常州信息职业技术学院

学生毕业设计（论文）报告

系 别： 软件与大数据学院

专 业： 移动互联应用技术

班 号： 互联171

学 生 姓 名： 程志祥

学 生 学 号： 17081630125

设计(论文)题目： 共享单车

指 导 教 师： 罗大晖

设 计 地 点： 常州信息职业技术学院

起 迄 日 期： 2019.06.01~2019.11.15

# 目录

[目录 I](#_Toc25007375)

[摘要 1](#_Toc25007376)

[Abstract 2](#_Toc25007377)

[第一章 绪论 3](#_Toc25007378)

[1.1 开发背景与意义 3](#_Toc25007379)

[1.2 开发可行性 4](#_Toc25007380)

[1.3 开发成本分析 4](#_Toc25007381)

[第二章 相关技术研究 5](#_Toc25007382)

[2.1 服务端开发技术 5](#_Toc25007383)

[2.1.1 Tomcat 5](#_Toc25007384)

[2.1.2 SSM框架 5](#_Toc25007385)

[2.1.3 HTTP 5](#_Toc25007386)

[2.1.4 数据库技术 6](#_Toc25007387)

[2.1.5 Restful架构 6](#_Toc25007388)

[2.2 前端技术 6](#_Toc25007389)

[2.2.1 JavaScipt 6](#_Toc25007390)

[2.2.2 JQuery 6](#_Toc25007391)

[2.2.3 Bootstrap 7](#_Toc25007392)

[2.3 阿里云Lot 7](#_Toc25007393)

[2.4 共享单车 7](#_Toc25007394)

[2.5 移动端 7](#_Toc25007395)

[2.5.1 Android 7](#_Toc25007396)

[第三章 系统需求 8](#_Toc25007397)

[3.1 功能性需求 8](#_Toc25007398)

[3.1.1 服务端 8](#_Toc25007399)

[3.1.2 共享单车服务器 9](#_Toc25007400)

[3.1.3 前端 9](#_Toc25007401)

[3.1.4 移动端 10](#_Toc25007402)

[3.2 非功能性需求 10](#_Toc25007403)

[3.2.1 性能需求 10](#_Toc25007404)

[3.2.2 安全性需求 10](#_Toc25007405)

[3.2.3 实用性需求 11](#_Toc25007406)

[第四章 系统设计与实现 12](#_Toc25007407)

[4.1 网站结构设计 12](#_Toc25007408)

[4.2 模块功能 12](#_Toc25007409)

[4.2.1 登录与注册 13](#_Toc25007410)

[4.2.2 实时数据展示 13](#_Toc25007411)

[4.2.3 用户操作功能实现 13](#_Toc25007412)

[4.2.4 分页查询实现 14](#_Toc25007413)

[4.2.5 消息系统实现 14](#_Toc25007414)

[4.3 前端界面设计 14](#_Toc25007415)

[4.3.1 登录注册 14](#_Toc25007416)

[4.3.2 主页数据展示 16](#_Toc25007417)

[4.3.3 用户操作页 16](#_Toc25007418)

[4.4 数据库设计 16](#_Toc25007419)

[4.4.1 数据表具体设计 17](#_Toc25007420)

[第五章 系统部署与测试 20](#_Toc25007421)

[5.1 项目部署上云服务器 20](#_Toc25007422)

[5.2 测试意义 20](#_Toc25007423)

[5.3 系统的测试环境配置 20](#_Toc25007424)

[5.4 测试记录 21](#_Toc25007425)

[5.4.1 登录与注册模块测试 21](#_Toc25007426)

[5.4.2 增删改查功能测试 22](#_Toc25007427)

[5.4.3 消息发布接收测试 22](#_Toc25007428)

[5.5 测试总结 23](#_Toc25007429)

[总结 24](#_Toc25007430)

[致谢 25](#_Toc25007431)

[参考文献 26](#_Toc25007432)

摘要

在大力倡导低碳出行，可持续发展的当今时代，共享经济开始在各个领域表现出其强有力的生命力。一种以互联网与传统的自行车相互结合而产生的新式交通出行方式——共享单车在这样的环境下油运而生。该业务在发展初期便得到了城市环境，交通部门的积极响应，更是以极快的速度占领了中低端人群对日常出行的需求市场。

共享经济的出现催生了大量的共享业务，共享单车相关业务蓬勃发展，人们对于包括共享单车在内的各种共享产品的需求也变得越来越多。

本文所研究的主要内容便是基于SSM框架的共享单车后台管理系统。经过对同类产品的功能分析以及对用户的使用调研后，对系统功能进行了详细的需求分析，最后采用了模块化开发的方式来进行实现。在实现上，本次研究以Eclipse EE和Tomcat作为开发工具，以JavaWeb开发技术作为主要开发技术，数据存储库则采用当前主流使用关系型数据库。

在本次系统的研究中，将整个系统分为数据展示，用户管理，设备管理，积分管理四个模块，依次实现对产品使用情况的展示，用户的增删操作，设备的查询修改，以及积分商城后台商品的状态管理等功能。

**关键字：共享单车；SSM框架；JSP；MySQL**

Abstract

In the current era of vigorously advocating low-carbon travel and sustainable development, the sharing economy has begun to show its strong vitality in various fields.

A new mode of transportation generated by the combination of Internet and traditional bicycles -- Shared bikes were born in such an environment.

In the early stage of its development, the business has received positive response from the urban environment and the transportation department, which has occupied the market of daily travel demand of the middle and low-end people at a very fast speed.

The emergence of the sharing economy has spawned a large number of sharing businesses, and bike-sharing related businesses are booming. People's demand for various sharing products including bike-sharing is also becoming more and more.

The main content of this paper is the background management system of Shared bikes based on SSM framework.

After analyzing the function of the same kind of products and investigating the use of users, the system function is analyzed in detail, and finally modular development is adopted to realize it.

In terms of implementation, this study takes Eclipse EE and Tomcat as the development tools, takes JavaWeb development technology as the main development technology, and the data repository adopts the current mainstream relational database.

In the study of this system, the whole system is divided into four modules: data display, user management, equipment management and integral management, which successively realize the display of product usage, user's operation of adding and deleting, inquiry and modification of equipment, as well as the status management of goods in the background of integral mall and other functions.

