

本科生毕业论文（设计）



题 目 **基于微信的智能共享车位系统**

学 院  **计算机学院**

专 业  **计算机科学与技术**

学生姓名  **李治成**

学 号  **2014141462126** 年级 **2014**

指导教师  **潘薇**

教务处制表

二Ο一七年五月十五日

**基于微信的智能共享车位系统**

专业 计算机科学与技术

学生 李治成 指导老师 潘薇

**[摘要]** 近几十年来，我国盲目快速推动城市化进程，而各地的城市化进程过程中，却没有相应的在城市基础设施建设以及城市管理方面加强改进，停车位建设远远赶不上汽车增长的速度。依托互联网平台，民间资本的涌动加上国家政策支持，停车行业迎来了属于自己的春天，出现的智能车库平台也发展比较快，已出现了很多移动app管理车库平台，但是对于微信这个新宠，有关智能车库的公众号或者微信小程序少之又少，满足不了大多数客户的需求。

因此，本项目是基于智能停车产业界出现的现实用户体验问题提出来的，旨在通过分析研究智能停车人机交互用户体验设计，在一定程度上缓解和优化现有的问题，让众多停车用户能够对基于移动互联网和车联网技术的停车系统产生信任和依赖，成为他们的停车好帮手。

**[主题词]** 微信公众号；停车； 优化

**Based-Wechat Intelligent Shared Parking System**

Major : Computer Science and Technology

Student : Li Zhicheng Adviser : Pan Wei

[Abstract] Recent years, our country has been rapidly promoting the process of urbanization. However, in the process of urbanization in various places, there has been no corresponding improvement in urban infrastructure construction and urban management. The construction of parkinglots is far behind the speed of automobile growth. Relying on the Internet platform, the surge of private capital and the support of national policies, the parking industry ushered in its own spring, and the emerging smart garage platform also developed relatively quickly. There have been many mobile application management garage platforms, but this is the new favorite of WeChat.The number of public accounts or WeChat applets about smart garages are few and cannot meet the needs of most customers.

Therefore, this project is based on the real user experience problems that emerged in the smart parking industry. It is designed to analyze and research the user experience design of intelligent parking man-machine interaction, to alleviate and optimize existing problems to a certain extent, so that many parking users are able to generate trust and dependence on parking systems based on mobile and car networking technologies, and become a good helper for their parking.

[Keyword] WeChatOfficial Accounts; parking; optimization

目录

[第一章 绪论 1](#_Toc483347318)

[1.1 项目背景 1](#_Toc483347319)

[1.1.1传统停车场利用情况分析 1](#_Toc483347320)

[1.1.2目前市场上移动智能停车场软件应用的比较和分析 1](#_Toc483347321)

[1.1.3国内的停车发展现状分析以及未来发展 2](#_Toc483347322)

[1.1.3项目研究意义 2](#_Toc483347322)

[1.2 本文的主要工作和结构框架 2](#_Toc483347323)

[1.2.1 论文的主要工作 2](#_Toc483347324)

[1.2.2 论文的结构框架 3](#_Toc483347325)

[第二章微信公众号的技术分析 4](#_Toc483347326)

[2.1 ssh框架介绍 5](#_Toc483347331)

[2.2.1spring技术 5](#_Toc483347332)

[2.2.2springMVC技术 5](#_Toc483347333)

[2.2.3 hibernate技术 5](#_Toc483347334)

[2.2 前端技术开发 5](#_Toc483347335)

[2.2.1javascript技术 5](#_Toc483347332)

[2.2.2jquery技术 5](#_Toc483347333)

[2.2.3 百度地图开发 5](#_Toc483347334)

[2.1微信公众号开发 4](#_Toc483347327)

[2.4 本章小结 6](#_Toc483347336)

[第三章 应用需求分析 7](#_Toc483347337)

[3.1系统应用需求分析 7](#_Toc483347338)

[3.2性能需求 8](#_Toc483347339)

[3.3公众号设计目标 8](#_Toc483347340)

[3.4 本章小结 8](#_Toc483347341)

[第四章 应用概要设计 9](#_Toc483347342)

[4.1 应用功能模块 9](#_Toc483347343)

[4.1.1 用户关注公众号模块 11](#_Toc483347344)

[4.1.2 停车场出租管理模块 11](#_Toc483347345)

[4.1.3 车主停车管理模块 11](#_Toc483347346)

[4.1.4 管理员更改菜单栏模块 12](#_Toc483347347)

[4.1.5 登录注册模块 12](#_Toc483347348)

[4.1.6其他模块说明 12](#_Toc483347349)

[4.2 项目框架总体架构 13](#_Toc483347350)

[4.2.1视图层与控制层之间传递数据 11](#_Toc483347344)

[4.2.2项目数据流传递方式... ...11](#_Toc483347345)

[4.2.3 hibernate接口 11](#_Toc483347346)

[4.2.4 微信与服务器交互 12](#_Toc483347347)

[4.3数据库设计 14](#_Toc483347351)

[4.3.1接口设计 11](#_Toc483347346)

[4.3.2数据库操作设计 12](#_Toc483347347)

[4.4本章小结 17](#_Toc483347354)

[第五章 应用详细设计和实现 18](#_Toc483347355)

[5.1公众号详细设计与实现 18](#_Toc483347356)

[5.1.1微信请求校验程序设计与实现 19](#_Toc483347357)

[5.1.2 一键关注公众号设计与实现 20](#_Toc483347358)

[5.1.3 公众号信息的发送与回复设计与实现 20](#_Toc483347359)

[5.1.4自定义微信菜单模块 20](#_Toc483347360)

[5.1.5 用户查找停车场模块 21](#_Toc483347361)

[5.1.6 用户管理停车场模块 21](#_Toc483347362)

[5.1.7 用户预约停车场模块 22](#_Toc483347363)

[5.1.8 登录注册模块 22](#_Toc483347364)

[5.3 本章小结 26](#_Toc483347369)

[第六章 测试与成果展示 27](#_Toc483347370)

[6.1 测试环境 27](#_Toc483347371)

[6.2 功能性测试 27](#_Toc483347372)

[6.3 兼容性与性能测试 28](#_Toc483347373)

[6.4 成果展示 28](#_Toc483347374)

[6.5 本章小结 34](#_Toc483347381)

[第七章 总结 35](#_Toc483347382)

[参考文献 36](#_Toc483347383)

[声 明 37](#_Toc483347384)

[致 谢 38](#_Toc483347385)

[附录1 39](#_Toc483347386)

[附录2 45](#_Toc483347387)

# 第一章 绪论

## 1.1 项目背景

随着我国城市经济和汽车工业的迅速发展，拥有私家车的家庭越来越多，而与此相对应的是城市停车状况的尴尬，特别是大城市的机动车拥有量的增长速度远远超过停车基础设施的增长速度，泊位与车数量的比例严重失调，由此带来停车难、违章停车、停车贵、停车管理困难等一系列问题日益突出，轿车露宿街头的景象在许多城市随处可见。解决好出行问题已经成为城市发展的重点之一，我们必须重视城市停车难的问题，并积极探求解决的措施。

智能停车系统建设已经纳入智慧城市建设板块，成为具有经营性的城市智能交通项目，未来，城市停车将逐步进入智能时代，通过智能停车方案解决人工管理、施工难度大、运维复杂、车主体验差、收费管理混乱等问题，在这个阶段智能停车方案将深入到城市每个角落，影响每个车主和停车运营厂商。

微信，是时下最受关注的移动互联网产品。经过长时间的发展，微信的注册已经超过8亿，成为移动互联网最重要的入口之一，微信公众平台是腾讯公司在微信基础上新增的功能模块，通过这一个平台，个人与企业都可以打造一个微信公众平台，并实现与特定群体的文字、拖以及语音的全方位沟通与交流。公众平台主要面向名人、政府、媒体、企业等，在这里可以通过微信渠道将品牌推广给上亿的微信用户，减少宣传成本，提高品牌知名度，打造更具影响力的品牌形象。

### 1.1.1传统停车场利用情况分析

传统停车场仍占市场主导，目前传统停车场建设规模仍占接近90%的市场份额，新型的机械式停车场的建设虽一片火热，但整体市场占比仅10%左右，传统的停车场管理系统只解决了出入口控制的问题，对于停车场停车信息，停车场实时数据等功能望尘莫及，而且在收费这个环节上也存在缴费方式单一、人工管理效率低下、存在收费漏洞等问题，更别说进行停车场整体的系统整合及资源优化配置了。目前国内的停车场尤其是大厦的停车场，收费管理部分做得尤为完善，但是对于一些小型停车场、私有停车场的应用管理，功能比较完善的，也是基于APP应用开发的，但是对于微信平台，公众号与小程序开发出来的功能太少，要么界面太丑，应用信息不够完善，用户宁愿用APP而不用微信平台提供的服务。

### 1.1.2 目前市场上移动智能停车场软件应用的比较和分析

目前，应用市场上关于此类型的软件鱼龙混杂，本文选取国内外比较受欢迎的三款app应用PP停车、ETCP停车、宜停车加以分析。这三款软件都是针对停车场开发的软件，能够实现停车场查找、管理、车位市场发布等功能，对比其最基础的功能可以发现可以有以下这些功能：最基本的功能是定位当前位置、实时查看周边停车位数量及停车费用、导航到指定停车位、手机支付停车费用（提供停车券活动）、分享app等功能。这三款软件早早地进入了市场，得到了一些投资者的青睐，而且积累了一定量的停车场客户。有较多的收费停车场的客户，就会对停车场的业务相比其他后来者要熟练不少。其次，在线支付功能易用，支持微信、支付宝支付，而且可以直接充值到app。有良好的业务能力，足以满足一般停车需求，不过这些应用软件数据库还并不是很大。

