**渤海大学本科毕业论文（设计）**

**基于Spring Boot的美酷理发店管理系统的设计与实现**

**The Design and Development of MeikuBarber Shop Management System Based on Spring Boot**

学 院： 信息科学与技术学院

专 业： 计算机科学与技术（软件开发）

学 号： 16301303

学 生 姓 名： 石晴

入 学 年 度： 2016

指 导 教 师： 杨一柳 李梦琪

完 成 日 期： 2020年5月2日

渤海大学

Bohai University

# 摘 要

美发是当今社会不可或缺的热门行业，它看似是一种服务行业但背后的管理业务也不亚于其服务的压力，例如：理发物资的采购入库、货品的出售、会员的注册、顾客的消费、员工的工作状况等。大到盈亏统计，小到每一笔消费，都是需要有记录可以追寻的。繁琐的记录工作无形之中加大了管理人员的压力。

基于此社会背景，本论文以解决上述问题为目的，以理发店的实际管理业务流程为基础业务需求，结合实际情况，设计与实现了一款理发店管理系统，并命名为“美酷理发店管理系统”。此系统涵盖了理发店基本的管理业务，库存管理，会员管理，员工管理，消费管理，盈亏统计等。让繁琐的工作简单化，可以很大程度的减小管理人员的工作压力。

经历多次调研和分析之后，从技术与实际意义两个角度确定系统开发的可行性。本系统最终选择搭建Maven项目工程，采用前后端分离的形式，后端选用了Java语言为本系统的开发语言，选用Spring Boot为开发框架，集成Mybatis，并将系统的交互数据存储到MySQL数据库中。前端选用Vue框架，集成Element组件。

本系统选题来源于生活，从行业现状出发，最终结果达到了缓解理发店管理人员的压力的目的。对于设计而言，实现了简洁，方便，操作性强的特点。对于功能而言，满足于理发店的基本需求。

关键词**：**理发店；系统；管理；Spring Boot

**The Design and Development of MeikuBarber Shop Management System Based on Spring Boot**

# Abstract

Hairdressing is an indispensable hot industry in today's society. It seems to be a service industry, but the management business behind it is no less than the pressure of its service, such as: the purchase and warehousing of hairdressing materials, the sale of goods, the registration of members, the consumption of customers, the working conditions of employees, etc. From profit and loss statistics to every consumption, we need to have records to trace. Tedious work increases the pressure of managers.

Based on this social background, in order to solve the practical problems, this paper designs and implements a barber shop management system, named "meiku barber shop management system", which is based on the actual management business process of barber shop and combined with the actual situation. This system covers the basic management business of barber shop, such as inventory management, member management, employee management, consumption management, profit and loss statistics, etc. Simplify the tedious work and greatly reduce the work pressure of managers.

After many investigations and analyses, the feasibility of system development is determined from two aspects of technology and practical significance. The system finally chooses to build Maven project, which adopts the form of front and back-end separation. The back-end uses Java language as the development language of the system, spring boot as the development framework, integrates mybatis, and stores the interactive data of the system into MySQL database. Vue framework is selected for the front-end to integrate element components.

The topic of this system comes from life, starting from the current situation of the industry, the final result is to alleviate the pressure of barber shop managers. As far as the design is concerned, it realizes the characteristics of simplicity, convenience and strong operability. For function, meet the basic needs of barber shop.

**Key Words：**Barber shop; System; Management; Spring Boot

目 录

[摘 要 I](#_Toc39189003)

[Abstract II](#_Toc39189004)

[1 绪论 1](#_Toc39189005)

[1.1 选题背景 1](#_Toc39189006)

[1.2 课题研究的目的 1](#_Toc39189007)

[2 相关技术和工具 2](#_Toc39189008)

[2.1 技术方面 2](#_Toc39189009)

[2.1.1 Spring Boot框架 2](#_Toc39189013)

[2.1.2 Vue框架 3](#_Toc39189014)

[2.4 开发工具 3](#_Toc39189015)

[2.4.1 开发工具介绍 3](#_Toc39189016)

[2.5 可行性分析 4](#_Toc39189017)

[3 需求分析与设计 6](#_Toc39189018)

[3.1 问题分析 6](#_Toc39189019)

[3.2 需求分析 6](#_Toc39189020)

[3.2.1 登录页面 6](#_Toc39189021)

[3.2.2 功能管理主页面 6](#_Toc39189022)

[3.3 系统功能模块设计 7](#_Toc39189023)

[3.4 数据库设计 8](#_Toc39189024)

[3.4.1 数据库的概念结构设计 9](#_Toc39189025)

[3.4.2 数据库的总体设计 10](#_Toc39189026)

