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181

学 生 姓 名： 蔡嘉辰

学 生 学 号： 18081630119

设计(论文)题目： 家庭记账本

指 导 教 师： 解志君

设 计 地 点： 常州信息职业技术学院

起 迄 日 期： 2020.06.01~2020.11.15

# 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

随着经济的不断发展，全国人均收入的提高，消费水平也在随之升高，而传统的手书记账本存在便携性，保存风险性的缺点，所以开发一款记账app就有着十分重要的意义。

本论文主要介绍了家庭记账本系统开发的整个过程。利用了移动软件开发技术相关原理，采用了oop（面向对象）的编程方法，其主要任务为后台数据库的建立、维护以及后端服务端的开发维护和安卓端的开发，使得家庭记账本自动化，高效化以及信息化。

该软件是数据库采用Mysql，后端基于Springboot，移动端采用Android系统作为载体的一款软件系统。经过大量对应用市场记账本的调研，对系统功能进行了较为详细的需求分析。在开发工具选择为IntelliJ IDEA，Mysql，Navicat，Android Studio，PostMan，以Web后端开发和Android开发作为主要的技术栈，数据库则为主流的开源数据库Mysql。

在系统研究中，后端系统分为用户管理、账单管理、资金管理三个模块，具体实现为用户的登录、用户的注册、用户密码的修改、资金的添加，修改，删除以及账单的添加修改删除。在Android方面分为账单页面、资金页面、饼状图统计显示页面和个人信息展示页面。

**关键字：家庭记账本；后端；Android；Springboot；Mysql**

Abstract

With the continuous development of economy, the improvement of national per capita income, the consumption level is also rising, and the traditional hand Secretary account book has the shortcomings of portability and risk preservation, so it is of great significance to develop an accounting app.

This paper mainly introduces the whole process of the system development. The main task is to establish and maintain the backstage database, the development and maintenance of the back-end server and the development and maintenance of the Android terminal, which makes the home bookkeeping Book automatic, efficient and information-based.

The software is a software system that uses MySQL as the database, springboot as the back-end, and android as the carrier for the mobile terminal. After a lot of research on the application market bookkeeping book, the system function is analyzed in detail. In the development tools, IntelliJ idea, mysql, Navicat, Android studio, postman are selected. Web back-end development and Android development are the main technology stacks, and MySQL is the mainstream open-source database.

In the system research, the back-end system is divided into three modules: user management, bill management and fund management. The specific implementation is user login, user registration, user password modification, fund addition, modification, deletion and bill addition, modification and deletion. In Android, it is divided into bill page, fund page, pie chart statistics display page and personal information display page.

**[Keyword]: Home bookkeeping; back end; Android; springboot; MySQL**

第一章 **绪论**

1.1 **开发背景与意义**

理财管理软件在如今强调资金管理，强调经济效益的当代社会越来越重要，而资金管理一直运用于企业中，能体现企业的资金状况，使用状况，以及对未来资金使用的预期情况，是一个企业长足发展的一个方面。若一个企业没有高效的资金管理系统，则将可能没用良好的发展前景。而在中国经济飞速发展，家庭收入与消费水平水涨船高的今天，资金管理软件不仅仅为企业所需要，在普通家庭也是必不可少的。而此次的家庭记账本则是针对家庭用户，使用这个家庭记账本，我们就可以记录或者查询任意时刻的收入与开销，我们仅需动动手指，点击屏幕，即可以看到具体的收支信息，为家庭资金的管理提供便捷的帮助，有利于资金的管控。

1.2 **开发可行性**

在本次的项目研究中，所选用的的工具为如今最受欢迎的IntelliJ IDEA，使用用户数十分广泛，框架方面选用的为Springboot，也是近期大热的后端框架，其核心思想为约定大于配置，帮助我们进行了很多配置，同时Springboot又集成了大量的第三方库配置，几乎可以开箱即用，简化了SSM框架的配置复杂性。

同时，家庭记账本并不是一个庞大的项目工程，且网络中已有众多记账本项目，若有相关问题无法自行处理，则可以利用百度，Google等搜索引擎与众多提问平台进行提问解决。

