数据表结构

根据实体-关系E-R图，更为cate+设计了user，order，food，community，comment，collection，address，banner 8个表

1.用户表，此表包含id(作为主键)，电话，密码，用户名，代金券，性别，默认收货地址id，余额，头像，积分等10个字段；具体如4.5\_1表所示；

4.5\_1表 用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 约束 | 缺省值 |
| Id | int(11) | 否 | PRI | auto\_increment |
| PHONE | varchar(11) | 否 | UNI |  |
| PASSWORD | varchar(32) | 否 |  |  |
| USERNAME | varchar(15) | 是 |  | 空 |
| CASH | float(4.1) | 是 |  | 空 |
| GENDER | int(1) | 是 |  | 0 |
| DEFAULT\_ADDRESS\_ID | int(11) | 是 |  | 空 |
| BALANCE | float(5,1) | 是 |  | 0.00 |
| ICON | varchar(255) | 是 |  | img/user.svg |
| INTEGRAL | int(6) | 是 |  | 0 |

2.订单表，此表有订单id(主键)，食物id，用户id，代金券，商店名，食物名，单价，食物花费，购买数量，包装费，配送费，满减优惠价格，总花费（不算代金券抵扣），带支付，用餐人数，口味偏好备注，下单时间，收货地址，支付方式等字段，具体如4.5\_2表；

4.5\_2表 订单表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 约束 | 缺省值 |
| ID | int(11) | 否 | PRI | auto\_increment |
| ORDER\_ID | varchar(64) | 否 | UNI |  |
| FOOD\_ID | int(11) | 否 |  | 0 |
| USER\_ID | int(11) | 否 |  | 0 |
| CASH | float(4,1) | 是 |  | 空 |
| STORE\_NAME | varchar(20) | 是 |  | 空 |
| FOOD\_NAME | varchar(30) | 否 |  |  |
| PRICE | float(6,1) | 否 |  |  |
| FOOD\_COST | float(6,1) | 否 |  |  |
| BUY\_NUMBER | int(2) | 否 |  |  |
| PACK\_FEE | float(3,1) | 是 |  | 0.5 |
| FREIGHT | float(3,1) | 是 |  | 空 |
| FAVORABLE\_PRICE | float(3,1) | 是 |  | 空 |
| TOTAL\_COST | float(7,1) | 否 |  |  |
| PAYMENT | float(7,1) | 是 |  | 空 |
| PEOPLE\_NUMBER | int(2) | 是 |  | 1 |
| OTHER | varchar(30) | 是 |  | 空 |
| STATE | int(1) | 是 |  | 0 |
| ORDER\_DATE | datetime | 是 |  | 0000-00-00 00:00:00 |
| RECEIVER\_ADDRESS | varchar(50) | 是 |  | 空 |
| PAY\_METHOD | varchar(10) | 是 |  | 空 |

3.食物表，此表定义了食物id，食物分类，评分，更新时间，经度，纬度，所在地的省市区，详细地址，浏览量，图片地址，食物名称，食物单价，满减优惠应满足的价格，优惠价格，其他优惠说明，食物介绍，商店名等字段，具体见4.5\_3表所示；

4.5\_3表，食物表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 约束 | 缺省值 |
| Id | int(11) | 否 | PRI | auto\_increment |
| CLASSIFY | varchar(10) | 否 |  | meishi |
| SCORE | float(2,1) | 是 |  | 5.0 |
| UPDATE\_TIME | datetime | 否 |  | 0000-00-00 00:00:00 |
| LONGITUDE | float(10,7) | 是 |  | 0.0000000 |
| LATITUDE | float(10,7) | 是 |  | 0.0000000 |
| ADDRESS | varchar(30) | 是 |  | 空 |
| DETAIL\_ADDRESS | varchar(30) | 是 |  | 空 |
| VIEW | int(6) | 是 |  | 0 |
| IMG\_URL | varchar(255) | 否 |  |  |
| NAME | varchar(30) | 否 |  |  |
| PRICE | float(6,1) | 是 |  | 0.0 |
| REACH\_PRICE | float(7,1) | 是 |  | 0.0 |
| FAVORABLE\_PRICE | float(4,1) | 是 |  | 0.0 |
| OTHER\_FAVORABLE | varchar(30) | 是 |  |  |
| CONTENT | text |  |  |  |
| STORE\_NAME | varchar(20) | 是 |  | 空 |

4.社区表，此表定义了话题id，用户id，话题分类，话题标题，话题内容，话题摘要，话题的浏览量，话题发布时间，是否删除标志等字段，具体如4.5\_4表所示；

4.5\_4表，社区表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 约束 | 缺省值 |
| Id | int(11) | 否 | PRI | auto\_increment |
| USER\_ID | int(11) | 否 |  | 0 |
| TOPIC | varchar(10) | 是 |  | 空 |
| TITLE | varchar(30) | 否 |  |  |
| CONTENT | text | 否 |  |  |
| INTRODUCE | varchar(255) | 是 |  | 空 |
| VIEW | int(8) | 是 |  | 空 |
| PUBLISH\_DATE | datetime | 是 |  | 0000-00-00 00:00:00 |
| IS\_DELETE | int(2) | 是 |  | 0 |

5.评论表,此表定义了评论id，评论用户的id，所评论话题id，用户评论时间， 评论的内容等字段，具体如4.5\_5表所示；

4.5\_5表 评论表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 约束 | 缺省值 |
| Id | int(11) | 否 | PRI | auto\_increment |
| USER\_ID | int(11) | 否 |  | 0 |
| TOPIC\_ID | int(11) | 否 |  | 0 |
| TIME | datatime | 是 |  | 0000-00-00 00:00:00 |
| CONTENT | text | 否 |  |  |

