

河 南 科 技 大 学

毕 业 设 计（论 文）

题目 基于 SSM 的报价管理系统
的设计与实现

姓 名	<u>段刘振</u>
学 院	<u>软件学院</u>
专 业	<u>软件工程</u>
指导教师	<u>杨继松</u>

2020 年 5 月 24 日

基于 SSM 的报价管理系统的设计与实现

摘 要

伴随着计算机技术不断发展，越来越多的企业开始使用计算机来帮助解决实际中的问题。传统的报价管理大多采用 Excel 表和人工的方式进行，这种方式不仅数据的准确性较低而且容易出错。随着企业的不断发展扩大，数据量就会越来越大，工作量也以几何倍数增加，再人为的管理这些数据就会变得非常艰难。为了解决这一惨状，开发一个可以对企业各种数据进行管理的系统就显得非常重要。本文将设计一个报价管理系统，可以解决传统方式数据准确性低且容易出错的弊端，并且可以给产品准确合理的定价。

在此次项目的设计和实现过程中，使用到的操作系统是 Window10, 使用到的开发工具是 Eclipse, 运用 MVC 三层开发模式, 选用的是当下流行 SSM 框架+MySQL 数据库实现。系统中主要涉及到了以下的模块，分别是客户管理、产品管理、订单管理、报价管理等四个模块，并且对各个模块添加权限的判断。最后系统的功能模块基本上都得到了实现，并且具备一定的稳定性和安全性，系统功能将在下面的章节中进行介绍。

关键词：报价管理，Java Web，SSM 框架，MySQL 数据库，MVC 开发模式

DESIGN AND IMPLEMENTATION OF QUOTATION MANAGEMENT SYSTEM

ABSTRACT

With the continuous development of computer technology, more and more enterprises begin to use computers to help solve practical problems. The traditional quotation management mostly adopts excel table and manual method, which not only has low accuracy but also is easy to make mistakes. With the continuous development and expansion of enterprises, the amount of data will become larger and larger, and the workload will increase in geometric multiples. It will be very difficult to manage these data artificially. In order to solve this situation, it is very important to develop a system that can manage all kinds of data in the enterprise. This paper will design a quotation management system, which can solve the problem of low accuracy and easy to make mistakes in the traditional way of data, and can give accurate and reasonable pricing to products.

In the design and implementation process of this project, the operating system used is window10, the development tool used is eclipse, the MVC three-tier development mode is used, and the current popular SSM framework + MySQL database is selected for implementation. The system mainly involves the following modules, which are customer management, product management, order management, quotation management and other four modules, and the judgment of adding authority to each module. Finally, the functional modules of the system are basically realized, and have certain stability and security. The system functions will be introduced in the following chapters.

KEY WORDS: Quotation Management, Java Web, SSM Framework, MySQL DataBase, MVC Development Pattern

目 录

前 言	1
第 1 章 绪论	2
§ 1.1 研究课题的目的与背景	2
§ 1.2 国内外同类设计的概况	2
§ 1.3 本文主要研究的内容	3
第 2 章 系统分析	4
§ 2.1 需求分析	4
§ 2.2 可行性研究	4
第 3 章 系统设计	6
§ 3.1 系统总体结构与功能	6
§ 3.1.1 系统的设计目标	6
§ 3.1.2 系统的模块设计	6
§ 3.2 系统流程图	7
§ 3.3 数据库设计	7
§ 3.3.1 数据库设计	8
§ 3.3.2 数据库表设计	9
第 4 章 系统详细设计	13
§ 4.1 系统各模块功能设计与简介	13
§ 4.1.1 用户登录注册	13
§ 4.1.2 客户信息录入	14
§ 4.1.3 客户信息修改	15
§ 4.1.4 客户信息查看	16
§ 4.1.5 产品类型管理	17
§ 4.1.6 产品信息录入	17
§ 4.1.7 产品信息修改	18
§ 4.1.8 产品信息查看	19

§ 4.1.9 订单信息录入	19
§ 4.1.10 订单信息查看	20
§ 4.1.11 报价信息录入	20
§ 4.1.12 报价信息查看	21
第 5 章 系统测试	22
§ 5.1 系统的功能测试	22
§ 5.1.1 测试登录注册功能	22
§ 5.1.2 测试客户录入功能	23
§ 5.1.3 测试客户信息修改功能	24
§ 5.1.4 测试客户信息查看功能	25
§ 5.1.5 测试删除客户功能	25
§ 5.1.6 测试产品类型录入	26
§ 5.1.7 测试产品信息录入	27
§ 5.1.8 测试产品信息查看	27
§ 5.1.9 测试订单信息录入	28
§ 5.1.10 测试订单信息查看	28
§ 5.1.11 测试报价信息录入	29
§ 5.1.12 测试报价信息查看	29
§ 5.2 系统测试结论	30
结 论	31
参考文献	32
致 谢	34
附 录	35

前 言

从世界上第一台计算机诞生到今天，计算机的发展已有 74 年的历史，跟计算机相关的技术也是越来的越完善。到了 21 世纪互联网已经成为了各个行业发展必不可少的关键因素。计算机技术的飞速发展和在全球范围的普及应用正在给人类生活带来革命化变化。网络技术的发展也取得了巨大的成就，为程序的开发提供了很好的技术支持。基于这个背景，越来越多的企业开始建立各种网站和系统来简化工作。通过计算机技术的方式，客户可以根据自己的需要在系统上通过货比三家选择适合自己的产品，而企业则可以将自家的产品放在系统上，最后通过与分析产品的销售量以及客户的反馈来实现对产品价格的精准把控。

报价管理系统的主要目标是对企业的各种数据进行管理操作，减少企业管理客户信息、产品信息、订单信息、报价信息的工作量，同时可以通过参考对比，对产品进行精准的定价。报价管理系统使用当下流行的 Java 语言开发，开发使用的到的技术较为简单，使用到的是当下流行的 SSM（Spring、SpringMVC、Mybatis）框架，数据库使用到了 Mysql，前端使用到了 JavaScript 和 JQuery。

第1章 绪论

§1.1 研究课题的目的与背景

随着互联网技术应用的不断深入，因为时间、成本等多种阻碍因素的存在，以往传统的销售方式在如今激烈的商业竞争环境下已经显得力不从心。为了解决传统的销售方式在当今环境下的弊端，越来越多的企业开始建立属于自己的网站或者系统，他们将自家的产品放在上面进行报价销售，客户则可以通过浏览网站的方式寻找到自己需要的产品，最后货比三家，选择适合自己的产品，而企业则可以通过销售量来觉察到产品价格的定位是否太高或者太低，当然也可以浏览别的企业相同产品的定价，做后做出准确的定价。从而更多的销售产品。

产品报价系统根据以上需求，产品的管理、用户的管理、订单的管理、报价的管理。最后对于此系统的每一个模块，都通过了很多次全面的测试，各个功能稳定可靠。系统要实现产品报价的主要功能。产品报价系统作为新的报价管理的方式和传统的报价方式存在很大的差异，但就其方便快速性，必将取代传统的报价方式，应用到实际上肯定也能发挥出巨大的作用。

§1.2 国内外同类设计的概况

随着我国经济持续而又快速的发展，企业就必须要对市场变化进行快速的改变，竞价推广已经普及了大多数的企业，市场竞争也是越来越激烈，企业面对竞争如此激烈的市场。对于国内来说产品的报价还是使用传统的 Excel 和人工的方式已经不足以满足对报价的精准定位，更不利于对销售利润的控制。对于国外来说，因为人家的科技水平发达，比我们更早的进入了信息化时代。所以我们在信息化可谓是落后一大截，我们必须做出相应的转变。新的方式就是公司与客户的联系通过互联网的方式实现。企业如何给客户一个较为准确的报价，如果有这么一个平台公司把自家的产品以及价格放到上面，并且大家都可以看到，那么消费者就可以通过对比选择适合自己的产品，公司也能够更加准确的掌握价格的定位。

由此可观，虽然只是一个简单的报价管理，但是过程是比较复杂的。对于一个企业来说报价订单管理系统也是非常重要的。对企业来说，随着企业不断的发展壮大订单数量和客户量也会逐渐增多，面对如此多的数据量的时候，就无法对这些数据进行管理，也就无法对每个客户的报价情况做到精准掌握，那么在下次再和客户合作的时候就很难会报错价格，不是很清楚上次与该客户成交时的价格。如果是通过线下的口头报价的方式，到与客户成交的周期比较长,没有办法立即反应当时的一个情况,如果企业及时用了报价订单管理系统的话就会很好的解决这些问题。

§1.3 本文主要研究的内容

开发一个系统,首先需要选择一门开发语言,本系统选择的语言是 Java,使用到的数据库是 Mysql, 框架选用的则是当下流行的 SSM (Spring+SpringMVC+Mybatis), 本系统的主要功能有:

客户管理: 该模块主要是用来对客户的信息进行管理的, 主要包括名称、电话、联系地址、联系人等信息。

订单管理: 该模块主要是用来对客户下的订单进行管理的, 一条订单主要包括下单客户的名称、产品的名称、产品的数量、订单的金额等信息。

产品管理: 该模块主要是用来管理企业中产品的类别和产品信息, 包括产品的名称、产品的单位、产品的价格等信息。

报价管理: 该模块主要主要是用来录入报价信息的, 主要包括报价人、客户的名称、产品的名称、报价的时间等信息。

第2章 系统分析

§2.1 需求分析

对于销售公司来说，最大的需求就是能够对公司的各种信息进行管理。首先，因为公司有很多的产品种类，所以需要对产品进行归类划分并进行管理。当公司新推出一个系列的产品后，后台管理人员可以根据需要将新的产品类别录入到数据库中进行存储。对于客户而言，他关注的就是公司有哪些产品类别，所以客户需要可以对公司的产品类别进行查看。

同样还需要能够对产品进行管理操作，当公司推出新的产品后，需要将新的产品信息录入到数据库中，也要能够就产品的基本信息可以进行修改操作，也需要可以从数据库中将产品的信息删除掉以及根据查询条件查询产品的基本信息。对于客户来说，只能查询公司的产品。公司还需要对客户进行管理，主要包括将客户添加到数据库、对客户的信息进行修改、对客户信息删除操作及根据条件查询客户的信息等。对于报价信息和订单信息需要进行与客户信息管理类似的操作，其功能与客户信息管理基本相似。

§2.2 可行性研究

首先，在社会可行性方面，对于一家主要从事销售的公司，最重要的就是卖出更多的产品获取更多的利润。而在产品卖出前，有一个很重要的环节，那就是对产品进行报价。因为对于同一个产品来说，消费者通常会为了省钱选择价格较低的。通过该系统可以将公司的产品类型、产品、客户、报价信息以及订单信息进行高效可靠的管理，从而提高销售公司的竞争力以及销售能力。

其次，在技术可行性方面，为了提高开发的效率，本系统选择的是 **SSM** 框架，在本科学习期间已经基本掌握 **SSM** 框架以及 **JavaEE** 的开发模式，并且熟练使用 **Maven** 进行工程所需依赖包的管理，使用 **Mysql** 数据库保存系统数据。

最后，在开发环境可行性发面，使用的电脑环境是 Window10 系统。开发中使用到的工具有 Eclipse、WebStorm、Navicat、NotePad++等。

第3章 系统设计

§3.1 系统总体结构与功能

§3.1.1 系统的设计目标

经过调查与研究，预期的报价管理系统应达到以下目标：

（1）实现登录注册功能。

（2）系统能够稳定运行的同时也要安全可靠。

（3）报价管理系统是后台管理系统，对管理员来说，权限是最大的，拥有全部功能操作权限。对普通员工来说，只能录入和修改信息。对于客户，仅仅只能浏览和查询信息。

§3.1.2 系统的模块设计

用户登录注册：登录成功以后获取登录用户的权限，根据权限的不同则会显示不同的菜单导航栏。

客户管理模块：该模块只有管理员和操作员拥有权限，并且操作员只有录入新客户、修改客户信息的功能。

产品管理模块：该模块三个权限的用户都可以看到，但是普通用户只能根据查询条件浏览产品信息。

订单管理模块：该模块也只有管理员和操作员可以看到，同样操作员只有录入新订单、修改订单信息的功能。

报价管理模块：该模块三个权限的用户都可以看到，但是普通用户只能浏览报价信息。

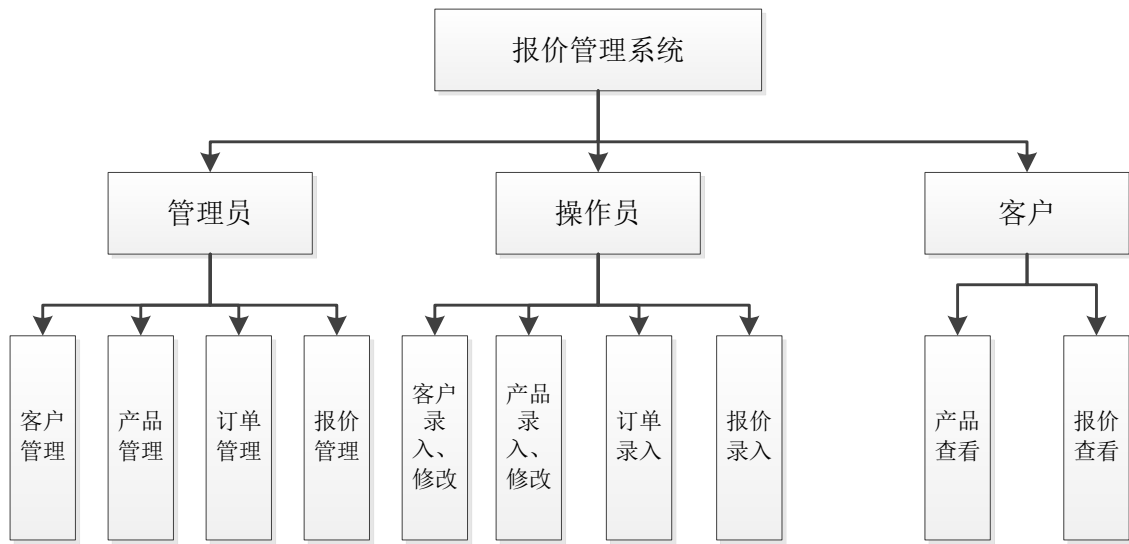


图 3-1 模块总体结构图

§3.2 系统流程图

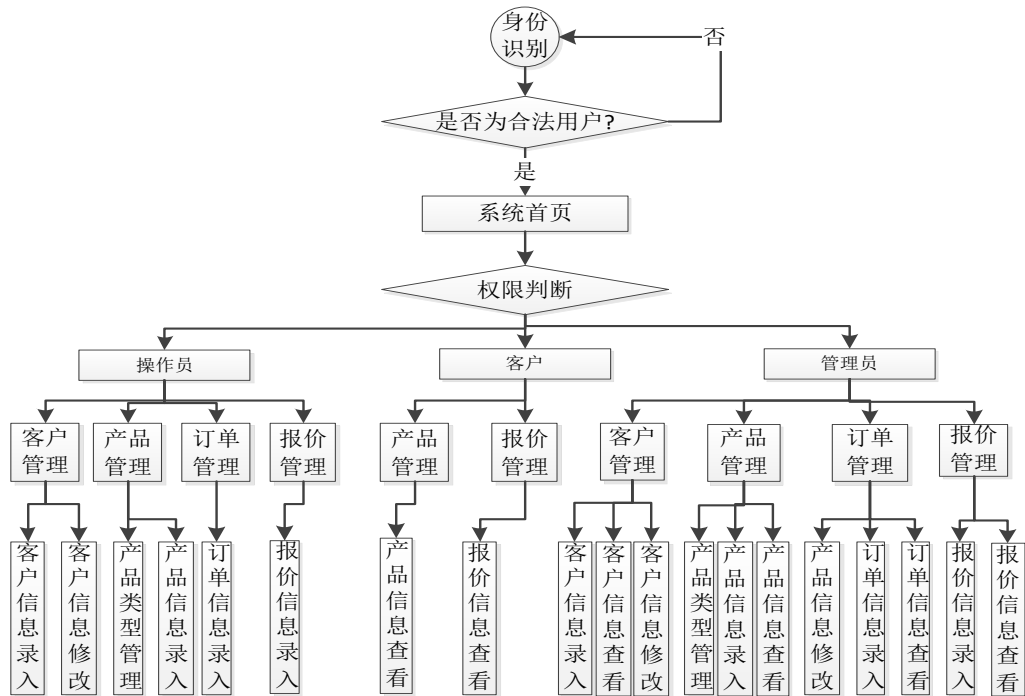


图 3-2 系统流程图

§3.3 数据库设计

§3.3.1 数据库设计

首先数据库的设计是在满足三大范式的基础下完成的，因此表结构设计合理且有相对较少的冗余。在本系统中，需要存储产品的信息、产品类型的信息、客户的信息、报价的信息、订单的信息和用户的信息等，所以需要对这些信息创建相应的数据库表，分别是客户信息基本表、产品类型信息基本表、产品信息基本表、订单信息基本表、报价信息基本表以及用于存储登录用户的用户信息基本表。但是有的表可能会有关联的表，比如：客户信息就需要为客户设置客户所属的行业、客户地址表（存储省市县）、订单信息会涉及到产品单位的产品单位信息表，所以系统有区域信息表、行业信息表、产品单位信息表。具体实体关系图如图 3-3 所示。