在经过权衡思考以及分析程序功能后，本次开发切实可行。

1.3 **开发成本分析**

为了让系统进行更好的开发，我对资金的家庭收支情况进行了了解，一般家庭的收入开销，与其他各种繁杂费用很复杂，很难清晰明了记录，而一般家庭也是采用手动记账，这样使得记账变得繁琐，采用记账软件可以进行无纸化记账，非常的简单快捷。

知识技术：本系统需要开发人员有一定的Android开发能力，同时需要掌握MySQL数据库和后端服务器程序的开发能力，同时需要bug的调试能力以及使用Linux的相关技术。

时间成本：家庭记账本系统并不复杂，一个月左右即可完成，其中包括了可行性研究，需求分析，项目框架的选择，ui库的获取以及开发完成后的测试。

金钱成本：服务器的成本。

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 实现了对 Servlet 和 JavaServer Page（JSP）的支持，并提供了作为 Web 服务器的一些特别支持功能。由于 Tomcat 本身也包含一个 HTTP 服务器，所以它可以作为一个单独的 Web 服务器，适合在此次记账本系统中使用。

2.1.2 SpringBoot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减少代码冗余,提高运行速度,便于维护,规范编程。

在本次项目中，所采用的框架是SpringBoot框架，SpringBoot是一个全新框架，使用来简化Spring的初始搭建和开发过程，使用了特定的方式来进行配置，让开发人员不在需要定义样板化的配置。此框架不需要配置xml，依赖于像MAVEN这样的构建系统。

相较于SSM框架，SpringBoot有如下几个优点：

1.减少了开发者的很多时间并提高了生产力。

2.避免了编写大量的样板代码，注释和XML配置。

3.优化了Spring框架的弊端。

4.可以快速进行整合第三方框架。

5.代码少了、配置文件少了、不需要对第三方框架烦恼了、项目精简了，对整个团队的开发及维护来说，更大的节约了成本。

6.使用Java开发基于Spring的应用程序非常容易。

2.1.3 HTTP (HyperText Transfer Protocol）

HTTP可以使浏览器更加高效，使网络传输减少。它不仅保证计算机正确快速地传输超文本文档，还确定传输文档中的哪一部分，以及哪部分内容首先显示(如文本先于图形)等

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，有如下特点：

1.可以处理拥有上千万条记录的大型数据；

2.支持常见的SQL语句规范；

3.可移植行高，安装简单小巧；

4.良好的运行效率，有丰富信息的网络支持；

5.调试、管理，优化简单（相对其他大型数据库）。

（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 是阿里巴巴开源平台上一个数据库连接池实现，结合了 C3P0、DBCP 等 DB 池的优点，同时加入了日志监控。

Druid 可以很好的监控 DB 池连接和 SQL 的执行情况，天生就是针对监控而生的 DB 连接池。

本次研究的主题是家庭记账本，需要大量使用数据库的CRUD，而增删改查难免会出现错误，所以我们只需进入Druid后台即可查看SQL代码的执行情况。

2.2 移动端

2.2.1 Android

如今Android系统作为市场拥有率最多的系统，Google及开源社区对其维护更新也使得其越来越安全易用，Android中包括了四大常用组件和丰富而强大的系统控件， 可以使得Java开发者轻松编写应用，并且拥有很高的稳定性与易维护性。

2.3 服务器

2.3.1 Linux

Linux作为服务器系统，具有比Windows更强大的安全性与稳定性，因为其是免费开源，因而Linux服务器在应用开发上更能节约成本，再者Linux服务器可以部署几乎所有的业务，使得开发更加自由。

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：

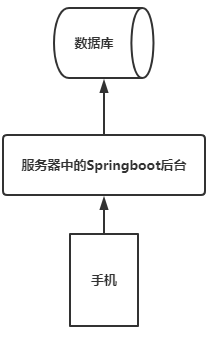


图3-1数据流程图

3.1.1 服务端

在本次设计中，将服务器端分为登录，注册，资金管理，账单管理四个模块。

（一）登录模块：要求记账本系统能够根据用户所键入的用户名进行数据库中的数据进行比对判断输入的账号是否存在以及是否匹配。如果账号验证成功则将从登录页面转入首页，如用户名与密码不匹配则弹出错误提示。

