**分类号：TP311.1 U D C：D10621-408-(2017) 3137-0**

**密 级：公 开 编 号：2013124055**

**成都信息工程大学**

**学位论文**

**基于springboot框架的网上商场购物平台的设计与实现**

|  |  |
| --- | --- |
| **论文作者姓名：** | **蒲俊鹏** |
| **申请学位专业：** | **网络工程** |
| **申请学位类别：** | **工学学士** |
| **指导教师姓名（职称）：** |  |
| **论文提交日期：** | **2021年06月09日** |

**基于springboot框架的网上商场购物平台的设计与实现**

**摘要：**随着社会经济的发展，手机不再是简单的通信工具，智能手机开始普及，变成了许多用户形影不离的生活必需品，用户对手机依赖的同时也对手机应用程序提出了更高的要求。由于交通网络的不断变更，人们产生了地图指引方向和路线的需求。无论人们开车还是步行，去一个陌生的地方，都会选择使用导航APP作为导向指引。

行车辅助APP是基于Android平台开发的应用，使用百度地图的开源API实现地图展开、地点搜索与定位等功能，使用 XML 语言进行基本界面View 的绘制，通过 Java将 View 和 Activity、Fragment连接起来实现页面逻辑，采用第三方BMOB云服务器来存储数据。最终实现了路线导航，推荐车辆服务，查看历史信息，分享等功能。

**关键词：**安卓；出行；地图；记录；服务

**Driving Assistance Android APP**

**Abstract**

With the development of social economy, China mobile phone is no longer a simple communication tools. The intelligent mobile phone began to spread, and many become users’ inseparable necessities of life. Users of mobile phone depend on mobile phone application, at the same time, put forward higher requirements on it. For the traffic network continuing to change, people need direction and route direct. Whether people driving or walking, when they go to a strange place, they will choose to use APP as navigation.

Driving Assistance is an application which based on Android platform, using Baidu maps API open source to location search and positioning. And using XML language to make the basic interface of View, Activity, and fragment connected to the page logic by Java. And using third party BMOB cloud server to store data. Finally, I realize the route navigation, vehicle service, view historical information, sharing and other functions.

**Key words:** android; travel; map; record; service

**目录**

论文总页数：30页

[1 引言 1](#_Toc485032439)

[1.1 课题背景 1](#_Toc485032440)

[1.2 国内外研究现状 1](#_Toc485032441)

[1.3 本课题研究方法和工作小结 1](#_Toc485032442)

[2 开发环境及相关技术工具介绍 2](#_Toc485032443)

[2.1 开发环境 2](#_Toc485032444)

[2.2 相关技术 2](#_Toc485032445)

[2.2.1 Java 技术 2](#_Toc485032446)

[2.2.2 Android 技术 3](#_Toc485032447)

[2.3 开发工具 3](#_Toc485032448)

[2.3.1 JDK 3](#_Toc485032449)

[2.3.2 Android SDK 3](#_Toc485032450)

[2.3.3 百度地图API 3](#_Toc485032451)

[2.3.4 Bmob云端服务器 4](#_Toc485032452)

[2.3.5 OkHttp 4](#_Toc485032453)

[2.3.6 SharedSDK 4](#_Toc485032454)

[3 系统需求分析和设计 4](#_Toc485032455)

[3.1 需求分析 4](#_Toc485032456)

[3.2 系统功能模块设计 5](#_Toc485032457)

[3.2.1 引导模块 5](#_Toc485032458)

[3.2.2 服务模块 5](#_Toc485032459)

[3.2.3 个人信息模块 5](#_Toc485032460)

[3.2.4 其他功能模块 6](#_Toc485032461)

[3.3系统架构与流程设计 6](#_Toc485032462)

[3.3.1 引导模块流程 6](#_Toc485032463)

[3.3.2 服务模块流程 7](#_Toc485032464)

[3.3.3 个人信息模块流程 8](#_Toc485032465)

[3.3.4 其他功能模块流程 9](#_Toc485032466)

[3.4 数据库设计 9](#_Toc485032467)

[4 系统的实现 10](#_Toc485032468)

[4.1 引导模块的实现 10](#_Toc485032469)

[4.1.1欢迎页子模块 10](#_Toc485032470)

[4.1.2 登录子模块 11](#_Toc485032471)

[4.1.3 主页搜索子模块 12](#_Toc485032472)

[4.2 服务模块的实现 13](#_Toc485032473)

[4.3 个人信息模块的实现 14](#_Toc485032474)

[4.4 其他功能模块的实现 16](#_Toc485032475)

[4.4.1 历史信息子模块 16](#_Toc485032476)

[4.4.2 分享模块子模块 17](#_Toc485032477)

[4.4.3 意见反馈子模块 17](#_Toc485032478)

[4.4.4 清理缓存子模块 18](#_Toc485032479)

[5 系统展示 19](#_Toc485032480)

[5.1 欢迎界面 19](#_Toc485032481)

[5.2 登录界面 19](#_Toc485032482)

[5.3 主页搜索界面 20](#_Toc485032483)

[5.4 服务界面 21](#_Toc485032484)

[5.5 个人信息界面 23](#_Toc485032485)

[5.6 历史信息展示界面 24](#_Toc485032486)

[5.7 分享界面 24](#_Toc485032487)

[5.8 意见反馈界面 25](#_Toc485032488)

[5.9 清理缓存界面 26](#_Toc485032489)

[结 语 27](#_Toc485032490)

[参考文献 28](#_Toc485032491)

[致 谢 29](#_Toc485032492)

[声 明 30](#_Toc485032493)

# 1 引言

## 1.1 课题背景

当代网络技术高速发展，网络覆盖越来越宽阔，大众人群消费水平普遍提升，快递行业发展蓬勃，人们更倾向足不出户地在网上购买自己所需要各种各样的商品。其中，网上购物服务流程完善，，消费者能方便快捷地达到购物目的，同时网上购物的法律规章十分健全，这样购物者的消费能得到完善的法律保护。另外，网上售货模式也为大量人群提供了自主创业的机会和就业岗位，于是一个完整全面的购物平台由消费者和卖家需求油然而生。