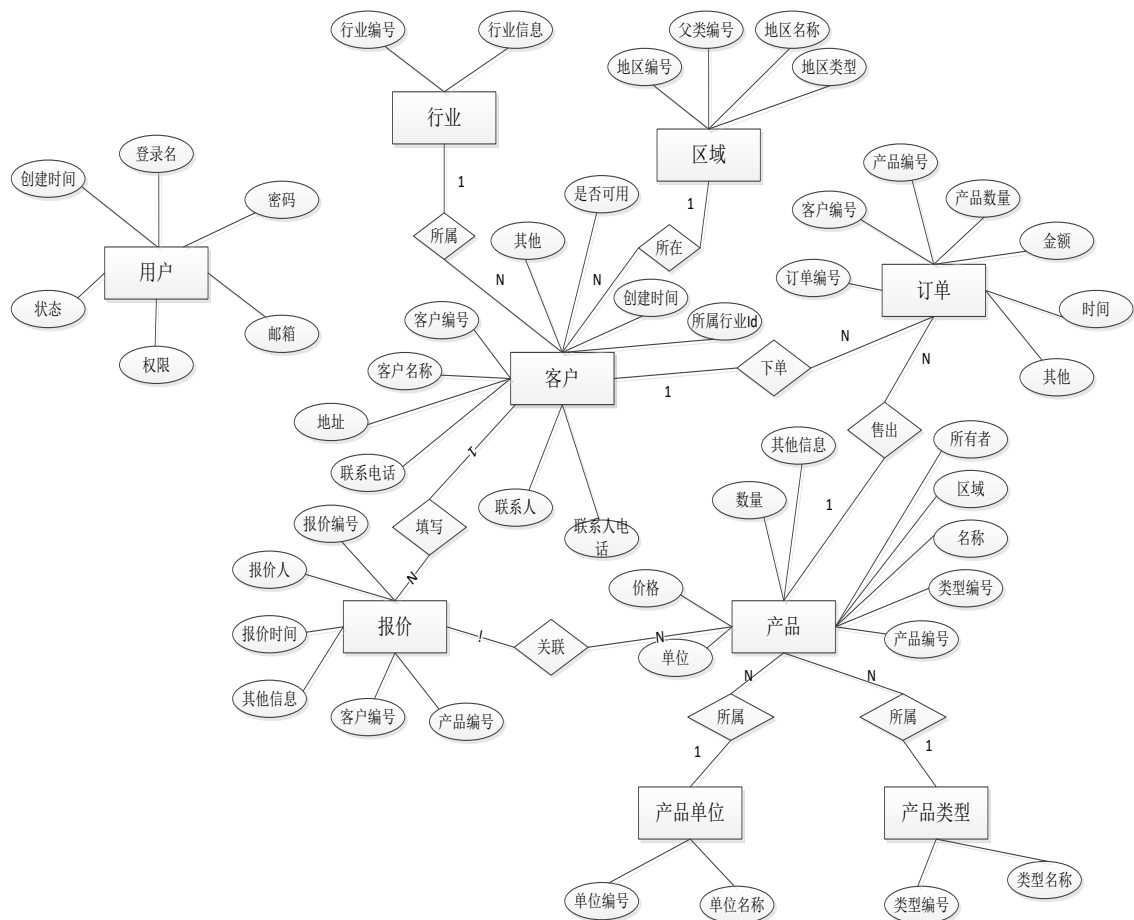


图 3-3 E-R 图

§3.3.2 数据库表设计

根据以上分析，系统共需使用 9 张表，以下是本系统设计中所需要的主要的表的设计。

用户表用来存储用户的相关信息也可以用来验证登录信息是否正确，详情见表 3-1。

表 3-1 用户表

tb_user		
字段名	类型	说明
username	varchar(50)	登录名
password	varchar(100)	登录密码
email	varchar(50)	邮箱
grade	varchar(6)	权限等级
createDate	datetime	创建时间
address	varchar(255)	用户地址

客户信息表主要是用来对公司的客户的信息进行存储的，详情见表 3-2。

表 3-2 客户信息表

tb_customer		
字段名	类型	说明
customerNo	varchar(30)	客户编号
customerName	varchar(50)	客户名称
address	varchar(200)	地址
phone	varchar(11)	联系电话
relationman	varchar(50)	联系人
relationmanPhone	varchar(11)	联系人电话
industryId	varchar(20)	所属行业 Id
createDate	datetime	客户创建时间

state	int	是否可用 1 可用 0 不可用
otherInfo	varchar	其他信息

区域信息表用来存储客户所在省市县信息的，详情见表 3-3。

表 3-3 区域信息表

tb_area		
字段名	类型	说明
areaId	varchar(20)	地区Id
parentId	varchar(20)	父类Id
areaName	varchar(30)	地区名称
areaType	varchar(10)	地区类型

行业信息表用来存储客户所属行业的，详情见表 3-4。

表 3-4 行业信息表

tb_industry		
字段名	类型	说明
industryId	varchar(20)	行业Id
industryName	varchar(100)	行业名称

产品类型信息表，用来储存产品类型信息，详情见表 3-5。

表 3-5 产品类型信息表

tb_producttype		
字段名	类型	说明
producttypeNo	varchar(10)	产品类型编号
producttypeName	varchar(50)	产品类型名称

产品表是用于存储产品信息与产品类型信息表关联，详情见表 3-6。

表 3-6 产品信息表

tb_product		
字段名	类型	说明
productNo	varchar(30)	产品编号
producttypeNo	varchar(10)	产品所属类型编号
productName	varchar(50)	产品名称
productArea	varchar(100)	产品所在区域
productOwner	varchar(50)	产品所有者
unit	varchar(50)	产品单位
price	decimal(10,2)	产品价格
quantity	int	产品数量
otherInfo	varchar(500)	产品其他信息

产品单位表用来存储产品单位信息，详情见表 3-7。

表 3-7 产品单位表

tb_unit		
字段名	类型	说明
unitId	int	单位Id
unitName	varchar(255)	单位名称

订单信息表用来存储订单信息，详情见表 3-8。

表 3-8 订单信息表

tb_order		
字段名	类型	说明
orderNo	varchar(30)	订单编号
customerNo	varchar(30)	下单客户编号
productNo	varchar(30)	产品编号
productCount	decimal(10,2)	产品数量

orderAmount	decimal(10,2)	订单金额
orderTime	datetime	下单时间
otherInfo	varchar(500)	订单其他信息

报价信息表用来联系用户与收货地址的，详情见表 3-9。

表 3-9 报价信息表

tb_quotation		
字段名	类型	说明
quotationNo	bigint(30)	报价编号
quotationMan	varchar(20)	报价人
quotationTime	datetime	报价时间
otherInfo	varchar(500)	报价其他信息
customerNo	varchar(30)	客户编号
productNo	varchar(30)	产品编号

第4章 系统详细设计

§4.1 系统各模块功能设计与简介

§4.1.1 用户登录注册

在地址栏输入 localhost:8088/qms 会默认打开登录页面。如图 4-1

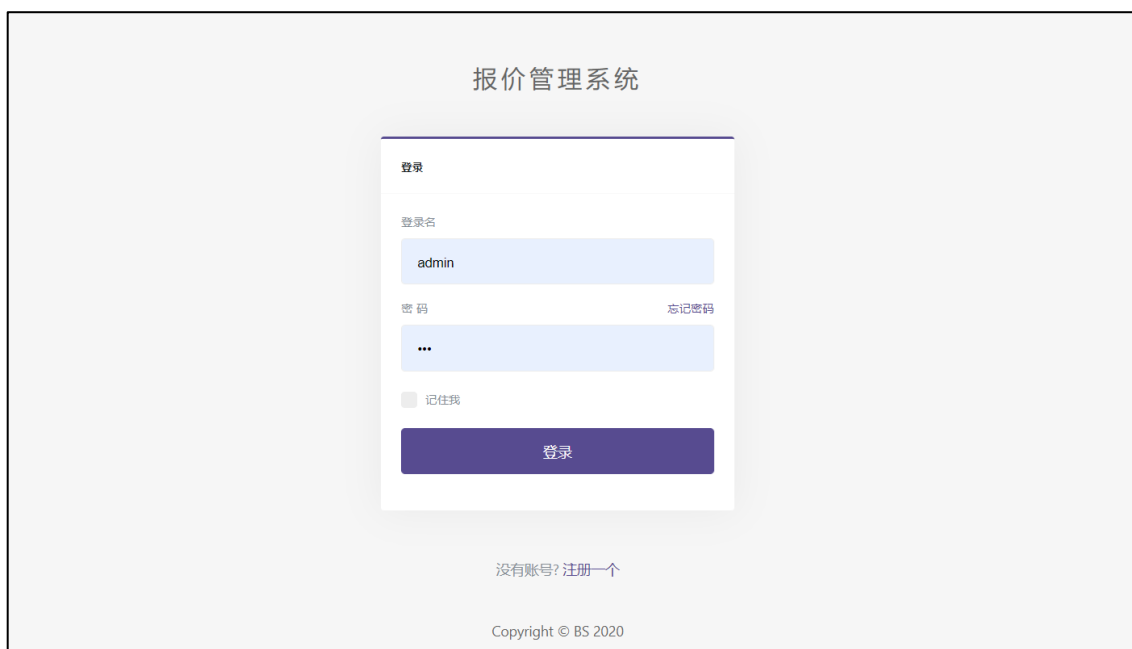


图 4-1 系统登录界面

在文本框“登录名”和“密码”中输入登录名和密码之后，点击“登录”按钮，就会到数据库中查询登录名和密码是否存在并且输入是否正确，如果登录名和密码都正确即登录成功，进入报价管理系统主页。若不成功则当前页面刷新，需要重新输入正确的登录名和密码。但是如果没有用于登录该系统的账号的话，可以点击“注册一个”按钮跳转来到用户的注册页面。如图 4-2

报价管理系统

注册

登录名

zhangsan

✓

邮箱

dlz0729@126.com

发送验证码

验证码有效时间 2:0

请输入验证码

密码

...

确认密码

...

✓

注册

取消

Copyright © BS 2020

图 4-2 系统注册界面

当输入的登录名经过校验未被注册过，则在登录名文本框后方提示“√”，如果输入的登录名是注册过的，则在登录名文本框后方提示“登录名已存在”。如果输入的邮箱合法，点击“发送验证码”按钮就可以给输入的邮箱发送验证码（6 位随机生成的纯数字），验证码的有效时间是两分钟，“请输入验证码”文本框上方会出现两分钟的倒计时，如果密码与确认密码一致，则在确认密码文本框后方提示“√”，验证码输入正确并且两次密码一致，点击注册按钮即可完成注册。如果期间超过两分钟后点击注册会提示验证码已失效，可以重新发送验证码。如果输入的验证码不正确，会提示验证码不正确。当完成注册之后，页面跳转至登录页面。期间如果我们不需要注册了，点击“取消”按钮，页面跳转至登录页面。

§4.1.2 客户信息录入

该功能只有管理员和操作员可以看到，当点击二级菜单“客户信息录入”，页面跳转至客户信息录入界面。如图 4-3

填写信息

客户名称

河南科技大学

联系方式

0394-123456

所属行业

信息传输、软件和信息技术服务业

省/直辖市

河南

市/市辖/县辖

洛阳

县/区

涧西区

详细地址

河南科技大学西苑校区

联系人姓名

张三

联系人电话

12346678900

其他信息

保存

取消

图 4-3 客户信息录入界面

页面中所属行业下拉框是页面加载时，查询行业信息表（tb_industry）后绑定的，我们可用为客户选择对应的行业。页面中的省/直辖市，市/市辖/县辖，县/区是一个省市县三联动的下拉框。同样，页面加载的时候会查询区域信息表（tb_area），优先绑定省/直辖市，当我们选择一个省/直辖市，触发事件，查询该省/直辖市下的市/市辖/县辖，县/区同理。当输入框的值都录入完毕之后，点击“保存”，就会将客户信息保存到客户信息表（tb_customer）中。保存成功之后页面会刷新，表单内容被清空，页面不跳转。如果点击“取消”，页面跳转至主页。

§4.1.3 客户信息修改

该功能只有管理员和操作员可用看到，操作员只能通过二级菜单“客户信息修改”跳转至客户信息修改页面，管理员不仅能通过二级菜单，还能在客户信息查看页面，选中某一条数据，点击编辑按钮，这条数据会被带到客户信息修改页面对应的文本框中。如图 4-4

是第一页或者已经是最后一页了，就不能再往前或者往后翻页。表头上面是五个可用于模糊查询的参数，分别是根据客户编号唯一查询，根据客户名称模糊查询，根据联系人姓名模糊查询，根据查询时间起查询注册时间大于参数时间的数据，根据查询时间止查询注册时间小于参数时间的数据。点击查询按钮后对满足条件的数据进行筛选，最后分页展示。