目前国内停车位APP服务的比较多，对于微信公众号少之又少。app停车主要侧重在闲置车位出租上，ETC停车场主要侧重找车位，还有的一些停车APP则是什么违章查询之类的，但是对于一些小型 停车场、私有停车场的应用管理，功能比较完善的，也是基于APP应用开发的，但是对于微信平台，有较大的开发利用价值。

### 1.1.3 国内的停车发展现状分析以及未来发展

国内的停车应用比较多，ETCP、好停车、吾舍停车等，但是一直做不起来，具体分析有以下几个原因：

1.就目前市场上的应用来看，目前的难点在线下车位渠道上，缺少相应的政策支持，大部分物业、商圈、市政是不愿意把车位拿出来贡献的，需要找到利益切合点才好开展工作。

2. 产业联调太长，如果和小区打通，涉及物业、闸机厂商、业主委员会，再加上提供车位的人，利益分配难统一，谈判成本过高；

3. 闸机设备的厂商太多太杂，要和这么多的厂商打通不容易，且有的用的是车牌识别，有的是蓝牙卡，门槛不高，各个地方又有很多的小厂商。

4. 目前的车位共享其实更多的是租赁经济，一个车位运营商去和停车场合作，把车位拿过来再租给需要车位的人，并不是真正的C2C共享，少了一方的参与者，是完全重资产模式，需要不断的融资去拿车位，市场不容易爆发。

智能车库发展方向：

一是将更多的高科技使用到停车设备中，韩国和美国的车库行业将机电技术成果随时转化和移植到车库产品中，使车库产品更新很快，也使车库的存取车速度加快，存车量变大，从而逐步代替老式停车设备。

二是越来越重视停车设备的性价比和经济性。在保证停车设备的性能质量前提下，生产厂家更重视如何提高产品的实用性和降低单车位造价。

### 1.1.4 本项目的研究意义

停车场智能化管理的现状将成为未来的发展趋势，随着我国劳动力成本的快速上升，停车场的管理人员越来越少，自动化程度越来越高，无人化服务将逐渐普及，所以，安装[智能化](http://link.zhihu.com/?target=http%3A//www.sztxsy.net/a/zhinenhuaxiliechanpin/)[停车场管理系统](http://link.zhihu.com/?target=http%3A//www.sztxsy.net/index.html)将成为未来大型停车场的一种趋势。

随着智能停车场的发展，很多地方的停车场将实现联网共享数据，可通过场内停车量、停车收费额、客流方向等数据了解车主们的需求，目的是使人们停车缴费更方便。由于现在停车难和乱收费问题已是人们最为头疼的问题，为了解决这一问题，停车场管理将逐步走向智能化。停车场智能化管理主要体现在停车场安装的智能停车场管理系统，停车场管理系统包括出入口管理、车位引导、反向寻车、车牌自动识别、自助缴费等很多个子系统组成。同时采用中央收费模式、远距离识别技术等来实现智能化管理的功能。在一些大型的停车场内我们可以看到人们从停车到缴费整个过程只需短短的几分钟时间即可完成，不再像之前那样，把大部分时间都浪费在停车上面。

其次停车场的发展趋势将逐步走向智能化、无人化。所以传统的停车场管理已不能满足人们的需求，很多地方的停车场开始进行改造，升级为车牌识别系统，同时停车场内安装车位引导和反向寻车及自助缴费系统，这样进出停车场的车辆可以通过车位引导系统快速的找到停车位，返回停车场时，由于场内车辆较多，一时间不能找到自己爱车停放的位置，通过反向寻车系统，车主只需几分钟时间就可找到车辆的存放位置，同样缴费方式也比之前更加的多样化，人们可以选择在停车场的出口处通过停车场收费系统的提示进行缴费，也可以通过场内的自主缴费机缴费，这样不但不会出现乱收费现象，还可以节省车主排队缴费的时间。

微信公众平台，作为微信最重要的部分，以“再小的平台也有自己的平台为宣传口号”，吸引了众多个人与企业加入，争先在微信平台上推广自己，到目前为止，每年都在以80000个左右的速度增长，微信公众平台经过这么久的发展，受到了广大开发者的青睐。

## 1.2 本文的主要工作和结构框架

### 1.2.1 论文的主要工作

本文主要介绍系统的功能、实现、性能以及相关技术。开发此公众号主要是从目前市场出发，借助微信强大的连接能力，将停车系统、商场、服务及用户连通互动，实现online和offline的真正无缝融合。下面是关于本项目的一些介绍：

1）选用公众号，而不是传统的APP

传统的APP，不仅占手机内存，还需要损耗手机流量，增加了用户的抵触情绪，支付功能一般都是第三方支付，需要填写很多资料，比较繁琐复杂，启用微信公众号，一切就变得更加简单，公众号有推送功能，界面简洁，易于操作，便捷支付，这个方面来说比传统的APP方便。目前在国内，公众号是最佳的吸粉神器，通过公众号车主可以很快速的使用相应服务，公众号有天然的优势，可以有很多的信息推送以及增值服务可供选择，后期的维护更加方便。

2）实时获取周围停车场，提供到停车场的路径导航

相比传统的微信公众号而言，根据定位情况，加载出来周围所有停车场，网页的限制，导航去停车场往往只能调用第三方，而本公众号，提供了路径导航，根据终点，提供去此终点的路径导航，用户可以根据自己的情况选取路线。

3）更加注重系统性能

比起传统的项目，完全不考虑数据库的性能，往期只要涉及到数据库的增、删、改、查的时候完全不考虑数据库的压力，在项目中，应用了hibernate的一些性能对数据库进行了优化，减少了对数据库的操作，很大程度上提高了系统的性能，在用户量很大的情况下，依旧可以保持良好的运行。

基于以上三点的设计方案，本论文开展的主要工作有：

1. 学习微信公众号开发的方法，了解微信的相关的技术，搭建微信平台；
2. 利用百度地图提供的接口，解决停车场位置、定位问题。
3. 项目主要利用java语言编写，采用ssh（spring，springMVC，hibernate）框架，最大程度上优化系统性能。
4. 采用瀑布模型开发方法，从用户需求分析开始，通过策划、建模、构建与部署的过程，最终提供一个完整的软件。

### 1.2.2 论文的结构框架

第一章绪论介绍了项目背景，当前国内外停车场的发展状况，分析国内停车场发展不起来的原因，阐述微信公众号优点，项目的开发意义。

第二章介绍开发微信公众号相关的技术。

第三章通过用例图分析该智能车库应用的用户需求，描述本应用的功能并提出应用设计目标

第四章采用UML建模，通过需求用例分析对应用的总体架构进行概要设计

第五章阐述微信智能车库应用的详细设计以及实现，包括系统性能设计，界面设计以及类的设计。

第六章介绍了应用测试，列出了主要的测试用例和测试结果

第七章对本项目进行了总结，介绍了应用存在的不足以及发展方向。

# 第二章 微信公众号的技术分析

本公众号主要利用微信提供的API进行开发，因为接口限制，申请微信测试号来解决接口问题。以eclipse为开发工具，利用Apache tomcat v9.0搭建服务器。客户端利用javascript、jquery、css、html、bootstrap开发的响应式页面，优化界面，服务端利用java语言编写，采用ssh（spring、springMVC、hibernate）框架简化开发流程。利用mysql来保存用户的数据。客户端与服务器交互采用json、xml格式数据，springMVC提供的模型、视图、控制层的开发，方便了界面与服务器的交互。

## 2.1 微信公众平台介绍

现如今，微信已经不再只承担着交流沟通、娱乐大众的功能，微信公众号的推出将微信逐渐转变成个人、商家、企业单位用来营销的重要工具。对于公众号平台的开发，官方只给出了PHP示例代码，本项目使用的是java语言开发。公众平台的处理程序其实就是一个web项目，负责接收并相应微信服务器发送的http请求。只要能支持动态web开发的编程语言都能用于公众号开发。

## ssh框架介绍

SSH框架一般指的是Struts、Spring、Hibernate，后来Struts2代替了Struts，最近5年，Struts2已经被Spring MVC代替，而Hibernate基本也被iBatis/MyBatis代替。本项目采用的是spring，springMVC,hibernate的框架。使用框架的目的提高程序的可读性和维护性，框架的核心思想是按照关注点分离，按照MVC的框架进行分离。

### 2.1.1 spring技术

Spring基于IOC和AOP的结构J2EE系统的轻量级的开发框架，框架的基础版本在2 MB 左右，使用 Spring 框架可以创建好性能、易于测试、可重用的代码，为容易而快速的开发出耐用的 Java 应用程序提供了全面的基础设施。

Spring涉及到相关技术：

IOC（Inversion Of Control）反转控制是Spring的基础，对象的生命周期由Spring来管理，直接使用Spring去获取对象，创建对象由以前的程序员自己构造变成了交由Spring创建。

DI（Dependency Inject）依赖注入.拿到的对象的属性，已经被注入好相关值，可以直接使用。

AOP 即Aspect Oriented Program面向切面编程，能够选择性的、低耦合的把切面和核心业务功能结合在一起的编程思想。在面向切面编程的思想里面，把功能分为核心业务功能和周边功能。所谓的核心业务，比如登陆，增加数据，删除数据都叫核心业务。性能统计，日志，事务管理等等是周边业务，周边业务也被定义为切面。