## 1.2 国内外研究现状

在国内外有很多优秀的网上购物商城。

亚马逊公司（Amazon）是美国的一家覆盖全球的电子商务公司。在最开始，亚马逊仅仅是一个网上书店，但是很快它跟随时代潮流开始全方面多领域发展，其包含业务覆盖：食品、书籍、DVD销售、视频、软件、电视游戏、化妆品、服装、家具、玩具、保健品、珠宝、电子产品等。

沃尔玛百货公司（Walmart）是一家全世界连锁店和存储式商店的跨国销售公司。该公司大规模涉足[零售业](https://baike.baidu.com/item/%E9%9B%B6%E5%94%AE%E4%B8%9A/3173194)，同时还是是世界上雇佣员工最多的公司。其经济规模巨大，有先进的物流模块，甚至购买卫星以构成自主的物流系统。除此之外，沃尔玛百货有限公司采购成本低，销售业务广泛，管理标准化。

淘宝网是亚太地区深受大众人群喜爱的网络零售，该网站由阿里巴巴公司成立于2003年5月。淘宝网是中国深受人群欢迎的网购零售平台，现如今它拥有近5亿之多的注册用户，而且每天至少6000万的固定访客。

## 1.3 本课题研究的意义

很多消费者和商店管理者都会购物平台反馈用户体验，但这些知名网站在多个版本更新中都未完善。要想在众多优秀的网上购物商场中脱颖而出，本系统必须弥补这些网站现未注意到的微小瑕疵细节。

本系统不仅能满足消费者的基本购物需要，还能在全方面考虑消费者感受、让消费者得到更好的购物体验。同时开发者也能根据网上商店管理者提出的管理需求，为管理者提供具有完整、可靠的管理功能的网上购物管理平台。除此之外，对于开发者而言，本系统开发工作量小，但所含功能完善，足以满足小众人群使用。

## 1.4 本课题研究的方法

首先去各大网上购物网站调研，参考网站门户界面框架、功能，体验多次完整的购物流程，模拟多种购物场景，构想设计本系统的独特之处。然后向网购人群发起问卷调查，根据问卷报告进行需求分析，搭建数据库，引入测试数据。下一步选择开发工具和搭建开发环境，开始本系统的开发。在开发过程中，主要分布式开发门户模块(ShopOnline-portal)和商店管理模块(ShopOnline-management)两个主要模块，同时搭建公用代码模块(ShopOnline-common)、利用mybatis逆向工程根据数据库信息自动生成实体类和mapper文件模块(ShopOnline-generator)。随后进行完整的开发，在这一过程中，不断测试代码的正确性，不断完善代码的健壮性。最后完成系统开发，完成论文编撰说明本系统。

# 2 开发环境及相关技术工具介绍

## 2.1 开发环境

Android Studio 是一个Android集成开发工具，基于IntelliJ IDEA. 类似 Eclipse ADT，Android Studio 提供了集成的 Android 开发工具用于开发和调试。

Android Studio主要由IntelliJ IDEA、Gradle、ProGuard三个部分组成，其中Gradle和ProGuard都是Android Studio基于IDEA提供 。

本系统使用Android Studio进行编写。

## 2.2 相关技术

### 2.2.1 SpringBoot

Java语言主要有以下特点:简单指令集、面向对象编程、分布式管理、鲁棒特性、可移植。

(1).面向对象：Java不同于C语言的面向过程，它是把所有的事物都当作一个对象，然后以这个对象来进行一系列操作，负责各自的任务，对象是一种抽象的概念，封装了它自身所具有的性质特征。

(2).分布性：Java是面向网络的语言，开发人员可以通过JDK里封装的HttpConnection和HttpClient等类的方法，便捷地对 TCP/IP协议进行处理。

(3).简单性：由于Java是面向对象的语言，并且提供一些基本的方法来完成对应的简单明了动作指令，因此开发人员可以较为轻松地编写适用于各种情况的程序。

(4).鲁棒性：在软件编译和运行代码的时候，Java会优先进行错误排查，以消除不必要的错误，它还有可以对内存进行优先级管理的垃圾回收机制，便于进行内存管理。

(5).可移植性：Java可以存在于各种设备，也可以在OS、Windows、Linux等操作系统中任意移植，因为Java可以实现与不同平台的接口，具有平台无关性。

### 2.2.2 Mysql

Android系统是目前世界市场最受欢迎的移动操作系统， Android可分为四层架构。

(1). Linux内核层：内核层为Android设备硬件提供了底层驱动，如显示驱动、音频驱动、照相机驱动、蓝牙驱动等。

(2). 系统运行库层：如 SQLite库提供了数据库的支持，Webkit库提供了浏览器内核的支持等。

(3). 应用框架层：构建应用程序时可能用到多种 API，Android 自带的一些核心应用就是使用这些API完成的，开发者也可以通过使用这些API来构建自己的应用程序。

(4). 应用层：所有安装在手机上的应用程序都属于这一层，比如系统自带的程序，以及自己写的程序。

## 2.3 开发工具

### 2.3.1 JDK

JDK是一个程序开发环境。它由一个处于操作系统层之上的运行环境还有开发者编译，调试和运行用Java语言写的applet和应用程序所需的工具组成。

JDK中还包括完整的JRE，也被称为private runtime。包括了用于产品环境的各种库类，以及给开发员使用的补充库，如IDL库。

JDK由一个标准类库和一组建立，测试及建立文档的Java实用程序组成。

### 2.3.2 Maven

Android SDK是谷歌提供的 Android开发工具包，这一工具包里有相关的 API用来开发 Android程序。

### 2.3.3 Redis

百度地图API给用户免费提供地图应用接口，包括JavaScript API、Web服务API、Android SDK、iOS SDK、定位SDK、车联网API、LBS云等多种开发工具与服务，具有基本地图展现、搜索、定位、逆/地理编码、路线规划、LBS云存储与检索等功能。通过该接口，就能轻松的获取用户的地理位置、周围信息等数据，并将数据应用到所需要的APP中。