§4.1.5 产品类型管理

该功能只有管理员和操作员可以看到，当点击二级菜单“产品类型管理”，页面跳转至“产品类型管理”页面。如图 4-6



图 4-6 产品类型管理界面

页面左侧展示出数据库中已经存在的所有的产品类型，页面右侧可以新增产品类型，挡在输入框输入内容后，会先校验该产品类型是否已经存在，如果不存在便可以新增，如果产品类型名称是已经存在的则页面提示，产品类型名称已存在，并且清空输入框，点击“取消”按钮的话，页面跳转至主页。

§4.1.6 产品信息录入

该功能只有管理员和操作员才能够看到，点击“产品信息录入”页面跳转至产品信息录入页面。如图 4-7

填写信息

产品名称

跑步机

产品所属类型

保健休闲类

产品所属区域

上海

产品单位

件

产品单价

3600.99

产品数量

100

产品所有者

张三

其他信息

产品其他的描述信息

保存

取消

图 4-7 产品信息录入界面

对需要新增的产品输入各种参数以后，点击“保存”，即添加成功，如果，带年纪“保存”之前未输入参数，则无法提交表单信息，如果点击“取消”页面跳转至主页。

§4.1.7 产品信息修改

该功能只有管理员和操作员可以看到，管理员有两种方式进入该页面，分别是二级菜单“产品信息修改”、产品信息查看页对面每条数据修改操作的按钮。操作员只能通过二级菜单“产品信息修改”进入产品信息修改页面。如图 4-8

产品编号

1000017

查询

产品信息修改

产品编号

1000017

产品名称

B1产品

所属产品类型

B类

产品所属区域

上海

产品单位

个

价格

101.0

数量

500

产品所有者

dlz

其他

其他信息

保存

取消

图 4-8 产品信息修改界面

页面中产品信息修改部分的产品编号文本框是只读的，填写需要修改的内容后，点击保存，即可修改成功，修改成功根据权限判断页面跳转，如果是管理员跳转至产品信息查看页面，如果是操作员跳转至主页。

§4.1.8 产品信息查看

该功能管理员和客户可以看到，但是客户不能对数据进行修改，点击菜单选项“产品信息查看”，页面跳转至产品信息查看页面，管理员权限的页面如图 4-9

产品信息									
	产品编号	产品名称	产品类别	产品所属区域	产品所有者	产品单位	产品价格	产品数量	操作
01	1000011	A1产品	F类	北京	dlz	码	1	100	
02	1000012	A2产品	A类	上海	dlz	只	101	500	
03	1000013	B1产品	B类	上海	dlz	个	101	500	
04	1000014	B2产品	B类	上海	dlz	个	101	500	
05	1000015	B3产品	D类	浙江	dlz	英尺	101	1111	

图 4-9 产品信息查看界面

管理员和客户都可以根据查询条件对产品信息进行查询。

§4.1.9 订单信息录入

该功能只有管理员和操作员才可以看到，点击二级菜单“订单信息录入”页面跳转至订单信息录入页面。如图 4-10

填写信息

客户名称

产品类型

产品名称

产品数量

请选择

请选择

▼

请输入数字

其他信息

保存

取消

图 4-10 订单信息录入界面

其中客户名称、产品类型、产品名称都是下拉框可以进行选择，产品数量文本框进行了 js 校验，是只能输入数字的。点击保存按钮，订单信息录入成功。点击取消，页面跳转至主页。

§4.1.10 订单信息查看

该功能只有管理员可以看到，当登录的账号是管理员权限的，点击订单信息查看，页面跳转至订单信息查看页面。如图 4-11

订单编号	客户姓名	地址	电话	产品名称	产品所有者	数量	单位	订单金额	下单时间	操作	
01	e200425102104	xxxx软件公司	上海市徐汇区/崑山路719号	0394-46548	跑步机	张三	222	件	799419.78	2020-04-25 12:08:37	[icon]
02	e1111111112	xxxx软件公司	上海市徐汇区/崑山路719号	0394-46548	B1产品	dlz	10	个	1010	2020-04-19 21:29:16	[icon]
03	e20200419209135	XX软件公司	河南科技大学西苑校区	0394-123456	跑步机	张三	222	件	799419.78	2020-04-19 14:52:32	[icon]
04	e20200419585674	XX软件公司	河南科技大学西苑校区	0394-123456	跑步机	张三	222	件	799419.78	2020-04-19 14:47:40	[icon]
05	e20200419913335	xxxx软件公司	上海市徐汇区/崑山路719号	0394-46548	B1产品	dlz	2	个	202	2020-04-19 14:46:24	[icon]
06	e20200419346565	xxxx软件公司	上海市徐汇区/崑山路719号	0394-46548	B2产品	dlz	10	个	1010	2020-04-19 14:45:22	[icon]
07	e20200419866472	xxxx软件公司	上海市徐汇区/崑山路719号	0394-46548	B2产品	dlz	10	个	1010	2020-04-19 14:25:23	[icon]
08	e20200419602889	xxxx软件公司	上海市徐汇区/崑山路719号	0394-46548	A1产品	dlz	10	码	10	2020-04-19 14:22:03	[icon]

图 4-11 订单信息查看界面

页面数据采用数据分页的方式展示数据，默认每页显示 8 条数据。可以根据查询条件对数据进行筛选，比如，通过订单编号进行唯一查询、通过客户名称、产品名称模糊查询。最后将满足条件的数据展示在页面上。管理员可以点击每条数据的详细按钮查看订单的完整信息。

§4.1.11 报价信息录入

该功能只有管理员和操作员可以看到，如果登录的账号权限是管理员或操作员，点击纪报价信息录入，页面跳转至报价信息录入页面。如图 4-12

图 4-12 报价信息录入界面

填写好报价信息之后，点击保存，即可完成报价信息的录入，点击取消的话，页面跳转至主页。

§4.1.12 报价信息查看

该功能只有管理员和客户可以看到，点击报价信息查看，页面跳转至报价信息查看页面，如果登录账号的权限是管理员，还有删除报价信息的权限。如图 4-13

	报价编号	客户名称	产品名称	报价时间	报价人	其他信息	操作
01	111111111125	xxxx软件公司	跑步机	2020-05-01 07:08:12	张三	忘后能多次合作	
02	111111111124	xxxx软件公司	跑步机	2020-05-01 07:06:38	张三	忘后能多次合作	
03	111111111123	xxxx软件公司	跑步机	2020-05-01 06:41:12	张三	忘后能多次合作	
04	111111111122	XX	B2产品	2020-05-01 06:30:15	张三	忘后能多次合作	
05	111111111121	xxxx软件公司	B1产品	2020-05-01 06:27:16	张三	忘后能多次合作	
06	111111111118	xxxx软件公司	B1产品	2020-05-01 05:52:15	张三	忘后能多次合作	
07	111111111116	xxxx软件公司	B2产品	2020-05-01 05:51:38	张三	忘后能多次合作	
08	111111111119	xxxx软件公司	A2产品	2020-04-10 05:52:29	张三	忘后能多次合作	

图 4-13 报价信息查看界面

页面同样采用数字分页的方式对报价信息进行分页展示，默认每页 8 条数据，如果是管理员，可以删除某条数据，只需要点击删除按钮，即可将该条数据删除。

第5章 系统测试

此系统的开发过程还使用到了 Maven，可以对项目中使用到的 jar 包进行统一的管理，使用 JSP 页面实现视图的展示，JSP 页面中使用到了 EL、JSTL 表达式来简化代码量，前端使用到了 Ajax、Jquery。最后系统功能基本上实现。现在在本地计算机上进行功能测试，在测试描述中，在前面模块的测试中进行详细展示，后面若有类似则不再展示。

以下仅展示部分测试内容。

§5.1 系统的功能测试

§5.1.1 测试登录注册功能

进入注册页面之后，填写注册信息之后，点击注册按钮，若输入的验证码不正确，页面弹窗给出提示信息。如图 5-1

The image shows a web application interface for registration. At the top, a modal dialog box displays the message "localhost:8088 显示" and "验证码不正确" (Verification code is incorrect), with a blue "确定" (Confirm) button. Below this, the registration form is visible. It includes fields for "登录名" (Username) with the value "adminss" and a green checkmark, "邮箱" (Email) with the value "dlz0729@126.com", and "密码" (Password) and "确认密码" (Confirm Password) fields, both with green checkmarks. A "发送验证码" (Send verification code) button is next to the email field. A timer "验证码有效时间 1:33" is shown. At the bottom, there are two large buttons: a red "注册" (Register) button and a dark blue "取消" (Cancel) button.

图 5-1 注册界面

§5.1.2 测试客户录入功能

进入客户信息录入界面，如果未填写客户信息就点击“保存”按钮，则非空字段会在文本框的下面提示“请输入该字段”并且文本框的边框变为红色。如图 5-2

填写信息

客户名称

请输入客户名称

联系方式

请输入联系方式

所属行业

请选择

请选择客户所属行业

省/直辖市

请选择

请选择省/直辖市

市/市辖区/县镇

请选择市/市辖区/县镇

县/区

详细地址

请输入详细地址

联系人姓名

请输入联系人姓名

联系人电话

请输入联系人电话

其他信息

保存

取消

图 5-2 客户信息录入界面

§5.1.3 测试客户信息修改功能

客户信息修改功能可对客户的信息进行修改，但要求某些信息不能为空，如果为空的话，会在文本框下方进行提示“请输入该字段”并且文本框的边框变为红色。如图 5-3

修改客户信息

客户编号

c20200329337431

客户名称

xxx软件公司

所属行业

请选择

请选择
教育业
化工业
农、林、牧、渔业
采矿业
制造业
电力、热力、燃气及水生产和供应业
建筑业
交通运输、仓储和邮政业
住宿和餐饮业
信息传输、软件和信息技术服务业
金融业
房地产业
科学研究和技术服务业
水利、环境和公共设施管理业
居民服务、修理和其他服务业
其他