[3.5 系统流程图 12](#_Toc39189027)

[4 开发环境的准备 15](#_Toc39189028)

[4.1数据库安装 15](#_Toc39189029)

[4.1.1 安装数据库 15](#_Toc39189030)

[4.2 开发工具安装及环境变量的配置 15](#_Toc39189031)

[4.2.1 安装Idea 15](#_Toc39189032)

[4.2.2 JDK安装与环境变量配置 15](#_Toc39189033)

[4.2.3 Navicat Premium安装 16](#_Toc39189034)

[4.2.4 VsCode安装 16](#_Toc39189035)

[5 系统设计与实现 17](#_Toc39189036)

[5.1 数据库的实现 17](#_Toc39189037)

[5.2 功能模块的设计与实现 17](#_Toc39189038)

[5.2.1 用户登录 17](#_Toc39189039)

[5.2.2 修改密码 18](#_Toc39189040)

[5.2.3 统计详情模块 19](#_Toc39189041)

[5.2.4 消费管理模块 21](#_Toc39189042)

[5.2.5 会员管理模块 23](#_Toc39189043)

[5.2.6 人员管理模块 24](#_Toc39189044)

[5.2.7 库存管理模块 26](#_Toc39189045)

[6 系统测试 28](#_Toc39189046)

[6.1 系统测试 28](#_Toc39189047)

[6.2 测试发现的问题与改进 29](#_Toc39189048)

[结 论 32](#_Toc39189049)

[参 考 文 献 33](#_Toc39189050)

[附 录 34](#_Toc39189051)

# 1 绪论

## 1.1 选题背景

美发行业在中国有着源远流长的历史，早在清朝就有剃头的说法，到现在演变成我们如今的理发、发型设计、美发。虽然面对长久的发展历史，但是在信息化的现代，相对于国外，并没有全面实现平台管理的自动化、信息化。而美发行业由于互联网技术的介入，使得越来越多的新型技术浮出水面。在如此背景下一个高质量的美发管理系统的性能高低，操作是否简便，是否可以直观的观察到盈利情况，是否可以实现自动化管理便成为重中之重。系统既要前台简洁，还需要后台处理大量的逻辑，并且还有很多功能。如果没有一个准确的、简洁的系统，仅仅依靠人工操作，不仅效率低而且容易出差错。而发展迅速的时代，人们对视觉要求也逐渐增高。面对如此激烈的竞争市场，采用美观、高效率的信息化管理系统更是赢家的选择。

## 1.2 课题研究的目的

在互联网发达的时代，信息化管理逐渐走进美发行业，快节奏的生活使得人们厌恶日复一日做着繁琐而又复杂的操作，一款快捷的、高效率的、多功能的管理系统便直击眼球。

本课题主要的目的就是简化理发店店长的管理工作，方便理发店店长直观的查看收益情况和员工的业绩，同时对于店员来说，提高收银效率。使整个理发店实现信息化管理。美观简洁大气的画面又无形之中为工作人员带来了美好的心情。

# 2 相关技术和工具

一个优秀的项目从最开始需要考虑很多问题。例如，针对美酷理发店管理系统的需求和功能点，选择何种技术进行开发，选用何种开发工具进行编写代码，选用何种数据库来完成数据存储，这些问题都需要反复斟酌。而我们就需要在繁多的技术中找到最适合美酷理发店管理系统的，在繁多的开发工具中，找到相对效率最高的编辑器，只有这样才可以让系统更有含金量，更高效的运行，拥有更多的客户。

## 2.1 技术方面

系统采用前后端分离的方式来实现项目的开发。前后端分离是当今时代项目开发发展的趋势，即前端页面通过ajax调用后端的接口并使用json数据进行交互。前后台关系如下图2.1所示：

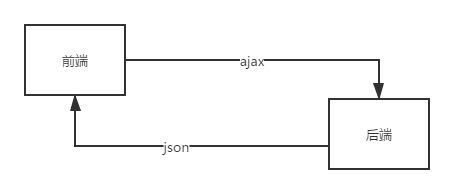


图2.1 前后台关系

而采用这种方式的好处是：

1. 数据的处理可以放到前台完成，这样可以大大地降低服务器的压力；
2. 后台报错不会直接反映到前台，这样错误的接收相对来说比较友好；
3. 前后台的职责清晰，可维护性比较高。