**[Keyword]:**Share bikes; SSM framework; JSPTS; MySQL

第一章 **绪论**

1.1 **开发背景与意义**

以方便人们日常出行，解决出行最后一段路为主要目的而诞生互联网网约型单车，共享单车以使用便利，无污染，使用价格低廉等优点得到了大众的欢迎和支持。随着互联网的不断发展，共享单车已经是治理通行拥堵的一个重要环节，共享单车的出现在帮助交通管理部门缓解拥堵问题的同时也促进了与自行车相关的产业链的良性发展，可谓是“一车多能”。

现如今，已经有越来越多的互联网移动手机用户加入到共享型经济的市场中来，根据艾媒咨询提供的数据可知，虽然从2017年以后共享单车系列产品的用户规模增长的速度开始逐渐缓慢下来，但总体市场仍保持着不断扩大的趋势。

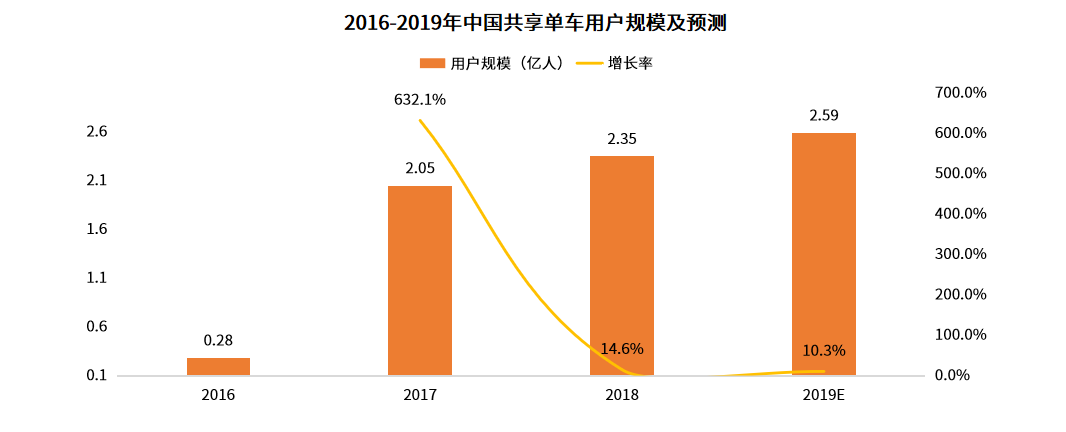


图1-1数据来源艾媒咨询

市场的扩大随着带来的便是对共享单车产品需求的增大，这直接导致了大量共享单车产品涌入市场，而产品的增多则很容易引起产品的管理瘫痪。以知名共享单车品牌摩拜单车举例：在2017年时就曾发生过大面积瘫痪的事件，引起了广大用户的强烈不满，该次共享单车瘫痪事故在媒体上备受关注，人们由此开始注意到共享单车系列产品的稳定性问题。

像摩拜瘫痪这样的事件即使是技术发展至今，我们依旧难以保证是否还会发生，所以本次研究的目的便是要在已有技术的基础上开发一款实用的具有高效功能的共享单车后台管理系统，以求能够帮助使用共享单车系统的管理员能够及时了解当前产品状态和使用情况，并及时对出现的问题做出解决和回应。

1.2 **开发可行性**

在本次研究中，所选用的工具和环境均已提前配置完成，系统开发所使用到的相关技术也大多数都是在大学期间已经学习过的基础知识，利用这些技术已经足以实现本系统所需的功能。

同时，由于共享单车管理系统已经是一个相对成熟的开发项目，在网络上已经有很多类似的项目，如果遇到无法处理的技术问题，可上网查找解决方案。

所以，经过严谨的分析和数据调查，本次开发是一定可行的。

1.3 **开发成本分析**

共享单车作为一个几乎成熟的共享类型业务，几乎已经不存在所谓的技术壁垒，所以作为一个已经逐渐趋向于成熟的产品，开发一款共享单车管理系统所需要的成本相较于共享单车出现初期已经大大的降低，但仍然有必要的成本付出，这主要体现在开发人员的知识技术和时间付出上。

知识技术：开发一款功能完备，具有实际使用价值的共享单车后台管理系统需要开发人员具有一定的前端开发能力，同时还需要熟练掌握数据库和后台服务器的开发设计，熟悉MQTT，HTTP等常见通信协议的连接原理，熟练掌握Servlet，JSP等开发技术同时还需要有一定的系统测试能力。

时间成本：开发一款完整的共享单车管理系统通常需要花费软件开发人员一到三个月的开发时间，因为这其中不仅包括了前期的可行性讨论，用户需求分析，还有中期的资源整合，个性化功能代码的编写和UI等个性化功能的设计以及后期的产品测试，测试后的上线运营等诸多事宜。

除此之外，还有比如共享单车实物测试，服务器租用等成本损耗。

第二章 相关技术研究

2.1 **服务端开发技术**

2.1.1 Tomcat

由于JavaWeb项目只有在Web应用服务器上才能正常使用服务，所以在本系统选取了Tomcat作为测试用服务器。Tomcat相比起其他的诸如jetty，Resin等服务器，使用起来更加稳定，也更容易上手，虽然Tomcat的效率没有Apache高，但由于Tomcat是Servlet容器，完全支持Servlet和Jsp请求，所以十分适合本系统开发。

2.1.2 SSM框架

为什么要选择框架开发？所谓框架，其实就是一种项目的半成品。在框架中有很多中功能组件可供开发者使用并用来完善自己的项目。框架可以帮助开发者完成项目底层的基础工作，不禁省去了代码编写的大量重复工作量，维护起来也更加直接和方便。

在本次研究中，所选用的框架便是SSM框架，相较于企业开发放弃的SSH框架，SSM框架也更加的简洁低量，已经逐渐成为企业级Web应用开发的主要选择。

2.1.3 HTTP

HTTP协议规定了当我们使用者在发起http请求时，所请求的数据包中都包含了什么样的数据，同时数据按照什么样的先后顺序存放在对应的数据包中。

HTTP协议的架构基于B/S，协议以用户本身使用的浏览器作为协议客户端并通过URL链接的方式向S端即服务器端发送用户请求。

HTTP协议使用与其他协议相比更加容易，服务器的程序大小也相对较小，由于HTTP在向服务器进行服务请求时，只需要传送请求的方法和请求相关联的路径，所以HTTP协议的通信速度相较于其他通信协议速度更加快速也更加灵活。