电话

0394-123456

地址

河南科技大学西苑校区

联系人

请输入联系

联系人电话

15994158499

其他

修改

取消

图 5-3 客户信息修改界面

§5.1.4 测试客户信息查看功能

点击客户信息查看，当展示的已经是首页或末页的时候，在点击上一页或者下一页的时候，页面弹窗给出提示信息“已经是第一页了！”，“已经是最后一页了！”。如图 5-4

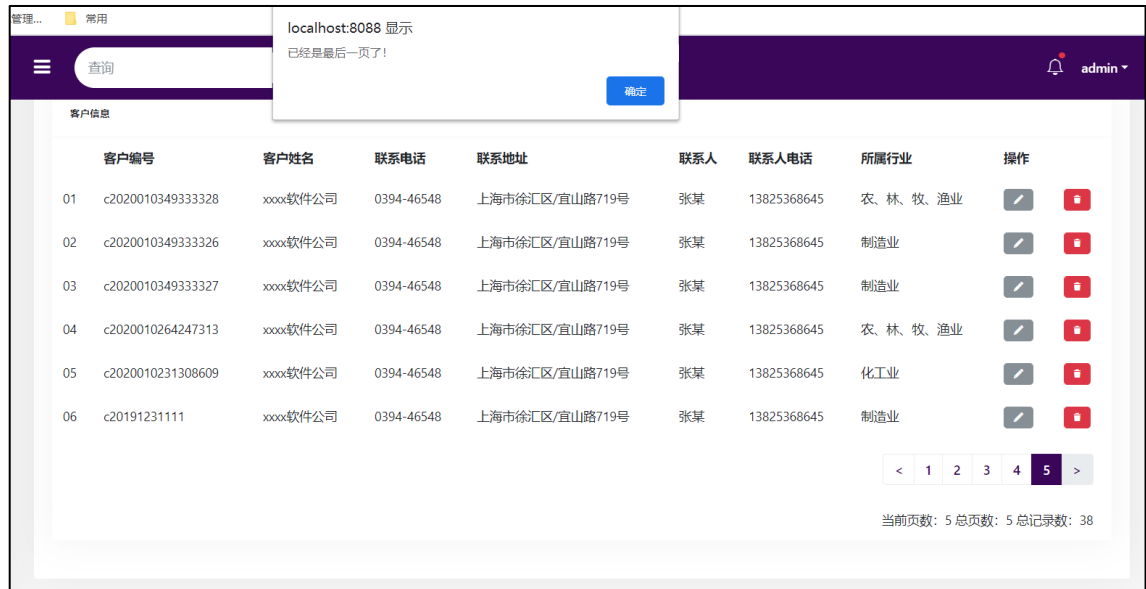


图 5-4 客户信息查看界面

§5.1.5 测试删除客户功能

点击单条数据操作列中的“删除”按钮，页面弹出“确定删除客户吗？”。如图 5-5

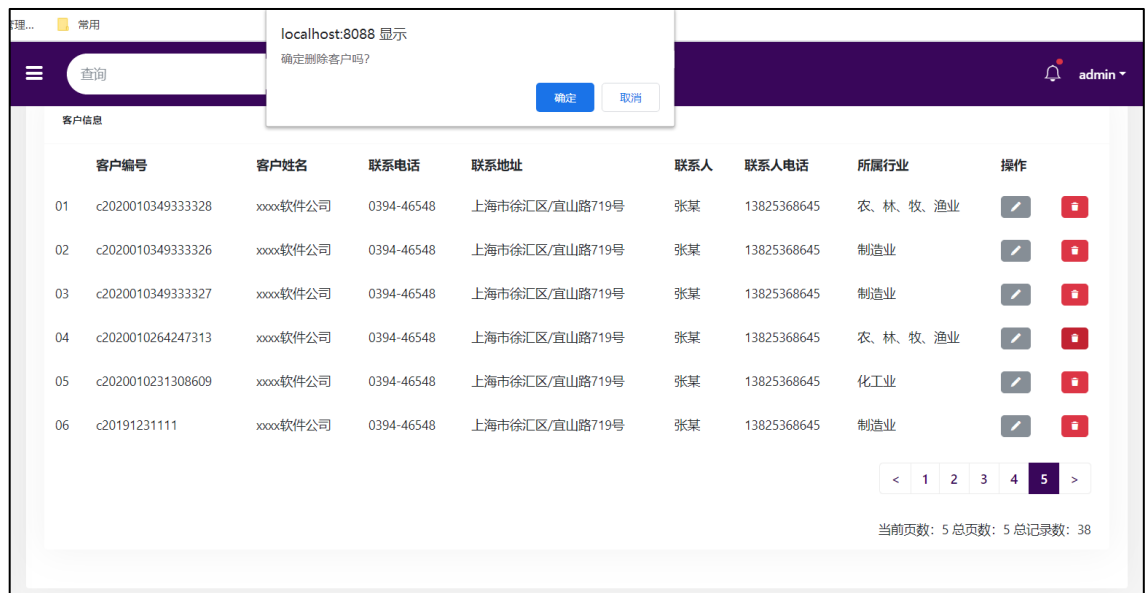


图 5-5 客户信息删除确认界面

点击“确认”，客户信息删除成功，页面给出“删除成功！”提示。如图 5-6

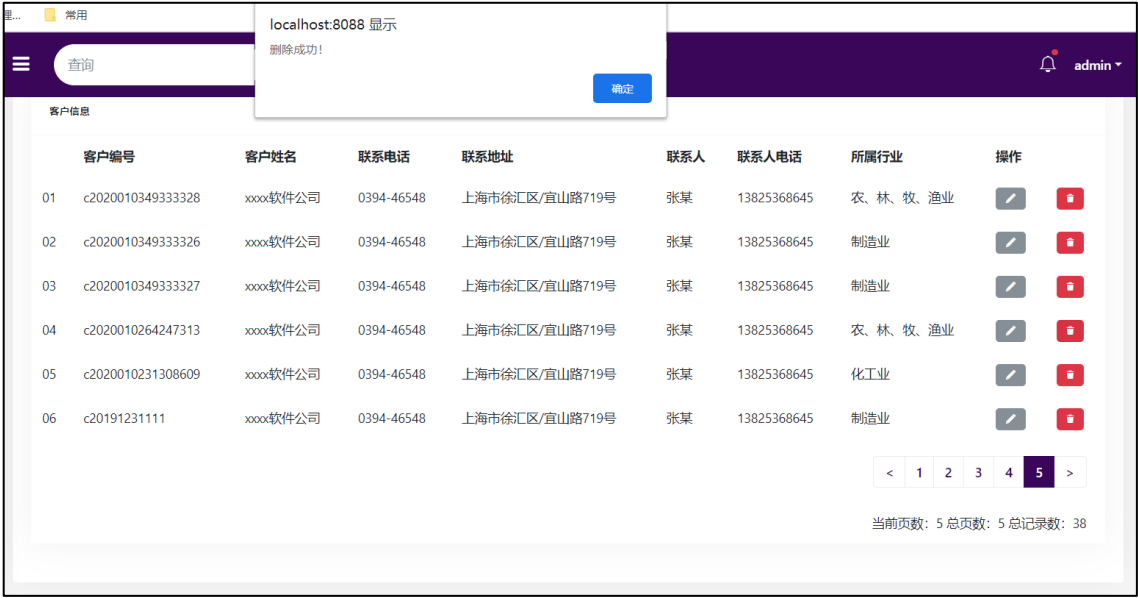


图 5-6 客户信息删除成功界面

§5.1.6 测试产品类型录入

如果输入的产品类型名称是已存在的，则页面弹窗给出提示“产品类型已存在”并且清空输入框。如图 5-7

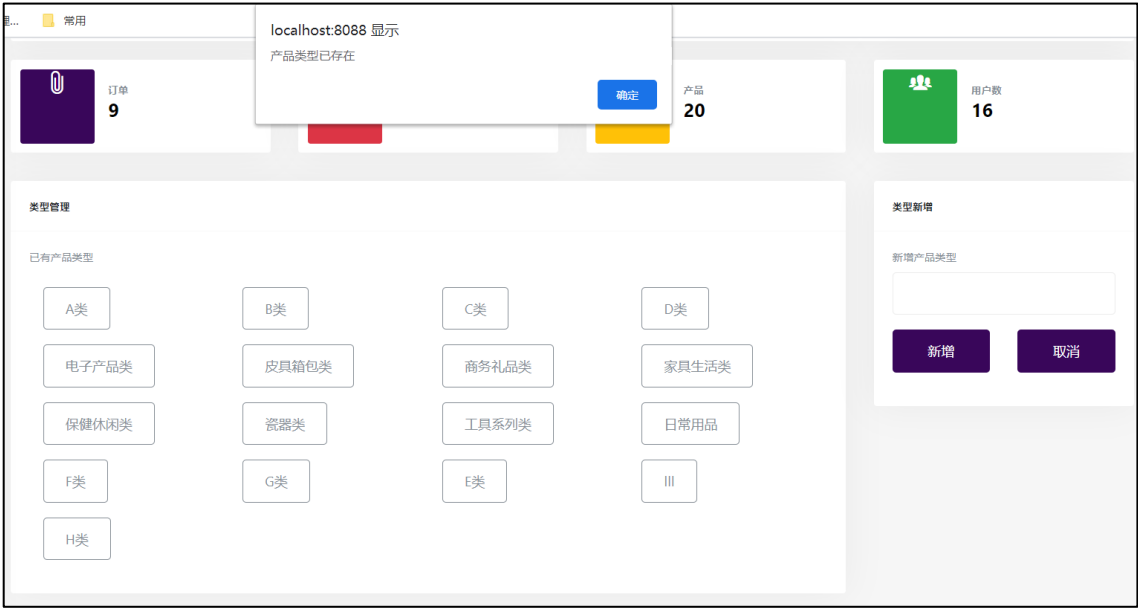


图 5-7 产品类型录入界面

§5.1.7 测试产品信息录入

进入产品信息录入页面，如果未填写产品信息就点击“保存”按钮，则非空字段会在文本框的下面提示“请输入该字段”并且文本框的边框变为红色。如图 5-8

图 5-8 产品信息录入界面

§5.1.8 测试产品信息查看

进入产品信息查看页面，页面对数据进行分页展示，默认第 1 页 8 条数据，数据展示上方为查询条件，可以根据其中一个或多个条件对数据进行筛选。如图 5-9

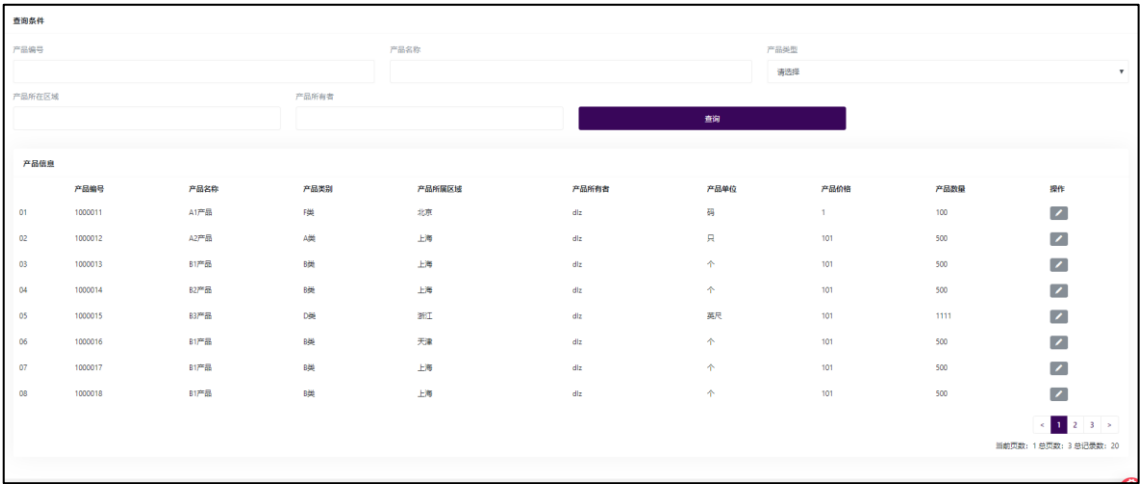


图 5-9 产品信息查看界面

§5.1.9 测试订单信息录入

进入订单信息录入页面，如果未填写订单信息就点击“保存”按钮，则非空字段会在文本框的下面提示“请输入该字段”并且文本框的边框变为红色。如图 5-10



图 5-10 订单信息录入界面

§5.1.10 测试订单信息查看

进入订单信息查看页面，页面对数据进行分页展示，默认第 1 页 8 条数据，数据展示上方为查询条件，可以根据其中一个或多个条件对数据进行筛选。如图 5-11

查询条件

订单编号

客户名称

产品名称

查询时间

年 / 月 / 日

查询

订单编号	客户姓名	地址	电话	产品名称	产品所有者	数量	单位	订单金额	下单时间	操作	
01	c200425102104	xxxx软件公司	上海市徐汇区/宜山路719号	0394-46548	跑步机	张三	222	件	799419.78	2020-04-25 12:08:37	
02	c111111112	xxxx软件公司	上海市徐汇区/宜山路719号	0394-46548	81产品	dz	10	个	1010	2020-04-19 21:29:16	
03	c20200419209135	XX软件公司	河南科技大学西苑校区	0394-123456	跑步机	张三	222	件	799419.78	2020-04-19 14:52:32	
04	c20200419585674	XX软件公司	河南科技大学西苑校区	0394-123456	跑步机	张三	222	件	799419.78	2020-04-19 14:47:40	
05	c20200419913335	xxxx软件公司	上海市徐汇区/宜山路719号	0394-46548	81产品	dz	2	个	202	2020-04-19 14:46:24	
06	c20200419346565	xxxx软件公司	上海市徐汇区/宜山路719号	0394-46548	82产品	dz	10	个	1010	2020-04-19 14:45:22	
07	c20200419866472	xxxx软件公司	上海市徐汇区/宜山路719号	0394-46548	82产品	dz	10	个	1010	2020-04-19 14:25:23	
08	c20200419602889	xxxx软件公司	上海市徐汇区/宜山路719号	0394-46548	A1产品	dz	10	磅	10	2020-04-19 14:22:03	

<

1

2

>

图 5-11 订单信息查看界面

§5.1.11 测试报价信息录入

进入报价信息录入页面，如果未填写订单信息就点击“保存”按钮，则非空字段会在文本框的下面提示“请输入该字段”并且文本框的边框变为红色。如图 5-12

填写信息

客户名称

产品类型

产品

报价人

请选择

请选择

请选择

请选择客户名称

请选择产品类型

请选择产品

请输入报价人

其他信息


保存

取消

图 5-12 报价信息录入界面

§5.1.12 测试报价信息查看

进入报价信息查看页面，页面对数据进行分页展示，默认第 1 页 8 条数据。页面展示的是管理员权限页面，可以删除报价信息。如图 5-13

报价信息							
	报价编号	客户名称	产品名称	报价时间	报价人	其他信息	操作
01	1111111111125	xxxx软件公司	跑步机	2020-05-01 07:08:12	张三	忘后能多次合作	
02	1111111111124	xxxx软件公司	跑步机	2020-05-01 07:06:38	张三	忘后能多次合作	
03	1111111111123	xxxx软件公司	跑步机	2020-05-01 06:41:12	张三	忘后能多次合作	
04	1111111111122	XX	B2产品	2020-05-01 06:30:15	张三	忘后能多次合作	
05	1111111111121	xxxx软件公司	B1产品	2020-05-01 06:27:16	张三	忘后能多次合作	
06	1111111111118	xxxx软件公司	B1产品	2020-05-01 05:52:15	张三	忘后能多次合作	
07	1111111111116	xxxx软件公司	B2产品	2020-05-01 05:51:38	张三	忘后能多次合作	
08	1111111111119	xxxx软件公司	A2产品	2020-04-10 05:52:29	张三	忘后能多次合作	

< 1 2 >

当前页数: 1 总页数: 2 总记录数: 10



图 5-13 报价信息查看界面

§5.2 系统测试结论

通过对上述功能的测试，各模块的功能基本上都达到了预期的效果，运行相对稳定，测试成功。

结 论

通过此次毕业设计的制作，我收获了很多东西。总结有以下几点：

第一，对于 SSM 框架的搭建更加的熟练，而且体会到使用框架开发效率的提高和方便。我们的项目中 controller 包存放的是前端控制器主要用于与页面进行交互，类和方法上使用注解进行唯一区分。Dao 层存放的是持久化的接口以及和接口唯一映射的 mapper 文件，我们在接口中定于操作数据库的抽象方法，而在 mapper 文件中定义该方法使用的 sql。Server 层存放的是业务逻辑的类，类里面可以写很多实现复杂业务的代码。

第二，就是对于 Mysql 数据库的使用，Mysql 做为系统数据存储的工具，通过本次毕业设计的制作，其中用到了表关联、模糊查询、动态 sql 等等，让我对于 sql 语句的使用更加的熟练。

第三，对于 Ajax、Jquery 的使用，项目中大多数地方使用到了 ajax 来与后台进行交互，对返回的结果进行处理，比如：数字分页，让我对 Ajax、Jquery 的使用更加的熟练。

第四，在注册功能中对于第三方接口的使用，能够熟练的使用 Java 发送邮件用来做校验。

在此次毕业设计的过程中，毫无疑问这次的毕业设计教会了我很多东西，极大的提升了自我学习能力。首先对 Java 面向对象的理解的更加深入，其次是对 Java 常用框架有了更加深入的理解，最后就是不能吃毕业设计锻炼了个人的耐心，因为项目开发的周期比较长，而且是独自开发，这在之前是没有过的，最后对于此次毕业设计的制作加大了我对开发的兴趣，在以后一定会更加努力的。

参考文献

- [1] 张永强. 计算机软件JAVA编程特点及其技术分析[J]. 计算机产品与流通. 2019 (01):23
- [2] 王越. JAVA编程语言在计算机软件开发中的应用[J]. 电子技术与软件工程. 2019 (01):35
- [3] 倪海顺. 计算机软件开发中的JAVA编程语言应用探讨[J]. 信息与电脑（理论版）. 2019（02）:60-61.
- [4] 曹文渊. JAVA语言在计算机软件开发中的引用[J/OL]. 电子技术与软件工程. 2019（02）:53-54
- [5] 吕萱. 计算机软件开发中Java编程语言的应用[J]. 信息与电脑（理论版）, 2018（08）:67-69.
- [6] 姚素红. 基于JAVA的WEB应用系统开发模式[J]. 江苏工程职业技术学院学报. 2018, 18（01）:6-9.
- [7] 石坤泉, 杨震伦. 基于MYSQL数据库的数据隐私与安全策略研究[J]. 网络安全技术与应用. 2017（01）:79+87.
- [8] 高杨, 赵立杰. 基于JSP+MySQL的物流管理系统的设计与实现[J]. 信息系统工程, 2016,（11）:141+143.
- [9] 刘永欢, 张磊, 王玉成. Android移动终端与PHP及MySQL数据通信设计[J]. 智慧工厂, 2016,（12）:82-84.
- [10] 庄小妹. 彩虹表在MySQL密码破解中的运用研究[J]. 长春师范大学学报, 2016,（10）:47-51.
- [11] 贾晓芳, 沈泽刚. Java Web应用开发中的常见乱码形式及解决方法[J]. 软件导刊, 2017,（04）:214-216.
- [12] 胡蕊蕊. Java Web程序运用中泛型DAO的作用[J]. 电子测试, 2017,（04）:46+49.
- [13] 贾文潇, 邓俊杰. Java Web应用中常见的错误和异常问题处理[J]. 电子测试. 2016,（07）:60+81.
- [14] 单东林, 张晓菲, 魏然. 锋利的jQuery [M]. 北京:人民邮电出版社, 2012

(03):50+52

- [15] 刘焱. jQuery在平台中的应用开发研究[J]. 信息系统工程. 2019 (01):20.
- [16] 刘志洋. AJAX技术在web程序开发中的运用探讨[J]. 轻纺工业与技术.2020 (02):25.
- [17] 徐雯, 高建华. 基于Spring MVC及Mybatis的Web应用框架研究[J]. 微型电脑应用. 2012, 28(7):1-4.
- [18] 林萍, 朱婵. 基于Ajax技术和JAVAE的分页查询优化. 计算机系统应用. 2017 (08):15.
- [19] 温立辉. AJAX异步交互技术浅析. 山东工业技术. 2017 (02):15.
- [20] 许家珩, 白忠建, 吴磊. 软件工程:理论与实践(第2版)[M]. 北京:高等教育出版社. 2009.
- [21] 邱吉雨. 基于Java开发Web项目的核心技术. 电子元器件与信息技术. 2018 (07):20.
- [22] 李鹏博,于立婷,王天琪. Java web软件框架技术探析. 通讯世界. 2016 (07):25.
- [23] 范成臣. 浅析在JAVA WEB应用程序中使用Servlet技术. 数字技术与应用.2013 (09):15.
- [24] 于亚芳,郭磊. JavaEE框架技术课程建设研究[J]. 电脑知识与技术. 2018 (10):19.