### 2.1.1 Spring Boot框架

本系统后端使用Spring Boot框架进行开发。Spring Boot是基于Spring之后更简单的一种框架。Spring是一个开源框架，它是在2003 年兴起的一个轻量级的Java 开发框架[1]。Spring是为了解决企业级应用开发的复杂性而创建的，使用Spring可以让简单的Java Bean实现之前只有EJB[2]才能完成的事情。而Spring Boot的设计是为了让用户尽可能快的执行 Spring 应用程序并且尽可能减少用户的配置文件。简单来说就是Spring Boot其实不是什么新的框架，它默认配置了很多框架的使用方式，就像Maven整合了所有的jar包，而Spring Boot则整合了所有的框架。

介绍一下Spring Boot的特性：

1. 无需手动管理依赖jar包的版本：Spring-boot-\*的jar包已对一些功能性jar包进行了集成；
2. 独立运行的Spring项目：Spring Boot默认将应用打包成一个可执行的jar包文件，用java -jar命令即可运行应用。或者在应用项目的主程序中运行main函数即可，不需要依赖tomcat、jetty等外部的应用服务器[3]。
3. 自动配置，无需xml：Spring Boot尝试根据你添加的jar依赖自动配置你的应用,使用@EnableAutoConfiguration或者@SpringBootApplication注解，配合@Configuration注解类，即可完成自动配置。
4. 外部化配置：Spring Boot可以使用properties文件，YAML文件，环境变量，命令行参数等来外部化配置；
5. 嵌入式servlet容器：在项目打包时会自动将servlet容器嵌入到应用jar中。

### 2.1.2 Vue框架

本系统前台使用Vue框架[4]进行开发。Vue是华裔工程师尤雨溪开发的前端JS库，它相对于普通前台开发来说，拥有以下特点：

1. 遵循MVVM模式[5]；
2. 编码简洁，体积小，运行效率高；
3. 它本身只关注UI，可以轻松引入Vue插件和其它第三方库开发项目。
4. MVVM模式是指模型-视图-视图模型。模型是指后端传递的数据；视图是我们所能看到的画面；视图模型是核心所在，是连接模型和视图的桥梁[6]。

## 2.4 开发工具

### 2.4.1 开发工具介绍

1. 后端编辑器：Idea

Idea的全称是IntelliJ Idea，它是一款轻量级的代码编辑器，它的智能化编码给程序员带来了非常大的便利。且具备以下优点：

1. 自动补充未完成的类或方法；
2. 代码不规范编辑器会自动加亮显示；
3. 占用空间小；
4. 自动检查代码，不使用的代码会给予提示。
5. 前端编辑器：VsCode

VsCode是一款强大的开源的、跨平台的代码编辑器，它拥有繁多的实用的插件，可以根据自己的需求下载。并且方便前台调试。

1. 数据库：MySQL

MySQL是一种开源的、免费的网络数据库。相对于其他数据库来说，它的特点是体积小、速度快

1. JDK：JDK1.8

JDK是由sun公司提供的Java开发工具包[7]。它为Java程序开发提供了编译环境和运行环境，一切基于Java开发的程序所需的代码编写都依赖于它。使用JDK可以将Java程序编译为字节码文件，即.class文件。且具备以下特点：

1. 速度更快：由于底层结构和JVM的改变，使得JDK1.8的速度提高。
2. 代码更少（增加了新的语法 Lambda 表达式）：增加新特性Lambda表达式的内部类改造，使得代码在书写上变得更加简洁。
3. 强大的 Stream API：增加了核心功能，使得代码调用方式变得更加简洁。
4. 便于并行
5. 最大化减少空指针异常 Optional。
6. 图形化工具：Navicat Premium

Navicat Premium是一款拥有很多粉丝的数据库图形化工具，它设计简单，适合新手操作，支持多种格式的数据迁移[8]。

1. 项目管理工具：Maven

Maven是Apache下的一个纯Java开发的开源项目，是项目构建和管理工具[9]。它的核心功能是合理叙述项目间的依赖关系，，就是通过pom.xml文件的配置获取jar包，而不用手动去添加jar包。它具备以下特点：

1. 使用Maven管理项目都有着同样的项目结构。
2. 便于统一维护jar包。

## 2.5 可行性分析

系统的正常运行性能不能仅仅是确定开发语言和开发工具这么简单，最重要的是要从经济、法律、技术等多个角度来完成可行性分析，以此来保证系统开发的可行性。针对于本系统选用的框架以及开发工具来说，具体分析如下：

本系统的开发仅仅需要一台电脑来作为开发的环境，而且全部的开发工具以及数据库等都是免费的，所以在经济层次上是可行的，并且可以接受；在技术方面vue和spring boot也都是免费的，而且在开发过程中所查的资料也是免费的，所以技术方面也可行；另外，本系统不涉及相关法律问题，在法律角度更是没问题的；本系统开发之后运行也仅仅需要一台服务器即可完成运行，所以在运行方面也是乐观的。