（二）注册模块：若用户无账号，按照账号注册用户名、密码、设计号、邮箱规范注册新账号，后台将新注册好的账号与密码保存在MySQL数据库中，用户即可以用刚注册的账号密码进行登录使用。

（三）资金管理：用户在Android端新建支付宝、微信、银行卡等资产后，服务器自动将相关信息保存在个人用户的数据字段中。

（四）账单管理：用户在Android端新建支付宝、微信、银行卡等资产后，服务器自动将相关信息保存在个人用户的数据字段中。

（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：

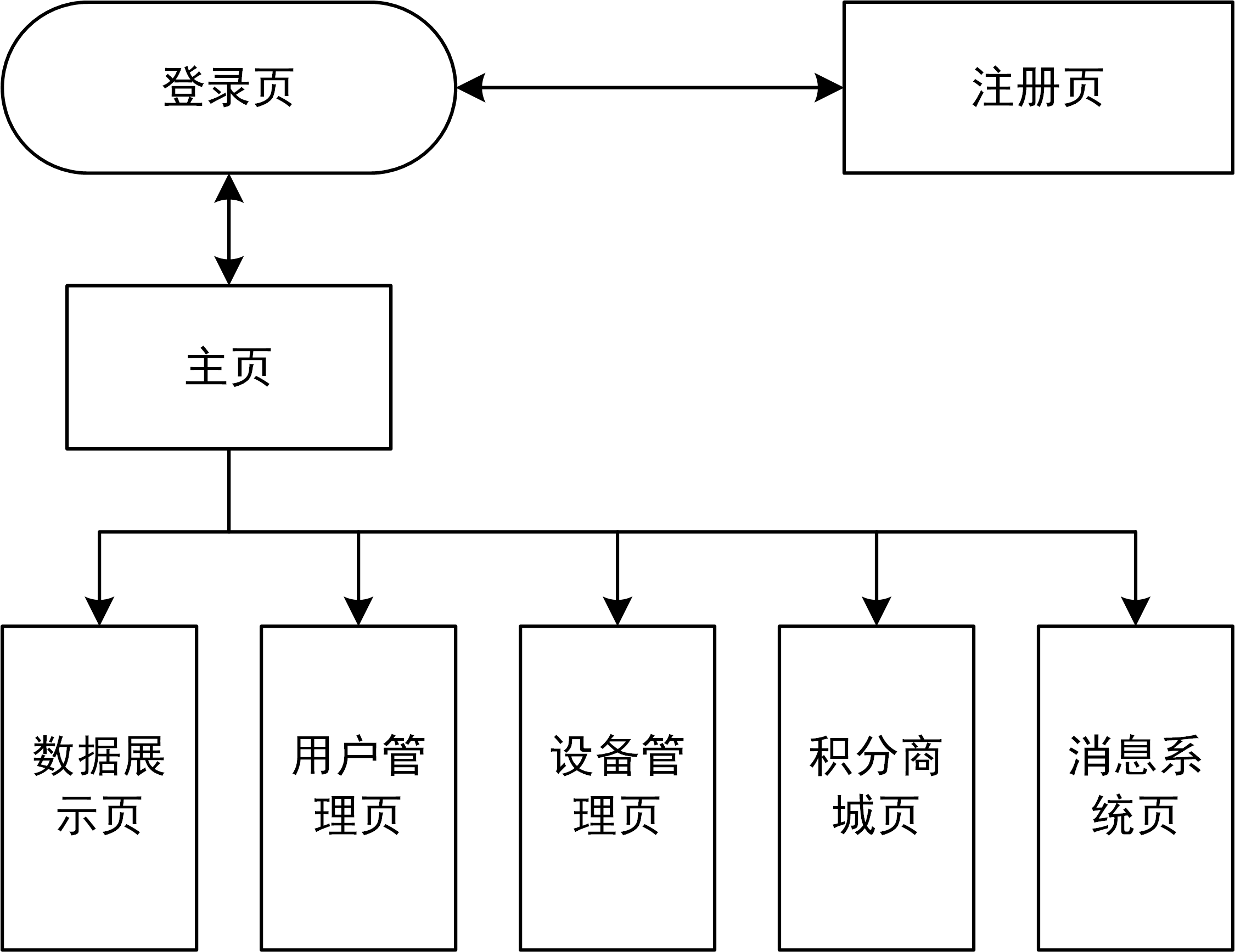


图4-1 网站结构设计图

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：

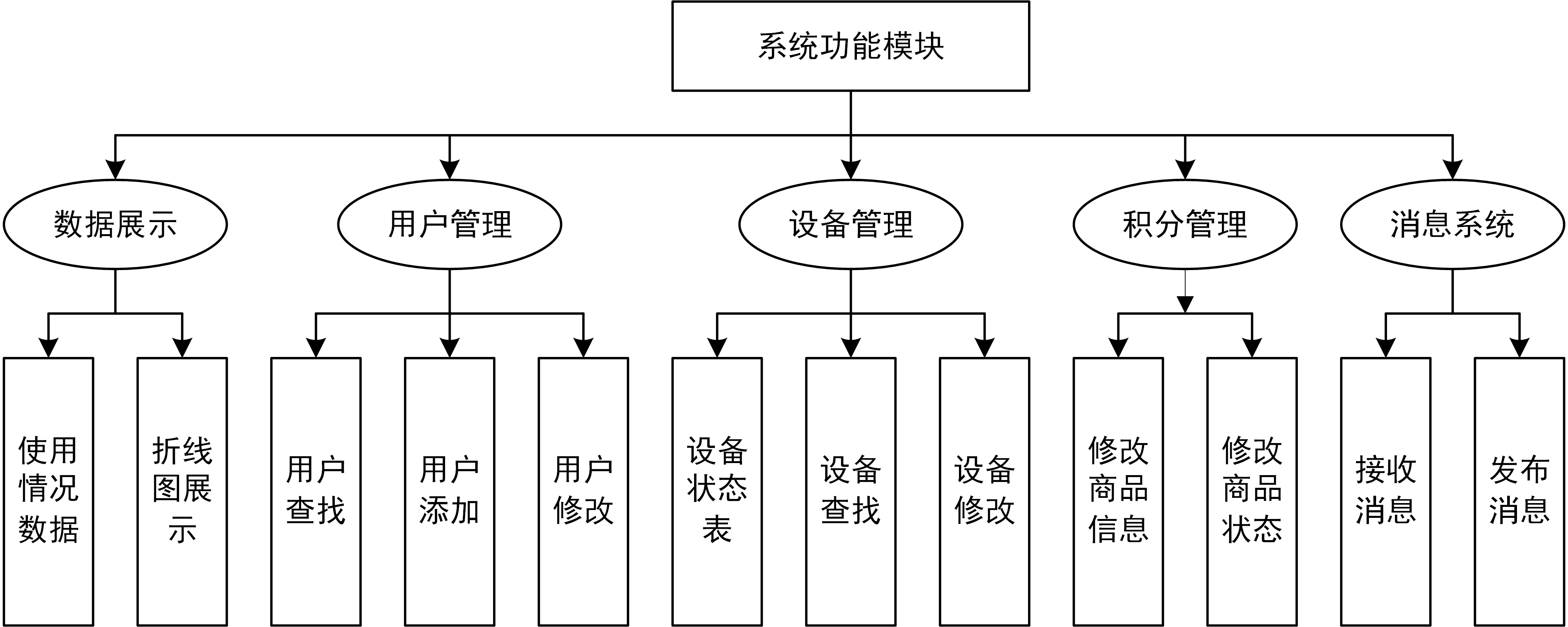


图4-2 模块功能设计图

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.