### 2.3.4 七牛云服务器

Bmob是一个第三方云服务器平台，具备可视化的云端数据表设计界面，方便快捷地建库建表。包括10种不同数据类型存储：如字符串，整型，数组等。

### 2.3.5 Mybatis

OKHTTP是一个针对Android和Java应用的Http和SPDY客户端。HTTP是现代网络应用的通用协议，它决定了我们如何交换数据或媒体。高效的HTTP通信可以提升数据加载速度并减小带宽。OKHttp是一个高效的HTTP客户端。

# 3 系统需求分析和设计

## 3.1 系统可行性分析

### 3.1.1 经济可行性分析

本文所述的网上购物商场仅需要以个人计算机作为运行平台，所需开发工具和存储服务器都可以使用免费版本，无高成本要求，而且利用浏览器便能访问测试本系统搭建的网站。因为网站的较低工作量，其运行和维护成本会变得很低。同时，商家的入住和广告商的投资会给本系统带来一些收益。综上所述，本系统完全符合经济可行性要求

### 3.1.2 技术可行性分析

本系统选用IntelliJ IDEA作为软件开发环境，该平台功能十分强大，整合多种开发工具，满足各种编程需求，完全足够本系统的开发。对于数据库方面，本系统选用mysql数据库来保存各种数据。在编程语言方面，本系统采用java语言开发，该编程语言是当前热门、十分成熟的高级编程语言，能保证编写的系统能稳定开发和运行。本系统运行所需的硬件要求极低，现市场上绝大主流配置的计算机都能轻易满足运行该系统的要求。综上所述，本系统完全满足技术可行性方面。

### 3.1.3 操作可行性分析

本系统同现各大网上购物商场有相似门户界面，且根据多数用户偏好构建页面框架，友好便利。对用商店管理界面设计，本系统引导语言简单易懂，商店管理员能立刻熟悉各个功能的使用方法，本系统能全方面满足管理员对商品等方面进行便利地管理。

## 3.2 功能需求

本系统网上购物商城主要功能有商店管理员对商品分类和商品进行增删改查，有消费者登录和商店管理者登录功能。消费者能浏览商城门户，根据商品分类或者搜索关键字查询自己所需的商品，并将选的商品加入购物车。门户首页的采用静态资源渲染，以便随时更新热门商品显示。而搜索界面的商品展示，则根据所搜索商品存储在数据库中的url地址查询对应在七牛云服务器存储图片进行动态渲染，以减少本地计算机存储资源量，增强本系统运行效能。同时商品分类信息和商品信息在第一次查询过后，会存储在redis缓存中，避免频繁访问同一资源，降低了对数据库的请求频率。

## 3.3 功能模块图

## 3.4 系统流程图

# 4 系统设计

## 4.1系统模块设计

### 4.1.1 公共类模块

公共类模块主要存放公共依赖文件、公共返回状态类(其中包括分页信息)、工具类。门户模块和商品管理模块依赖继承于公共模块，该两主模块仅需声明依赖名称，减少了依赖文件的冗余。同时两模块在调用方法后，都需要返回状态信息以标识方法是否执行成功。Cookie工具类用于存储用户登录信息，Json工具类用于对象与字符串之间的转化，Md5工具类用于加密用户秘密保障用户信息安全性，UUID工具类用于生成用户登录的唯一标识的凭据。

### 4.1.2 商品管理模块

商品管理模块是商家拥有者对自家商品进行管理的模块。商家在该模块可以安全单点登录，同时商家处于未登录状态无法直接通过url地址访问管理界面。商家能够对商品分类信息和商品信息进行增删改查操作。也能根据商品分类、商品品牌、商品名关键字搜索指定商品信息，同时能调节每页显示数据条数，分页查询商品信息。最后管理者能够安全退出以保证账号的安全性。

### 4.1.3 门户模块

门户模块是满足消费者进行自主购物的模块。消费者在未登录状态下，可以任意浏览商品信息，搜索指定商品。若消费者要查看自我购物车信息或者将商品添加到购物车中，需要登录个人信息才能执行成功。消费者的登录也会处于安全的单点登录状态，最后消费者也能安全退出以保证账号的安全性。

### 4.1.4 逆向工程模块

该模块通过配置mybatis-generator.xml文件来根据数据库表名和表中主键名，逆向生成对于实体类和mapper文件。

## 4.2 数据库设计

4.2.1 数据库实体图

4.2.2 数据表设计

数据库表采用Bmob云数据库，根据需求创建四张数据表，数据表中字段及字段类型，描述如下所示。

表1 历史记录（JiLu）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 备注 | 是否必须 |
| ID | Number | 记录ID | 为历史记录表的主键 |  |
| Start | String | 起始地址 |  |  |
| End | String | 结束地址 |  |  |
| Time | BmobDate | 搜索时间 |  |  |

表2 汽车服务店铺表（Weixiu、Meirong、Jiayou）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 备注 | 是否必须 |
| ID | Number | 店铺ID | 为店铺表的主键 | 是 |
| Pic File | BmobFile | 店铺图片 |  |  |
| Name | String | 店铺名称 |  |  |
| Addr | String | 店铺地址 |  |  |
| Price | Number | 店铺服务价格 |  |  |
| Distance | String | 店铺距离 |  |  |
| Num | Number | 购买人数 |  |  |

# 5 系统的实现

## 5.1 商品管理模块的实现

### 5.1.1管理员登录子模块

通过js、css和image静态资源文件对项目resource包下index.ftl渲染出一个管理员登录界面，其界面主要有登录账号、密码和验证码输入框。然后系统将用户输入的账号、密码和验证码发送至UserController调用方法验证信息的正确性。

*//ticket  
String ticket = ssoService.login(admin);  
// cookie  
if(!StringUtil.isEmpty(ticket)){  
 System.out.println("username:"+admin.getUserName());  
 //save in session  
 request.getSession().setAttribute("user",admin);  
 return cookieService.setCookie(request,response,ticket)?BaseResult.success():BaseResult.error();  
}  
System.out.println("admin is null ");  
return BaseResult.error();*