综上所述，本系统的可行性是良好的。

# 3 需求分析与设计

在系统还未进行开发之前，我们务必要对即将开发的系统进行全方位的思考，主要针对系统的功能和性能做出全方面的分析和设计。我们要将每个需求细化开来，找到需求点，并找出满足需求即实现功能的最佳方法和设计方案。

系统的开发过程其实就是满足需求的过程，需求是一切的根源，而设计方案就是为了更好的、更完美的、更高效的实现需求。这就是为什么找到需求是这么的重要。只有找到需求点，才能针对需求点进行开发设计，有条不紊地完成系统的开发，否则一切的开发都是徒劳的、毫无意义的。

## 3.1 问题分析

在进行开发之前我对项目进行了大体的规划，列出了以下问题：

1. 登录注册问题：是否采取普遍的员工店长注册账号，之后每个人登录系统。通过我的实地考察发现大多数的理发店仅仅能满足拥有一台电脑，用来记录消费情况，所以普遍的每个人都登录自己账号的方式不可取，所以我规定了两种身份：店长，员工。所有员工公用一个账号，而店长拥有自己的账号，拥有一些员工没有的权限。
2. 数据储存问题：怎么储存数据，怎么设计数据库性能更好。便于维护最终我采取某些表用单值代码表来存取相关的内容，以便于更简洁的与前台交互。

## 3.2 需求分析

本系统整理如下需求，需要实现的功能点如下所示：

### 3.2.1 登录页面

1. 实现两个角色的登录功能；
2. 修改密码功能；
3. 需要输入验证码验证功能；
4. 页面右上角动态展示登录人角色功能。

### 3.2.2 功能管理主页面

1. 统计详情模块：
2. 页面第一个模块展示近一年的收入情况概述；
3. 页面第二个模块展示当前年份前五个月每个月的收入及盈利情况的柱状图；
4. 页面第三个模块展示所有员工当月的消费记录数量；
5. 修改会员折扣按钮，实现动态改变会员折扣。
6. 消费记录模块：
7. 点击消费记录模块展示所有的消费记录（分页，每页8条数据）；
8. 页面右上角有添加图案，实现添加消费记录功能；
9. 添加消费记录以弹出框形式展示；
10. 添加消费记录的弹出框应包含以下信息项：电话号，会员的基本信息（姓名，），消费信息（），取消、查询和提交按钮。
11. 会员管理模块：
12. 点击会员管理展示会员信息；
13. 实现修改和删除会员的功能；
14. 实现用姓名，手机号模糊查询功能；
15. 添加充值按钮，对会员的会员卡进行充值功能。
16. 人员模块：
17. 展示员工的基本信息；
18. 实现对员工的基本信息进行增加、修改、删除操作。
19. 库存模块：
20. 展示商品的库存情况；
21. 对商品库存进行加减；
22. 通过手机号查询库存数功能；
23. 商品的模糊查询。

## 3.3 系统功能模块设计

本系统从登录功能开始，登录功能分为三？个部分，分别是：注销和登录操作。本系统一共有两种角色：店长和员工。店长拥有最大的权限，店长和员工的账号密码是默认的，登录页面有一个按钮用来修改账号的密码。登录之后系统会根据账号密码判断登录人的身份，从而赋予不同的权限，进行不同的操作。登录的用户还可以通过注销安全退出。

进入系统之后店长拥有全权限可以看到五个模块，分别是：统计详情，消费管理，会员管理，人员管理，库存管理。统计详情从两个方面进行展示，一是对店内收入角度进行数据统计，二是员工的业绩角度进行数据统计。消费管理主要的功能是：首先会展示所有人的消费记录，之后可以进行添加消费记录操作，在添加消费记录的同时会对库存、会员卡余额和员工的业绩进行加减。会员管理主要是可以对会员手机号进行修改，删除会员，添加新会员以及展示会员的基本信息。并且可以对会员卡进行充值。库存管理主要展示店内商品的库存情况，以及进价售价等价格进行记录、运算。人员管理则是对理发店内员工进行管理，记录员工信息，绩效，以及对员工的修改新增删除等操作。而店员的权限仅仅是消费管理、会员管理和库存管理。店员仅可以对这三个管理模块进行相关操作。