6.收藏表，此表定义了收藏id，用户id，食物id，推荐食物关联的id，社区话题id，是否收藏标志等字段，具体如4.5\_6表；

4.5\_6表 收藏表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 约束 | 缺省值 |
| Id | int(11) | 否 | PRI | auto\_increment |
| USER\_ID | int(11) | 否 |  | 0 |
| FOOD\_ID | int(11) | 是 |  | 0 |
| RECOMMENT\_ID | int(11) | 是 |  | 空 |
| COMMUNITY\_ID | int(1) | 是 |  | 空 |

7.收货地址表，此表定义了地址id，用户id，省市区，详细地址，收货人姓名，收货人性别，收货人电话等字段，具体见4.5\_7表；

4.5\_7表 收货地址表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 约束 | 缺省值 |
| Id | int(11) | 否 | PRI | auto\_increment |
| USER\_ID | int(11) | 是 |  | 空 |
| PROVINCE | varchar(30) | 是 |  | 空 |
| DETAIL\_ADDRESS | varchar(30) | 是 |  | 空 |
| RECEIVER\_NAME | varchar(10) | 是 |  | 空 |
| RECEIVER\_GENDER | int(1) | 是 |  | 1 |
| PHONE | varchar(11) | 是 |  | 空 |

8.banner表，用于存储首页banner 的信息，此表定义了banner的id，图片地址，排序序号，连接地址，是否上架标志，上架时间等字段，具体如表4.5\_8所示；

表4.5\_8 banner表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 能否为空 | 约束 | 缺省值 |
| Id | int(11) | 否 | PRI | auto\_increment |
| ORDER\_NUMBER | int(2) | 是 |  | 空 |
| IMG\_URL | varchar(255) | 否 |  |  |
| LINK\_URL | varchar(255) | 是 |  | 空 |
| IS\_ONSHELVE | tinyint(1) | 否 |  | 1 |
| ONSHELVE\_TIME | datetime | 否 |  | 2017-04-04 18:15:34 |

5.系统功能设计

5.1系统功能模块设计

安功能模块划分，系统大致可以分为登录注册，首页，推荐，社区，和个人中心这5打模块，同时各模块当中有细分模块；

本系统采用前端和后台分开开发的模式，目的是为了更好的调试和维护升级，相互不影响，开发更为方便，前端用ajax向后端发送请求，后端接收请求，返回处理之后的结果；

5.1.1注册模块设计

注册时，需要用户填写手机号，和登录密码，当用户点及注册时，前端通过javascript获取用户输入的手机号码和密码，先监测用户的输入是否为空，如果为空，则用toast的方式提醒用户输入相关信息，如果用户填写信息不为空，则监测用户输入的手机号是否合法，这里用的是js的正则表达式进行监测，如果用户输入的不是合法的手机号，则toast提示用户重新输入手机号，如果用户输入合法，则用ajax将用户填写的数据发送到后端，后端用一个servlet接收处理，由于后端框架是springMVC+Hibernate，所以只需要新建一个类，并加入Spring 的注解@Controller 和对应的请求地址@RequestMapping，则后台就会匹配的请求路径，并做相应处理；

本系统请求的注册接口地址为：/user/doregister 。

请求参数为见5.1\_1表

5.1\_1表 注册请求参数

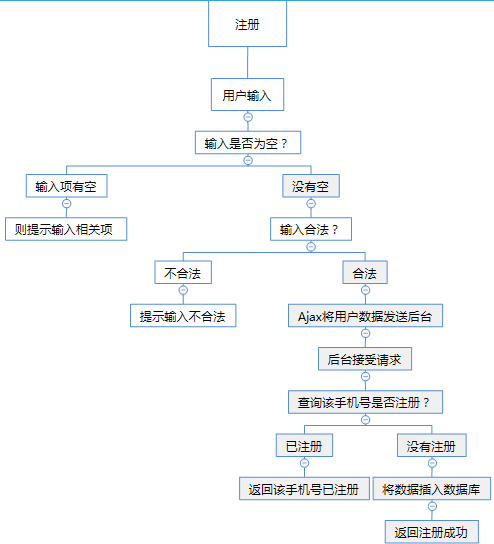
|  |  |  |
| --- | --- | --- |
| **元素** | **数据类型** | **说明** |
| phone | String | 电话号码 |
| password | String | 密码（经过md5加密） |

注册后端响应内容如5.1\_2表

5.1\_2表 注册响应内容

|  |  |  |  |
| --- | --- | --- | --- |
| **元素** | **子元素** | **数据类型** | **说明** |
| header | success | boolean | 成功标志：如果为false，表示失败，详情见errorCode和errorInfo元素说明。 |
| errorCode | String | 错误编码 |
| errorInfo | String | 错误信息描述 |

注册模块功能处理逻辑，如5.1\_3图。



5.1\_3图 注册模块功能设计

5.1.2登录模块设计

登录时，需要用户用注册过的手机号和注册的密码进行登录，登录时，先检测用户输入时候为空，如果为空则提示输入相关项，如果不为空，则检测输入的手机号是否合法，如果用户输入的手机号合法，则请求登录接口，如果输入不合法，则提示输入正确的手机号，当登录成功之后，后台返回用户的信息，前端接收到后台返回的信息之后，将用户的手机号和用户一点存入sessionStorage中，供系统其他地方需要时使用；

登录请求的接口地址为：/user/dologin

请求的参数有手机号和密码，请求参数说明如5.1\_4表；

5.1\_4表 登录请求参数表

|  |  |  |
| --- | --- | --- |
| **元素** | **数据类型** | **说明** |
| phone | String | 电话号码 |
| password | String | 密码（经过md5加密） |