致 谢

伴随着六月份的到来，大学四年学习时光已经接近尾声，而回望这四年，大一入学似乎就在昨天，可转眼我们就要离开生活了四年的地方。在这段时间里，我们欢笑过、挥泪过、奋斗过、拼搏过，大学的时光是充实而又丰富的，我们既有满满当当的课程，又有轻松愉快的课下生活。和蔼可亲的老师、奋发向上的同学们、互帮互助的室友，这些都是我们人生中积累的财富。

在此我想对我的学校，我的父母、亲人们，我的老师和同学们表达我由衷的谢意。感谢我的父母对我大学四年学习的支持；感谢河南科技大学软件学院给了我四年深造的机会，能够让我继续的学习和提高；感谢软件学院的老师和同学们四年来的关心和鼓励。老师们课堂上的激情洋溢，课堂下的谆谆教诲；同学们在学习中的认真热情，生活上的热心主动，所有这些都让我的四年大学生活充满了感动，这次毕业论文设计我得到了很多老师和同学的帮助，尤其是我的毕设指导老师杨继松老师，他在我做毕业设计的每个阶段，从选题到查阅资料，论文提纲的确定，中期论文的修改，后期论文格式调整等各个环节中都给予了我悉心的指导。在这里我想向他们表达我深深的谢意。

大学生活一晃而过，回首走过的岁月，心中倍感充实，当我写这篇论文的时候，有一种如释重负的感觉，感触良多。

附 录

项目核心代码

```
/**
 * 系统用户 前端控制器
 *
 * @author 段刘振
 *
 */
@Controller
public class UserController {

    @Resource(name = "userService")
    private UserService userService;

    @RequestMapping(value = { "/login", "", " " })
    public String login() {
        return "/login";
    }

    @RequestMapping(value = "/user/toIndex")
    public String index(HttpServletRequest request, HttpServletResponse
response) {
        HttpSession session = request.getSession();
        User user = (User) session.getAttribute("u");
        String pageName = "/index";
        if (user == null) { // 未登录
            pageName = "redirect:/login";
        }
    }
}
```

```
    }  
    return pageName;  
}  
  
@RequestMapping(value = "/user/doLogin")  
public String doLogin(HttpServletRequest request, HttpServletResponse  
response, Model model, @Validated User user) {  
    User u = userService.doLogin(user);  
    String pageName = "redirect:/login";  
  
    if (u != null) { // 用户存在  
        pageName = "/index";  
  
        HttpSession session = request.getSession();  
        // 设置 session 永不过期  
        session.setMaxInactiveInterval(-1);  
        session.setAttribute("u", u);  
  
        session.setAttribute("orders", userService.getOrders());  
        session.setAttribute("productTypes",  
userService.getProductTypes());  
        session.setAttribute("products", userService.getProducts());  
        session.setAttribute("users", userService getUsers());  
  
        // 得到是否记住我 CheckBox 的状态  
        String rem = request.getParameter("remember");  
        System.out.println("checkBox 状态: " + rem);  
  
        // 创建本地 Cookie 用来存储登录名、密码
```

```
Cookie username = new Cookie("username", user.getUsername());
Cookie password = new Cookie("password", user.getPassword());
if (rem != null) {
    username.setMaxAge(999999999);
    password.setMaxAge(999999999);
} else {
    password.setMaxAge(0);
}
// 将登录名、密码存放到 response 中
response.addCookie(username);
response.addCookie(password);
} else { // 登录信息不存在
    // System.out.println("登录失败");
    // request.setAttribute("errorMsg", "用户名或密码错误！");
}
return pageName;
}

@RequestMapping(value = "/user/doLogout")
public String doLogout(HttpServletRequest request, HttpServletResponse
response) {
    HttpSession session = request.getSession();

    User u = (User) session.getAttribute("u");
    if (u != null) { // 判断 session 中存的 u 是否为空
        session.removeAttribute("u");
    }
    return "redirect:/login";
}
```

```
@RequestMapping(value = "/user/toRegister")
```

```
public String toRegister() {
```

```
    return "/register";
```

```
}
```

```
@RequestMapping(value = "/esc")
```

```
public String escRegister() {
```

```
    return "/login";
```

```
}
```

```
@RequestMapping(value = "/user/valUsername")
```

```
@ResponseBody // ResponseBody 标注返回值是 json 类型
```

```
public Map<String, String> valUserName(HttpServletRequest request,  
HttpServletResponse response) {
```

```
    String username = request.getParameter("username");
```

```
    // System.out.println(username);
```

```
    String name = userService.selUserName(username);
```

```
    Map<String, String> map = new HashMap<>();
```

```
    if (name != null && name != "" && name != " ") { // 用户名已存在
```

```
        map.put("msg", "用户名已存在");
```

```
    } else {
```

```
        map.put("msg", "✓");
```

```
    }
```

```
    return map;
```

```
}
```

```
@RequestMapping(value = "/user/sendCode")
```

```
@ResponseBody
```

```
public Map<String, String> sendCode(HttpServletRequest request,
HttpServletRequest response) throws Exception {
    String email = request.getParameter("email");
    Map<String, String> map = new HashMap<>();
    // 发送邮件 返回发送的 6 位纯数字验证码
    String valCode = SendEmailUtils.sendEmail(email);

    // 判断验证码是否生成成功
    if (!StringUtils.isEmpty(valCode)) {
        map.put("msg", "验证码生成成功");
    } else {
        map.put("msg", "验证码生成失败");
    }
    HttpSession session = request.getSession();
    // 将验证码存放到 session 中
    session.setAttribute("valCode", valCode);

    // 设置定时 两分钟后清除 session 中存放的验证码
    Timer timer = new Timer();
    timer.schedule(new TimerTask() {

        @Override
        public void run() {
            String valCode = (String) session.getAttribute("valCode");
            // 判断非空 防止报空指针异常
            if (!StringUtils.isEmpty(valCode)) {
                session.removeAttribute("valCode");
                // System.out.println("验证码被从 session 中清除了");
            }
        }
    });
}
```

```
        timer.cancel();
    }
}, 2 * 60 * 1000);
return map;
}

@RequestMapping(value = "/user/doRegister")
// 返回 json 数据类型 需要使用该注解
@ResponseBody
public Map<String, Object> doRegister(HttpServletRequest request,
HttpServletResponse response) throws Exception {
    Map<String, Object> map = new HashMap<>();
    HttpSession session = request.getSession();
    // String viewName = "/";
    // 从 session 得到验证码
    String valCode = (String) session.getAttribute("valCode");
    // System.out.println("session 存的注册码为: " + valCode);
    // 从表单得到验证码
    String valCode2 = request.getParameter("valCode");
    // session 中的验证码不为空 验证码未过期
    if (!StringUtils.isEmpty(valCode)) {
        if (valCode.equals(valCode2)) { // 验证码正确
            String username = request.getParameter("username");
            String email = request.getParameter("email");
            String password =
MD5Utils.MD5Encode(request.getParameter("password"), "utf-8");
            // 3 普通浏览用户 1 可用
            User user = new User(username, password, email, 3, 1, new
Date());
```

```
        int result = userService.doRegister(user);

        if (result > 0) { // 注册成功
            map.put("msg", "注册成功");
        } else {
            map.put("msg", "注册失败");
        }
    } else {
        map.put("msg", "验证码不正确");
    }
} else {
    map.put("msg", "验证码已失效！");
}

return map;
}

@RequestMapping(value = { "user/repwd" })
public String toRePwd(HttpServletRequest request, HttpServletResponse
response) {
    return "repwd";
}
}

/**
 * 客户信息管理 前端控制器
 *
 * @author 段刘振
 *
 */
@Controller
```

```
public class CustomerController {

    @Resource(name = "customerService")
    private CustomerService customerService;

    @RequestMapping(value = "/customerInfo/entry")
    public String toEntryCustInfo(HttpServletRequest request,
    HttpServletResponse response) {
        String pageName = "/customer/customerInfoAdd";
        if (IsLoginUtils.isLogin(request, response).length() > 0) {
            pageName = IsLoginUtils.isLogin(request, response);
        }
        return pageName;
    }

    @RequestMapping(value = "/customerInfo/see")
    public String seeCustomerInfo(HttpServletRequest request,
    HttpServletResponse response) {
        String pageName = "/customer/customerInfoSearch";
        if (IsLoginUtils.isLogin(request, response).length() > 0) {
            pageName = IsLoginUtils.isLogin(request, response);
        }
        return pageName;
    }

    @RequestMapping(value = "/customer/getIndustry")
    @ResponseBody
    public Map<String, Object> getIndustry() {
        Map<String, Object> map = new HashMap<String, Object>();
```



```
List<Industry> list = customerService.getIndustryInfo();  
if (list.size() > 0) {  
    map.put("list", list);  
}  
return map;  
}
```

```
@RequestMapping(value = "/customer/getProvince")  
@ResponseBody  
public Map<String, Object> getProvince() {  
    Map<String, Object> map = new HashMap<String, Object>();  
    List<Area> list = customerService.getProvince();  
    if (list.size() > 0) {  
        map.put("list", list);  
    }  
    return map;  
}
```

```
@RequestMapping(value = "/customer/getCity")  
@ResponseBody  
public Map<String, Object> getCity(HttpServletRequest request,  
HttpServletResponse response) {  
    Map<String, Object> map = new HashMap<>();  
    String provinceId = request.getParameter("provinceId");  
    List<Area> list = customerService.getCity(provinceId);  
    if (list.size() > 0) {  
        map.put("list", list);  
    }  
    return map;  
}
```

```
}

@RequestMapping(value = "/customer/getCounty")
@ResponseBody
public Map<String, Object> getCounty(HttpServletRequest request,
HttpServletResponse response) {
    Map<String, Object> map = new HashMap<>();
    String cityId = request.getParameter("cityId");
    List<Area> list = customerService.getCounty(cityId);
    if (list.size() > 0) {
        map.put("list", list);
    }
    return map;
}

@RequestMapping(value = "/custInfo/save")
public String doSave(HttpServletRequest request, HttpServletResponse
response, @Validated Customer customer) {
    String pageName = "/customer/customerInfoAdd";
    if (IsLoginUtils.isLogin(request, response).length() > 0) {
        pageName = IsLoginUtils.isLogin(request, response);
    }

    // 省/直辖市
    String province = request.getParameter("province");
    // 市/市辖/县辖
    String city = request.getParameter("city");
    // 县/区
    String county = request.getParameter("county");
```

```

Area provinceArea = customerService.getAreaById(province);
Area cityArea = customerService.getAreaById(city);
Area countyArea = customerService.getAreaById(county);

StringBuffer sBuffer = new StringBuffer();
if ("直辖市".equals(provinceArea.getAreaType())) {
    sBuffer.append(provinceArea.getAreaName() + "市");
    if ("市辖".equals(cityArea.getAreaType())) {
        // sBuffer.append(cityArea.getAreaName());
        if ("区".equals(countyArea.getAreaType())) {
            if ("区".equals(countyArea.getAreaName().substring(countyArea.getAreaName().length() - 1,
                countyArea.getAreaName().length())) {
                sBuffer.append(countyArea.getAreaName());
            } else {
                sBuffer.append(countyArea.getAreaName() + "区");
            }
        }
    } else if ("县辖".equals(cityArea.getAreaType())) {
        // sBuffer.append(cityArea.getAreaName());
        if ("县".equals(countyArea.getAreaType())) {
            // sBuffer.append(countyArea.getAreaName() + "县");
            if ("县".equals(countyArea.getAreaName().substring(countyArea.getAreaName().length() - 1,
                countyArea.getAreaName().length())) {
                sBuffer.append(countyArea.getAreaName());
            }
        }
    }
}

```

```

        } else {
            sBuffer.append(countyArea.getAreaName() + "县");
        }
    }
}

if ("省".equals(provinceArea.getAreaType())) {
    if ("台湾".equals(provinceArea.getAreaName()) || "香港"
.equals(provinceArea.getAreaName())
|| "澳门".equals(provinceArea.getAreaName())) {
        sBuffer.append(provinceArea.getAreaName());
        sBuffer.append(cityArea.getAreaName());
    } else if (provinceArea.getAreaName().length() >= 3) {
        if ("自治区"
.equals(provinceArea.getAreaName().substring(provinceArea.getAreaName().le
ngth() - 3,
        provinceArea.getAreaName().length())) {
            sBuffer.append(provinceArea.getAreaName());
            sBuffer.append(cityArea.getAreaName());
            sBuffer.append(countyArea.getAreaName());
        }
    } else {
        sBuffer.append(provinceArea.getAreaName() + "省");
        if ("市".equals(cityArea.getAreaType())) {
            sBuffer.append(cityArea.getAreaName() + "市");
            if ("县".equals(countyArea.getAreaType())) {
                if ("县"
.equals(countyArea.getAreaName().substring(countyArea.getAreaName().lengt
h() - 1,

```

```

        countyArea.getAreaName().length())) {
            sBuffer.append(countyArea.getAreaName());
        } else {
            sBuffer.append(countyArea.getAreaName() + "县");
        }
    } else if ("区".equals(countyArea.getAreaType())) {
        if ("区
".equals(countyArea.getAreaName().substring(countyArea.getAreaName().length() - 1,

        countyArea.getAreaName().length())) {
            sBuffer.append(countyArea.getAreaName());
        } else {
            sBuffer.append(countyArea.getAreaName() + "区");
        }
    }
} else if ("市辖".equals(cityArea.getAreaType())) {
    // sBuffer.append(cityArea.getAreaName());
    if ("区".equals(countyArea.getAreaType())) {
        // sBuffer.append(countyArea.getAreaName() + "区");
        if ("区
".equals(countyArea.getAreaName().substring(countyArea.getAreaName().length() - 1,

        countyArea.getAreaName().length())) {
            sBuffer.append(countyArea.getAreaName());
        } else {
            sBuffer.append(countyArea.getAreaName() + "区");
        }
    }
} else if ("县辖".equals(cityArea.getAreaType())) {

```

```

        // sBuffer.append(cityArea.getAreaName());
        if ("县".equals(countyArea.getAreaType())) {
            // sBuffer.append(countyArea.getAreaName() + "县");
            if ("县
.equals(countyArea.getAreaName().substring(countyArea.getAreaName().length() - 1,
        countyArea.getAreaName().length())) {
            sBuffer.append(countyArea.getAreaName());
        } else {
            sBuffer.append(countyArea.getAreaName() + "县");
        }
    }
}

customer.setAddress(sBuffer.toString() + "/" + customer.getAddress());
customer.setState(1);
customerService.doSaveCustomerInfo(customer);
return pageName;
}

@RequestMapping(value = "/customer/getAllCustomer")
@ResponseBody
public Map<String, Object> getCustomer(HttpServletRequest request,
HttpServletResponse response) {
    Map<String, Object> map = new HashMap<>();
    Integer startRow = Integer.parseInt(request.getParameter("startRow"));
    Integer pageSize = Integer.parseInt(request.getParameter("pageSize"));
    String customerNo = request.getParameter("customerNo");