### 2.1.2 springmvc

Spring Web MVC是一种基于Java的实现Web MVC设计模式的请求驱动类型的轻量级Web框架，即使用了MVC架构模式的思想，将web层进行职责解耦，基于请求驱动指的就是使用请求-响应模型。SpringMVC是Spring框架的一个模块，SpringMVC和Spring无需通过中间整合层进行整合。Spring MVC 分离了控制器、模型对象、分派器以及处理程序对象的角色，这种分离更容易进行定制。

Spring 的 Web MVC 框架是围绕DispatcherServlet进行设计，把请求分派给处理程序，同时带有可配置的处理程序映射、视图解析、本地语言、主题解析以及上载文件支持。应用控制器其实拆为处理器映射器(Handler Mapping)进行处理器管理和视图解析器(View Resolver)进行视图管理，页面控制器是非常简单的Controller接口，只有一个方法ModelAndView handleRequest(request, response)。Spring 为其提供一个控制器层次结构，可以派生子类。

### 2.2.3 hibernate

使用JDBC做数据库相关功能开发会做很多重复性的工作，比如创建连接，关闭连接，把字段逐一映射到属性中。Hibernate是数据访问层的框架，对JDBC进行了封装，是真对数据库访问提出的面向对象的解决方案。使用Hibernate可以直接访问对象，自动将访问转化成SQL执行，从而达到简介访问数据库的目的，简化数据访问层的代码开发。

## 2.2 前端技术开发

### 2.2. javascript

JavaScript是世界上最流行的脚本语言，在电脑、手机、平板上浏览的所有的网页，以及无数基于HTML5的手机App，交互逻辑都是由JavaScript驱动的。JavaScript依赖于浏览器本身，与操作环境无关，只要计算机能运行浏览器，并支持JavaScript的浏览器，就可正确执行，从而实现“编写一次，走遍天下”的梦想。

本项目的前端交互应用了javascript语言，提高良好的交互性能，减少了后台数据的压力，对表单的判断等都可由前端直接执行判断。在调试的时候直接使用浏览器进行调试，方便开发。

### 2.2.2 jquery

JQuery是一个javascript的框架，是对javascript的一种封装。使用jquery可以消除浏览器差异，不需要写冗长的代码来针对不同的浏览器来绑定事件，编写AJAX等代码，可以简洁的操作DOM的方法，轻松实现动画、修改CSS等各种操作。

本项目前端使用了jquery来实现遍历HTML文档，操作DOM、事件处理、执行动画等操作，相比javascript更加简洁，更加方便。

### 2.2.3 百度地图开发

百度地图JavaScript API是一套由JavaScript语言编写的应用程序接口，可在网站中构建功能丰富、交互性强的地图应用，支持PC端和移动端基于浏览器的地图应用开发，且支持HTML5特性的地图开发，支持HTTP和HTTPS，免费对外开放，可直接使用。在申请百度AK之后，接口无限制使用。

本项目开发的应用百度地图进行定位，提供路径导航，获取用户的地理位置信息，并为其展示周边的停车场。相比腾讯地图，百度地图开发文档更加完善，有丰富的demo，能很快的进行开发应用。

## 2.3微信公众号开发

凭借大量活跃用户，微信已成为商家重要营销平台之一。商家为庞大用户群提供定制化服务的迫切需求，吸引了大量开发者/开发商投入到微信公众平台开发中。学习和掌握微信公众平台微信的火爆让我们大家都不敢坐视不理，不管是企业还是个人开发者。

微信公众号可分为一下三类：

订阅号：主要偏于为用户传达资讯（类似报纸杂志），认证前后都是每天只可以群发一条消息；

服务号：主要偏于服务交互（类似银行，114，提供服务查询），认证前后都是每个月可群发4条消息；

企业号：主要用于公司内部通讯使用，需要先有成员的通讯信息验证才可以关注成功企业号；

根据公众号类型进行相应的认证开发，本项目偏向于服务号开发，可以与用户进行交互，

提供良好的性能，以确保服务质量。

## 2.4 本章小结

本章主要介绍微信公众号开发及其需要的技术。首先介绍了微信公众号平台基础知识，结合项目的开发，ssh（spring，springMVC,hibernate）框架涉及到的概念，包括数据交换格式，前端开发用到的主要技术。

# 第三章 应用需求分析

根据上文项目背景与应用分析的简要阐述，设计项目的基本需求，本章将会通过需求分析来进一步确定应用的主要功能及设计目标。

## 3.1系统应用需求分析

根据项目的要求，结合公众号的开发，本应用需要提供以下功能：

（1）用户注册登录：用户在关注公众号之后，需要完善自己的信息，进行注册登录；

（2）根据用户位置定位，地图上显示停车场：根据当前所在位置，显示最近的20个停车场；

（3）停车场搜索：用户根据需要，输入位置，根据位置显示最近的20个停车场；

（4）停车场管理：用户发布停车场，管理停车场，进行修改，删除；

（5）管理车辆：用户需要添加车辆，对自己的车辆进行管理。

（6）预定停车场：用户根据需要，可以对停车场进行预定，取消预定。

根据用户的需求，相应的用例图如下3-1。

（7）管理员模块：对用户信息进行审核，验证。

（8）更改公众号菜单：对公众号的界面菜单就行更改，删除。

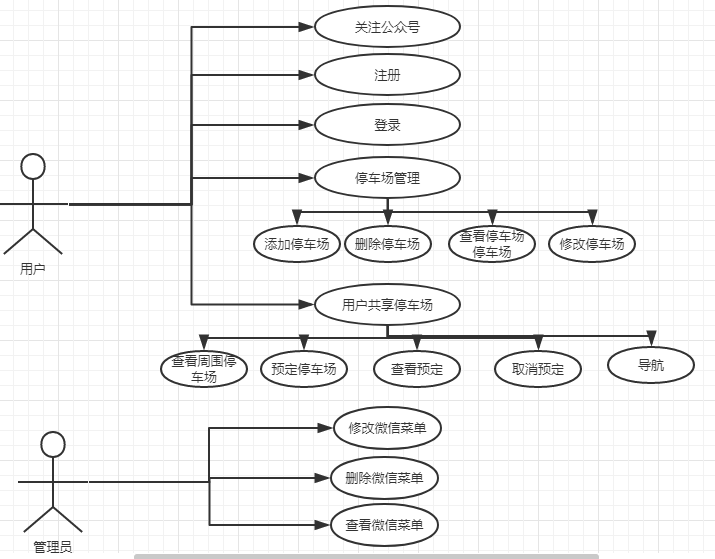


图3-1应用整体用例图

## 3.2性能需求

系统的性能在开发设计过程中是一个非常重要的过程，好的系统不仅可以提升系统的质量，还可以提升用户的认可程度，尤其在数据库的设计与开发，应尽可能减少对数据库的操作，得到尽可能多得数据，减少系统压力，以防因访问量过多，系统崩溃。根据系统设计的目标，本公众号的性能需求如下：

1. 系统性能需求

本应用是公众号，运行在微信上边，在公众号上访问服务器反应时间要求较高，尤其应能及时准确的加载定位，实时传递应用数据，不能要用户等待过长，如果有等待不能影响其他用户的使用。该系统需要多次请求获取数据，对数据的交互以及准确度要求非常高。

1. 运行性能需求

该应用主要运行在移动设备上边，由于设备功能资源限制，应减少运行时所占内存，降低所占用的cpu资源。

1. 界面功能需求

本应用主要针对广大的车主，界面要求尽可能的布局合理，简洁美观，方便用户操作。

1. 安全性需求

系统需确保用户数据安全，用户停车记录、个人信息等保密。

1. 实时性能需求

实时性能至关重要,在软件的需求分析中充分考虑,以保证开发出的应用软件的实时性能。这些实时性能的体现一部分来源于实时操作系统的实时性能,另一部分则依赖于应用软件本身的设计和代码。本应用实时性能要求较高，需设计满足要求的系统。

## 3.3 公众号设计目标

根据上述用例和需求，该公众号的具体设计目标为：

（1）实时加载用户位置，停车场数据

（2）用户能及时进行停车场预约、取消，查看自己的订单

（3）根据用户的位置，提供到停车场的导航

（4）用户能方便添加、查看、删除、修改停车场

（5）界面简洁，易于操作

开发实现中上述目标的优先级别依次降低。

## 3.4 本章小结

本章明确了应用的功能性需求，同时描述了性能需求并确定公众号设计目标。

# 第四章 应用概要设计

“微信智能停车场”是基于微信开发的公众号，主要是针对广大车主及有停车位的用户开发的应用，旨在为广大车主提供停车场停车，为广大有停车位的用户出租停车场。该应用主要解决车主找停车位难，有停车位的用户难出租的问题。本章将根据需求分析进行系统模块与功能设计，以及数据库设计的详细说明。

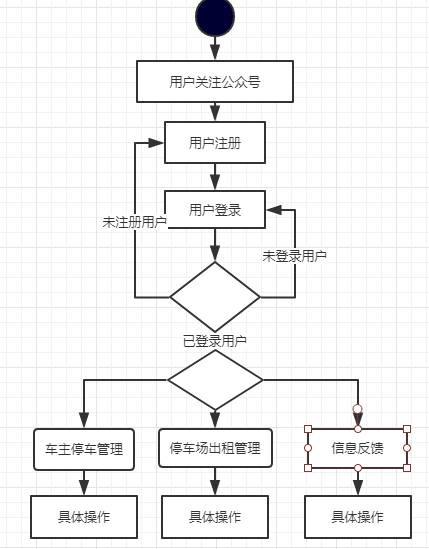
## 4.1 应用功能模块

根据用户需求， 本项目所包含的功能模块如下图4-1所示：



图4-1应用功能模块

整个项目主要分为两个部分，客户端部分和服务端部分，客户端主要负责与用户进行交互，服务端主要负责数据的存储于处理。系统操作流程如下：

 图4-2 系统流程图

用户模块主要分为登录注册、停车场出租管理、车主停车管理、信息反馈。

### 4.1.1 用户关注公众号模块

用户关注公众号，进入公众号之后，其主要操作如下：

扫二维码关注：用户扫描二维码之后，关注公众号成功，用户可以进行对公众号的操作。

### 4.1.2停车场出租管理模块

停车场出租管理模块，主要针对有停车场的个人或者单位，根据需要出租停车场，为广大车主找到停车位。主要操作如下：

1）出租停车场添加：公司或者个人根据需要出租停车场，需要输入信息：停车场位置、停车场名称、停车场收费、停车场描述、停车场类型。停车场类型分为：大车、小车。停车场收费分为：起始价与每小时收费价格。