登录响应内容包括登录是否成功，如果成功，则返回内容包含用户信息，登录相应内容如表5.1\_5；

表5.1\_5 登录响应内容

|  |  |  |  |
| --- | --- | --- | --- |
| **元素** | **子元素** | **数据类型** | **说明** |
| header | success | boolean | 成功标志：如果为false，表示失败，详情见errorCode和errorInfo元素说明。 |
| errorCode | String | 错误编码 |
| errorInfo | String | 错误信息描述 |
| body | id | int | 用户编号 |
| phone | String | 电话号码 |
| username | String | 用户名 |
| defaultAddressId | int | 默认收货地址 |
| cash | float | 代金券 |

登录模块功能设计处理逻辑如图5.1\_6所示；

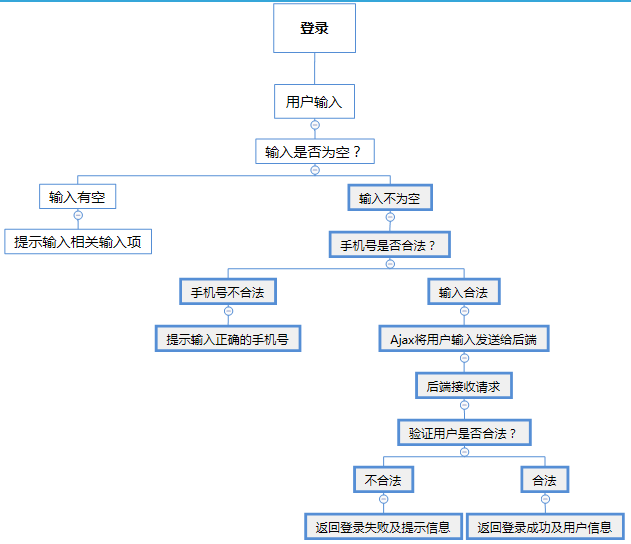


图5.1\_6 登录处理逻辑图

5.1.3首页、推荐、社区、个人中心模块功能设计

首页、推荐、社区、个人中心这4个模块的初始化基本相同，就是当用户点击跳转到相应页面时，前端会发送一个ajax请求到后台，请求初始化内容，后台返回初始化数据，前端通过javascript异步构造页面代码并进行初始化界面渲染。

基本处理逻辑相同，进入页面，请求初始化数据，然后进行页面渲染，基本逻辑处理如图5.1\_7所示；

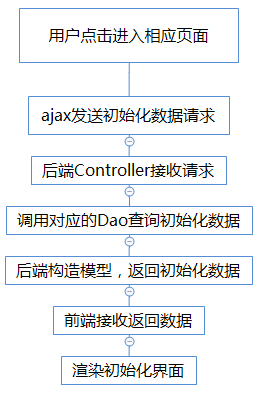


图5.1\_7 主页面初始化逻辑处理

5.1.3 购买美食模块功能设计

在用户浏览美食的过程中，如果有自己喜欢的美食，用户可以点击查看详情，并且可以下单购买，在用户点击购买下单时，先检测sessionStorage中是否存在用户信息，包括用户id和用户的手机号如果不存在，则表明用户没有登录，则跳到登录页面，如果存在，则将下单请求发送到后台，后台接收用户的处理，构造订单的model并返回数据，在订单页，需要用户选择收货地址，支付方式（默认在线支付），用餐人数（默认为1人），口味偏好等备注，其中选择收货地址为必选，如果用户在提交订单前没有选择收货的地址和联系人等信息，则提示用户选择收货地址。

在选择收货地址时，如果用户之前没有设置收货地址，则用户可以通过新增收货地址的方式，新增一条收货地址，收货地址需要用户填写，收货人姓名、联系电话、及详细地址。

为了更好的适合某些用户的口味偏好，增加了口味偏好等备注，如有的用户喜欢吃辣，有的用户不喜欢吃辣，这是为了商家能给用户提供更合适口味的美食，用户可以在备注区备注自己喜欢的口味偏好，方便商家在准备食物时候，给不同用户配置不同口味的套餐，实现更加人性化的服务。

购买美食中设计的接口有请求美食详情，当用户查看美食时，需要把美食的各种信息展示出来，需要用到该接口，第二个接口是加入购物车的接口，当用户选择好美食，并且设置好购买数量之后，点击加入购物车，或把当前用户选择的美食和购买数量发送到后端处理，后端构造订单模型数据返回，前端接收订单数据，调用订单模块进行渲染，订单主要给用户确认自己所购买的美食的一个确认，当用户选择好收获地址之后，并确认订单信息之后，就可以进行下单付款，由于本系统在开发上，没有做过相关的认证，没有接通支付功能，在本系统的支付功能设计上只是一个模拟的功能。

购买美食的逻辑处理如图5.1\_8所示；



图5.1\_8 购买美食功能模块设计

## 5.2系统关键用例设计

由于系统大部分的功能差不多，大同小异，基本原理是前端用ajax发送求情，后台接收请求，处理数据之后返回数据模型，前端接收返回数据进行渲染显示，前后端的交互基本全用到ajax进行交互。我就简单挑几个典型的按钮进行详细设计说明。

5.2.1 用户登录与注册的详细设计

没有账号的用户可以注册新账号，注册时，要求用户输入手机号phone字段，和密码passwor字段，密码需要输入两次，第二次为确认密码，点击注册按钮时，先检测用户输入项是否存在空项，如果有则提示，然后检测用户输入的手机号是否为合法的手机号，如果不合法，则用js模拟原生android的toast提示方式，向用户提示输入正确的手机号，这里验证手机号用的是js正则表达式，规则为用户输入的手机号必须以13或15或18开头，后面是9为数字，具体正则表达式为：

var reg = /^(((13[0-9]{1})|(15[0-9]{1})|(18[0-9]{1}))+\d{8})$/;

然后将用户的输入的手机号和正则表达式匹配，利用test()方法进行匹配，当用户的基本语法是reg.test(phone)；其中的phone参数是用户输入的手机号；