其中ssoService服务的login()方法实现验证信息正确性和实现单点登录。ssoService首先验证账号是否存在。若存在，则用该账号在数据库中存放的md5加密盐加密登录密码，再对比密码是否匹配。若全部匹配，则用UUIDUtil工具类生成一个随机的ticket并返回作为该用户的登录唯一票据，同时将该存入redis和cookie中以备单点登录使用。

*//login()*

*if(!admin1.getPassword().equals(Md5Util.getMd5WithSalt(admin.getPassword(),admin1.getEcSalt()))) {  
 System.out.println("pwd is invalid");  
 return null;  
}  
ValueOperations<String,String> valueOperations = redisTemplate.opsForValue();  
String ticket = UUIDUtil.getUUID();  
System.out.println("ticket:"+ticket);  
valueOperations.set(userTicket+":"+ticket,JsonUtil.object2JsonStr(admin1),30, TimeUnit.MINUTES);*

当用户处于未登录状态，直接通过url地址访问管理界面，会被设置的拦截器拦截至登录界面。该功能在intercepter包下的ManagerLoginIntercepter.java类实现，同时在config包下的MvcConfig.java类进行添加配置该拦截器，同时分离出静态资源设置允许该类通过。当用户已经登录时，通过cookie中存储信息得到票据，并通过validdate()方法验证票据的正确性取得用户对象，再将对象存入session以备输出个人信息使用。

*//config intercepter:*

*String ticket = CookieUtil.getCookieValue(request, "userTicket");  
System.out.println("ticket:"+ticket);  
if(!StringUtil.isEmpty(ticket)){  
 Admin admin = ssoService.validate(ticket);  
 request.getSession().setAttribute("user",admin);  
 ValueOperations<String, String> valueOperations = redisTemplate.opsForValue();  
 valueOperations.set(userTicket+":"+ticket, JsonUtil.object2JsonStr(admin),30, TimeUnit.MINUTES);  
 return true;  
}  
System.out.println("turn to login");  
response.sendRedirect(request.getContextPath()+"/login");*

当用户访问其他网页或者关闭浏览器后，在缓存生效时间30分钟时间内，再次访问管理主页，都无需登录验证，同时缓存时间重新刷新。

若用户点击主页的安全退出，则调用logout()方法删除redis中的缓存、调用deleteCookie()方法删除cookie中缓存的票据信息、调用remove()方法删除session中的用户对象，则用户再次访问管理界面需要重新验证登录信息。

*//logout safely:*

*ssoService.logout(ticket);  
request.getSession().removeAttribute("user");  
cookieService.deleteCookie(request,response);*

这就是本系统实现的完整的管理员单点登录功能，极大的减除了用户多次访问管理界面的需要重新验证信息的麻烦问题，同时安全退出则保障了用户信息安全。

### 5.1.2 管理商品分类子模块

点击进入商品模块的管理商品分类界面，需要渲染商品分类页面。首先通过地址映射调用GoodsCategoryController中的的categoryList()方法获得所有的分类信息。其中调用goodsCategoryService中的selectCategoryListForView()方法先查询redis缓存中是否已存在缓存的分类信息，若有则直接返回分类信息。若无，则先查询所有的顶级分类，用list接收查询返回对象，然后将顶级分类对象循环转化为vo对象，其中vo对象增加的List<GoodsCategoryVo> children字段以放置子分类信息。再根据父分类id循环查询各自的子分类(二级分类)信息并存放至父分类的children字段中。之后依次类推查询三级分类并存放。最后将顶级分类信息放入redis缓存中，返回list。

*//looking for buffer:*

*String gcvListJson = valueOperations.get(goodsListKey);  
if(!StringUtil.isEmpty(gcvListJson)){  
 return JsonUtil.jsonToList(gcvListJson,GoodsCategoryVo.class);  
}*

*//save in redis*

*valueOperations.set(goodsListKey,JsonUtil.object2JsonStr(gcvList));*

*GoodsCategoryController*

*//save in model*

*model.addAttribute("gcvList",goodsCategoryService.selectCategoryListForView());*

前端freemarker模板利用<#list></list>标签得到存放在model中list的数据信息，通过三层标签循环输出数据，渲染分类列表信息。

*//print the information*

*<#list gcvList as gcv1>*

*<td>${gcv1.id}</td>*

*<#list gcv1.children as gcv2>*

*<td>${gcv2.id}</td>*

*<#list gcv2.children as gcv3>*

*<td>${gcv3.id}</td>*

*</#list>  
 </#list>  
</#list>*

利用图片渲染分类信息是否推荐和是否显示的方法是：给每个对应标签加入唯一的id，再添加js方法在加载界面资源时执行循环渲染对应图片。

*//set id*

*<input type="hidden" id = "isHot${gcv1.id}" value="${gcv1.isHot}">*

*<input type="hidden" id = "isShow${gcv1.id}" value="${gcv1.isShow}">*

*//set images*

*if ("1" == $("#isHot"+i).val()) {  
 //$("#imgHot"+i).attr("src","${ctx}/static/images/yes.png");  
 document.getElementById("imgHot"+i).src = "${ctx}/static/images/yes.png";  
  
 console.log($("#imgHot").attr("src"));  
 } else  
 $("#imgHot"+i).attr("src", "${ctx}/static/images/cancel.png");  
 if ("1" == $("#isShow"+i).val()) {  
 document.getElementById("imgShow"+i).src ="${ctx}/static/images/yes.png";  
 }else  
 $("#imgShow"+i).attr("src","${ctx}/static/images/cancel.png");  
}*

在实现添加商品分类功能过程中，最主要的两个部分是多选框级联查询和图片上传服务。对于实现两个级联选择的多选框，首先需要在进入添加商品分类界面时，将顶级分类信息查询并循环添加到顶级分类多选框中，给每个<option>设置value字段值并添加onchange()方法。当进行选择顶级分类后，本系统去查询以该顶级分类为父分类的子分类，请求成功后，将返回数据进行拼接成html代码，根据下一级多选框id将拼接的<option>标签设置进入多选框中，同时将一隐藏输入框的值设置为该顶级分类的id作为父分类id、将另一隐藏输入框的值设置为2作为分类等级。选择二级分类时，只需更新两个隐藏输入框的值。