由于HTTP协议在每次建立连接后都只会对一个用户请求进行处理，所以一旦用户的请求被完成，HTTP协议会立即自行终止与用户的连接，以此来减少传输所花费的时间，同时又由于该协议是无状态的，在进行消息传输前并不需要进行信息处理，所以还有应答快速的特点。

2.1.4 数据库技术

（1）MySQL

作为一个开放型关系数据库，MySQL与其他数据库相比，MySQL独特的多层架构设计模式更适应于应对各种不同的开发场景。

（2）Druid连接池

在本次数据库技术开发中，使用的连接组件是Druid连接池。Druid连接池作为当前最流行的数据连接池，与DBCP和C3P0等数据库连接池相比，Druid在功能和性能等方面更有优势。

同时由于本次研究的主题是共享单车系统项目开发，在数据库的连接使用次数上比其他类型的项目要多的多，Druid在SQL执行后产生的日志记录相比起其他连接池要少得多，对数据库的产生的负担要小得多。

2.1.5 Restful架构

对MVC架构进行改进后诞生的一种新的架构，由已经定义好的接口对应不同的服务。该架构将服务器分成前后端两个部分，前端向开发者提供无模型视图，后端为前端提供需要接口。选择Resful架构进行项目开发，能够帮助开发团队 更方便的进行并行开发。

2.2 前端技术

2.2.1 JavaScipt

主流web脚本语言，能够对网页的响应式行为进行自定义编程，增加网页的交互性。

2.2.2 JQuery

基于JavaScript的优秀框架，上手容易，学习门槛低。JQuery能够利用强大的选择器取得DOM元素，然后对应这些DOM元素的赋取值，增删改等操作进行封装成相应的方法，同时还有开发者提供了诸如ajax，isArray等工具类方法。

2.2.3 Bootstrap

基于HTML，CSS和JS开发的优秀前端框架，独特的栅格式布局设计是的Bootstrap完全适用于手机，平板等多种设备的响应式网页的前端实现，其核心文件包括bootstrap.min.css样式文件，Jquery文件以及min.js等脚本文件。

2.3 阿里云Lot

相较于普通的服务器，阿里云服务器采用分布式存储，支持云镜像，能够同时存储多份数据并且能够确保存储数据的安全性，具有极好的稳定性和安全性，能够帮助开发者更加方便的管理海量设备，这一点对于共享单车项目是极其重要的。  
 相比其他的物联网接入，阿里云的设备接入通信能力更加安全也更加稳固。用户可以在云端上传大量的数据，随后云端可以自主调用API，实现消息下行传递到任何一个设备。实现远程设备控制。  
运用Aliyun IoT有以下优势：  
 设备连接：可以连接百万级别单车数量，使硬件软件一体化，并且连接稳定牢靠。  
 高并发通信推送：具有可以支撑海量消息推送接受的通信能力，方便随后项目产品的日后升级更新和多方面扩展的问题。

2.4 共享单车

共享单车最重要的技术点主要在它自身所携带的智能锁上，在智能锁中包括了定位，通讯等重要功能模块。

目前在市场上投放的单车中所使用的单车锁一共分为两种。第一种是通过车身上张贴的设备二维码来进行云端即时解锁，而另一种的开锁方式类似商场中的临时保险柜。但不同的是需要通过手机扫描二维码来惊醒获取密码在对机械密码锁进行输入密码手动开锁。

2.5 移动端

2.5.1 Android

Android中包括了四大常用组件和丰富而强大的系统控件，而它所具有的的多层架构特点也极大的方便了开发者进行多层对应开发，在代码编写上也更加规范，编写简单，可维护性强。