```

```
String customerName = request.getParameter("customerName");
String relationman = request.getParameter("relationman");
String startDate = request.getParameter("startDate");
String endDate = request.getParameter("endDate");

// 获得满足条件的总的记录数
int dataNumber = customerService.getDataNumber(customerNo,
customerName, relationman, startDate, endDate);
// System.out.println("获得满足条件的总的记录数" + dataNumber);
List<CustomerVO> list = customerService.getCustomer(startRow,
pageSize, customerNo, customerName, relationman,
startDate, endDate);
map.put("dataNumber", dataNumber);
map.put("list", list);
return map;
}

@RequestMapping(value = "/customer/custSearch")
public String custSearch(HttpServletRequest request, HttpServletResponse
response) {
    String pageName = "/customer/customerInfoUpdate";
    if (IsLoginUtils.isLogin(request, response).length() > 0) {
        pageName = IsLoginUtils.isLogin(request, response);
    }
    return pageName;
}

@RequestMapping(value = "/customer/getCustomerByCustomerNo")
@ResponseBody
```

```
public Map<String, Object>
getCustomerByCustomerNo(HttpServletRequest request, HttpServletResponse
response) {
    Map<String, Object> map = new HashMap<>();
    String customerNo = request.getParameter("customerNo");
    CustomerVO customerVO =
customerService.getCustomerByCustomerNo(customerNo.trim());
    if (customerVO != null) {
        map.put("customerVO", customerVO);
    } else {
        map.put("customerVO", null);
    }
    return map;
}

@PostMapping(value = "/customerInfo/update")
public String customerInfoUpdate(HttpServletRequest request,
HttpServletResponse response,
    @Validated Customer customer) {
    String pageName = "index";
    if (IsLoginUtils.isLogin(request, response).length() > 0) {
        pageName = IsLoginUtils.isLogin(request, response);
    }
    // 修改
    customerService.updateCustomerByCustomerNo(customer);
    // 判断登录权限 决定返回的页面
    User user = (User) request.getSession().getAttribute("u");
    if (user != null && user.getGrade() == 1) {
        pageName = "/customer/customerInfoSearch";
    }
}
```



```
    }  
    return pageName;  
}  
  
@RequestMapping(value = "/customer/reAvailable")  
@ResponseBody  
public Map<String, String> reAvailable(HttpServletRequest request,  
HttpServletResponse response) {  
    Map<String, String> map = new HashMap<>();  
    String customerNo = request.getParameter("customerNo");  
    int result = customerService.reAvailable(customerNo);  
    if (result > 0) {  
        map.put("msg", "客户恢复成功");  
    } else {  
        map.put("msg", "客户恢复失败");  
    }  
    return map;  
}  
  
@RequestMapping(value = "/customer/delCustomer")  
@ResponseBody  
public Map<String, String> delCustomer(HttpServletRequest request,  
HttpServletResponse response) {  
    Map<String, String> map = new HashMap<String, String>();  
    String customerNo = request.getParameter("customerNo");  
    int result = customerService.delCustomer(customerNo);  
    if (result > 0) {  
        map.put("msg", "删除成功！");  
    } else {
```

```
        map.put("msg", "删除失败！");
    }
    return map;
}

@RequestMapping(value = { "/customer/editCustomer" })
public String toUpdateCustomer(HttpServletRequest request,
    HttpServletResponse response) {
    String pageName = "/customer/customerInfoUpdate";
    if (IsLoginUtils.isLogin(request, response).length() > 0) {
        pageName = IsLoginUtils.isLogin(request, response);
    }
    String customerNo = request.getParameter("customerNo");
    CustomerVO customerVO =
customerService.getCustomerByCustomerNo(customerNo);

    // System.out.println(customerVO.toString());
    request.setAttribute("customerVO", customerVO);
    return pageName;
}

@RequestMapping(value = { "/customer/esc" })
public String Esc(HttpServletRequest request, HttpServletResponse
response) {
    String pageName = "index";
    if (IsLoginUtils.isLogin(request, response).length() > 0) {
        pageName = IsLoginUtils.isLogin(request, response);
    }
    // 判断登录权限 决定返回的页面
```

```
User user = (User) request.getSession().getAttribute("u");
if (user != null && user.getGrade() == 1) {
    pageName = "/customer/customerInfoSearch";
}
return pageName;
}
}

/**
 * 订单管理 前端控制器
 *
 * @author 段刘振
 *
 */
@Controller
public class OrderController {

    @Resource(name = "orderService")
    private OrderService orderService;

    @RequestMapping(value = { "/order/toOrderAdd" })
    public String toOrderAdd(HttpServletRequest request, HttpServletResponse
response) {
        String pageName = "/order/orderInfoAdd";
        if (IsLoginUtils.isLogin(request, response).length() > 0) {
            pageName = IsLoginUtils.isLogin(request, response);
        }
        return pageName;
    }
}
```

```
@RequestMapping(value = { "/order/getAllCustomer" })
@ResponseBody
public Map<String, Object> getAllCustomer(HttpServletRequest request,
HttpServletResponse response) {
    Map<String, Object> map = new HashMap<>();
    List<Customer> list = orderService.getAllCustomer();
    map.put("list", list);
    return map;
}

@RequestMapping(value = { "/order/getProduct" })
@ResponseBody
public Map<String, Object> getAllProduct(HttpServletRequest request,
HttpServletResponse response) {
    Map<String, Object> map = new HashMap<>();
    String productTypeNo = request.getParameter("productTypeNo");
    List<Product> list = orderService.getProduct(productTypeNo);
    map.put("list", list);
    return map;
}

@PostMapping(value = { "/orderInfo/save" })
public String doOrderAdd(HttpServletRequest request, HttpServletResponse
response, @Validated Order order) {
    String pageName = "/order/orderInfoAdd";
    if (IsLoginUtils.isLogin(request, response).length() > 0) {
        pageName = IsLoginUtils.isLogin(request, response);
    }
}
```

```
        orderService.doOrderAdd(order);

        return pageName;
    }

    @RequestMapping(value = { "/order/toOrderInfo" })
    public String toOrderInfo(HttpServletRequest request, HttpServletResponse
response) {
        String pageName = "/order/orderInfo";
        if (IsLoginUtils.isLogin(request, response).length() > 0) {
            pageName = IsLoginUtils.isLogin(request, response);
        }
        return pageName;
    }

    @RequestMapping(value = "/order/getAllOrderInfo")
    @ResponseBody
    public Map<String, Object> getCustomer(HttpServletRequest request,
HttpServletResponse response) {
        Map<String, Object> map = new HashMap<>();
        Integer startRow = Integer.parseInt(request.getParameter("startRow"));
        Integer pageSize = Integer.parseInt(request.getParameter("pageSize"));

        String orderNo = request.getParameter("orderNo");
        String customerName = request.getParameter("customerName");
        String productName = request.getParameter("productName");
        String orderTime = request.getParameter("orderTime");

        // 获得满足条件的总的记录数
```

```
        int dataNumber = orderService.getDataNumber(orderNo, customerName,
productName, orderTime);

        // System.out.println("获得满足条件的总的记录数" + dataNumber);

        List<OrderVO> list = orderService.getOrder(startRow, pageSize,
orderNo, customerName, productName, orderTime);

        map.put("dataNumber", dataNumber);

        // System.out.println("总记录数: " + dataNumber);

        // System.out.println("当前页及记录数: " + list.size());

        map.put("list", list);

        return map;
    }
}
```

```
@RequestMapping(value = { "/order/orderInfoByOrderNo" })

public String orderInfo(HttpServletRequest request, HttpServletResponse
response) {

    String pageName = "/order/updateOrder";

    if (IsLoginUtils.isLogin(request, response).length() > 0) {
        pageName = IsLoginUtils.isLogin(request, response);
    }

    String orderNo = request.getParameter("orderNo");

    OrderVO orderVO = orderService.getOrderVOByOrderNo(orderNo);

    if (orderVO != null) {
        request.setAttribute("orderVO", orderVO);
    }

    return pageName;
}
}
```

```
@RequestMapping(value = { "/order/getOrder" })

@ResponseBody
```

```
public Map<String, Object> getOrder(HttpServletRequest request,
HttpServletRequest response) {
    Map<String, Object> map = new HashMap<>();
    String orderNo = request.getParameter("orderNo");
    OrderVO orderVO = orderService.getOrderVOByOrderNo(orderNo);
    if (orderVO != null) {
        map.put("orderVO", orderVO);
    }
    return map;
}

@RequestMapping(value = { "/order/esc" })
public String orderEsc(HttpServletRequest request, HttpServletResponse
response) {
    String pageName = "index";
    if (IsLoginUtils.isLogin(request, response).length() > 0) {
        pageName = IsLoginUtils.isLogin(request, response);
    }
    // 判断登录权限 决定返回的页面
    User user = (User) request.getSession().getAttribute("u");
    if (user != null && user.getGrade() == 1) {
        pageName = "/order/orderInfo";
    }
    return pageName;
}
}

/**
 * 产品信息管理 前端控制器
```

```
*
* @author 段刘振
*
*/

@Controller

public class ProductController {

    @Resource(name = "productService")

    private ProductService productService;

    @Resource(name = "customerService")

    private CustomerService customerService;

    @RequestMapping(value = { "/product/proSearch" })

    public String productInfo(HttpServletRequest request, HttpServletResponse
response) {
        String pageName = "index";

        if (IsLoginUtils.isLogin(request, response).length() > 0) {
            pageName = IsLoginUtils.isLogin(request, response);
        }

        // 判断登录权限 决定返回的页面

        User user = (User) request.getSession().getAttribute("u");

        if (user != null) {
            pageName = "/product/productInfo";
        }

        return pageName;
    }

    @RequestMapping(value = "/product/getProductType")
```



```
@ResponseBody
public Map<String, Object> getProductType() {
    Map<String, Object> map = new HashMap<String, Object>();
    List<ProductType> list = productService.getProductType();
    if (list.size() > 0) {
        map.put("list", list);
    }
    return map;
}

@RequestMapping(value = "/product/getAllProduct")
@ResponseBody
public Map<String, Object> getCustomer(HttpServletRequest request,
HttpServletResponse response) {
    Map<String, Object> map = new HashMap<>();
    Integer startRow = Integer.parseInt(request.getParameter("startRow"));
    Integer pageSize = Integer.parseInt(request.getParameter("pageSize"));
    String productNo = request.getParameter("productNo");
    String productTypeNo = request.getParameter("productTypeNo");
    String productName = request.getParameter("productName");
    String productArea = request.getParameter("productArea");
    String productOwner = request.getParameter("productOwner");

    // 获得满足条件的总的记录数
    int dataNumber = productService.getDataNumber(productNo,
productName, productArea, productOwner, productTypeNo);
    // System.out.println("获得满足条件的总的记录数" + dataNumber);
    List<ProductVO> list = productService.getProduct(startRow, pageSize,
productNo, productName, productArea,
```

```
        productOwner, productTypeNo);
    map.put("dataNumber", dataNumber);
    // System.out.println("总记录数: " + dataNumber);
    // System.out.println("当前页及记录数: " + list.size());
    map.put("list", list);
    User u = (User) request.getSession().getAttribute("u");
    map.put("grade", u.getGrade());
    return map;
}

/**
 * 修改产品信息
 *
 * @param request
 * @param response
 * @return
 */
@RequestMapping(value = { "/product/updateProduct" })
public String updateProduct(HttpServletRequest request,
    HttpServletResponse response) {
    String pageName = "/product/updateProduct";
    if (IsLoginUtils.isLogin(request, response).length() > 0) {
        pageName = IsLoginUtils.isLogin(request, response);
    } else {
        String productNo = request.getParameter("productNo");
        if (productNo != null && productNo != "") {
            ProductVO productVO =
productService.getProductByProductNo(productNo);
            // 将产品对象信息存放在 request 里