*//选择顶级分类返回成功时*

*if (result.length > 0) {  
 for (i = 0; i < result.length; i++) {  
 htmlStr += "<option value='" + result[i].id + "'>" + result[i].id + "、" + result[i].name + "</option>"  
 }  
 $("#" + next).html(htmlStr);  
 $("#level").val(2);  
} else {  
 $("#level").val(2);  
 layer.alert("无子分类");  
}*

*//选择二级分类后*

*// 修改parentId和level  
if (0==parentId){  
 $("#parentId").val($("#parent\_id\_1").val);  
 $("#level").val(2);  
 return;  
}  
// 修改parentId和level  
$("#parentId").val(parentId);  
$("#level").val(level);*

对于图片上传服务，本系统编单独编写了一个上传服务。首先需要在七牛云服务器注册申请一个储存对象，更新dns服务器地址以解析上传图片的的域名。当进行上传图片服务时，在本系统中利用传入的图片名和当下本地时间拼接成新的图片名，再根据与七牛云服务器的存储对象凭据对应，完成上传图片服务，最后使用上传成功的回调方法将图片的url设置在一个隐藏输入框中。

*//UploadController调用上传服务和修改图片名*

*String filename = file.getOriginalFilename();  
String date = DateTimeFormatter.ofPattern("yyyy-MM-dd").format(LocalDateTime.now());  
filename = date +" "+ System.currentTimeMillis() + " "+filename;  
//.substring(filename.lastIndexOf("."))  
return uploadService.upload(file.getInputStream(),filename);*

*//上传图片凭据*

*Configuration cfg = new Configuration(Region.region2());  
UploadManager uploadManager = new UploadManager(cfg);  
//...生成上传凭证，然后准备上传  
String accessKey = "z3SsvlYyolMz1e67lcv5PFRoT7QtEfO1LpkbjrEj";  
String secretKey = "3pqTW5qk3mRRIPnEyGQi4EhEAfaNiUy0OgKxq08i";  
String bucket = "shoponline-pjp";*

*//上传图片*

*fileResult.setSuccess("success");  
fileResult.setMessage("上传图片成功");  
fileResult.setFileUrl("http://app2.merborn.fun/"+fileName);*

*//上传成功回调设置*

*$("#image").val(data.response.fileUrl);*

最后点击提交按钮，调用serialize()方法对表单中各个输入框的值进行序列化，然后利用ajax请求将数据传入后台，后台根据传入数据与实体的字段名匹配封装成完整对象，再删除redis中分类信息缓存并将该对象存储到数据库中，完成添加商品分类操作。

*//前端发送ajax请求存储分类数据*

*$.ajax({  
 url: "${ctx}/goods/category/save",  
 type: "post",  
 data: $("#category\_form").serialize(),  
 dataType: "JSON",  
 success: function (result) {  
 if (result.code == 200) {  
 layer.confirm("保存成功", {btn: ['继续新增', '返回列表']},  
 function () {  
 window.location.href = "${ctx}/goods/category/add";  
 }, function () {  
 window.location.href = "${ctx}/goods/category/list";  
 });  
 } else {  
 layer.alert("保存失败");  
 }  
 },*

*//GoodsCategoryController调用方法存储新增分类*

*int resultCol = goodsCategoryService.categorySave(goodsCategory);  
return resultCol>0?BaseResult.success():BaseResult.error();*

*//删除Redis缓存，数据存入数据库  
redisTemplate.delete(redisTemplate.keys("goods\*"));  
return goodsCategoryMapper.insertSelective(goodsCategory);*

对商品分类删除功能，采用前端和后台同步删除的方式，保证了数据的一致性。同时前端异步删除标签，避免了删除后需要重新刷新页面的不便利操作。点击删除按钮，用调用del\_fun()方法，该方法调用一个递归方法returnResult()以便在删除父分类时，先删除子分类，到达根据外键关系级联删除数据的目的。

具体流程是：先将删除分类的id传入后台，后台查明该分类属于几级分类。若该分类为三级分类，后台则删除数据库中该分类信息，然后直接返回一个仅存一个对象且对象id为-1的列表，前端判断该分类无子分类，直接调用document.getElementById().remove()方法删除该分类所属的父标签。若该分类有子分类，则查询子分类信息并返回，前端循环递归查询每个子分类情况，直至从最底层分类开始往上删除。最后删除redis中缓存分类信息。

*//前端递归删除标签*

*if(result[0].id != -1 ){  
 for(var i = 0;i < result.length;i++){  
 //obj是当使用$(obj).parent().parent().remove();方法是使用的，现无需。  
 returnResult(null,del\_url,result[i].id);  
 console.log(result[i].id);  
 }  
}else  
 console.log("无子类数据");  
document.getElementById("gcvList"+id).remove();*

*//后台创建最底层分类标志信息*

*goodsCategory3.setId((short)-1);  
goodsCategories.add(goodsCategory3);*

*//后台查询预删除分类的信息  
GoodsCategory goodsCategory = goodsCategoryService.categorySelectOne(deleteId);*

*//后台删除分类信息*

*goodsCategoryService.categoryDelete(deleteId);  
//如果该分类为一级分类  
if(goodsCategory.getLevel() == 1){  
 //查询该分类的二级分类列表  
 List<GoodsCategory> list2 = goodsCategoryService.selectCategoryByParentId(deleteId);  
 if(list2.isEmpty())  
 return goodsCategories;  
 else  
 return list2;  
 //如果该分类为二级分类  
}else if(goodsCategory.getLevel() == 2) {  
 List<GoodsCategory> list3 = goodsCategoryService.selectCategoryByParentId(deleteId);  
 if(list3.isEmpty())  
 return goodsCategories;  
 else  
 return list3;  
 //如果该分类为三级分类  
}else{  
 return goodsCategories;  
}//删除Redis缓存  
redisTemplate.delete(redisTemplate.keys("goods\*"));  
return goodsCategoryMapper.deleteByPrimaryKey(id);*