第三章 系统需求

3.1 功能性需求

需求分析需要建立在对系统总体结构的基础上，本次研究项目的系统结构图如下：

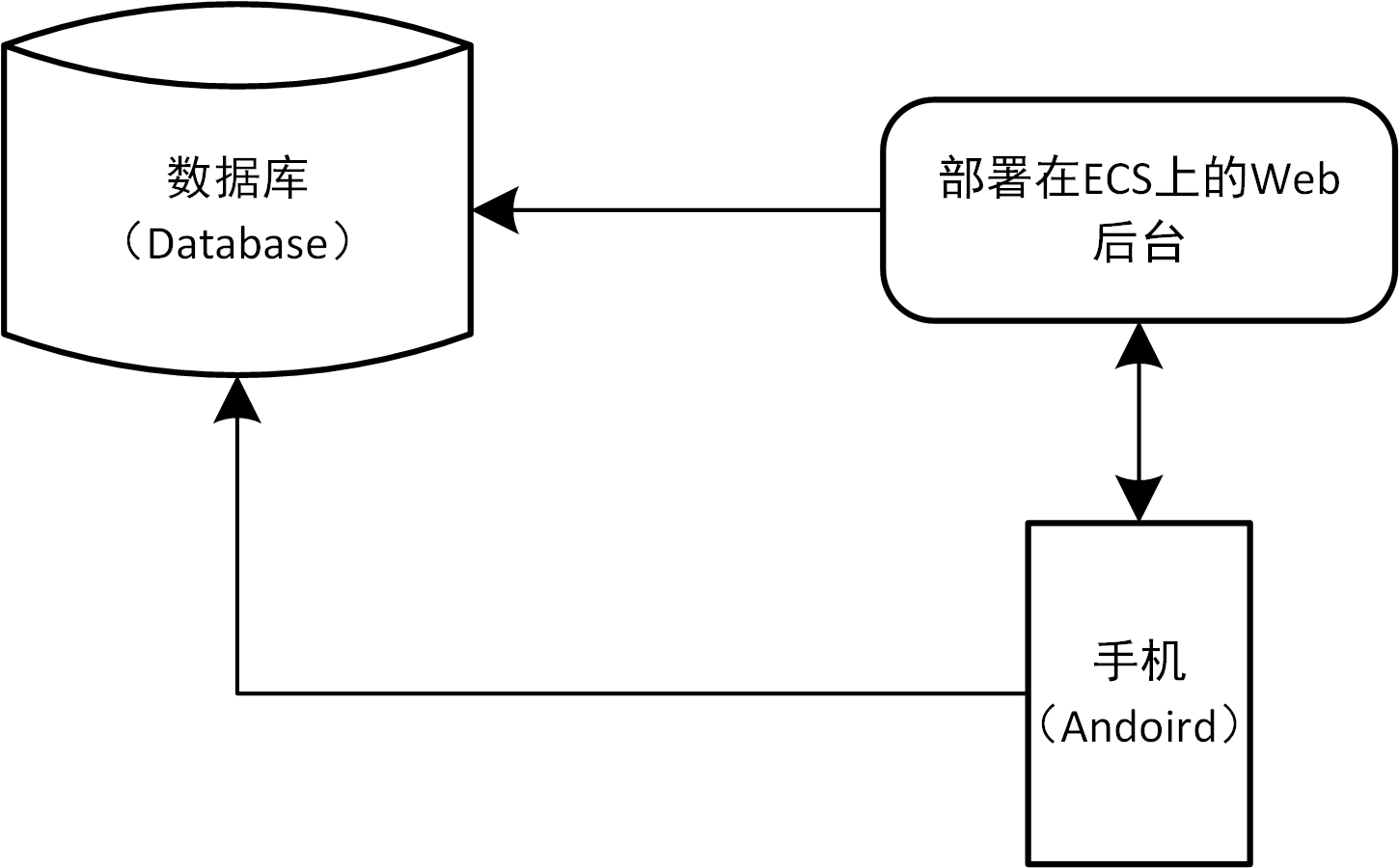


图3-1数据流程图

3.1.1 服务端

在本次研究中，将系统服务端分为登录，注册，实时数据展示，用户管理，设备管理，积分商城管理以及消息接受发布七个模块。

（一）管理员登录模块：要求系统能够根据用户输入的账号进行数据库中存储的账号数据进行比对判断管理员输入的账号是否能够通过安全验证 。如果管理员账号验证成功则在几秒后进行首页传送，如验证不匹配则弹出相应的错误弹窗并要求管理员进行重新验证 登录。

（二）管理员注册模块:要求管理员能在无账号情况下，按照账号注册规范注册新账号，系统将注册好的新账号保存在数据库中，管理员即可通过自己之前已经注册过的账号登录系统。

（三）实时数据展示：该功能模块位于系统主页，要求系统能够根据数据库内数据的变化以数字和图表的形式在页面上实时展示当前注册用户总数量，当前已租出设备数量，单日租出平均时长，当前租金收入总数等信息。

（四）用户管理模块：对数据库中已经存在的用户进行状态查看，以及实现用户增加修改等基础功能。

（五）设备管理模块： 功能要求管理员能够在设备管理页面进行设备的状态查看，并且能够对设备进行添加删除，对已有设备进行状态修改等操作。

（六）积分山城管理模块：功能要求管理员能够在积分商城管理页面对移动端的积分兑换商品进行修改，上架以及下架等操作。

（七）消息接受发布模块：要求管理员能够接受到来自用户向服务端发送的设备反馈信息，并且能够在web后台向已登录的客户端发布实时消息通知。

3.1.2 共享单车服务器

单车服务器作为整个单车系统的调控中心，其目的在于方便管理员对系统进行管理和为用户移动端提供响应服务。当移动端向服务器发送HTTP请求时，服务器向移动端返回一段JSON数据，移动端对服务器返回的数据再进行解析，解析完成后通过手机发送控制开锁信息，最后成功达到关锁功能。

3.1.3 前端

前端是管理员对系统最直观的操作面，要求前端页面能够简洁明了的展示出系统所具备的功能选项。

本系统主要有七个页面，分别是登录，注册，主页，用户操作页，设备管理页，积分商城页和消息系统页。

与系统功能一一对应分析后，可知以下结果：

要求登录页有输入管理员账号的文本框，同时还要有登录注册按钮用于跳转其他页面，要求系统在管理员在输入账号密码后能够利用JS对文本内容进行简单的非空判定和格式判定。

主页作为数据展示页，并不需要管理员对该页面进行直接操作，但要求系统能够根据数据库中的数据进行及时更新并将数据通过折线图的形式直观的表现出来。

用户操作页和设备管理页需要实现管理员对用户账户或单车设备的搜索和增加删除操作，同时还要对数据库返回的表数据进行简单的分页处理，所以该页面上应该存在搜索框，分页按钮，增删按钮等简单控件。

积分商城页需要能够展示当前库存商品的状态，同时应该具备商品状态修改和信息修改等功能组件，具体以按钮控件进行实现。

消息系统页，能够利用WebScoket与用户之间接受和发送消息，要求存在消息接收框和文本编辑框等控件。

3.1.4 移动端

共享单车APP作为用户主要使用的移动端系统，需要有定位单车，查看周围上线单车，扫码开始骑行等基本功能。

用户在初次使用APP时需要注册新账户，新注册的账户通过Restful架构将账号打包成gson数据传递给服务端，服务端接受到移动端传递过来的gson数据后，将gson解析成账号数据保存到服务端的数据库中。

3.2 非功能性需求

3.2.1 性能需求

系统的性能需求主要体现在响应速度和延时时间上。由于本系统的开发对象为共享单车项目，在对时间的要求较高。

响应速度：当管理员密码输入正确后，要在一秒内进行正确页面跳转。管理员对在前端页面对后台进行操作时，要求能做到实时进行数据表的更新。

用户使用移动端对共享单车进行扫码后，要求服务器能在三秒能做出请求响应，并向设备返回开锁信息。

延时时间，当用户向服务端反馈消息时，要求整个过程的时间不得超过五秒，以便管理员对用户的信息进行及时的处理。

3.2.2 安全性需求

管理员输入的账户信息需要封装起来，不能使用明文传输。同时要求系统能够对账户进行库比对，必须要正确的账户才能进入系统。

用户在操作移动端，要求对用户的账户进行核对，同时服务器要对用户提出的开锁请求进行核对，保证开锁请求对应的是正确的单车。

同时由于共享单车项目的用户量巨大，要求数据库能对数据进行并行处理，及时处理日志信息防止过载崩溃。

3.2.3 实用性需求

要求系统中所有的消息请求都经过了规范定义，方便服务器端对消息请求进行集中管理，管理员能够快速的发现消息请求中的错误。

代码要按照SSM规范进行编写，方便开发者后期的维护等。

第四章 系统设计与实现

4.1 网站结构设计

由于本系统的使用者主要是共享单车后台系统管理员，为了使系统的功能使用起来更加直接便捷，本系统采用的是网站结构化开发，在层次上要求做到清晰明了，内容和相关主题要衔接明显，在功能的分配上也要做到各个网页分工明显。