点击注册按钮，验证通过之后，先将用户输入的密码经过js写的MD5加密算法生成32位密码，然后构造ajax请求，将用户的手机号和密码发送到后台，在后台@Controller注解的类下用@RequestMapping注解解析请求处理路径，接收对应的请求。

后台接收到请求之后，先解析请求中传过来的手机号和参数，然后调用UserDao的queryByPhone()查询方法，用手机号查询，如果查询返回结果集大于0，则表示该手机号已经注册过该系统，则放回手机号已注册，前端提示用户手机号已注册。

如果查询返回接口等于空，则表示该手机号没有注册过该系统，则调用UserDao的add()方法，将用户的手机号和密码插入数据库中，返回注册成功，跳到登录页面。

在登录页面，用户可以用已经注册过的手机号和密码或者刚注册的密码进行登录，登录需要用户填写手机号和登录密码，同样，在点击登录按钮之前，需要检测用户输入项是否为空和用户输入的手机号是否合法，如果这些验证都通过，同样显示将用户输入的密码进行MD5加密成32为字符串密码，然后构造登录的ajax请求，将用户的手机号和加密的密码发送至后台验证。

后台同样根据@RequestMapping注解的请求处理路径，自动调用对应的方法处理登录请求。

后台接收登录请求发送过来的手机号和密码，先调用UserDao的queryByPhone()查询用户记录，如果记录条数为零，怎返回该手机号没有注册；如果有记录，则比较用户输入的密码和数据库中密码是否相等，如果不相等，则返回密码错误，否则返回登录成功和用户信息，这里放回的用户信息中去掉用户的密码。

前端接收到登录成功返回的数据后，先调用js的工具类，将用户的信息存入sessionStorage中，包括用户的id和用户的手机号，以供其他地方使用，和验证用户登录状态的标志，然后跳到系统的首页，请求首页初始化内容。

这里登录成功之后，选择把用户的信息保存到sessionStorage中，考虑到前端和后台分开开发，要记录用户的信息，只能用本地存储，而sessionStorage的好处就是当用户退出应用或者关闭浏览器之后，sessionStorage中的信息就会自动清空，比把用户信息存在cookie中长期有效的安全性稍要高一点，用户不必为防止被盗号，而每次退出应用前先清理浏览器。

5.2.2 主页面中模块调用与切换

在本系统主页面包括首页、推荐、社区、个人中心的设计中，我采用模块发开发，将各个模块抽离出来做成单个的界面，当用到哪个模块时，直接调用几个，这系统代码的编写上用到了gulp和grunt作为前端自动化工具，大大提高了前端的开发效率，使用到的功能有文件合并打包，和文件压缩，基于文件流的调试方式，减少文件的保存操作，较少了过多的重复而繁琐调试操作。

模块的开发，是先把模板做成来，当有请求时，向后台请求数据，通过js渲染模板代码，将模板代码插入到DOM文档中，进行显示，减少了开发中的耦合性。

如首页的食物列表，食物列表内容分为美食、食材、食谱、其他4大类，但是这4类的列表结构基本相同，所以就把列表部分的代码抽取出来作为一个模板，当点击到对应的分类的时候，通过js发送请求对应分类的食物数据，后端接收请求返回该分类的数据信息，前端接收到后端返回的数据后，进行模板代码的构造，然后把构造好的代码插入文档流的对应部分进行UI渲染。

如用户从美食tap切换到食材tap时，通过js获取食材tap上的data-classify值，data-classify表示的是该分类的字段名，然后够着ajax请求，将classify参数发送到/index/food/getFoodList 接口，后台接收到请求后，解析classify参数，调用FoodDao的queryByClassify() 方法查询该分类的所有事物列表，并返回给前端。

前端接收到后台返回的食物列表数据之后，用for循环解析食物列表数据，调用模板进行代码的进行构造，之后将代码插入DOM文档流中进行显示；

在各个页面当中，js模块化也显得尤为的重要，目的是减少系统的耦合性和相互依赖，当某个模块的改动，对其他模块的影响降到最小。

在js代码的模块化编写是，采用闭包的开发方式，防止命名空间的污染和变量的影响。

6.系统编码实现

6.1系统整体框架搭建

本系统是前端开发语言是HTML5+css3+javascript编写，用到的前端框架式MUI，前端自动化工具有grunt，gulp两个，主要用户代码的合并和压缩，后台开发语言为java，后台框架式SpringMVC + Hibernate，SpringMVC主要用户Controller控制层和Service业务层的调用，HIbernate主要用户数据库的增删改查操作，数据库使用的是Mysql，环境是tomcat8，jdk使用的是jdk1.8，开发工具是eclipse 和 sublime text，eclipse主要用于后台代码的编写，sublime text主要用户前端代码的编写。

在开发之前，第一步要做的事情就是环境的搭建，首先是SpringMVC的环境的构建。

6.1.1 SpringMVC框架的搭建

想在eclipase中选择 file > New > Dynamic Web Project ,新建一个动态的web项目，并勾选新建web.xml文件。

搭建SpringMVC 先到spring的官网下载spring的jar包，本系统使用的spring把嫩版本是spring-4.3.4.RELEASE，将使用到的jar包拷贝到上面新建的项目的WEB-INFO下的lib目下。

然后配置springMVC，打开web.xml文件，配置spring的处理分发器，配置如下，只需在web.xml中加入如下配置：

<servlet>

<servlet-name> dispatcherServlet </servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name> contextConfigLocation </param-name>

<param-value> classpath:dispatcher-servlet.xml </param-value>

</init-param>

<load-on-startup> 1 </load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name> dispatcherServlet </servlet-name>

<url-pattern> / </url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name> dispatcherServlet </servlet-name>