```

```
        request.setAttribute("productVO", productVO);
    }
}

return pageName;
}

@RequestMapping(value = { "/product/toAddProduct" })
public String addProduct(HttpServletRequest request, HttpServletResponse
response) {
    String pageName = "/product/addProduct";
    if (IsLoginUtils.isLogin(request, response).length() > 0) {
        pageName = IsLoginUtils.isLogin(request, response);
    }
    return pageName;
}

@PostMapping(value = { "/product/doAddProduct" })
public String doAddProduct(HttpServletRequest request,
HttpServletResponse response, @Validated Product product) {
    String pageName = "/product/addProduct";
    if (IsLoginUtils.isLogin(request, response).length() > 0) {
        pageName = IsLoginUtils.isLogin(request, response);
    }
    productService.doAddProduct(product);
    return pageName;
}

/**
 * 跳转至产品类型管理页面
```

```
*  
  
* @param request  
* @param response  
* @return  
*/  
  
@RequestMapping(value = { "/product/toProductType" })  
public String toProductType(HttpServletRequest request,  
HttpServletRequest response) {  
    String pageName = "/product/productType";  
    if (IsLoginUtils.isLogin(request, response).length() > 0) {  
        pageName = IsLoginUtils.isLogin(request, response);  
    }  
    return pageName;  
}  
  
/**  
 * 验证产品类型名称  
 *  
 * @param request  
 * @param response  
 * @return  
 */  
  
@RequestMapping(value = { "/product/checkProductTypeName" })  
@ResponseBody  
public Map<String, String> checkProductTypeName(HttpServletRequest  
request, HttpServletResponse response) {  
    Map<String, String> map = new HashMap<String, String>();  
    String productTypeName = request.getParameter("productTypeName");  
    int result = productService.checkProductTypeName(productTypeName);
```

```
        if (result > 0) {
            map.put("msg", "no");
        } else {
            map.put("msg", "ok");
        }
        return map;
    }

    /**
     * 新增产品类型
     *
     * @return
     */
    @PostMapping(value = { "/product/addProductType" })
    public String addProductType(HttpServletRequest request,
        HttpServletResponse response) {
        String pageName = "/product/productType";
        if (IsLoginUtils.isLogin(request, response).length() > 0) {
            pageName = IsLoginUtils.isLogin(request, response);
        }
        String productTypeName = request.getParameter("productTypeName");

        // 生成 10 位主键
        String productTypeNo = RandomNumberUtils.randomNumber(10);
        int result = productService.checkProductTypeName(productTypeName);
        if (result == 0) { // 类型不存在
            productService.addProductType(new ProductType(productTypeNo,
                productTypeName));
        }
    }
}
```

```
    }  
    return pageName;  
}  
  
/**  
 * 获得所有的产品类型  
 *  
 * @param request  
 * @param response  
 * @return  
 */  
  
@RequestMapping(value = { "/product/getAllProductType" })  
@ResponseBody  
    public Map<String, Object> getAllProductType(HttpServletRequest request,  
    HttpServletResponse response) {  
        Map<String, Object> map = new HashMap<String, Object>();  
        List<ProductType> productTypeList =  
productService.getAllProductType();  
        map.put("productTypeList", productTypeList);  
        return map;  
    }  
  
/**  
 * 获得所有的产品单位  
 *  
 * @param request  
 * @param response  
 * @return  
 */
```

```
@RequestMapping(value = { "/product/getUnit" })
@ResponseBody
public Map<String, Object> getProductUnit(HttpServletRequest request,
HttpServletResponse response) {
    Map<String, Object> map = new HashMap<String, Object>();
    List<Unit> unitList = productService.getProductUnit();
    map.put("unitList", unitList);
    return map;
}

/**
 * 修改产品信息
 *
 * @param request
 * @param response
 * @param productVO
 * @return
 */

@PostMapping(value = { "/product/doUpdateProduct" })
public String doUpdateProduct(HttpServletRequest request,
HttpServletResponse response,
    @Validated ProductVO productVO) {
    productService.updateProduct(productVO);
    String pageName = "index";
    if (IsLoginUtils.isLogin(request, response).length() > 0) {
        pageName = IsLoginUtils.isLogin(request, response);
    }
    // 判断登录权限 决定返回的页面
    User user = (User) request.getSession().getAttribute("u");
```

```
        if (user != null && user.getGrade() == 1) {
            pageName = "/product/productInfo";
        }
        return pageName;
    }

    /**
     * 根据产品编号 获得产品信息
     *
     * @return
     */
    @RequestMapping(value = { "/product/selProductInfo" })
    @ResponseBody
    public Map<String, Object> getProductByNo(HttpServletRequest request,
        HttpServletResponse response) {
        Map<String, Object> map = new HashMap<String, Object>();
        String productNo = request.getParameter("productNo");
        ProductVO productVO =
productService.getProductByProductNo(productNo);
        map.put("productVO", productVO);
        return map;
    }
}

    /**
     * 报价信息管理 前端控制器
     *
     * @author 段刘振
     */
```



```
*/  
  
@Controller  
public class QuotationController {  
  
    @Resource(name = "quotationService")  
    private QuotationService quotationService;  
  
    @RequestMapping(value = "/quotation/quotationInfoAdd")  
    public String toQuotationInfoAdd(HttpServletRequest request,  
    HttpServletResponse response) {  
        String pageName = "/quotation/quotationInfoAdd";  
        if (IsLoginUtils.isLogin(request, response).length() > 0) {  
            pageName = IsLoginUtils.isLogin(request, response);  
        }  
        return pageName;  
    }  
  
    @RequestMapping(value = "/quotation/quotationInfo")  
    public String toQuotationInfo(HttpServletRequest request,  
    HttpServletResponse response) {  
        String pageName = "/quotation/quotationInfo";  
        if (IsLoginUtils.isLogin(request, response).length() > 0) {  
            pageName = IsLoginUtils.isLogin(request, response);  
        }  
        return pageName;  
    }  
  
    @PostMapping(value = { "/quotation/doAdd" })
```

```
public String doQuotationInfoAdd(HttpServletRequest request,
HttpServletResponse response,
    @Validated Quotation quotation) {
    String pageName = "/quotation/quotationInfoAdd";
    if (IsLoginUtils.isLogin(request, response).length() > 0) {
        pageName = IsLoginUtils.isLogin(request, response);
    }
    quotationService.quotationAdd(quotation);
    return pageName;
}

@RequestMapping(value = "/quotation/getAllQuotation")
@ResponseBody
public Map<String, Object> getQuotation(HttpServletRequest request,
HttpServletResponse response) {
    Map<String, Object> map = new HashMap<>();
    Integer startRow = Integer.parseInt(request.getParameter("startRow"));
    Integer pageSize = Integer.parseInt(request.getParameter("pageSize"));

    // 获得满足条件的总的记录数
    int dataNumber = quotationService.getDataNumber();
    // System.out.println("获得满足条件的总的记录数" + dataNumber);
    List<QuotationVO> list = quotationService.getQuotation(startRow,
pageSize);
    map.put("dataNumber", dataNumber);
    // System.out.println("总记录数: " + dataNumber);
    // System.out.println("当前页及记录数: " + list.size());
    map.put("list", list);
    User u = (User) request.getSession().getAttribute("u");
```

```
        map.put("grade", u.getGrade());  
        return map;  
    }  
  
    @RequestMapping(value = { "/quotation/del" })  
    @ResponseBody  
    public Map<String, String> delQuotation(HttpServletRequest request,  
    HttpServletResponse response) {  
        Map<String, String> map = new HashMap<String, String>();  
        String quotationNo = request.getParameter("quotationNo");  
        int result = quotationService.delQuotation(quotationNo);  
        if (result > 0) {  
            map.put("msg", "删除成功！");  
        } else {  
            map.put("msg", "删除失败！");  
        }  
        return map;  
    }  
}
```

英文资料翻译

基于 Mybatis Plus 的 Web 后端模板引擎的研究与应用(译文)

摘

要

:

在web开发中，对后端进行编码，包括业务逻辑和接口集成，需要花费大量的时间，具有机械化和高重复性。为了解决后端web编码中存在的问题，本文分析了SSM（Spring、SpringMVC和MyBatis）框架的优缺点和原端分离的开发模式。通过在PHP web开发中集成模板引擎，实现了web后端代码生成技术，有利于实现web编码的人工智能，节省了软件工程的时间。

关键词：后端开发，代码生成，模板引擎

一、 导言

软件与网络背景

随着信息技术的发展，软件的使用挤进了人们的生活，包括娱乐和办公。与硬件不同，软件由一系列计算机数据和指令表组成。软件不仅是计算机中的程序，文件也是软件的组成部分。软件包括操作软件和应用软件。众所周知，android和ios操作系统是最流行的操作软件。Web是一种存在于操作软件中的应用软件。

文献综述

软件开发包括文献综述应用开发、程序开发、电话应用开发和 web 开发。本文介绍了 web 服务的发展（称为后端开发）基于 Java Web。在网络的发展中，设计模式包括 MVC（模型，视图和控制器）模式，Web 窗体，网页等，但是开发人员最常用的模式是 MVC 模式。有学者认为 MVC 模式具有低耦合性的优点，并且该项目中的数据高度一致。但也有人认为这使得编码冗余，评估数据效率低下。基于 MVC 模式的开发模式是一种前端与后端分离的模式。

使用这种模式,专业人员将负责 web 开发的不同部分,开发分为前端和后端,与接口交互。目前,最流行的技术框架是 SSM(Spring, SpringMVC and MyBatis) 框架近年来得到了广泛的应用。不同于 SSH(Spring, Struts and Hibernate) 框架,它能更好地处理请求和响应,适合 RESTful 编码风格和 MVC 模式。然而,这种开发模式的缺点是每个模型的接口具有相同的功能接口,在项目中产生大量的高重复代码,浪费了开发人员大量的时间。在 PHP 的开发过程中,这个问题是由模板引擎来处理的。由于 Java 和 PHP 的不同,更少的 Java 开发人员使用模板引擎生成代码。这项技术确实有效地解决了这个问题。

二、Web开发的运行机制

MVC 模式

MVC 模式是 20 世纪 80 年代出现的一种交互式界面结构设计模式,也是一种有效的设计模式。MVC 设计模式强调开发人员根据用户的输入、数据模型和信息显示以模块化的方式设计软件。它将应用软件分为以下三个部分。

模型是应用软件的业务逻辑,包括数据和模型。模型是应用程序的核心功能,它封装了数据和问题解决过程。用户通过调用控制器调用模型,使模型能够为视图提供数据,减少代码冗余。

视图是应用程序软件的一部分,用于处理显示不同信息的数据。视图是用户看到并交互的界面。对于视图,忽略数据和问题解决过程,用户可以通过调用控制器来更新视图,从中向模型发送服务请求。

控制器是应用软件中处理交互的部分。控制器将输入的鼠标、键盘等信息转换为模型和视图的服务请求,并将模型的变化响应到视图中。用户只能通过控制器与软件交互。

一组视图和控制器组成一个用户界面,一个模型可以有許多视图。如果用户通过控制器更改模型,控制器会将此更改扩展到所有视图,从而实现显示的统一性。

端部分离发展模式

Web 页面是早期的设计模式之一，它要求开发人员能够熟练掌握整个开发过程。它还使开发人员了解了 web 开发的所有技术，导致开发人员之间职责不明确。因此，开发人员需要熟练掌握 web 项目中难以维护的混乱代码。

显然，MVC 模式改变了很多。它最显著的改进是端分离开发模式，即项目开发分为前端开发和后端开发两部分。

在这种开发模式中，前端工程师更关注页面的表达、用户体验、兼容性等，后端工程师更关注高性能、安全性和业务逻辑。显然，与原来的 Web 表单、Web 页面等开发模式相比，这种开发模式并没有降低对开发人员的技术要求，而是提高了开发的标准。

SSM 框架

Spring 是一个开源 java 框架。它是为降低企业项目开发的复杂性而开发的一个框架。Spring 的核心功能是 IOC（控制反转）和 AOP（面向方面编程）。由于这两个核心功能，几乎任何 Java 应用程序都可以从中受益。Spring 是一个轻量级、低侵入性的开源框架，它可以与其他框架兼容。

SpringMVC 是 Spring 的一个子项目，具有与 Spring 相同的低侵袭特性。它是后端开发和前端开发完全分离的框架。也就是说，几乎任何视图都可以使用。此外，Spring 使 java web 的组件在 SpringMVC 中得到更好的管理。

Mybatis 是 Apache 的一个开源项目 iBatis，后来被移植到 google 代码中并重命名为 Mybatis。MyBatis 封装了几乎所有 JDBC 操作，扩展了参数设置和返回结果集的功能。通过解析配置文件和对应的映射文件，扩展了原有的 JDBC 函数，使得应用程序和数据库之间的交互更加简单。

在 SSM 框架下，后端开发分为三个层次。它们是层控制器、层服务和层 DAO。层控制器用于处理相关请求，调用相关服务完成业务逻辑，选择相应视图作为对用户的反馈。服务层封装业务逻辑，调用层 Dao 并预处理异常。MyBatis 集成了层服务，将数据库操作为层 DAO，保存数据。

问题分析

一般来说，SSM 框架的后端开发过程是数据库设计、实体类设计、层 DAO 开发、层服务开发、层控制器开发、前端集成和项目完成。这样，虽然减少了前端和后端的耦合，提高了代码的可维护性，但仍然存在如下问题。

三、研究方法与解决方案

研究方法

有了上述问题，不难发现，编写这样的代码是一种简单的机械劳动。因此，缩短简单 CRUD 代码的开发周期，提高编码效率是主要的解决方案。

同样的问题，另一种编程语言 PHP 有一个非常有效的解决方案，那就是模板引擎。Java 也使用了相应的技术，但由于这两种语言之间的差异，这不是 Java 的常规开发方法。

案例分析

由于 PHP 的模板引擎不同于 Java 所使用的模板引擎，本文忽略了 PHP 模板引擎。Java 的模板引擎主要用于 JSP（Java 服务器页面）。目前只有两个流行的模板引擎：FreeMaker 和 Velocity。这两个模板引擎的编程语言很简单。

在考虑了相关引擎的框架之后，我们找到了一个名为 MyBatis Plus 的开源框架。这个框架是对 MyBatis 的一个增强，它改进了 MyBatis 的功能，但没有改变 MyBatis 的原始结构。通过该框架的扩展功能，我们可以通过模板引擎生成层 DAO、层服务和层控制器的代码。

案例重构

案例重构最重要的是，这是一个开源框架。许多人共享框架的用例，官方网站上有官方文档。我们需要做的是对启动类和模板进行编码。

图 2 控制器层模板

对于启动类，启动类中有 20 多个属性配置。我们为配置选择最重要的属性，而默认情况下配置其他属性以保持开发简单。在这种情况下，我们最终修改了属性并成功地创建了它们，开发人员只需要修改数据库连接相关连接的配置。

对于 MyBatis Plus 的模板，生成的层服务和层 DAO 使用 MyBatis Plus 的模板。我们只需要编写层控制器的模板，它只有五个接口函数。

代码生成后，此模板将生成相应的类。类中有五个函数，分别是插入实体、按 id 更新实体、按 id 删除实体和为页面选择实体。

四、应用程序

数据库设计

在JDK8、Spring Boot2.0、Maven3.6和MySQL5.7的开发环境下，这里是解决用户信息管理问题的一个典型的数据库实例，并建立了相应的数据库表，如表1所示。

列名	数据类型	长度	不允许为空	主键	注释
id	int	11	True	True	User ID
userName	varchar	255	True	False	User Name
password	varchar	255	True	False	User Pwd

表 1 学生表

创造依赖

groudId	artifactId	version
com.baomidou	mybatis-plus-generator	3.0.7.1
org.apache.velocity	mybatis-plus	3.0.7.1
com.baomidou	velocity-engine-core	2.0
org.apache.velocity	mysql-connector-java	8.0.15

表 2 依赖表

从表 2 可以看出，该项目有四个依赖项。前两个依赖项是 MyBatis Plus 的 jar 包。第三个依赖是模板引擎的 jar 包。最后一个是 Java 与数据库的连接。

在创建信息表和依赖项之后，我们将对启动类进行编码。生成器包的代码生成器中有一个启动类，其中有一个主函数，用于配置数据源、模板文件路径和输出文件路径。将模板放置在配置指定的路径下。最后，运行 main 函数完成代码生成。

包	文件
java.per.example.demo.generator	CodeGenerator.java
resource.templatesMybatis	Controller.java

表 3 原始文件表

项目运行后的主要功能变化如下。在表 3 中列出了 case 项目的原始环境，在表 4 中，生成的 java 文件分为三部分。DAO 层的一部分是“java.per.example.demo.mapper”、“java.per.example.demo.entity”和“resource.mapper”中的文件。层服务的一部分是“java.per.example.demo.Service”和“java.per.example.demo.Service.impl”中的文件。包“java.per.example.demo.controller”属于层控制器。

包	文件
java.per.example.demo.controller	UserController.java
java.per.example.demo.entity	User.java
java.per.example.demo.mapper	UserMapper.java
java.per.example.demo.service.impl	UserServiceImpl.java
java.per.example.demo.service	UserService.java
resource.mapper	UserMapper.xml

表 4 新文件表

五、评估和未来工作

通过使用 MyBatis Plus 和开发模板，生成代码。使用该生成器，减少了代码的重复性，缩短了后端开发的周期。该代码生成器可以生成后端开发接口，有利于代码生成的人工智能。不过，也有一些缺点。它不能处理不同开发人员的不同编码风格的问题。此外，仅为 CRUD 生成的接口和一些特殊的接口需求如授权不能生成。

在上述研究中，MyBatis Plus 是这项技术不可或缺的。因此，我们打算减少技术对 MyBatis Plus 的依赖，并将此技术与 MyBatis Plus 分开。至于以后的工作，我们会收集开发人员的案例，分析他们的代码编写习惯。参考 Spring 的工作模型，通过深入学习和神经网络算法分析编码习惯，有利于代码生成的人工智能。

六、致谢

这项研究得到了以下项目的支持和资助。广东省大学生创新创业培训项目（201810566090）、广东海洋大学创新创业培训项目（CXXL2018171）、广东海洋大学大学生创新团队（CXTD2019003）。

七、工具书类

1. Steele J, To N. Android 开发人员的书籍：使用 Android 构建应用程序 SDK[M]. 2010.
2. Armstrong, Eric. Java Web 服务教程[M]. 高等教育出版社, 2003.
3. Wang Y , Guo C , Song L . 基于 j2ee 和 mvc 模式的电子商务系统体系结构[C] 国际电子商务管理会议 & 电子政务. 电气与电子工程师协会, 2009.
4. Ding Y H , Liu C H , Tang Y X . 基于 JAVA 的 MVC 模式[J]. 应用力学与材料, 2012, 198-199:537-541.
5. Han H , Lu J , Lu X . 虚拟接口机[J]. ACM SIGSOFT 软件工程注释, 2002, 27(3):88-92.