功能模块设计如图3.1所示：

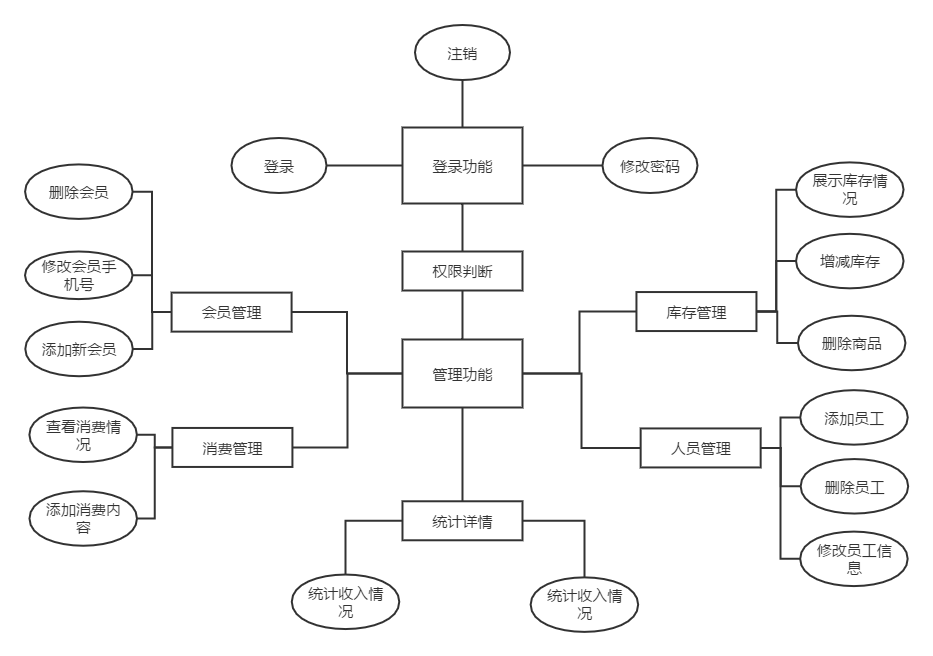


图3.1 功能模块设计图

## 3.4 数据库设计

系统的运行除了需要有良好的运行环境和严谨的逻辑结构。如何储存数据同样也是极为重要的。在开发过程中，数据库是一个极其重要的环节，最开始的数据库设计直接关乎到后期的需求开发工作，这部分则主要阐述本系统的数据库设计。

### 3.4.1 数据库的概念结构设计

在设计数据库之前必须要做的一件事就是理清各实体之间的关系，只有清晰的熟悉了解各个实体之间的关系，才能设计出逻辑清晰的数据库。本系统涉及到的实体有：人员、会员、消费记录、充值记录、商品这几个实体。

人员拥有姓名、性别、手机号、简介、人员编号属性，会员有会员编号、手机号、账户余额、积分、会员等级、性别、姓名、办理日期属性，商品有名称、商品编号、商品进价、商品售价、库存属性，消费记录拥有消费时间、会员价、消费金额、消费记录编号、会员名属性，充值记录拥有充值金额、充值日期、充值记录编号和备注这四个属性。

其中实体中人员和商品通过库存管理是多对多的关系，人员和会员是通过创建的多对多的关系，人员和消费记录通过创建形成多对多的关系，商品与会员之间通过购买形成多对多的关系，会员和充值记录通过充值操作形成了多对多的关系，会员和消费记录是多对多的关系，人员与充值记录通过创建形成多对多的关系。