修改商品分类信息分为两个部分完成。对于修改商品分类是否推荐和是否显示仅需点击显示状态图片就可以切换，修改分组和排序信息仅需在显示页面修改数值便可。但对于修改商品分类的具体信息，管理员需前往category-update.ftl页面才能修改。该页面查询到该分类的信息并填充到对应输入框中，管理员仅需修改输入框中数值，选择提交便可，后台将表单数据序列化并传入后台，后台进行更新数据操作达到修改分类信息目的。最后后台将存储在redis中的分类信息删除。

*//修改分类列表ishot，同修改分类是否显示  
function changeIshot(id,isHot,obj)  
{  
 var src = "";  
 if($(obj).attr('src').indexOf("cancel.png") > 0 )  
 {  
 src = '${ctx}/static/images/yes.png';  
 var hot = 1;  
 }else{  
 src = '${ctx}/static/images/cancel.png';  
 var hot = 0;  
 }  
 $.ajax({  
 url:"${ctx}/goods/category/changeIshot",  
 type: "POST",  
 data: {  
 id: id,  
 isHot: hot,  
 },  
 dataType: "JSON",  
 success: function(data){  
 $(obj).attr('src',src);  
 },  
 error:function (result) {  
 console.log(result);  
 }  
 });  
}*

*/// 修改分类列表的分组，同修改分类排序  
function updateCatGroup(id,obj)  
{  
 var value = $(obj).val();  
 console.log(value);  
 $.ajax({  
 url:"${ctx}/goods/category/changeCatGroup",  
 type: "POST",  
 data: {  
 id: id,  
 catGroup: value,  
 },  
 dataType: "JSON",  
 success: function(data){  
 layer.msg('更新成功', {icon: 1});  
 },  
 error :function (result) {  
 console.log("error");  
 console.log(result);  
 }  
 });*

*}*

### 5.1.3 管理商品子模块

该模块功能类同与商品分类模块，除此之外主要添加了三多选框级联查询、添加在线编辑器、商品列表显示分页、三个搜索条件查询商品信息等功能。

三多选框级联查询分类为新增商品时添加商品分类标签使用，功能实现类似于两多选框级联查询。添加在线编辑器的目的是管理者在添加新商品时，管理者能用多样的工具来具体描述商品的信息，这些商品详细信息描述将会转化html代码格式存储在数据库中。

//在线编辑器

<**textarea rows="6" cols="80" class="span12 ckeditor" id="goods\_content"  
 name="goodsContent" title=""**></**textarea**>

## 5.2 门面模块的实现

### 5.2.1 登录子模块

### 5.2.2 主页面渲染子模块

### 5.2.3 搜索子模块

### 5.2.4 添加购物车子模块

# 6 系统展示

## 6.1 后台管理模块展示

**6.1.1 登录界面展示**

访问后台管理平台的登录界面或者未登录状态下访问后台管理界面，会进入到该



图7 引导模块-欢迎页

**5.1.2 商品分类列表展示**

访问后台管理平台的登录界面或者未登录状态下访问后台管理界面，会进入到该

## 5.2 登录界面

点击欢迎页右上角的“跳过”按钮，或者等待6s，自动跳转到登录页面（用户第一次打开APP未登录状态下），用户只需要输入手机号码，点击“获取验证码”按钮，获取到验证码并填入，即可成功登录，如下图所示：