<url-pattern> /index </url-pattern>

</servlet-mapping>

以上配置主要配置，spring的接收处理器所有的请求先经过org.springframework.web.servlet.DispatcherServlet处理解析，在调用对应的Controller处理请求。

在开发项目的过程中还经常遇到的一个问题是乱码，在spring mvc开发中解决传值乱码，只需在web.xml中增加如下配置：

<filter>

<filter-name> setCharacterEncoding </filter-name>

<filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>

<init-param>

<param-name> encoding </param-name>

<param-value> UTF-8 </param-value>

</init-param>

<init-param>

<param-name> forceEncoding </param-name>

<param-value> true </param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name> setCharacterEncoding </filter-name>

<url-pattern> /\* </url-pattern>

</filter-mapping>

接下来是spring mvc的住配置文件，dispatcher-servlet.xml，在resources文件夹下创建dispatcher-servlet.xml。

配置启用spring注解和spring扫描的包路径：

<context:component-scan base-package="com.cate.\*" />

<!-- 启用mvc注解 -->

<mvc:annotation-driven />

<mvc:default-servlet-handler/>

<context:annotation-config />

配置视图解析器：

<bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<property name="order" value="1"/>

<property name="prefix" value="/WEB-INF/view/" />

<property name="suffix" value=".jsp" />

</bean>

同样的方法到Hibernate的官网下载HIbernate的jar包，将使用到的jar包复制到WEB-INF目录下，在resources下新建hibernate.cfg.xml并配置，配置详情如下：

配置数据库的用户名和密码、数据库连接驱动还有数据库连接地址及Hibernate与数据交换数据使用的方言：

<property name="connection.username">root</property>

<property name="connection.password">\*\*\*\*\*\*</property>

<property name="connection.driver\_class">com.mysql.jdbc.Driver</property>

<property name="connection.url">jdbc:mysql://localhost:3306/cate</property>

<property name="dialect">org.hibernate.dialect.MySQLDialect</property>

在Hibernate中，实体和数据库的映射用的是注解的方式，所以在hibernate.cfg.xml中还需加入对应实体类的mapping：如用户实体和数据库对应的配置需加入如下配置：

<mapping class="com.cate.entity.User"/>

6.2登录注册模块设计

登录注册主要使用ajax和后台进行数据交互，在提交钱获取用户的输入，并进行有效验证，js获取用户的输入，先去掉输入两端的字符串，再进行有效验证。

登录注册的基本原理相同，这里就用登录作为例子，详细说明登录的实现过程。

界面代码，只需要在前端界面代码放置两个输入框，类型分别是text 和 password，如：<input type="number" id="phone"> ，设置id属性为了在js中能通过id值来获取改标签，从而获取输入框的值value，如获取用户的输入的手机号码的js方法为：mui("#phone")[0].value，此处的mui为MUI框架提供的选择器，跟jquery是同一个原理，如果使用原生js方法获取用户输入的手机号的方法可以是：querySelector("#phone")[0].value ，或者使用：document.getElementById("phone").value 同样可以获取到用户输入的手机号。

获取用户输入的密码方式，跟获取手机号的方式形同，这里不在赘述；

验证用户输入的合法性，首先先判断用户输入项中是否存在为空，这里判断用户输入是否为空之前，先将获取到的用户输入数据去掉两端的空格，再进行判断是否等于空，去掉字符串两端的空格使用正则表达式：

str.replace(/(^\s\*)|(\s\*$)/g, "")

str为要处理的字符串。

下一步是验证用户输入的手机号的合法性，也是用到正则表达式，前面 5.2.1 用户登录与注册的详细设计 中已经介绍过验证手机号的方法，这里不在重复赘述验证过程。

所有验证合法性都通过后，创建ajax请求，在本系统中，使用MUI框架封装号的ajax方法，具体使用如下：

mui.ajax(urlUtil.getRequestUrl("login"),{

data: {

phone: phone,

password: password

},

type: "post",

dataType: "json",

success: function(data){

if(data.header.success){

loginAndRegister.setUserInfo(data.body);

}else{

util.toast(data.header.errorInfo);

}

},

error: function(xhr, type, errorThrown){

util.toast(type + "错误，注册错误，请稍后重试");

}

})

其中urlUtil.getRequestUrl("login") 方法是获取登录的接口地址，urlUtil是自己封装的一个工具类，主要包系统所用的接口地址，通过urlUtil的getRequestUrl(param)方法，即可放回接口地址,其中param是接口名，urlUtil的一些核心方法，如获取当前项目访问连接的协议名、主机号、和端口号的方法如下：

window.document.location.href.substring(0, window.document.location.href.indexOf(window.document.location.pathname))

如项目登录页访问地址为：http://127.0.0.1:8080/cate/login，则该方法放回的是：http://127.0.0.1:8080。

在urlUtil中把所有的接口地址放在一个object中，提供一个按接口名返回接口地址的方法，具体代码设计如下：

//接口地址

requestPath: {

//首页轮播

initSlid: "/index/getSlides",

//登录

login: "/user/dologin",

...