6. Lara J A , Lizcano D , María Aurora Martínez, et al. 开发前端 Web 2.0 技术, 在未来的 Internet 中访问服务、内容和事物[J]. 未来一代计算机系统, 2013, 29(5):1184 - 1195.
7. Yunrui Q . 仓库管理系统的前端和后端分离[C] 2018 第 11 届智能计算技术与自动化国际会议 (ICICTA) 。IEEE 计算机协会, 2018.
8. Mitchell J G . 弹簧系统概述[C] Compcon Spring 94, 论文摘要. 电气与电子工程师协会, 2002.
9. Mak G , Long J , Rubio D . Spring @MVC[J]. 2010.
10. Vaidyanath K . MyBatis3 的 Java 持久性——MyBatis 实用指南, 一个简单而强大的 Java 持久性框架[J]. 2013.
11. Bai Y. 开发 Java Web 服务访问数据库[M] 用 Java 实现实用数据库编程. 2011.
12. Zhang D , Wei Z , Yang Y . 基于 Spring MVC 和 Mybatis 的轻量级 MVC 框架研究[C] 2013 年第六届计算智能与设计国际研讨会论文集第 1 卷. 电气与电子工程师协会, 2013.
13. Amanatidis T , Chatzigeorgiou A . PHP web 应用的演进研究. [J]. 信息& 软件技术, 2016, 72(4):48-67.
14. Nakaike T , Kondoh G , Nakamura H , et al. JSP 拆分提高执行性能 [C] 应用与互联网国际研讨会. 电气与电子工程师协会, 2004.

Research and Application of Template Engine for Web Back-end Based on Mybatis-Plus(原文)

Abstract: In web development, it takes much time to code for back-end including business logic and interface integration, which is machanized and high-repeated. In order to address the problem in back-end web coding, this paper analyzes the advantages and disadvantages of SSM(Spring, SpringMVC and MyBatis) framework and development pattern of the original and end separation. By integrating the template engine in PHP web development, we achieve the technology of code generation for back-end of web, which benefits for artificial intelligence of web coding and saves the time on software engineering.

Key words: Back-end Development, Code Generation, Template Engine

I. Introduction

Background of Software and Web

With the development of information technology, the use of software crowds into people's life including in entertainment and office work. Differing from hardware, software is made of a series of computer data and instruction list. Software is not only the program in the computer, files are also the parts of software. Software includes operating software and application software. As we know, android and ios operating system are the popular operating software. Web is a kind of application software existing in the operating software.

Literature Review

Application development, program development, phone application development and web development are included in software development.

This paper introduces the development of web service(which is called back-end development) based on the Java Web. In the development of the web, design patterns include MVC(Model, View and Controller) pattern, Web Form, Web Page and so on, but the most popular pattern used by developers is MVC pattern. Some scholars think that MVC pattern has the advantage of low coupling and the data in this project is highly uniform. However, others think that it makes the coding redundant and the assessed data inefficient. Based on the MVC pattern, the development pattern is a kind of pattern with the separation of front-end and back-end. Using this pattern, the professionals will be responsible for the different parts of web development and the development is divided into front-end and back-end interacted with interfaces. At present, the most popular technical framework is SSM(Spring, SpringMVC and MyBatis) framework and it is widely used in last few years. Differing from SSH(Spring, Struts and Hibernate) framework, it handles requests and responses better and is suitable for RESTful coding style and MVC pattern. Nevertheless, this development pattern has a disadvantage that the interfaces for each Model have the same functional interface, causing so many high-repeated codes in the project and it wastes the developers much time. In the development of PHP, this problem is handled by Template Engine. Because of the differences between Java and PHP, less Java developers generate codes with template engine. This technology does deal with the problem effectively.

II. Operation Mechanism of Web Development

MVC Pattern

MVC Pattern is an interactive interface structure design pattern presented in the 1980s and it is also an effective design pattern. The MVC design pattern emphasizes that developers design their software in a modular

fashion based on users' input, data model and information display. It divides an application software into the following three parts.

Model is the business logic of application software including data and model. Model is the core functionality of the application and it encapsulates the data and problem solving process. User calls the Model by calling Controller so that the Model can provide data for Views to reduce the code redundancy.

View is the part of the application software that handles the display of data, which displays different information. View is an interface seen and interacted by users. For the View, data and problem solving process are ignored, from which users can send service requests to the Model by calling the Controller to update the View.

Controller is the part of the application software which handles interactions. Controller converts the inputted information such as mouse and keyboard into service request for Model and View, and it responses the changes of Model to the View. And users only interact with software by Controller.

One set of View and Controllers make up a user interface and a Model can have many Views. If user change the Model through Controller, Controller will spread this change to all Views, so as to achieve the unity of the display.

Development Pattern of End Separation

Web Page is one of the early design pattern, which requires the developers to be skilled at the whole development process. It also makes the developers know all technology about the web development, resulting the ambiguous duties among the developers. Consequently, developers need to be skilled at the codes in chaos of web project that is hard to be maintained.

Obviously, MVC pattern changes a lot. The most distinct improvement of it is end Separation development pattern, which means that project development are divided into two parts: front-end development and back-end development.

In this development pattern, front-end engineers pay more attention to page expression, user experience, compatibility, etc and back -end engineers are more concerned with high performance, security and business logic. Obviously, compared with the original development pattern like Web Form and Web Page, such a development pattern doesn't reduce the technical requirements for the developers but higher standards.

SSM Framework

Spring is an open source java framework. It is a framework developed to reduce the complexity of enterprise project development. The core functions of Spring are IOC(Inversion of Control) and AOP (Aspect-Oriented Programming). Because of these two core functions, almost any Java application can benefit from it. And Spring is a lightweight and low-intrusive open source framework that makes it compatible with other frameworks.

SpringMVC is a sub-project of Spring with the same low-invasive features as Spring. It is the framework that back-end development and front-end development are completely separated. Namely, almost any View can be used. Besides, Spring makes components of java web are better managed in SpringMVC.

Mybatis is an open source Project iBatis of Apache, which was later migrated to google code and renamed MyBatis. MyBatis encapsulates almost all JDBC operations and extends the functionality of parameter settings and return result sets. By parsing the configuration files and the corresponding

mapping files, the original JDBC function is extended, making the simple interaction between the application and the databases.

With SSM framework, the back-end development is divided into three layers. They are Layer Controller, Layer Service and Layer DAO. Layer Controller is used for processing relevant requests, calling relevant service to complete the business logic and selecting the corresponding view as the feedback to the users. The Service layer encapsulates the business logic, calling Layer Dao and pre-processing the exceptions. MyBatis integrates Layer Service to operate the database to as Layer DAO, persisting the data.

Problem Analysis

In general, the process of back-end development in SSM framework is database design, entity class design, Layer DAO development, Layer Service development, Layer Controller development, integration with front -end and project completion. In this way, although the coupling between the front-end and back-end is reduced and the code maintainability is high, there remain some problems shown as follows.

High code repeatability. In this development pattern, almost every operation of the table in the database has CRUD (Create, Retrieve, Update and Delete). Regardless of Layer DAO, Layer Service or Layer Controller, the code repeatability there is high. Moreover, as the underlying development of back-end, the development cycle of this stage can't be shortened. This kind of development make developers busy coding instead of taking more complex business logic into account.

Long cycle for project development. For enterprise projects, there may be hundreds of tables in the database. It takes a long time to develop the CRUD, which delays the business logic development, causing large cycle development of the project.

III. Research Method and Solutions

Research Method

With the problems above, it is not difficult to find that the writing of such code is simple but a kind of mechanical labor. So the main solution is to shorten the development cycle of the code for simple CRUD and improve the efficiency of coding.

As the same problem, another programming language PHP, has a very effective solution and that is the template engine. The corresponding technology is also used for Java but this is not a regular development method for Java due to the difference between the two languages.

Case Analysis

For the template engine for PHP is different from what is used by Java, the PHP template engine is missed in this paper. Java's template engine is mostly used in JSP (Java Server Pages). Currently there are only two popular template engines: FreeMaker and Velocity. And the programming languages for these two template engines are simple.

After considering the framework to the relevant engine, we found an open source framework called MyBatis-Plus. This framework is an enhancement to MyBatis which improves the functionality of MyBatis but doesn't change the original structure of MyBatis. Through the extended function of this framework, we can generate the code of Layer DAO, Layer Service and Layer Controller by template engine.

Case Refactoring

Most importantly, this is an open source framework. Many people share the use cases of the framework and official documents are on the official website. What we need to work is coding the start-up class and the template.

For the start-up class, there are more than 20 attribute configurations in the start-up class. we choose the most important properties for configuration, while other properties are configured by default to keep the development simple. In this case, we finally modify the properties and successfully created them where developers only need to modify the configuration of the database connection related connections.

For the template of MyBatis-Plus, the generated Layer Service and Layer DAO use the template from MyBatis-Plus. We only need to code the template of the Layer Controller which only has five interface functions.

From Figure.2, we can see the template of the Layer Controller. After the code generation, the corresponding class will be generated by this template. There are five functions in the class including inserting an entity, updating an entity by id, deleting an entity by id and selecting entities for page respectively.

IV. Application

Database Design

With the development environment of JDK8, Spring Boot2.0, Maven3.6 and MySQL5.7, here is a classic case of database that solving the problem of user information management and the corresponding table of database is established as shown in Table.1.

Table.1 Student Table

Column Name	Data Type	Length	Not Null	Primary Key	Annotation
id	int	11	True	True	User ID

userName	varchar	255	True	False	User Name
password	varchar	255	True	False	User Pwd

Creating Dependence

Table.2 dependence table

groudId	artifactId	version
com.baomidou	mybatis-plus-generator	3.0.7.1
org.apache.velocity	mybatis-plus	3.0.7.1
com.baomidou	velocity-engine-core	2.0
org.apache.velocity	mysql-connector-java	8.0.15

From Table.2, there are four dependences for the project. The first two dependences are the jar package of MyBatis-Plus. The third dependence is the jar package of template engine. The last one is the connection between Java and database.

After creating the information table and the dependence, we then code the start-up class. And there is a start-up class in the CodeGenerator of the generator package in which there is a main function and configures the data source, template file path and output file path. Place the template under the path specified by the configuration. Finally, run the main function to complete the code generation.

Table.3 original file table

Package	File
java.per.example.demo.generator	CodeGenerator.java
resource.templatesMybatis	Controller.java

The changes of the project after ran the main function is as follows. From Table.3, original environment of case project is listed and from Table.4, the

generated java files are divided into three parts. The part of Layer DAO is the files in the “java.per.example.demo.mapper”, “java.per.example.demo.entity” and “resource.mapper”. The part of Layer Service is the files in the “java.per.example.demo.service” and “java.per.example.demo.service.impl”. The package “java.per.example.demo.controller” is belong to the Layer Controller.

Table.4 new file table

Package	File
java.per.example.demo.controller	UserController.java
java.per.example.demo.entity	User.java
java.per.example.demo.mapper	UserMapper.java
java.per.example.demo.service.impl	UserServiceImpl.java
java.per.example.demo.service	UserService.java
resource.mapper	UserMapper.xml

V. Evaluation and Future Work

By using MyBatis-Plus and developing template, code generated is created. With this generator, the code repeatability is reduced and the cycle of development is shortened in back-end development. This code generator can generate interfaces of back-end development, which is beneficial for the artificial intelligent of code generation. However, there are some disadvantages. It can't deal with the problem of different coding style of different developers. Besides, the generated interfaces just only for CRUD and some special requirement of interface like granting can't be generated.

In the research above, MyBatis-Plus is indispensable for this technology. So we intend to reduce the dependence of technology on MyBatis-Plus and separate this technology from MyBatis-Plus. As for future work, we will collect the cases of developers and analyze their code writing habits. After

we refer to Spring's working model, coding habits can be analyzed by deep learning and neural network algorithm, which is beneficial to artificial intelligence of code generation.

VI. Acknowledgments

This research is supported and funded by the following projects. Guangdong Province Innovative Entrepreneurship Training Project for College Students (201810566090), Guangdong Ocean University Innovative Entrepreneurship Training Project(CXXL2018171) and Guangdong Ocean University Innovative Team for College Students(CXTD2019003).

VII. References

- 1.Steele J, To N. The Android Developer's Cookbook: Building Applications with the Android SDK[M]. 2010.
- 2.Armstrong, Eric. The Java Web services tutorial[M]. Higher Education Press, 2003.
- 3.Wang Y , Guo C , Song L . Architecture of e-commerce systems based on j2ee and mvc pattern[C] International Conference on Management of E-commerce & E-government. IEEE, 2009.
- 4.Ding Y H , Liu C H , Tang Y X . MVC Pattern Based on JAVA[J]. Applied Mechanics and Materials, 2012, 198-199:537-541.
- 5.Han H , Lu J , Lu X . Virtual Interface Machine[J]. ACM SIGSOFT Software Engineering Notes, 2002, 27(3):88-92.
- 6.Lara J A , Lizcano D , María Aurora Martínez, et al. Developing front-end Web 2.0 technologies to access services, content and things in the future Internet[J]. Future Generation Computer Systems, 2013, 29(5):1184–1195.
- 7.Yunrui Q . Front-End and Back-End Separation for Warehouse Management System[C] 2018 11th International Conference on Intelligent Computation Technology and Automation (ICICTA). IEEE Computer Society, 2018.

- 8.Mitchell J G . An overview of the spring system[C] Compcon Spring 94, Digest of Papers. IEEE, 2002.
- 9.Mak G , Long J , Rubio D . Spring @MVC[J]. 2010.
- 10.Vaidyanath K . Java Persistence with MyBatis 3 - A Practical Guide to Mybatis, a Simple Yet Powerful Java Persistence Framework[J]. 2013.
- 11.Bai Y. Developing Java Web Services to Access Databases[M] Practical Database Programming with Java. 2011.
- 12.Zhang D , Wei Z , Yang Y . Research on Lightweight MVC Framework Based on Spring MVC and Mybatis[C] Proceedings of the 2013 Sixth International Symposium on Computational Intelligence and Design - Volume 01. IEEE, 2013.
- 13.Amanatidis T , Chatzigeorgiou A . Studying the evolution of PHP web applications.[J]. Information & Software Technology, 2016, 72(4):48-67.
- 14.Nakaike T , Kondoh G , Nakamura H , et al. JSP splitting for improving execution performance[C] International Symposium on Applications & the Internet. IEEE, 2004.