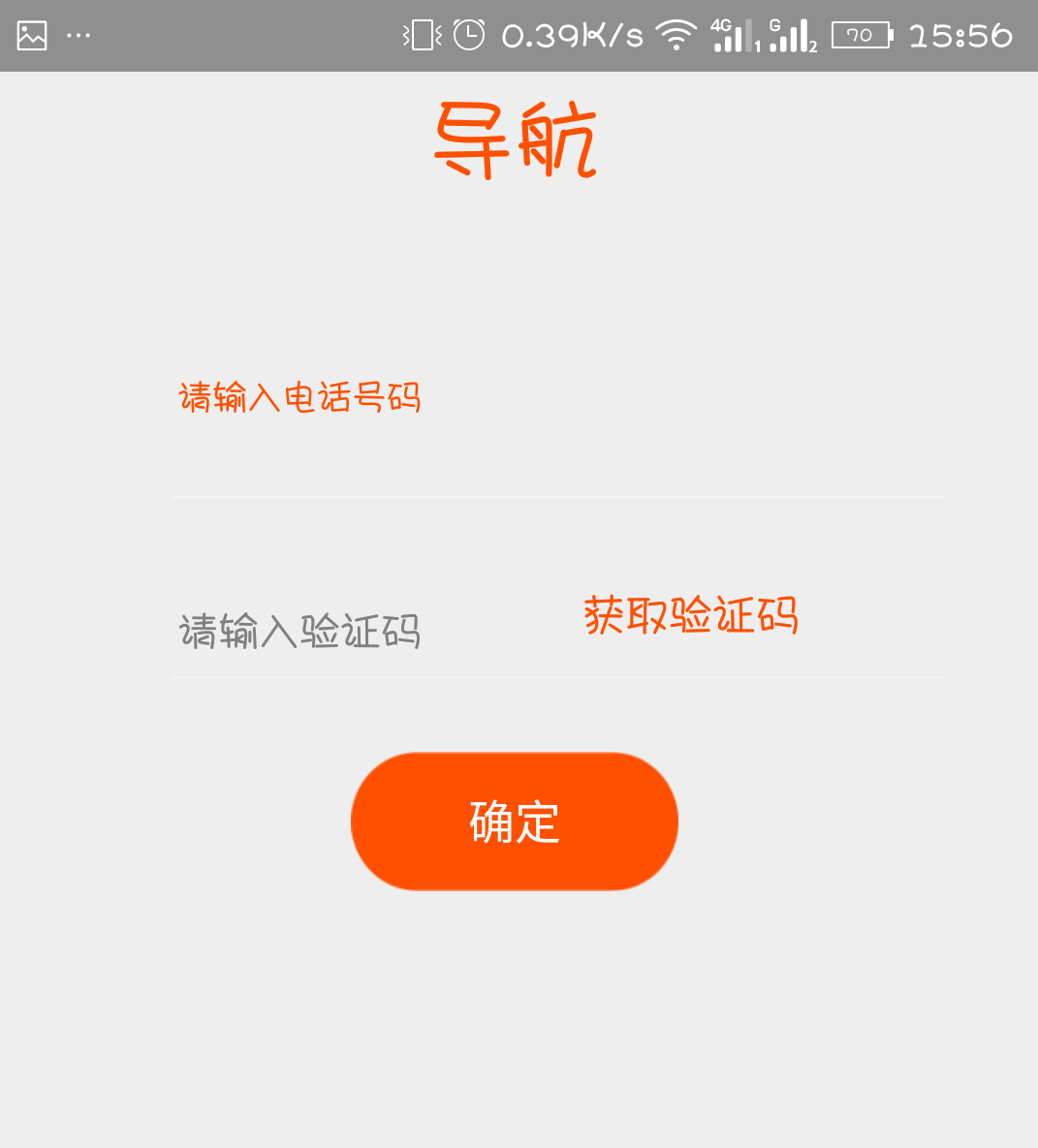


图8 引导模块-登录页

## 5.3 主页搜索界面

用户登录成功以后，就跳转到主页面，用户当前所在位置在打开APP时后台自动获取，所以无需输入。可以在顶部的搜索框输入目的地，如下图所示：

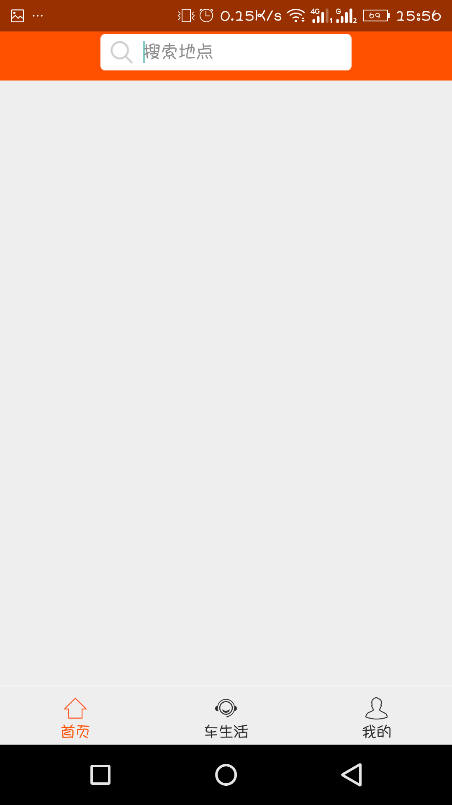


图9 引导模块-主页面

出现一系列带有输入的关键字的地点列表，点击列表某一条目，出现地图选择框，选择所用地图，开启调用百度地图，如下图所示：



图10 引导模块-搜索页面

进入地图页面后，蓝色箭头标记即为用户所在位置，红色图标表示用户要到达的终点，点击屏幕下方的“开始导航”就可进行导航，如下图所示：



图11 引导模块-地图导航

## 5.4 服务界面

点击底部tab的“车生活”按钮，或者横向滑动屏幕，就可进入车生活页面，车生活页面包括三部分：汽车美容，加油，维修保养。点击不同的图标会进入不同的页面，展示对应的店铺推荐列表，如下图所示：

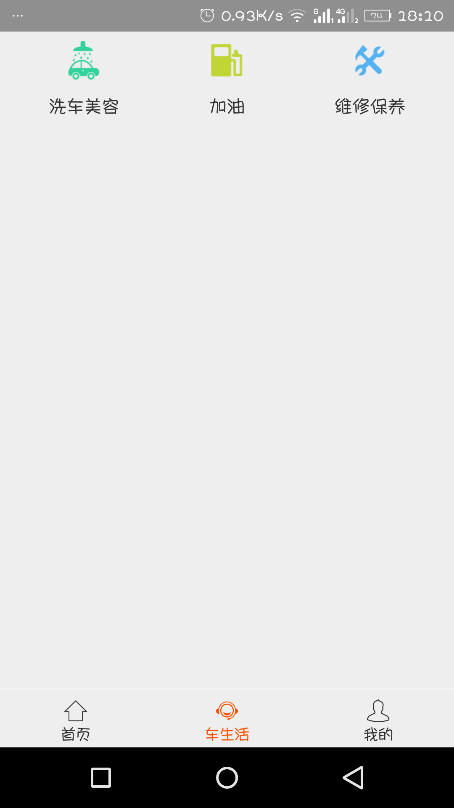


图12 服务模块-车生活页面

点击不同的图标会进入不同的页面，展示对应的店铺推荐，以列表形式给出，上面显示店铺的图片，名称，店铺地址，价格，购买人数以及距离等相关信息，如下列各图所示：



图13 服务模块-洗车美容



图14 服务模块-加油



图15 服务模块-维修

## 5.5 个人信息界面

横向滑动屏幕或者点击“我的”tab，进入个人页面，点击右上角编辑按钮，即可进入个人资料修改界面，分别有用户头像、姓名、性别、手机号、以及地址的信息。用户头像是登录以后系统自动随机选取图片进行显示，手机号是用户登录时输入的，不能修改。如下图所示：



图16 个人页面模块-个人资料编辑

## 5.6 历史信息展示界面

在“我的”页面，还提供了历史信息一栏，点击进入，即可展示用户每一次的历史搜索记录，记录按照时间先后顺序以列表形式进行排列，包括每次搜索的起始地，结束地以及搜索时间等信息，如下图所示：



图17 其他功能模块-历史信息

## 5.7 分享界面

用户如果想要对APP进行分享操作，可以点击“分享APP”一栏，在屏幕下方就会出现QQ空间、微信朋友圈、新浪微博等社交软件的标识，如下图所示：



图18 其他功能模块-分享APP

点击其中任何一个，就可以在对应的社交软件平台上进行转发分享，如下图所示：



图19 其他功能模块-转发界面

## 5.8 意见反馈界面

如果对APP有什么好的建议或意见的话，用户还可以点击“意见反馈”一栏进行留言，并提交。用户在输入框内输入意见内容和联系方式，点击“提交”按钮，意见就会存储到服务器，如下图所示：