},

//根据操作名，返回需要请求的路径

getRequestUrl: function(opt){

var \_url = "";

for(var item in this.requestPath){

if(item == opt){

\_url = this.baseRequestPath + this.requestPath[item];

break;

}

}

return \_url;

}

这里的getRequestUrl()方法是根据请求的接口名，返回接口的请求地址，

requestPath 和 getRequest()方法输入urlUtil的内部变量和方法，在调用这些变量或者方法时，需要在urlUtil的作用域下使用，如获取登录接口的接口路径：

urlUtil.getRequestUrl("login");

后台新建一个类，使用@Controller注解，将改类声明为一个Controller层，使用@RequestMapping(value=)，后台登录核心集体代码设计如下：

@Controller

@RequestMapping(value="/dologin")

public void dologin(PrintWriter out, @PathParam("phone") String phone, @PathParam("password") String password){

response.setContentType("application/json; charset=utf-8");

out.print(userCenter.login(phone, password));

}，

后台接收到前端发过来的请求之后，调用service层userCenter的login方法进行登录的业务处理，在业务处理层只需要在service类加上@Service注解，即可把该类声明为业务处理层，登录的业务处理核心代码如下：

@Override

public JSONObject login(String phone, String password) {

map = new HashMap<String, Object>();

user = userDao.queryByPhone(phone);

if(user == null){

header.setSuccess(false);

header.setErrorInfo("用户不存在");

}else{

if( !password.equals( user.getPassword() ) ){

header.setSuccess(false);

header.setErrorInfo("密码错误");

}else{

header.setSuccess(true);

UserInfo userInfo = new UserInfo();

userInfo.setCash(user.getCash());

userInfo.setDefaultAddressId(user.getDefaultAddressId());

userInfo.setId(user.getId());

userInfo.setPhone(user.getPhone());

userInfo.setUsername(user.getUsername());

userInfo.setBalance(user.getBalance());

map.put("body", userInfo);

}

}

map.put("header", header);

return JSONObject.fromObject(map);

}

在login方法中，调用userDao的queryByPhone()方法，根据手机号查询用户信息进行比较。

处理好之后的返回结果放在一个Map集合中，使用JSONObject.fromObject()将Map转化为json对象返回，改Map中主要有两对键值对，header和body，header中主要放处理业务成功与否的标志，如果错误，则header中的errorInfo为处理失败的原因描述，body是需要放回的数据，如登录成功之后，会返回用户的信息到前端，用户的信息是一个User对象，将User对象使用map.put(key, value)将用户信息对象添加到Map集合中，放回的用户信息过滤掉用户密码，即在登录成功之后，不需要返回用户的登录密码。

后台返回成功之后，前端ajax的success回调方法会接收后端返回的数据，success回调方法的data参数为后端返回的数据，这里放回数据形式为json对象，

然后在seccess方法中先判断返回数据的header.success是否为true，如果header.success的值为false则表示登录失败，则用util.toast(data.header.errorInfo)向用户提示出错信息，如果为true，则表示登录成功，则调用util.setSessionStorage(item)方法将用户的id和用户的手机号存储在sessionStorage中，接着跳转到首页，具体实现存储用户信息到sessionStorage中的实现核心代码如下：

//登录成功后，设置用户信息

setUserInfo: function(data){

util.setSessionStorage("uid",data.id);

util.setSessionStorage("phone", data.phone);

window.location.href = "index";

}

上面就是整个登录功能设计编码的实现过程，由于代码量比较大，所以在叙述过陈中，只放了关键的代码。

6.3其他模块设计

在本系统中其他模块的实现原理跟登录注册的实现原理基本一致，大同小异，基本原理都是使用html+css实现界面的展示，用js作为前端的业务处理工具，后台使用java提供数据访问接口，数据接口在@Controller注解的类的方法上使用@RequestMapping注解配置接口路径，当前端请求接口时，后台会自动比较请求路径和@RequestMapping定义的路径，调用对应的处理方法响应前端的请求。

前端js主要是利用ajax与后台交互数据，请求后台数据的过程是异步的，前端将同一个功能模块的代码或者结构采用模块化的开发方式，减少各个模块的耦合性。

7.系统测试

7.1 系统功能性及性能测试

本次功能性测试的方法主要是通过黑盒测试件测试方法，主要测试系统的功能设计和系统结构设计。测试案例是按照系统的应具有的功能、并按照测试规范、规格和要求进行设计的，测试者通过输入有效或者无效的输入，测试系统的响应。

测试用例如表7.1所示:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 测试点 | 操作步骤 | 期望结果 | 实际结果 | 测试结果 |
| 1 | 注册信息填写为空时 | 1.打开注册界面  2.直接点击注册按钮(REGISTER) | 提示输入相关项 | toast提示请输入手机号 | pass |
| 2 | 使用不合法手机号进行注册 | 1. 打开注册页面   2.在手机号输入框中输入手机号，如12345678901，并随便填写注册密码  3.点击注册按钮(REGISTER) | 错误提示：手机号不合法 | toast提示：”请输入正确的手机号” | pass |
| 3 | 使用注册过的手机号进行注册 | 1.打开注册界面  2.使用已经注册过的手机号进行注册 | 错误提示该手机号已经注册 | toast提示”该手机号已注册 | pass |
| 4 | 登录时，输入为空 | 1.打开登录页面  2.直接点击登录按钮（LOGIN） | 提示必须输入手机号 | toast提示”请输入手机号” | pass |
| 5 | 使用错误密码登录 | 1.打开登录界面  2.输入正确的手机号  3.输入一个随机错误的密码 | 提示密码错误 | toast提示手机号或者密码错误 | pass |
| 6 | 首页分类类表 | 1.登录  2.进入首页  3.点击首页的分类选项卡 | 看是食物分类类表是否正确获取 | 点击类别菜单（美食，食材，食谱，其他），页面列表显示不同内容，并且跟所点击的分类相同 | pass |
| 7 | 验证下单功能 | 1.打开首页  2.点击美食菜单  3.点击美食列表查看美食详情  4.点击加入购物车  5点击购买数量  6点击立即支付 | 点击支付之后，能正确的生成订单信息，总价格的计算会自动根据购买的数量计算 | 当点击加入购物车是，购买的数量自动变为1，当点击购买数量的调整按钮是，总价格随着购买的数量在实时改变 | pass |
| 8 | 不登陆的情况是否能下单 | 1.进入系统  2.点击个人中心 菜单按钮，跳转到个人中心  3.查看自己的登录状态，如果为已登录，则点击退出按钮，退出登录  4.点击首页按钮，操作上面的编号7的操作步骤 | 如果没有登录，怎不能下单购买 | 当编号7操作步骤的加入购物车时，系统自动跳转到登录页 | pass |
| 9 | 验证支付行为 | 1.进入系统  2.登录  3.进入首页，点击美食  4.点击美食列表查看详情，依次点击加入购物车、立即支付、提交订单 | 拥有支付功能 | 点击提交订单，弹出待付款信息和支付按钮，点击支付，直接跳转到个人中心 | 由于没有实质的认证，本系统中的支付方试是模拟的 |
| 10 | 测试系统功能的完善性 | 1.进入系统  2，点击首页，界面的按钮各点击一遍。  3.进入推荐页，所有的推荐信息随意点击一遍  4.点击社区按钮，所有话题随意点击一遍  5.点击进入个人中心页，所有的选项各点击一遍 | 各个功能有很好的串接 | 所有按钮都能正常的跳转，但是公能比较单一，不够丰富，购买美食时，不能同时购买多家商品，社区评论是只能统一留言回复，不能@的形式回复 | 本系统在开发行确实存在许多问题，还需要进一步的完善，功能也只是基本简单的实现，功能太过于单一 |