网站的整体结构设计如图：

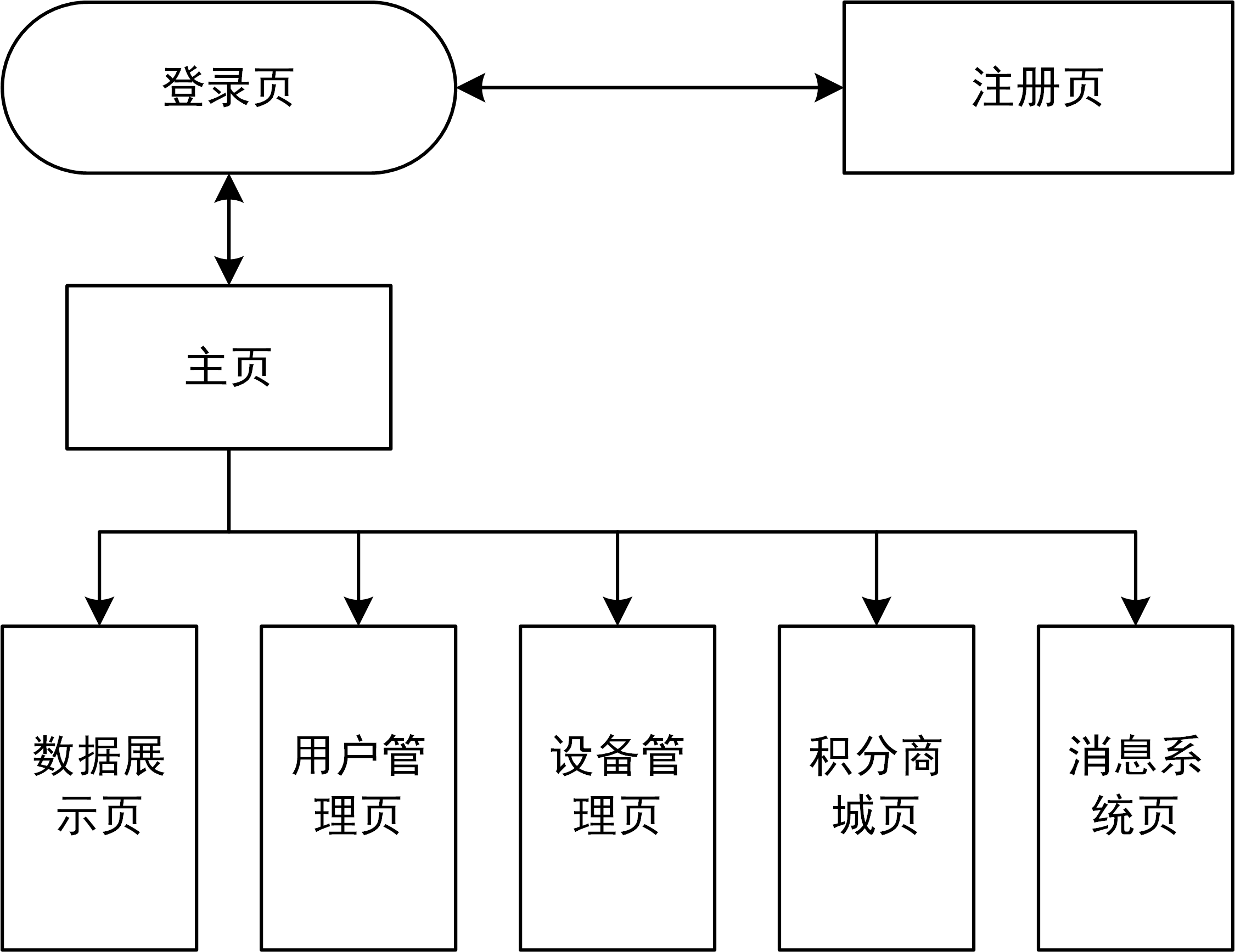


图4-1 网站结构设计图

4.2 模块功能

由于本系统的主要面相于管理员使用，主要的功能在于用户修改，设备修改，租金修改，积分修改以及客户端消息的发布接收，具体的模块结构如下图：

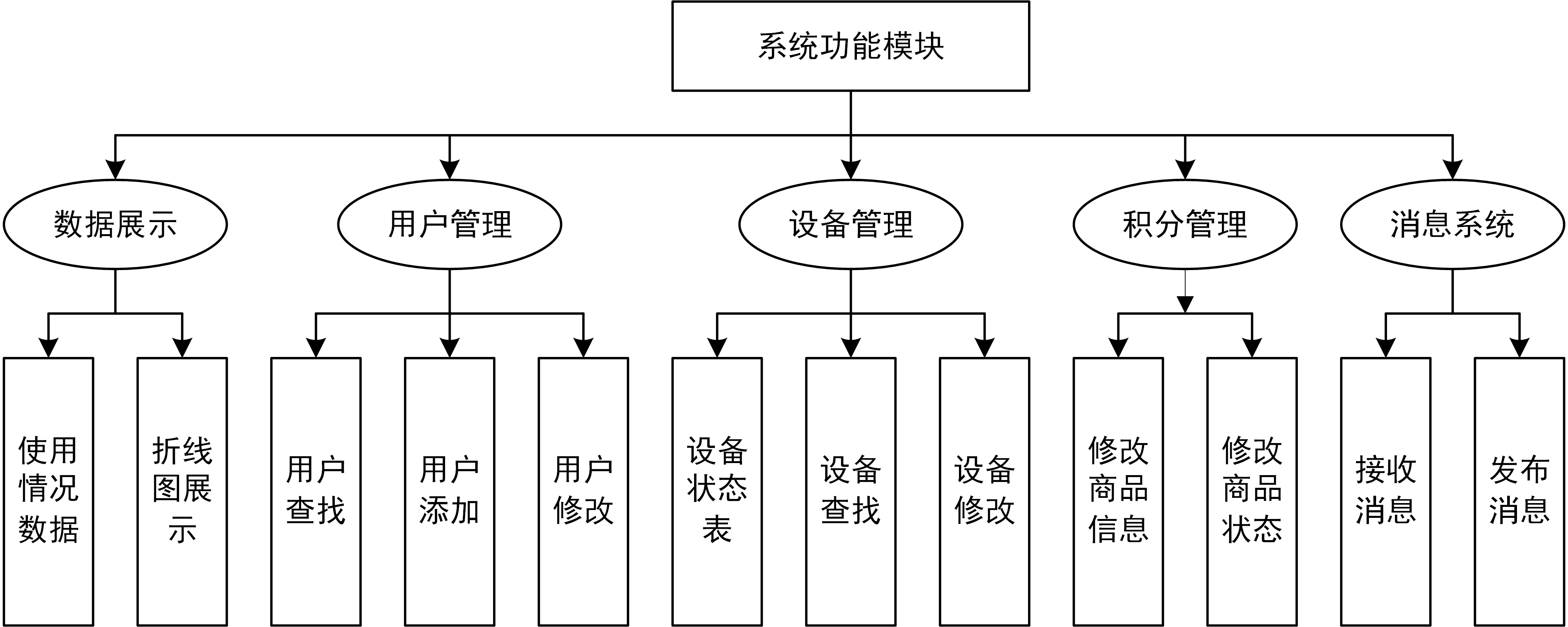


图4-2 模块功能设计图

4.2.1 登录与注册

由于本系统主要面向于共享单车系统管理员使用，所以需要使用管理员账户进行注册，在注册成功后，再次登录系统，会根据数据库表中已经提前注册好的管理员信息数据进行搜索比对，比对成功才能登入系统。

登录流程如下：

图片包含 物体

描述已自动生成

图4-3 登录流程图

4.2.2 实时数据展示

为了实现在主页通过图表实时展现数据的功能。本系统使用了INSPINIA中已经写好的js代码为主Echarts工具为辅的方式进行首页数据页开发。Echarts为开发者提供可供二次开发的十分美观的图形界面，使用起来非常简单，同时由于Echarts基于HTML5.，兼容性也足够好。

在具体开发中，通过创建Bean对象+Dao+Servlet对象的方式将数据库中的数据在jsp页面中实时展示出来。该功能能够让管理员实时观测到当前的数据库基本信息，在一定程度上方便管理员对共享单车后台数据进行监测。

4.2.3 用户操作功能实现

由于用户的增删等修改操作和设备的修改操作开发原理相同，所以这三个模块便以对用户的增删改查实现作为示例。

在本系统中，为了遵循Web应用程序的框架开发规范，采用Controller控制层+Dao数据接口层+Entity实体+impl数据接口实现各功能后台模块，Crud语句以及jsp前端页面进行实现操作页面。

4.2.4 分页查询实现

本系统实现分页查询功能用到了Dbutils工具，该工具中封装了JDBC的大量应用代码，极大简化了开发者在进行JDBC编码时需要编写的代码量，能够加快程序开发的速度，方便维护。

首先在数据持久层中实现总记录条数和满足分页条件记录的查询功能 ，然后在逻辑层中调用到中总的和每页记录的查询，通过总的记录数得到全部的页数，最后完成PageBean类（存放分页信息的类）的设置，并将其返回给实现类。

4.2.5 消息系统实现

订阅发布模式最普遍的应用场景就是消息系统的设计。消息系统中主要包括了三个组件信息发送者（Pub），信息接收者(Sub)以及核心组件——消息代理（Broker）。

本系统消息订阅发布模块的架构图设计如下：

图片包含 物体

描述已自动生成