数据库E-R图如图3.2所示：

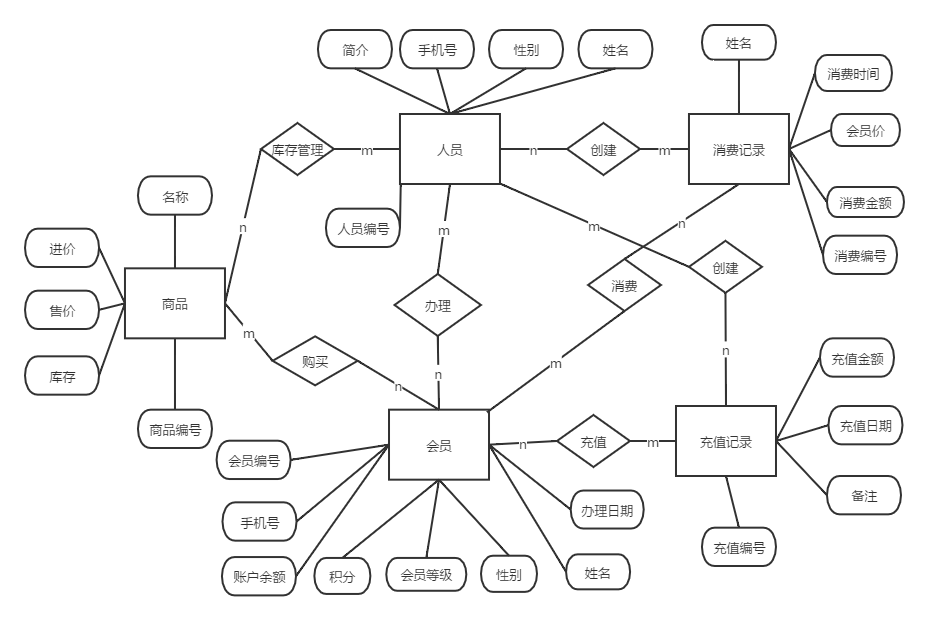


图3.2 E-R图

### 3.4.2 数据库的总体设计

根据本系统的各实体之间的关系，我将本系统分为七个表，分别为会员表、人员表、消费表、充值记录表、商品表、登录表、单值代码表。下面将具体阐述每个表的设计：

1. Vip（会员）表，会员表包含了会员的ID，会员名字，会员性别，电话号码，会员积分，会员等级，会员办理日期，办理人，账户余额这九个字段，会员积分和账户余额设置为数字类型，账户余额可能会出现小数所以设计成float类型。会员办理日期设置为日期类型，其余均为varchar类型。其中会员ID为主键，且不可以为空。其中办理人为外键，关联人员表。如表3.1所示：

表3.1会员表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 允许为空 | 主键 | 含义 |
| cid | varchar | 40 | 否 | 是 | 会员id |
| c\_name | varchar | 40 | 是 | 否 | 会员名 |
| c\_sex | varchar | 40 | 是 | 否 | 会员性别 |
| c\_tel | varchar | 40 | 是 | 否 | 电话号码 |
| n\_integral | int | 40 | 是 | 否 | 会员积分 |
| c\_grade  t\_date  c\_updateName  f\_balance | varchar  date  varchar  float | 40  40  40  40 | 是  是  是  是 | 否  否  否  否 | 会员等级  会员办理日期  办理人  账户余额 |

1. Purchase（消费）表，消费表包含了消费记录ID，消费时间，会员价格，消费金额，消费者ID，商品ID，员工ID，支付方式这八个字段。金额类的数据类型规定为float类型，日期类定义为日期类型，支付方式定义为int类型，存储的是单值代码，其余为varchar类型。其中消费记录ID为主键，且不可为空。其中商品ID、员工ID、消费者ID和支付方式为外键，分别关联商品表、人员表、会员表和单值代码表。如表3.2所示：

表3.2消费表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 允许为空 | 主键 | 含义 |
| cid | varchar | 40 | 否 | 是 | 消费记录id |
| t\_date | date | 40 | 是 | 否 | 消费时间 |
| f\_hymoney | float | 40 | 是 | 否 | 会员价格 |
| f\_money | float | 40 | 是 | 否 | 消费金额 |
| c\_customerId | varchar | 40 | 是 | 否 | 消费者id |
| c\_goodsIdc\_personnelId  n\_codeid | varchar  varchar  int | 40  40  10 | 是  是  是 | 否  否  否 | 商品id  员工id  支付方式 |

1. Personnel（人员）表，人员表包括了员工ID，员工姓名，员工性别，员工电话，简介，薪资这六个字段。其中薪资为float类型，其余均为varchar类型。员工ID为本表的主键，且不为空。如表3.3所示：

表3.3人员表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 允许为空 | 主键 | 含义 |
| cid | varchar | 40 | 否 | 是 | 员工id |
| c\_name | varchar | 40 | 是 | 否 | 员工姓名 |
| c\_sex | varchar | 40 | 是 | 否 | 员工性别 |
| c\_tel | varchar | 40 | 是 | 否 | 员工电话 |
| c\_remarks | varchar | 40 | 是 | 否 | 简介 |
| f\_salary | float | 40 | 是 | 否 | 薪资 |

1. Goods（商品）表，商品表由商品ID，商品名称，商品单价，库存，商品进价这五个字段构成。商品单价和商品进价为float类型，库存为int类型，其余均为varchar类型。其中商品ID为主键，且不可以为空。如表3.4所示：

表3.4商品表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 允许为空 | 主键 | 含义 |
| cid | varchar | 40 | 否 | 是 | 商品id |
| c\_name | varchar | 40 | 是 | 否 | 商品名称 |
| f\_price | float | 40 | 是 | 否 | 商品单价 |
| n\_number | int | 40 | 是 | 否 | 库存 |
| f\_cost | float | 40 | 是 | 否 | 商品进价 |