7.2 系统评价

对系统进行测试之后，系统确实存在很多问题，还需要进一步完善，当用户是首次进入系统时，后台会先去缓存数据导致整个加载过程比较慢，而系统大部分的数据加载都是通过ajax异步加载的，当用户的网络不是很好的时候，看到的信息是静态模版信息，等于什么内容都没有，但是后台的首次加载缓存一般需要的时间很长，这是给用户的体验并不是很好，用户会可能认为系统出了很大的BUG，半天没有响应。

后期进过系统的进一步优化，和进一步的完善功能，基本上把测试中的问题解决了，已能够满足基本的业务需求，并达到了预期的功能目标。

## 7.3 系统存在的问题

系统依然有一些问题以目前在时间进度、知识储备等因素下难以完成。举例如下：

体统功能做得不够细，比如购买美食，当购买美食提交订单时，在选择收货地址时，原本的想法是直接获取当前手机的位置，或者直接在地图上点选收货地址，但是当接入地图之后，需要手机的定位功能权限，如果用户在一开始进入系统之前就禁止使用手机定位，那么在选择收货地址页面就会报错，因为所有的页面都在同一个一面调用的模块，相当于同一个页面，当有地方报错的时候，会影响到其他功能的使用，所以只能在增加收货地址的时候，采用手动输入的方式，但这样降低了用户体验。

还有购买美食的时候，不能同时购买多家的美食，有的用户如果想同时购买多家的美食，只能一家一家的选择购买，完成支付之后，才能去购买下一下的美食。

社区的评论留言只是简单的留言功能，没有点对点针对某个人的回复，这样的留言回复没有亮点，功能太过于简单，因为在实现这个功能的时候，自己确实缺乏设计经验，没有设计出一条很好的解决方案，包括数据库的设计，前端架构和后台架构的实现方式。

总结

从开题报告到项目开发完毕，总的花费时间将近两个月，经过这两个月的努力，项目的开发终于圆满完成，在开发中遇到了很多问题，总是在不断摸索尝试，总结经验，积累知识，当我遇到问题时不断通过百度，google，还有两大社区csdn和stockOverflower去查找问题和解决问题，在这个项目过程中我学到了很多，这也是我第一次尝试使用新框架Hibernate，将Hibernate和SpringMVC整合作为后端开发框架，spring的Controller注册很方便的将一个类声明为控制层，取代了传统的servlet开发方式，省去了在web.xml中一个servlet类对应一个url的配置，大大提高了开发的方便性和开发效率，Hibernate将数据库自动映射到实体类，将操作数据编程了操作对象，开发时不用担心记不住数据库字段怎么定义的。

通过本次毕设项目的开发，通告了自己的综合应用能力，解决问题的能力。

# 致 谢

作者在设计（论文）期间都是在我的导师马垚老师全面、具体指导下完成进行的。马垚老师渊博的学识、敏锐的思维、民主而严谨的作风使学生受益非浅，并终生难忘。

感谢马垚老师等在毕业设计工作中给予的帮助。

感谢我的学友和朋友对我的关心和帮助。

# 参考文献

1. （美）麦克法兰著，俞黎敏译 .CSS实战手册（第2版）.电子工业出版社2010年
2. （美国）霍根（Brian P. Hogan） 著，吴珂 译. 写给程序员的Web设计书. 人民邮电出版社 .2011年08月
3. （美）威廉姆斯 著，李静 译.写给大家的Web和版式设计书. 机械工业出 版社. 2008年05月
4. 邵丽萍，张后扬，王英 著 .网页编程技术（第2版）.清华大学出版社 . 2007年09月
5. 李烨 著. 别具光芒CSS属性、浏览器兼容与网页布局. 人民邮电出版社 . 2008年10月
6. 耿祥义编著．JSP基础编程［M］．清华大学出版社，2004．55-162
7. 孙鑫编著．Java Web开发详解［M］．电子工业出版社，2006．189-274
8. 蔡剑，景楠.Java Web应用开发：J2EE和Tomcat[M].北京：清华大学出版社，2004.
9. 傅鹏，殷旻昊.基于Structs+Spring+Hibernate+Ajax技术的科研管理系统设计

[10] 陈正举.基于HIRBERNATE的数据库访问优化[J].计算机应用与软件,2012, 29(7):145-149.