2）查看已出租的停车场：在界面查看已出租的停车场。

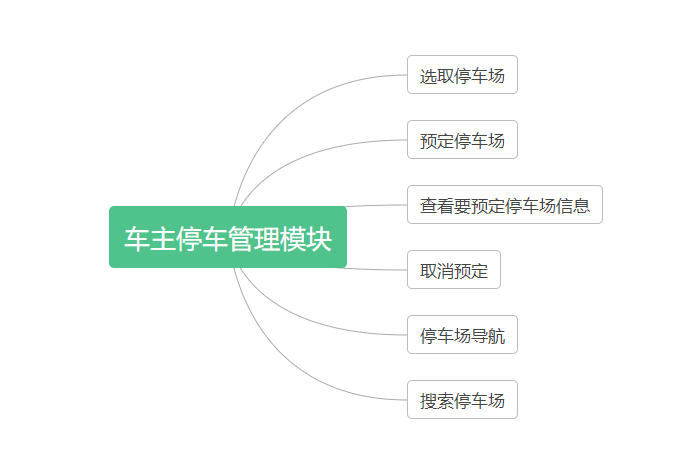
3）修改停车场信息：用户修改已添加的停车场信息。

4）删除停车场： 用户删除已添加的停车场。

### 4.1.3 车主停车管理模块

该模块是系统的重要功能，将根据用户定位在地图上加载最近的停车场，用户根据需要选择停车场。

停车场模块功能图

 图4-3停车功能模块

1）选取停车场：将根据用户的定位在地图上加载离用户最近的停车场，用户可以在地图上选取停车场。

2）查看预定停车场信息：在界面加载的停车场，用户点击之后查看停车场信息，在地图页面信息包括：停车场名称、停车场距离、停车场位置、停车场剩余停车位、停车场收费情况。当点击之后将跳转到详细界面，显示更加详细的停车场信息。

3）搜索停车场：在主界面输入要搜索位置之后，点击搜索，将根据定位显示距离定位最近的停车场。

4）预定停车场：用户在确定要选取的停车场之后，进行预定。

5）停车场导航：在用户预定停车场之后，为用户提供导航用户根据路劲导航到停车场。

### 4.1.4 管理员更改菜单栏模块

管理员模块目前设计的主要是修改微信菜单的功能，该功能主要是针对停车场管理员的需求，更改公众号的菜单栏，使菜单栏更加人性化，满足管理员的需求。

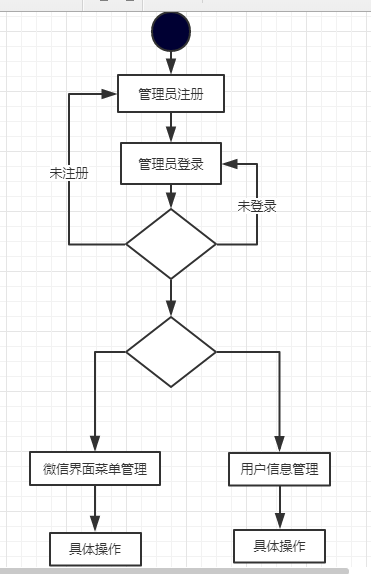


图4-4管理员的操作流程图

### 4.1.5 登录注册模块

用户与管理员都需要登录注册模块，只有登录注册之后才能使用本公众号提供的功能。

### 4.1.6 其他模块说明

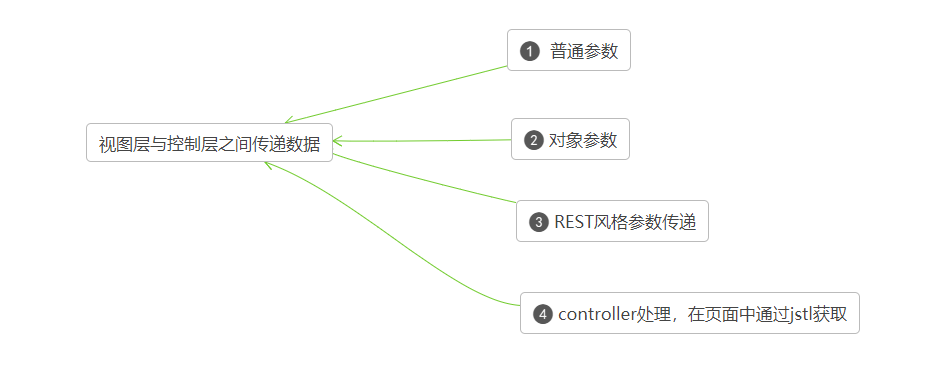
其他模块主要包括投诉与联系我们的功能，提供给用户投诉停车场与反馈信息的功能。

## 4.2项目框架总体架构

本项目公众号服务端采用了SSH框架来实现， spring框架的主要优势之一就是其分层架构，分层架构允许使用者选择使用哪一个组件，同时为*J2EE*应用程序开发提供集成的框架。Spring使用基本的JavaBean来完成以前只可能由EJB完成的事情。然而，Spring的用途不仅限于服务器端的开发，从简单性、可测试性和松耦合的角度而言，任何Java应用都可以从Spring中受益。Spring的核心是控制反转（IoC）和面向切面（AOP），传统的B/S架构的Web应用，我们一般可以把它分为表现层、业务逻辑层、数据链路层三层，在springMVC的设计模式中，这三层又分别对应着View、Controller和Model。Hibernate是一个开放源代码的对象关系映射框架，它对JDBC进行了非常轻量级的对象封装，使得Java程序员可以随心所欲的使用对象编程思维来操纵数据库。Hibernate可以应用在任何使用JDBC的场合，既可以在Java的客户端程序使用，也可以在Servlet/JSP的Web应用中使用，最具革命意义的是，Hibernate可以在应用EJB的J2EE架构中取代CMP，完成数据持久化的重任。下面介绍视图层与控制层之间传递数据方式、项目数据流传递方式与hibernate核心接口。

### 4.2.1视图层与控制层之间传递数据

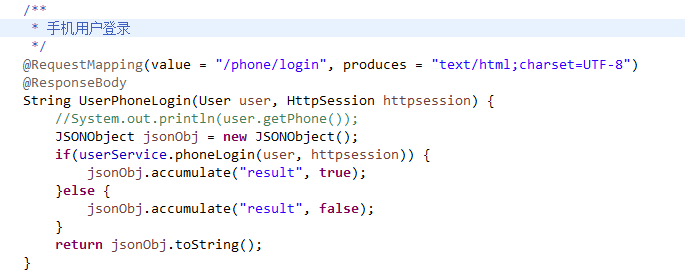
视图层主要负责与用户进行交互，用户输入数据与现实数据在视图层进行显示，标准的MVC模式里，通过request对象来做数据交换。在SpringMVC中，框架提供基于request对象之上的方式来交换数据，当然可以通过SpringMVC中获取request对象的方法来得到request对象之后，像以往那样传递数据。但SpringMVC提供了各式各样，用于不同情形的数据交换方式，提供了更加方便的方法。现面我们逐一介绍，

 图4-5视图层与控制层传递数据方式

1）普通参数

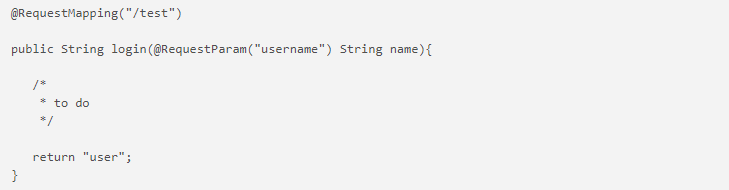
controller传递参数一般有URL和Form表单两种方式。

* 1. Form表单：

 图4-6 控制层获取form表单

在普通的Servlet中，通过request.getParameter的方法来逐一的获取form表单里元素的值，现在只需要函数参数的名称与form表单里元素的name属性相同，controller方法里就可获取值。

* 1. Form表单 或 URL?username=xxxx 方式

 图4-7 控制层获取参数

RequestParam()中的字符串与URL中传入的参数的名称或Form表单中对应元素的name属性的值相同，字符串name就是我们得到的想要的参数的值。但需注意运用此方式传值时，对应的参数的值不能为空，如果传入的URL中参数的值为空时，会报错。

2）对象参数

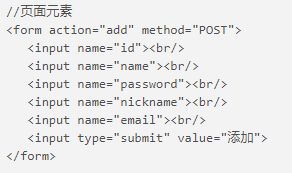
 对象参数的方式是通过页面直接提取对象数据

图4-8 控制层提取对象数据

注意：我们页面里input元素中的name属性中的id,name…都是直接对应contorller add方法里@ModelAttribute的参数user对象的属性字段，必须一致，不可不同。