1. Recharge（充值记录）表，充值记录表包括了消费记录ID，会员号，充值金额，充值方式，充值时间，备注这六个字段。充值金额和充值方式为数字类型，即int类型。其他字段则为varchar类型。消费记录ID为本表的主键，且不可以为空。会员号和充值方式为外键，分别关联会员表和单值代码表。如表3.5所示：

表3.5充值记录表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 允许为空 | | | 主键 | | 含义 |
| cid | varchar | 40 | | 否 | 是 | | 消费记录id | | |
| c\_tel | varchar | 40 | | 是 | 否 | | 会员号 | | |
| n\_rechargemoney | int | 40 | | 是 | 否 | | 充值金额 | | |
| n\_rechargeway | int | 10 | | 是 | 否 | | 充值方式 | | |
| t\_date | date | 40 | | 是 | 否 | | 充值时间 | | |
| c\_remarks | varchar | 200 | | 是 | 否 | | 备注 | | |

1. Login（登录）表，登录表包括了账号，密码，是否是管理员这三个字段。这三个字段都不涉及日期和数字所以定义为varchar类型。账号为本表的主键且唯一，不为空。如表3.6所示：

表3.6登录表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 允许为空 | 主键 | 含义 |
| cid | varchar | 40 | 否 | 是 | 账号 |
| c\_password | varchar | 40 | 是 | 否 | 密码 |
| c\_type | varchar | 40 | 是 | 否 | 是否管理员 |

1. Code（单值代码）表，单值代码表包括了代码值和代码名称两个字段。代码值为主键，唯一，不可以为空。代码值为int类型，代码值名称为varchar类型。如表3.7所示：

表3.7单值代码表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 允许为空 | 主键 | 含义 |
| n\_codeid | int | 10 | 否 | 是 | 代码值 |
| c\_name | varchar | 40 | 是 | 否 | 代码名称 |

## 3.5 系统流程图

本部分将阐述本系统的流程。

访问系统，首先我们看到的是登录页面，在这个页面你可以登录和修改密码。登录之后系统会判断你的身份。

如果登录的是店员账号，则仅仅可以看到会员管理，消费管理，库存管理页面。此时在会员管理页面店员可以添加会员，删除会员和修改会员的手机号功能。库存管理页面店员可以查看现在店内的商品库存情况，还可以添加商品种类。消费页面店员可以添加消费记录，在添加消费记录之前输入手机号查看该客户是否为会员，如果是会员则继续添加消费记录的操作，如果不是会员，店员可以为该客户办理会员，然后继续添加消费记录的操作。在这个页面还实现了为会员充值功能。操作之后点击注销可安全退出。

如登陆的身份为店长，除了员工的权限和可以操作的功能外，还可以看见统计详情和人员管理页面。统计详情页面主要是展示近五个月的收入情况对比和员工的业绩对比。人员管理页面店长可以维护店内人员（店员）的基本信息，例如添加店员，修改店员，删除店员等操作。之后同样可以点击右上角进行注销安全退出系统。美酷理发店管理系统的整个系统的流程图如图3.2所示：