图4-4 消息系统架构

本系统使用的是WebSocket来实现基于http协议的发布/订阅系统。仿照Mqtt协议模式来设计用户发布中心和订阅中心。

4.3 前端界面设计

在本次开发中，选取了Bootstrap的延伸产品INSPINIA（响应型管理主题）作为WEB主题。INSPINIA是在Bootstrap的基础进行模仿开发的，是一个完全式响应管理仪表板模板的引导型框架。而且INSPINIA和bootstrap一样包含了大量可重用的UI组件，可应用在各种类型的web程序上。

选取INSPINIA作为本系统的前端开发主题，能够极大的提升界面开发速度以及界面美观程度，剩下大量的时间成本。

4.3.1 登录注册

登录注册模块作为整个系统的初始模块，具体实现方式便是通过BootStrap样式文件进行输入框和注册按钮等控件的实现。

登录界面效果展示如下：

图片包含 屏幕截图

描述已自动生成

图4-5 登录界面

注册界面效果展示：

图片包含 屏幕截图

描述已自动生成

图4-6 注册页面

4.3.2 主页数据展示

主页作为后台数据展示页，并没有需要管理员进行操作的地方，所以该页面只有数据展示模块，该主页部分组件展示效果图如下：

图片包含 屏幕截图

描述已自动生成

图4-7 主页实时数据展示

4.3.3 用户操作页

用户操作页，在本页面中，同时还有选择单次展示用户数据条数的功能，可在十到一百的比例上进行选择。

由于用户操作页和设备管理页前端功能相似，不做赘述。

图片包含 屏幕截图

描述已自动生成

图4-8 用户操作功能图

4.4 数据库设计

在本次研究中，由于涉及到用户，设备，积分等系列信息，所以数据库中应该存在管理员账号储存表，用户综合信息表，单车信息表，使用情况表，积分统计表，五个表。

图片包含 屏幕截图

描述已自动生成

图4-9 数据库E-R模型图

4.4.1 数据表具体设计

管理员账号储存表：用户储存已有或新注册的管理员账号，包含了管理员注册的名称，管理员的邮箱地址以及管理登陆的密码。

表4-1 管理员账号储存表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名** | **字段类型** | **字段长度** | **主/外键** | **是否为空** | **描述** |
| adName | Varchar | 100 | 主键 | 否 | 管理员名称 |
| adEmail | Varchar | 100 |  | 否 | 管理员邮箱 |
| adPassword | Varchar | 50 |  | 否 | 管理员密码 |