3）REST风格参数的传递

在URL中使用REST风格进行传递参数，使用value = “/{值}”的方式 ，并添加@PathVariable注释的方法来传递参数值。

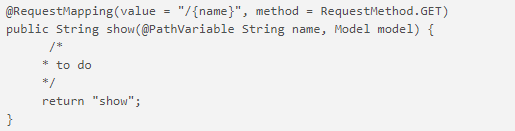


图4-8 REST风格参数的传递

1. 在页面中获取参数

SpringMVC中提倡的在controller中加入一个参数类型为Model的参数，在视图页面中通过jstl获取该参数的数值。

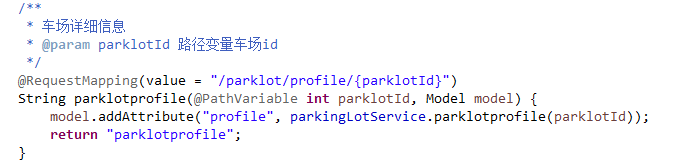


图4-9 Controller中参数示例

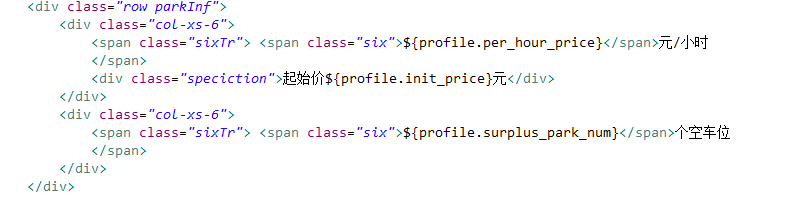
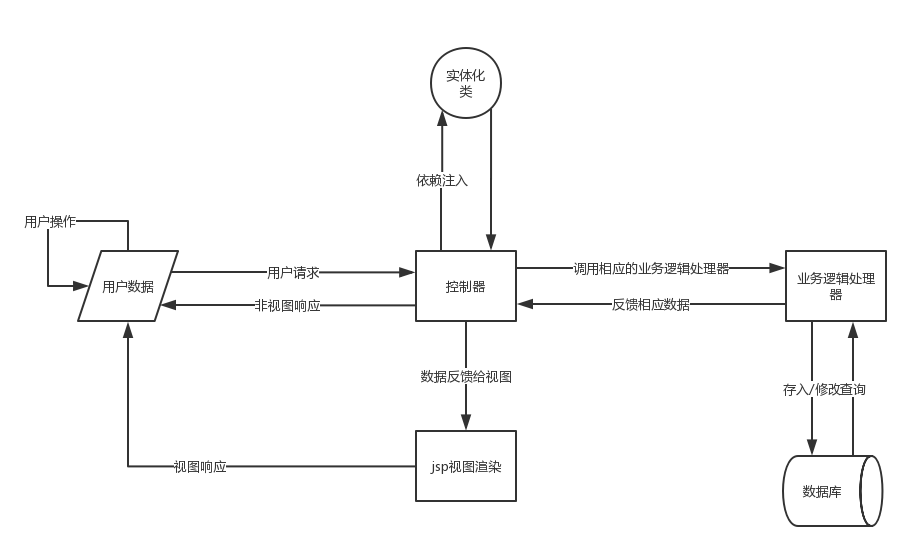


图4-10在jsp页面获取信息示例

### 4.2.2 项目数据流传递方式



当用户操作当前页面的时候，数据有两种流向，一种是前端到前端，用脚本或者缓存反馈给用户。当用户发起请求发送到微信服务器，发送post请求，所有请求都封装在request中，通过对request进行解析，根据请求的url交给控制器进行处理，如果请求中包含数据进行依赖注入，把数据注入实体类，不需要业务逻辑处理的直接返回页面，当请求中有数据需要业务层处理时，交给业务逻辑层进行处理，业务逻辑层通过dao层与数据库进行交互，得到需要的数据，请求完成之后，把数据封装在request中，把请求的数据返回给视图，有一部分是在业务逻辑层调用微信接口，通过返回的结果返回给控制器，没有与数据库进行交互。

### 4.2.3 hibernate 接口

Hibernate主要负责操作数据库，包括对数据库的增加、删除、修改、查找，核心接口主要有5个

1. SessionFactory接口

SessionFactory接口负责初始化Hibernate，专门用来创建session会话的工厂，创建一个sessionfactory，有两种创建方式，1.从XML文件读取配置信息构建SessionFactory，2.从Java属性文件读取配置信息构建SessionFactory，一般一个项目只想要一个sessionfactory，当有多个数据库的时，需为每个数据库指定sessionfactory。

1. Session

http中的session用来记录用户的访问信息，而hibernate中的session用来表示应用程序与数据库的一次交互，包含持久化方法，Session是一个轻量级对象（线程不安全），通常将每个Session实例和一个数据库事务绑定，每执行一个数据库事务，创建一个新的Session实例，在使用Session后，需关闭Session，session关系有三种状态，分别为：

Transient（自由/临时状态）：没进行持久化，数据库中没记录，没有与Session发生过关联。

Persistent（持久状态）：持久的实例在数据库中有对应的记录，在Session的缓存中。

Detached（游离状态）：持久的实例在数据库中有对应的记录，但不在Session的缓存中，曾经与Session发生过关联。

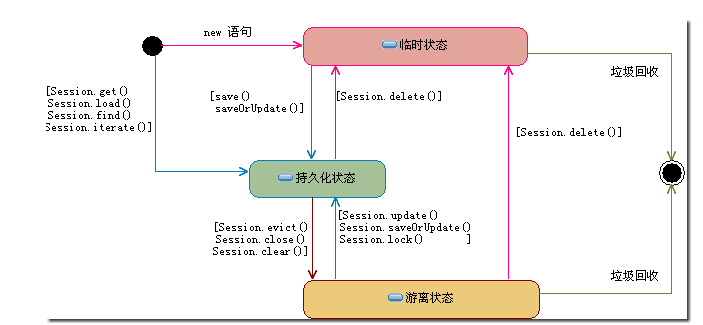


图4-12 Session状态图

### 4.2.4微信与服务器交互

公众平台信息交互流程如下：

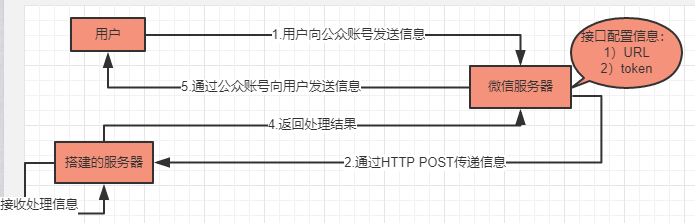


图4-13公众平台信息交互流程

1. 微信客户端向公众账号发送信息，微信服务器接收信息；
2. 微信服务器接收信息之后，将信息根据在接口配置中填写的url，通过HTTP POST方式传递到公众账号服务器上边；
3. 搭建的服务器接收信息之后进行逻辑处理，处理之后将信息返回给微信服务器；
4. 微信服务器接收信息将信息返回给公众号；

## 4.3数据库设计

数据库是项目的核心，停车场信息、个人信息等都保存在数据库里。好的数据库设计可以减少数据库的压力，提升系统性能，数据库设计时需遵循完整性、去冗余、解耦合等原则，下图是设计的E-R图：

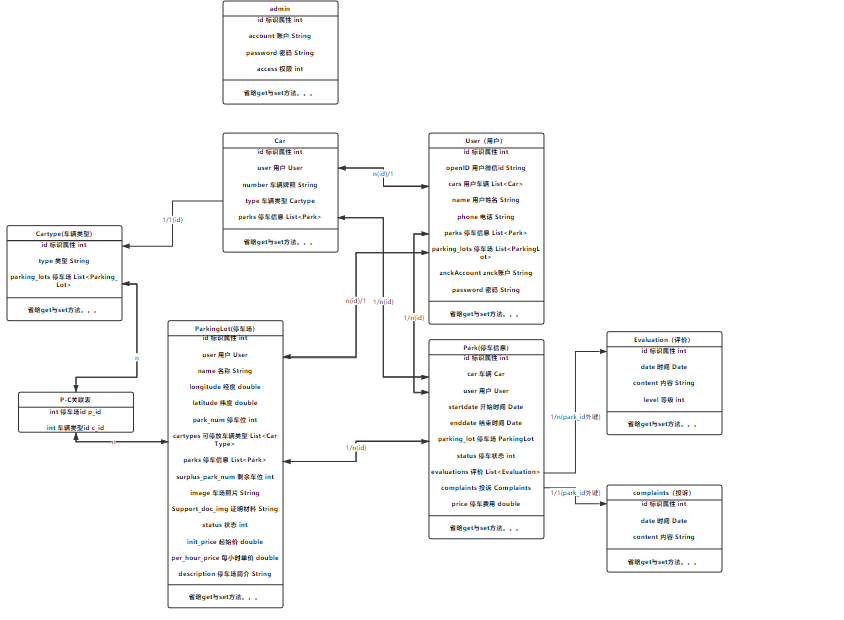


图4-14 E-R图

根据E-R图，形成的数据表分别为：

表4-15管理员信息表 admin

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 约束 | 意义 |
| id | int | 主码，自增，不为空 | 序号 |
| access | int | 不为空 | 权限 |
| password | String | 不为空 | 密码 |
| account | String | 不为空 | 账户 |

表4-16个人信息表user

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 约束 | 意义 |
| id | int | 主码，不为空 | 用户序号 |
| openID | string | 不为空 | 微信用户id |
| cars | List<Car> |  | 用户车辆 |
| name | String |  | 用户姓名 |
| phone | String |  | 导航 |
| parks | List<park> |  | 停车信息 |
| Parking\_lots | List<ParkingLot> |  | 停车场 |
| znckAccount | string | 外码（参照user表） | Znck账户 |
| password | string | 不为空 | 密码 |