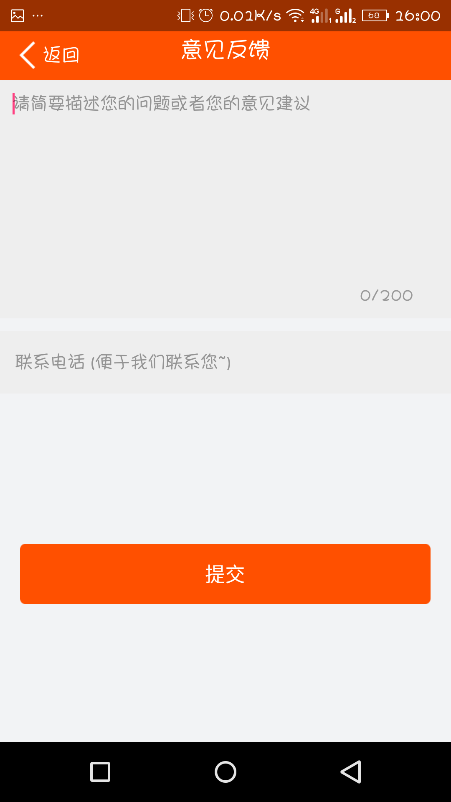


图20 其他功能模块-意见反馈

## 5.9 清理缓存界面

如果用户觉得软件的缓存太多，需要进行清理，便可点击“清理缓存”一栏进行清理。点击后，界面会显示一个环形进度条，用来提示用户正在清理，当清理完成后，会提示用户清理完成。如下图所示：



图21 其他功能模块-清理缓存

# 结 语

这次毕业设计，我巩固了在大学期间所学到的专业知识，学习了以前没有学习的Android编程技术。从选题到调查，再到分析设计与实现，我了解了一个软件的实现流程，也对 Android 开发技术有了了解。完成毕业设计的期间遇到了很多问题，自己解决不了我就上网查阅相关资料，或者请教他人。

本次设计，基本实现了行车辅助的相关功能，用户可以实现地点搜索，路线导航，以及根据需要或者推荐来查看汽车服务店铺。并且通过本次设计，大致了解了手机应用与服务器之间的交互，学习到了系统设计的完整性和逻辑性。系统难免有不足之处，比如UI界面的优化导致的用户体验，视图控制器逻辑上的最佳完善，这些都需要通过现实生活中的实践来进行修改。

通过本次毕业设计，我丰富了自身的学术知识，增强了探索问题和解决问题的能力。相信在今后的学习、工作中会有意想不到的效果。另外，在本次设计过程中，我也对程序设计和学术研究有了很浓厚的兴趣，在与导师和同学的沟通中，学习到了他人更灵活的思路。

# 参考文献

[1] 靳岩, 姚尚朗. Google Android开发入门与实战[M]. 人民邮电出版社, 2009.

[2] 朱桂英. Android开发应用从入门到精通[M]. 中国铁道出版社, 2011.

[3] 李刚.疯狂 Android 讲义[M].北京:电子工业出版社,2013.

[4] 陈昱, 江兰帆. 基于Google Android平台的移动开发研究[J]. 福建电脑, 2008, 24(11):156-157.

[5] 张利国, 龚海平, 王植萌. Android移动开发入门与进阶[M]. 人民邮电出版社, 2009.

[6] 潘功锋. Java技术在手机领域中的应用[J]. 计算机时代, 2002(7):43-44.

[7] 孙国峰, 胡景春. 基于Android的物联网控制系统的设计与实现[J]. 微型机与应用, 2013, 32(1):7-10.

[8] 朱根荣. 百度地图API应用数例[J]. 电脑编程技巧与维护, 2013(23):29-34.

[9] 施连敏, 盖之华. 基于Android的百度地图应用的搭建[J]. 技术与市场, 2014, 21(12):34-35.

[10] 韩超, 梁泉. Android系统原理及开发要点详解[J]. 2010.

[11] 高思, 赵博. 数据库设计流程浅析[J]. 中国科技博览, 2015(42):319-319.

[12] 冯秀珍, 郝鹏. 云计算环境下的信息资源云服务模式研究[J]. 计算机科学, 2012, 39(s2):110-114.

[13] 张潇丹, 李俊. 一种基于云服务模式的网络测量与分析架构[J]. 计算机应用研究, 2012, 29(2):725-729.

[14] 周治国, 李文印, 万吉,等. 基于Java的E—mail系统客户端的设计与实现[J]. 吉林大学学报:信息科学版, 2004, 22(2):169-172.

# 致 谢

本论文是在王娟老师的悉心指导和耐心帮助下设计完成的。王娟老师渊博的知识素养和严谨的教学态度使我获益颇多，对顺利完成本设计起到极其关键的作用。在此向她表示衷心的感谢！

最后向在百忙之中评审本文的各位专家、老师表示衷心的感谢！

作者简介：

姓 名：白萧 性别：女

出生年月：1995 年 07 月 19 日 民族：汉

E-mail: 1456412034@qq.com

# 声 明

本论文的工作是2016年12月至2017年6月在成都信息工程大学网络空间安全学院完成的。文中除了特别加以标注地方外，不包含他人已经发表或撰写过的研究成果，也不包含为获得成都信息工程大学或其他教学机构的学位或证书而使用过的材料。

关于学位论文使用权和研究成果知识产权的说明：

本人完全了解成都信息工程大学有关保管使用学位论文的规定，其中包括：

（1）学校有权保管并向有关部门递交学位论文的原件与复印件。

（2）学校可以采用影印、缩印或其他复制方式保存学位论文。

（3）学校可以学术交流为目的复制、赠送和交换学位论文。

（4）学校可允许学位论文被查阅或借阅。

（5）学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

除非另有科研合同和其他法律文书的制约，本论文的科研成果属于成都信息工程大学。

特此声明！

作者签名：

2017年 06 月 日