附录一：外文文献

**The Analysis of Automated HTML5**

**Offline Services (AHOS)**

The traditional web-based applications operate only when connected to the network. Many realities in a field require a web-based application that is applicable even in case of offline. We have proposed the Automated HTML5 Offline Service (AHOS), presented as the integration of advanced services, developing with HTML5 Application Programming Interfaces (APIs) to provide web-based applications with the ability to work offline. When the AHOS web-based application visits at the web server for the first time, the web server will notify the application the list of files required to be downloaded. Then after being downloaded, the web application can work successfully and continuously even though the network connection from the client to the server is unavailable. Moreover, if the connection to the server is reconnected, any changes that have been made during offline will be automatically uploaded. The present study describes the requirements and implementation stages of AHOS concept for web-based applications. Several current status and challenges of AHOS concept are also explained. The performance analyses of the AHOS concept are performed in a case of web-based maintenance Decision Support System (DSS) for Small and Medium Industries (SMIs). The results of the study are very useful in providing in-depth understanding of the advantages and limitations, and as the future directions in applying this AHOS concept to other web-based applications.

Currently, the use of web-based application spreads over the various fields. The web-based application is cross-platform.In other words, we can use web-based application with popular web browser. Most of the web browser that runs web applications can temporarily store (caching) the data of web documents, e.g. HTML pages and image files . This cache process aims to conserve the bandwidth, reduce the server load, or reduce delay between actions of users and the reactions of

the server (web lagging) .

A script program on client side e.g. JavaScript is increasingly being used today too. This type program is an important part of the Dynamic HTML (DHTML) concept,which allows the content changes based on user input,environmental conditions (e.g. time), or other variable with or without a request to the server. Ajax (Asynchronous JavaScript and XML) is an important addition to the JavaScript language that allows the web to communicate with the server in the background with or without a new request or rendering process . But the responsive web development has many constraints when dealing with low internet bandwidth or intermittent network connectivity.

The HTML5 is the most recent technological development that provides solutions for storing web pages, data, or resources on the client side . This technology can reduce the network load by reducing the client request to the web server so the web application performance can be improved . The HTML5 allows the client to access the application with poor networkconditions or even without any network connection.

Most of the traditional web-based applications require connectivity to the internet to be able to work. Those web-based applications are provided by carrying out a request to the server to obtain any resources or information . In this situation it becomes a considerable problem in the case of high latency or connectivity problems, so the user will be able to keep waiting for a response of the request, and get an error or unwanted results.

Our previous studies have proposed the automated HTML5 offline service (AHOS). The AHOS represent as the integrations of advanced services, developing with HTML5 Application Programming Interfaces (APIs) to provide web-based application with the ability to work offline. The AHOS concept has been successfully implemented on a case study of web-based maintenance Decision Support System (DSS) for Small and Medium Industries (SMIs).

The aim of the present study is to examine the AHOS concept from several aspects, i.e. requirements, implementation stages, and several current status and challenges, include the performance analysis of AHOS concept in a case of web-based maintenance DSS for SMIs.

附录二：中文翻译

**自动HTML5离线服务（AHOS）的分析**

传统的基于Web的应用程序只有在连接到网络时才能运行。一个领域的许多现实需要一个基于网络的应用程序，即使在脱机的情况下也是如此。我们提出了自动化HTML5离线服务（AHOS），作为高级服务的集成，开发与HTML5应用程序编程接口（API），以提供基于Web的应用程序脱机工作。当首次在Web服务器上访问AHOS Web应用程序时，Web服务器将通知应用程序需要下载的文件列表。然后，下载完成后，即使从客户端到服务器的网络连接不可用，Web应用程序也能正常工作。此外，如果与服务器的连接重新连接，则在离线过程中进行的任何更改都将自动上传。本研究描述了基于Web的应用程序的AHOS概念的要求和实施阶段。还解释了AHOS概念的几个现状和挑战。在基于Web的维护中小企业决策支持系统（DSS）的情况下，执行AHOS概念的性能分析。该研究的结果对于提供对优势和局限性的深入了解，以及将AHOS概念应用于其他基于Web的应用程序的未来发展方向非常有用。

目前，基于Web的应用程序的使用遍布全球各个领域。 基于Web的应用程序是跨平台的。换句话说，我们可以使用流行的基于Web的应用程序网页浏览器。 大多数运行Web的Web浏览器应用程序可以临时存储（缓存）Web的数据文件，例如 HTML页面和图像文件。 这个缓存流程旨在节省带宽，减少服务器负载，或减少用户行为与其反应之间的延迟服务器（web滞后）。

客户端的脚本程序，例如 JavaScript是今天越来越多地被使用。 这种类型的程序是一个动态HTML（DHTML）概念的重要组成部分，这允许基于用户输入的内容改变，环境条件（例如时间）或具有或没有请求到服务器。 Ajax（异步JavaScript和XML）是JavaScript的重要补充允许网络与服务器进行通信的语言具有或不具有新请求或呈现的背景过程。 但响应式Web开发有很多处理低互联网带宽时的约束间歇性网络连接。

HTML5是最新的技术开发，为客户端提供存储网页，数据或资源的解决方案。 该技术可以通过减少对Web服务器的客户端请求来减少网络负载，从而可以提高Web应用程序的性能。 HTML5允许客户端访问网络条件差的应用程序，甚至没有任何网络连接。

大多数传统的基于Web的应用程序需要连接到互联网才能工作。 通过向服务器执行请求以获得任何资源或信息来提供那些基于web的应用程序。 在这种情况下，在高延迟或连接性问题的情况下，成为一个相当大的问题，因此用户将能够继续等待请求的响应，并获得错误或不需要的结果。

我们以前的研究提出了自动化HTML5离线服务（AHOS）。 AHOS代表高级服务的集成，与HTML5应用程序编程接口（API）开发，为基于Web的应用程序提供脱机工作能力。 AHOS概念已经在基于Web的维护中小型行业决策支持系统（DSS）的案例研究中成功实施。

本研究的目的是从几个方面来考察AHOS概念，即要求，实施阶段以及若干现状和挑战，包括对基于Web的维护DSS SMI的AHOS概念的性能分析。