表4-17停车场信息表parkinglot

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 约束 | 意义 |
| id | Int | 主码，自增，不为空 | 停车场序号 |
| address | User | 不为空 | 地址 |
| Description | String |  | 描述 |
| Image | String |  | 照片 |
| Init\_price | Double | 不为空 | 起始价 |
| Latitude | Double | 不为空 | 经度 |
| Longitude | Double | 不为空 | 纬度 |
| Name | String |  | 停车场名称 |
| Park\_number | Int | 不为空 | 停车场数量 |
| Per\_hour\_price | Double | 不为空 | 起始价 |
| Status | Int | 不允许为空 | 状态 |
| Support\_doc\_image | String |  | 证明材料 |
| Surplus\_park\_num | String | 不为空 | 专门材料 |
| User\_id | int | 外键（参照user表的id），不为空 | 用户的id |

表4-18停车场信息表park

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 约束 | 意义 |
| id | Int | 主码，自增，不为空 | 停车场序号 |
| Estimatearrival | Date |  | 预计到达时间 |
| Car\_number | Int |  | 车牌号 |
| Enddate | Date |  | 结束时间 |
| Price | Double |  | 价格 |
| Startdate | Date |  | 开始时间 |
| Status | Int |  | 状态 |
| Parking\_id | Int | 外键，参照park表，不为空 | 对应park表的id |
| User\_id | int | 外键，参照user表，不为空 | 对应user表的id |

表4-19用户车辆car

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 约束 | 意义 |
| id | int | 主码，不为空 | 序号 |
| number | int |  | 车牌号 |
| whethercheck | int |  | 是否核对 |
| type\_id | int | 外码（参照cartype的id），不为空 | 类型id |
| user\_id | int | 外码（参照user的id），不为空 | 用户id |

表4-20车辆类型信息表cartype

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 约束 | 意义 |
| Id | Int | 主码，不为空 | 车辆类型序号 |
| type | int |  | 车辆类型 |

表4-21投诉信息表complaints

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 约束 | 意义 |
| id | Int | 主码，不为空 | 投诉信息序号 |
| Content | String |  | 投诉内容 |
| Date | Date |  | 投诉时间 |
| Complaint\_id | Int | 外键，参照parkinglot的id，不为空 | 投诉停车场对应的id |

表4-22评价信息表evaluation

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 约束 | 意义 |
| id | Int | 主码，外码（参照user\_msg），不为空 | 序号 |
| Cotent | string | 不为空 | 评价内容 |
| Date | Date | 时间不为空 | 评价时间 |
| Level | String |  | 评级 |
| Parkid | Int | 外键，参照park表的id，不为空 | 停车场对应的id |

上述是本项目要用到的大部分信息表，还有一些表不一一列举。

### 4.3.1接口设计

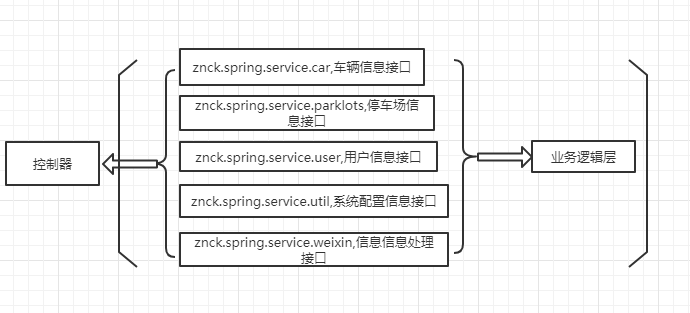


图4-23接口设计

接口设计的目的是方便改动与开发，可以轻松连接到业务逻辑层有哪些功能，方便处理。整个项目总工有五个大的接口包文件，里边有更加信息的实现。

### 4.3.2 数据库操作设计

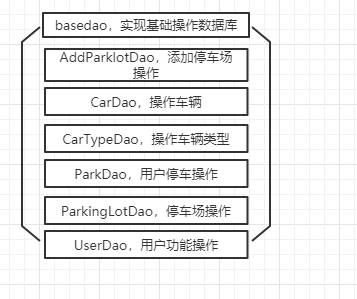


图4-24数据库设计

设计dao层的目的是方便操作数据库，方便改动且减少代码量。

## 4.5本章小结

第四章主要设计了项目的总体架构，阐述了系统功能，对ssh框架传递参数进行说明，描述了hibernate的常用接口。根据系统的需求、性能完成数据库的设计。

# 第五章 应用详细设计和实现

微信公众号开发是通过eclipse开发平台用java开发的应用，下边将信息介绍整个应用的详细设计。

## 5.1 公众号详细设计与实现

根据项目首先搭建微信公众开发平台，通过搭建平台，通过微信服务器访问已部署的服务器来实现，具体实现代码如图：

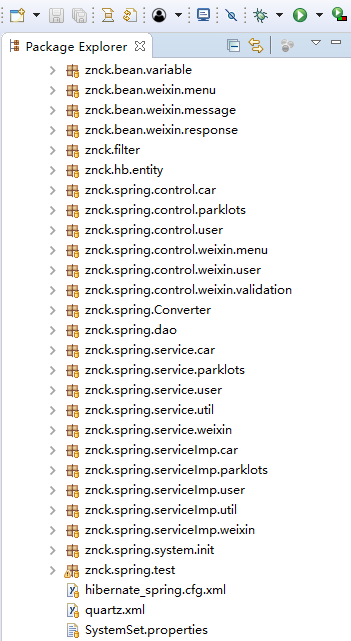
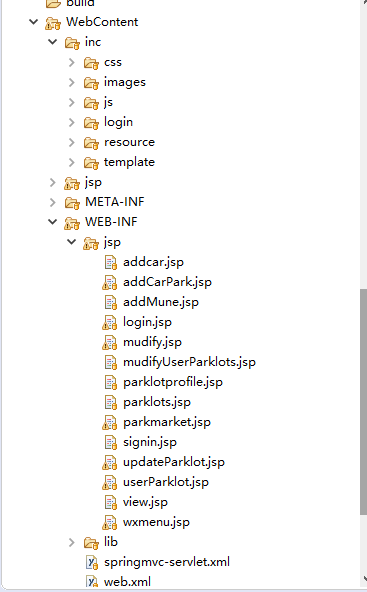
 

图5-1 代码目录结构

其中，inc目录中放置了项目中所用到的所有资源，包括图片、css、js等配置资源，WEB-INF/jsp下边是所有页面显示文件。

Java文件中放置需要的配置文件以及java代码，包括：

1）znck.spring.dao：BaseDao.java是其他所有操作数据库文件的父类文件，实现了基础的CRUD，其他操作数据库文件根据需要在这个文件基础上进行扩展。

2）znck.hb.entity：定义了所有Javabean，通过配置hibernate文件，实现自动生成数据表。

3）znck.spring.control.\*：实现逻辑处理的功能，接收前端传来的数据进行逻辑处理。

4）znck.spring.service.\*：接口文件，包的设计的目的是方便查看操作数据库用了哪些函数，一目了然。

5）znck.spring.serverceImp.\*:实现对应znck.spring.service.\*的方法，对数据进行操作。

6）znck.bean.weixin.message：封装微信的消息类型，包括按钮、文本、图片、语音、视频、地理位置、链接消息。

7）SystemSet.properties: 配置文件。

### 5.1.1微信请求校验程序设计与实现

为了实现请求校验流程，首先通过花生壳获取公网资源，填写URL和token，填写之后，需对请求者进行身份校验，以确保信息来自服务器，微信服务器将发送的GET请求填写到URL上：请求校验流程如下：

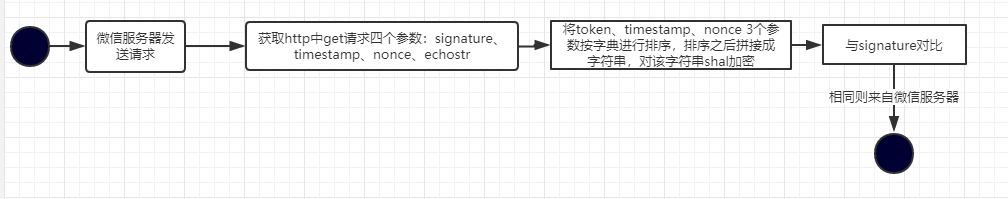


图5-2 请求校验流程

请求校验程序实现如下：



图5-3 请求校验程序

### 5.1.2一键关注公众号设计与实现

要扫码关注，首先就要生成二维码。二维码本质上就是存储数据的图片介质，其中的数据可以是 URL 也可以是文本等。通过二维码扫码工具就能扫出其中的数据。申请测试号之后，填写好测试号信息，接口配置信息配置成功之后，会自动生成测试号二维码，token与 URL 存储在该二维码中，微信扫码之后，会自动跳转到该 URL，用户确认关注之后，关注成功。