用户信息表：用于储存编号，设备名以及用户设备所注册的手机号。

表4-2 用户信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名** | **字段类型** | **字段长度** | **主/外键** | **是否为空** | **描述** |
| id | int | 11 | 主键 | 否 | 编号 |
| sn | Varchar | 128 |  | 否 | 设备名 |
| phone | Varchar | 128 |  | 否 | 用户手机号 |

设备信息表：包含用户识别号，单车识别号，单车上锁关锁时间，骑行开始时间点，位置坐标以及该数据产生时间点等信息。

表4-3 设备信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名** | **字段类型** | **字段长度** | **主/外键** | **是否为空** | **描述** |
| id | int | 11 | 主键 | 否 | 编号 |
| userId | int | 11 |  | 否 | 用户识别号 |
| bikeId | int | 11 |  | 否 | 单车识别号 |
| locked | tinyint | 1 |  | 否 | 上锁状态 |
| riding | tinyint | 1 |  | 否 | 骑行状态 |
| x | int | 16 |  | 否 | 坐标 |
| y | int | 16 |  | 否 | 坐标 |
| generateTime | datetime | 0 |  | 否 | 生成时间 |

使用情况表：包含用户识别,单车识别号，骑行开始时间，骑行终止时间，租金，令牌信息。

表4-4 使用情况表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名** | **字段类型** | **字段长度** | **主/外键** | **是否为空** | **描述** |
| id | int | 11 | 主键 | 否 | 编号 |
| userId | int | 11 |  | 否 | 用户识别号 |
| bikeId | int | 11 |  | 否 | 单车识别号 |
| startTime | datetime | 0 |  | 否 | 开始时间 |
| endTime | datetime | 0 |  |  | 结束时间 |
| fee | int | 5 |  | 否 | 租金 |
| uuid | varchar | 128 |  | 否 | 令牌 |

积分统计表：保存字段的编号，用户的使用识别号以及积分账户余额信息。

表4-5 积分统计表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名** | **字段类型** | **字段长度** | **主/外键** | **是否为空** | **描述** |
| id | int | 11 | 主键 | 否 | 编号 |
| userId | int | 11 |  | 否 | 用户识别号 |
| ib | int | 11 |  | 否 | 积分余额 |

第五章 系统部署与测试

5.1 项目部署上云服务器

1）注册账号后，在阿里云选择系统所需要的工具。为了节省资金成本，本系统所选择的服务器类型是学生版Windows Server 2008 ECS云服务器，虽然性能不是很高，但足以运载该系统的所有功能。

2）在用户控制台中打开自己的ECS实例并通过实例接入自己的远程桌面。

3）给自己的服务器配置对应运行环境，如JDK，Tomcat等，需要注意的是要和自己的Web程序开发环境相匹配。

4）将本地已经开发好的项目另导出成war包，并将生成的 war包导入到自己服务器下的Tomcat目录中存放运行工程的目录中，最后启动Tomcat。

5）通过公网IP地址访问服务器项目。

5.2 测试意义

系统测试是一个项目开发在结束前的一个十分重要的环节，有目的的测试和正确的测试方法更是一个项目必不可少的步骤。通过仔细的系统测试，能够帮助开发者发现在项目开发时被遗漏的错误和漏洞并及时做出项目修正，以求项目发布后产生错误的几率降到技术上的最低，避免维护时不必要的损失。

在测试时，我们需要对各个具体模块进行针对性的测试。

5.3 系统的测试环境配置

测试系统：Windows 10

CPU：Intel（R）Core（TM）i5-7300HQ CPU@2.50GHz

内存：8.00 GB

Web容器：Tomcat 9.0.7

网页浏览器：IE、Chrome

数据库：MySQL 8.0.12

数据库连接工具：Navicat Premium 12

5.4 测试记录

本次测试的主要测试模块主要集中在重要功能运行上，大致测试对象为登录注册，用户操作，设备操作以及消息发布接收模块，在对模块进行测试完毕后，将测试结果详细记录下来。

5.4.1 登录与注册模块测试

该模块作为系统的“门户”，必须要有较强的安全性。管理员在使用本系统前必须要经过数据库账号对比才能进入系统主页。所在，该模块必须要进行测试以确保不会出现低级错误。

该模块测试用例表如下所示：

表5-1 登录注册测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| **测试编号** | **测试对象** | **测试目的** | **测试结果** |
| 1 | 账号格式验证（标准：邮箱+数字密码） | 非规范账号是否能够登陆成功 | 测试成功，非规范账号登陆失败且弹出提示信息。 |
| 2 | 账号正确性验证 | 非已注册账号是否能够登陆成功 | 测试成功，非管理员账号无法登陆 |
| 3 | 注册新的账号 | 在注册页面输入新注册账号并提交注册后，数据库中是否保存了账号数据 | 测试成功，数据库中保存了新注册的账号信息 |
| 4 | 登录跳转 | 正确输入账号密码后是否正确跳转到主页 | 测试成功，登陆成功后成功跳转 |

5.4.2 增删改查功能测试

增删改查操作时本系统的核心操作功能之一，是系统管理员最直接操作数据的方式，必须要确保没有错误，所以要对着四个功能进行逐一测试。

该模块测试用例表如下：

表5-2 增删改查测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| **测试编号** | **测试对象** | **测试目的** | **测试结果** |
| 1 | 根据用户ID进行单一用户查找 | 测试是否查找成功或查找对象是否正确 | 测试成功，未发生错误 |
| 2 | 根据设备号查找设备 | 测试系统能够根据设备号查找对应的设备 | 测试成功，未发生错误 |
| 3 | 根据用户ID查找积分余额 | 测试系统能够根据用户ID查找该用户的积分余额 | 测试成功，未发生错误 |
| 4 | 删除选中用户 | 测试删除功能是能够成功删除对应用户 | 测试成功，未发生错误 |
| 5 | 删除选中设备 | 测试删除功能能够成功删除对应的设备 | 测试成功，未发生错误 |
| 6 | 修改用户信息 | 测试修改功能是否有错误 | 测试失败，原因是用户数据表中存在的外链没有与其他表的主键进行对应操作 |
| 7 | 修改设备信息 | 测试能否成功修改设备信息 | 测试成功，未发生错误 |