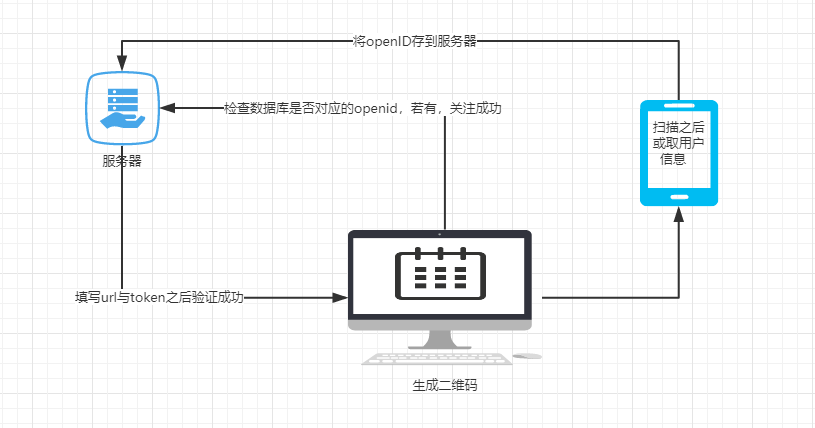


图5-4 扫描关注公众号

### 5.1.3 公众号消息的的发送与回复

微信服务器与公众账号之间以xml格式进行数据交换，通过解析与封装xml格式信息实现交互，按照微信说明文档先封装请求消息类型、事件类型与相应消息类型。

1）发送请求：当用户关注公众号事件发生之后，该请求会发送到微信服务器，微信服务器封装成xml数据向公众号服务器发送post请求，逻辑控制层从request中获取输入流，使用Dom4j读取输入流并且解析xml格式的请求消息；



图5-5 接收微信服务器请求程序

2）公众账号服务器接收信息：首先对接收的信息进行判断，之后对信息进行处理，返回xml格式进行响应，对响应信息进行了分装，使用XStream库文件实现java对象与xml之间的转换，当xml文件中包含“<”或者“&”时，需使用转义字符才能转换为java对象。当用户关注公众号时，扫码成功之后，会回复欢迎关注公众号。

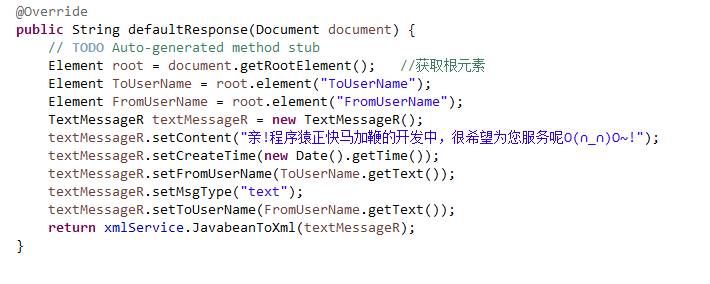


图5-6 接收微信服务器请求程序

### 5.1.4 自定义微信菜单模块

通过自定义菜单接口，实现对菜单的创建、修改、查询，方便用户自定义菜单栏，自定义菜单项通常被看做按钮，按钮的类型有click（点击事件）与view（访问网页）两种，但是有着不同的属性，项目中只用到view，view的属性有type、name、url三种属性，使用json格式进行交互。

1）获取接口访问权限

读取配置文件中AccessTokenUrl获取凭证接口请求地址，发起HTTPS GET请求调用该接口，在使用HttpUrlConnection类的getInputStream（）方法得到输入流，读取输入流中的内容，得到含有凭证信息的JSON字符串，通过JSON-lib开发包把json字符串转换为java对象，获取接口访问凭证access\_token和凭证存在时间expires\_in。

1. 创建菜单

自定义菜单是一个3x5结构的，即菜单最多只能有二级，一级菜单最多只能有3个，每个一级菜单下最多可以有5个二级菜单项。创建菜单实质是向<https://api.weixin.qq.com/cgi-bin/menu/create?access_token=ACCESS_TOKEN>地址以POST方式提交一个JSON格式的菜单字符串。

用户通过点击创建菜单，进入菜单添加页面，输入信息之后，按照微信开发文档说明把菜单信息封装成json数据格式提交，服务端中通过封装菜单结构，项目菜单只涉及到访问网页的类型的菜单，从配置文件中读取 MenuCreatUrl = <https://api.weixin.qq.com/cgi-bin/menu/create?access_token=，通过获取ACCESS_TOKEN构建菜单接口地址URL，>使用url.openConnection()建立连接，以post方式向输出流中写入前端传送过来的菜单结构，从输入流（菜单创建结果）读取相应内容并保存到buffer中。



图5-7创建菜单实现程序

3） 删除、查询菜单

删除与查询菜单与创建菜单基本步骤相同，需修改接口地址与提交方式。

用户发送查询与删除菜单的请求，逻辑层对请求进行处理，从配置文件中读取查询或者删除菜单的接口，查询接口为

MenuQueryUrl=<https://api.weixin.qq.com/cgi-bin/menu/get?access_token=ACCESS_TOKEN>，删除菜单接口为：

MenuDeleteUrl = https://api.weixin.qq.com/cgi-bin/menu/delete?access\_token= ACCESS\_TOKEN，以HTTPS GET方式调用该接口，查询成功时返回json数据格式的菜单数据，删除成功时返回true，否则返回false。

### 5.1.5 用户查找停车场模块

该功能主要实现根据用户的需求加载停车场，显示停车场详细信息，并且进行相关的操作。

1）根据用户位置加载停车场：当用户进入公众号点击菜单栏上停车场时，进入停车场页面，加载百度地图，首先根据用户的位置，用ajax实时请求百度地图实现实时定位，前端获取用户位置之后，将发送位置到ParkLotsControl.java控制层，在数据库中查询正在运行且剩余车位不为0的停车场，加载数据库之后，通过获取数据库中停车场的经纬度，利用百度地图距离计算方法，计算出距离定位直线距离最近的20个停车场，返回控制层，加载到百度地图上进行显示。

2）搜索停车场：该模块的实现与根据用户的定位加载停车场实现一样，根据用户的定位更改为根据搜索的位置加载，该功能待优化，搜索停车场时时根据用户所在城市进行加载，当要搜索另外一个城市的停车场时，需要输入完整的地址才能加载停车场，否则可能加载的位置不准确。

3）停车场信息：在地图上已加载用户需要的停车场，利用百度地图的api，对地图上的点添加事件，点击停车场之后在地图上可以看到部分停车场信息，当点击停车场信息之后，通过发送停车场的id查询该停车场的信息，跳转到新的页面加载更多信息。

4）停车场导航：该功能主要为用户提供停车场路径导航，限于百度地图api的限制，在网页中无法实时获取gps，百度地图没有提供实时导航的接口，根据用户位置和停车场的位置之后，调用百度地图路劲规划接口提供路径导航，根据用户的需要提供时间最少、路径最短的路径规划。

### 5.1.6 用户管理停车场模块

用户管理停车场，主要是对停车场进行添加、删除、修改、查看的操作，用户必须在登录之后才能使用该功能。

1）添加停车场：用户在登录之后，通过session保存用户信息，点击界面的我要加盟之后，输入信息，如果没有输入必要信息将提示无法提交信息，输入停车场详细信息之后，前端用ajax提交信息信息到服务器，控制层将停车场详细注入停车场类，业务逻辑通过session获取用户信息，调用parkinglot.sertUser(user)方法设置用户，调用dao层addParklotDao.save(parkinglot)方法添加到数据库中。

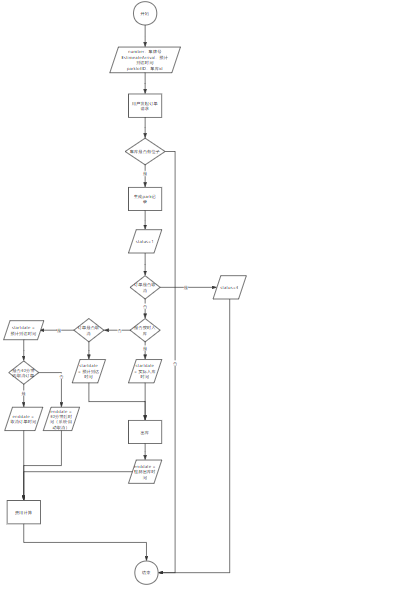
2）查看停车场：用户登录之后，通过session里边保存的信息对数据库里的停车场进行查询，返回List<ParkingLot>，前端通过jstl对List<ParkingLot>进行遍历，获取到所有停车场。

3）删除停车场：前端传入要删除停车场的id，设置要删除的ParkingLot对象，调用addParklotDao.delete(parkinglot)删除该对象，删除成功返回true，否则返回false。

4）修改停车场：用户进入要修改停车场信息的界面，修改信息界面通过jstl返回整个停车场对象，通过对值进行修改，提交之后，业务逻辑层调用dao层addParklotDao.update(parkinglot)方法实现修改。

### 5.1.7 用户预约停车场模块

根据用户的位置，用户根据需要了解停车场信息之后进行停车场预定，预定停车场流程图如下：



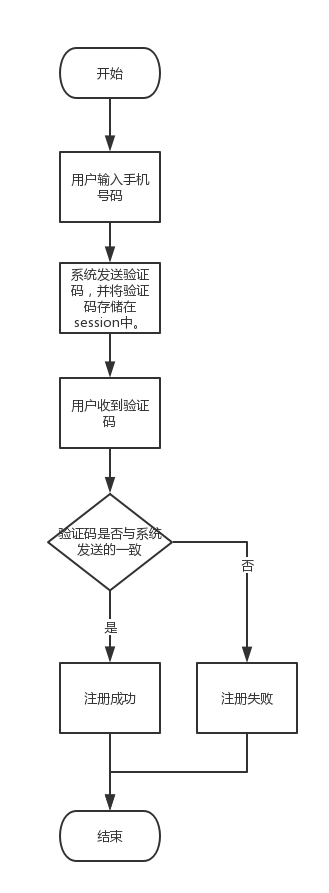
预定停车场：用户在地图上点击停车场查看信息之后，点击预定，利用百度地图获取驾车到该停车场的时间，用户输入车牌号之后，提交驾车时间，时间格式为yyyy-MM-dd hh:mm:ss，停车场id和车牌号码到控制层，控制层调用parkingLotService.parkOder(parklotId, park, httpsession)方法，在业务逻辑层调用parkDao.selectUnfinishedPark(park.getCar\_number())方法获取用户的订单，如果用户有订单则不能进行预定，如果为null，调用parkDao.save(park)方法把用户的信息保存到park表中，生成park记录，调用parkingLotDao.surplusParkSub(parklotID)停车场数量减一，表示预定成功。