5.4.3 消息发布接收测试

仿照MQTT协议消息系统的HTTP消息发布/订阅系统作为本系统与单车用户直接沟通的渠道，在系统中承担着及时了解用户反馈和需求的责任。所以测试该功能模块能否正常使用是极其重要的。

该模块尽管功能比较简单，但是在后台实现上具有一定的复杂性，任何一个环节连接失败都可能导致整个功能完全失去作用。由于本模块使用的是WebScoket+HTTP协议进行开发的，所以测试目标应集中在消息接发上，以此来查看是成功连接到移动端。

该模块测试用例表如下：

表5-3 消息系统测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| **测试编号** | **测试对象** | **测试目的** | **测试结果** |
| 1 | 向手机一条“helloworld”消息 | 测试发布系统能否发布消息并被设备成功接收 | 测试失败，手机端未成功接收消息 |
| 2 | 再次向手机发送一条Hello信息 | 在前面测试中对错误进行了修改，再次测试发布系统能够成功发布消息并被设备接收 | 测试成功，系统BUG已排除，发布系统功能已经能正常使用 |
| 3 | 手机向Web端发送一条“hello”消息 | 测试订阅系统能够接收到手机端消息 | 测试成功，能够成功接收。 |

5.5 测试总结

一个系统的正常生命周期应该指的是整个系统开发的完整过程，活动分配以及任务的总体结构设计等。在本次研究中，单车系统的开发包括了：开发背景意义分析，需求统计，详细功能设计，功能详细实现，部署系统，总结测试等几大阶段。

测试作为开发系统的最后阶段，是整个环节中不可或缺的关键所在。

经过长时间且仔细的模块一对一化测试，我发现一个系统不仅要在代码后台上实现功能运转，还要能够将该功能实际的运用到产品中。只有一次次的测试，在测试中不断的体验修改，才能将系统不断的修缮完整。

学会对已经完成的产品进行测试，能够帮助开发者更快地找到自身的不足，也能帮助开发者后期更加方便的去维护自己所开发的系统，同时在此过程中也能让我们领悟程序设计的真谛。

总结

在本次共享单车Java后台管理项目中，我学习到了很多新的知识。首先是关于web项目的开发知识。

为了方便后续的代码管理或者新开发者的加入，一个正规的web项目在进行正式开发前都需要选择所对应的开发规则或规范原则，比如在本次研究中，我便选择了以SSM作为项目主要框架。SSM框架使用人群非常之多，已经形成了相对默认的开发规范。

同时，由于Web项目又有前后端之分，大多数项目都比较复杂，所以一个人完成这样的项目对于开发者来说具有极大的挑战性。在更多时候，我们还是需要有一个团队来进行Web项目的开发，分工合作能够极大的提升开发效率。

除了开发规则或者说开发风格的选择以外，选择一个好的适合开发者使用的Web开发工具也是极其重要的，比如在本次研究中，我所选择的工具便是Eclipse，尽管对于JavaWeb项目开发来说，Eclipse已经显得有些老旧了，在某些功能上也不够便捷，但是由于在校期间我所接触的时间最久的工具便是Eclipse，所以我还是选择了其作为我的开发工具，因为对工具足够熟悉对一个开发者来说也是极其重要的。

经过这次研究，我不仅深刻的了解了web项目开发的流程，同时也接触到了在校期间未曾接触过的诸如HTTP，MQTT等通信协议，也了解了阿里云等企业开发平台，这对于我今后走上程序开发道路有着极大的帮助。

致谢

三个多月的时间，终于将这个知识量广泛设计复杂的管理系统开发完成。在此过程中，我尽管遭遇了诸多的困难，但在这个过程中也得到了很多老师和好友的热心帮助。尤其是在协议的连接和SSM框架搭建时，迟迟无法完成这两个关键部分的技术实现。在关键时候，指导老师罗老师及时为我解决了困惑，并且不惜耗费自己的私人时间在工作以外的时间指导我们。

同时还要感谢我的合作搭档以及室友张翼飞，和我一起共同奋战，共同攻克整个项目的开发难题，在这个过程中我们相互协作，相互分析，在开发中获得了很多有关于系统开发的知识，共同克服技术难题的经历也使得我们的友谊愈发深厚。

最后，还要感谢其他毕业设计组的同学，愿意在如此紧张的时间下向我伸出援手。除此之外，还有很多帮助我的人，就不一个一个的致谢了，由衷的感谢大家给我带来的所有帮助！

参考文献

[1]刘义忠,张伟.基于SSM框架的后台管理系统设计与实现[J].软件导刊,2019,18(02):68-71.

[2]魏书寒.基于SSM框架的图书管理系统的设计与实现[J].工业控制计算机,2017,30(07):133-134.

[3]李洋.SSM框架在Web应用开发中的设计与实现[J].计算机技术与发展,2016,26(12):190-194.

[4]徐芸.共享单车的起源与发展[J].智库时代,2018(46):172-173.

[5]张玉杰，张海涛，张婷婷. 基于MQTT的物联网系统消息发布/订阅方法研究[J].电视技术，2017,41(Z3):83-87.

[6]张雪敏.浅议MVC设计模式在JavaWeb中的作用[J].科技风,2018(22):58-59.

[7]刘泓杉.关于软件测试在Web开发中的运用探讨[J].电脑迷,2018(12):68.

[8]贾文潇,邓俊杰.基于Java的Web开发技术浅析[J].电子测试,2016(08):65+86.

[9]何梦花,张超群,金建.基于模块化开发思想的小型图书信息管理系统的设计与实现[J].信息与电脑(理论版),2017(18):107-109.

[10]柳青松.JAVA语言的开发平台及J2EE编程技术[J].电子技术与软件工程,2017(17):56.