在预定之后进行取消：用户点击取消订单之后，当取消得时间少于预计到达时间时，调用park.setStatus(ParkStatus.odered)方法，把状态status设置为3，调用parkDao.dalete(park)方法删除订单记录。当取消时间超过预计到达时间30分钟，则生成一条park记录，结束时间取消订单时间，收费标准为预计到达时间与取消时间这段的收费。当用户按时进入停车场，出库时间为实际检测到的时间，按照收费标准进行结算。第二种情况不能按时入口且30分钟内不取消订单，则预计到达时间为调用百度地图获取的时间预计到达时间来结算。

### 5.1.8 登录注册模块

该模块主要是针对用户登录与注册，通过手机号码方式进行登录与注册，方便用户。

1）用户注册：用户在界面点击完善信息之后，跳转到注册界面，通过使用“阿里大鱼”短信的api来实现验证码获取，当用户点击获取验证码时，发送手机号码到控制器，调用Verification\_code\_send(String phone, HttpSession httpsession)方法，生成四位验证码，把发送验证码的手机号码的与生成的验证码保存到session里边，并且设置在session里边存在时间为60秒，防止用户恶意注册，浪费系统性能，同时也避免造成资源浪费。流程图如下。



1. 用户登录

用户登录时输入手机号与密码，通过ajax发送数据到控制器，业务层调用PhoneUserLogin(User user)方法，查询数据库，如果数据库中有相同的密码与账号，返回第一个用户记录的信息。

## 5.3 本章小结

本章主要阐述了微信公众号应用于设计的具体细节，按照模块依次介绍了各个模块的详细设计与实现。

# 第六章 测试与成果展示

## 6.1 测试环境

项目开发完成之后进行测试，测试时需要一部手机，项目系统环境采用windows 10、jdk1.8、tomcat9.0、mysql5.6

## 6.2 功能性测试

本应用采用黑盒测试的方式进行，按照输入是否满足预期输出判断结果，下边是测试结果：

表6-2 测试用例与测试结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模块 | 前提条件 | 操作 | 预期结果 | 测试结果 |
| 关注公众号 |  | 扫描二维码，点击关注公众号 | 关注公众号成功 | 通过 |
| 查看公众账号菜单 | 管理员登录 | 点击查看菜单 | 显示所有菜单 | 通过 |
| 修改公众账号菜单 | 管理员登录 | 点击修改菜单，进入修改页面，输入信息 | 修改成功 | 通过 |
| 删除公众账号菜单 | 管理员登录 | 点击删除菜单，提示是否删除成功 | 删除成功，删除成功之后有提示 | 通过 |
| 注册 | 输入手机号且获取验证码，用户输入密码 | 在公众号界面点击完善信息，输入手机号之后，点击获取验证码，输入密码之后点击注册 | 获取验证码成功，输入信息之后跳转到登录界面 | 通过 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 登录 |  | 输入手机号与密码 | 点击登录之后跳转到停车场界面 | 通过 |
| 用户添加停车场 | 用户登录 | 点击添加停车场之后进入添加停车场页面，输入停车场信息点击添加 | 添加成功之后，跳转到用户个人所有停车场 | 通过 |
| 用户修改停车场 | 用户登录 | 点击要修改的停车场，输入修改信息，点击提交 | 修改成功之后，跳转到用户个人所有停车场 | 通过 |
| 用户删除停车场 | 用户登录 | 点击要删除的停车场 | 删除成功之后，跳转到用户个人所有停车场 | 通过 |
| 车主查看周围停车场 | 用户登录 |  | 该项操作开启或关闭 | 通过 |
| 查看停车场信息 | 用户登录 |  |  | 通过 |
| 预定停车场 | 用户登录 |  |  | 通过 |
| 取消预定 | 用户登录 |  |  | 通过 |
| 查看预定 | 用户登录 |  |  |  |

针对测试过程中发现的问题进行修改与优化后，本应用最终通过测试。

## 6.4 成果展示

下面将按用户使用流程来展示本应用主要功能的运行界面：

### 6.4.1 用户关注公众号

|  |  |
| --- | --- |
|  |  |
| 图6-1二维码 | 图6-2公众号主界面 |

### 6.4.2 用户登录注册

|  |  |
| --- | --- |
|  |  |
| 图6-3用户登录 | 图6-4用户注册 |
|  |  |

### 6.4.2 停车场管理

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 图6-3加载停车场 | 图6-4查看停车场信息 | 图6-5查看具体停车场信息 |
|  |  |  |
| 图6-6搜索地址 | 图6-7导航路线 | 图6-8路径导航 |
|  |  |  |
| 图6-9选择路线 |  | 图6-10预约停车场 |

### 6.4.3 用户管理停车场

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | |  | |
| 图6-15发布邀约 | 图6-16发布成功 | | 图6-17详细信息 | |
|  |  |  | |
| 图6-21邀约未完成 |  | 图6-23邀约点赞 | |

### 6.4.5 提醒与上课静音

|  |  |
| --- | --- |
|  |  |
| 图6-24上课静音开启 |  |
|  |  |
| 图6-27更改提醒时间 |  |

### 6.4.6 设置

## 6.5 本章小结

本章首先介绍了Android课表应用的测试情况，进而展示了课表规划结伴应用主要功能模块的实际运行结果。应用运行情况良好，界面设计简洁，操作简单，功能相对完善，能够满足用户需求。

# 第七章 总结

本文首先分析了当前国内外停车场发展现状，根据当前科技发展，提出了开发公众号开发垮平台移植的方法，同时针对现有停车场的优缺点进行改进。本文详细介绍了应用需求分析，以及技术实现方法，在实现过程中利用流程图，UML图帮助分析与理解。

本应用网页是响应式设计，实现了关注公众号获取用户信息、微信菜单修改、加载停车场、预定停车场、停车场导航等功能，实现了车位共享，极大的帮助用户已经广大车主，当仍然存在以下问题：

1）定位存在问题，定位不够准确，虽然可以通过ajax实时请求位置，但是定位偏差较大，由于地图开发限于api限制，无法使用实时导航功能，如果配合微信小程序开发将极大提高程序的应用性；

2） 搜索停车场带优化，需要开发新的搜搜方法，解决停车场名字相同位置不同的问题；

3）暂时缺少推送停车场信息给用户的功能；

4）还未能与用户实现互动，用户输入信息后可以根据要求自动回复相关信息。

本应用是工程项目，在这期间，学习了java框架，百度地图开发，微信开发技术，极大的锻炼了我的动手能力和自学能力，在一定程度上帮助了我的成长，这是一个好的开始，使我觉得它不会结束，我希望这次毕业设计会是一个相对圆满的句号

# 参考文献

[1] 单东林．锋利的jQuery［M］．张晓菲、魏然等．2012年7月第2版．北京：人民邮电出版社，2012：172．

[2] 李刚．疯狂的XML讲义［M］．2011年8月第一版．北京：电子工业出版社，2011：2．

[3] 明日科技．Java web从入门到精通［M］．2012年9月第1版．北京：清华大学出版社，2012：376．

[4] League of ExtraorDinary Developers．React 引领未来的用户界面开发框架［M］．版本2015年5月第一版．北京：电子 工业出版社，2015：5．

[5] Nicholas S.Williams．Java Web高级编程［M］．2015年6月第1版．北京：清华大学出版社，2015：107-110．

[6] 卢瀚．Java web开发实战1200例［M］.王春斌．2011年1月第1版．北京：清华大学出版社，2011：325．

[7] 蔺华．Web程序设计与架构［M］.王玉清．2011年6月第1版．北京：电子工业出版社，2011：489-490．

[8] David Flanagan．JavaScript权威指南［M］．2012年4月第1版．北京：机械工业出版社，2012：169-170．

[9] 王飞飞．Mysql从入门到精通［M］.崔洋、贺亚茹．2014年4月第2版．三河市：中国铁道出版社，2014：243．

[10] Craig Walls．Spring实战［M］．2016年4月第1版．北京：人民邮电出版社，2014：145-147．

# 声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得四川大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

本学位论文成果是本人在四川大学读书期间在导师指导下取得的，论文成果归四川大学所有，特此声明。

学位论文作者（签名）

论文指导教师（签名）

2017年 月 日

# 致 谢

转眼即逝的大学时光，当毕业论文提交的那一刻，大学也画上了一个句号，在这四年时光里，有过拼搏，有过奋斗，也有过怀疑人生。大学是连接社会的桥梁，在这里不仅学会了做人的道理，认识一群有追求的人，更多的是开阔了我的视野，为我生活打开了一扇窗。感谢母校的培养，感谢每一位老师的指导与教诲，也感谢在深夜里边一起写代码、一起拼搏、帮助我成长的所有同学。

论文的编写也接近尾声，非常感谢指导老师潘薇老师对我的指导，从选题到结束，潘老师一直给我提建议，对我项目帮助很大，同时也对我以后的生活工作给与鼓励，让我深受启发。借此机会，我谨向潘老师表示衷心的感谢！

# 附录1

这是啥？