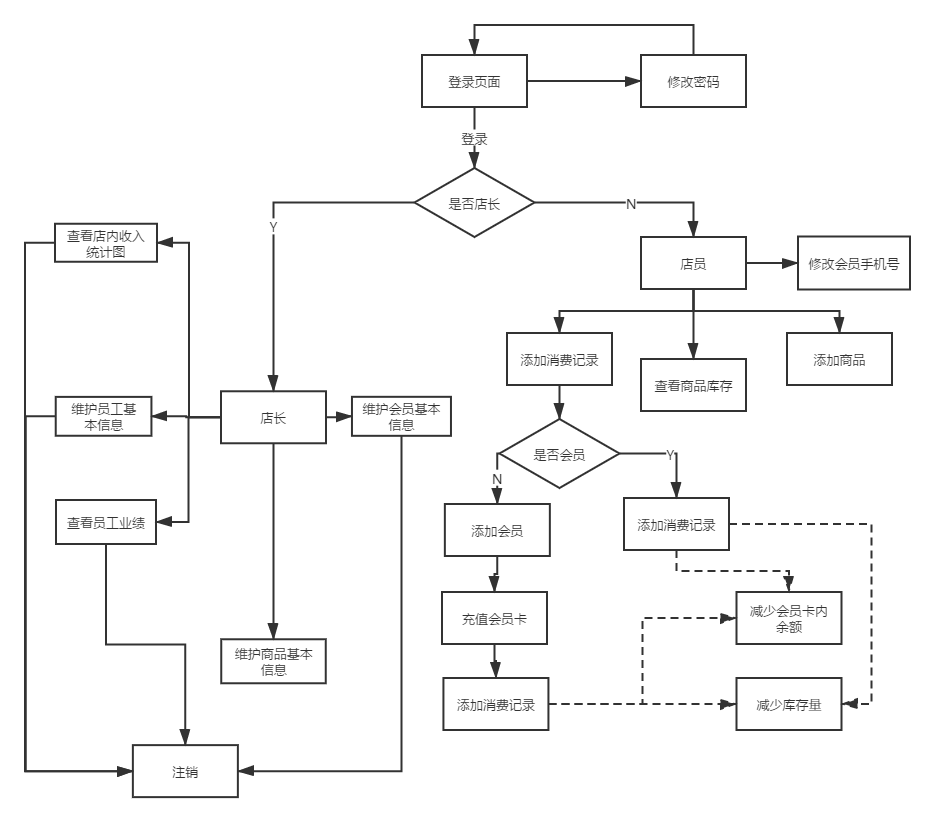


图3.2 系统流程图

# 4 开发环境的准备

## 4.1数据库安装

### 4.1.1 安装数据库

安装数据库之前要保证服务器的硬件配置满足数据库版本的要求，然后到官网下载安装包，点击setup.exe开始安装，之后点击下一步即可。在安装过程中需要注意以下几点：

1. 数据库的安装路径尽量以英文来命名；
2. 检查防火墙端口，如果未成功添加则需要手动添加；
3. 如果安装失败需要重新安装时需要将data文件夹中所有文件删掉。

## 4.2 开发工具安装及环境变量的配置

### 4.2.1 安装Idea

1. 访问官网，下载想要的Idea版本；
2. 点击下载后的安装包；
3. 选择Idea的安装目录；
4. 选择电脑对应的字节；
5. 完成安装。

### 4.2.2 JDK安装与环境变量配置

1. 分别安装JDK和JRE；
2. 在系统变量中新建名为JAVA\_HOME的变量,其变量值为JDK的安装目录；
3. 点击系统变量中Path变量，其变量值指向JDK和JRE的bin目录；
4. 运行cmd输入java -version若显示版本信息,则说明安装和配置成功。成功信息如图4.1所示：



图4.1 成功信息

### 4.2.3 Navicat Premium安装

1. 官网下载Navicat Premium；
2. 选择与电脑对应的字节版本；
3. 点击下一步即可完成安装。

### 4.2.4 VsCode安装

1. 访问官网，下载VsCode；
2. 选择与电脑匹配的字节；
3. 选择合适的安装目录；
4. 点击下一步即可。

# 5 系统设计与实现

## 5.1 数据库的实现

本系统数据库是集成MyBatis方式实现的，通过在pom文件中添加Mybatis依赖，并且在application.yml或者是application.properties配置文件中配置连接数据库的相关信息。本系统是将配置写在了application.properties中。这两个配置文件区别就是，一般application.yml放一些系统相关的配置，像我们自己添加的配置或者一些接口尽量不写在yml文件中。之后我们需要编写Mapper对应的接口，以及配置Mapper.xml文件。

## 5.2 功能模块的设计与实现

### 5.2.1 用户登录

首先映入眼帘的是我们的登录页面，在登录页面我们可以根据账号和初始密码进行登录，因为考虑到实际情况中，普遍理发店的电脑是所有员工一起用，而且如果每个人都有一个账号，每个人使用系统都需要登录的话，很浪费时间。所以本系统初始化了两个角色：店长和店员。在用户输入账户密码之后，需要输入正确的验证码才能正确的登录到系统。如果密码失败或者验证码错误则会对应提示密码或验证码错误。如果验证通过，则会在系统的右上角展示登录人的角色。

用户登录页面如图5.1所示：



图5.1 用户登录主页面

用户登录部分涉及到的最重要的就是对验证码的校验以及获取用户输入的账号密码到后台比对，并且将用户的信息存到缓存中，以便进入系统后展示在右上角。具体实现代码如下所示：

login() {

let checkResult = this.checkNull();

if (!checkResult) {

return;

}

if (this.test()) {

let url = '/axios/api/Login/' + this.input + '/' + this.password

axios.get(url).then(response => {

let ms = response.data;

if (ms != null && ms != "") {

// 将对象转换成json存在缓存中

window.localStorage.setItem("loginer", JSON.stringify(ms));

this.$router.push({ name: "home" });

} else {

this.$message.error("账号或密码错误！");

}

})

.catch(error => {

console.log(error);

});

}

},

### 5.2.2 修改密码

因为考虑到实际，只给系统两个账号两个角色，我们最开始会给用户初始密码，所以会涉及到后续用户真正使用的时候需要将初始密码进行修改，所以系统实现了修改密码的功能。页面展示如图5.2所示：

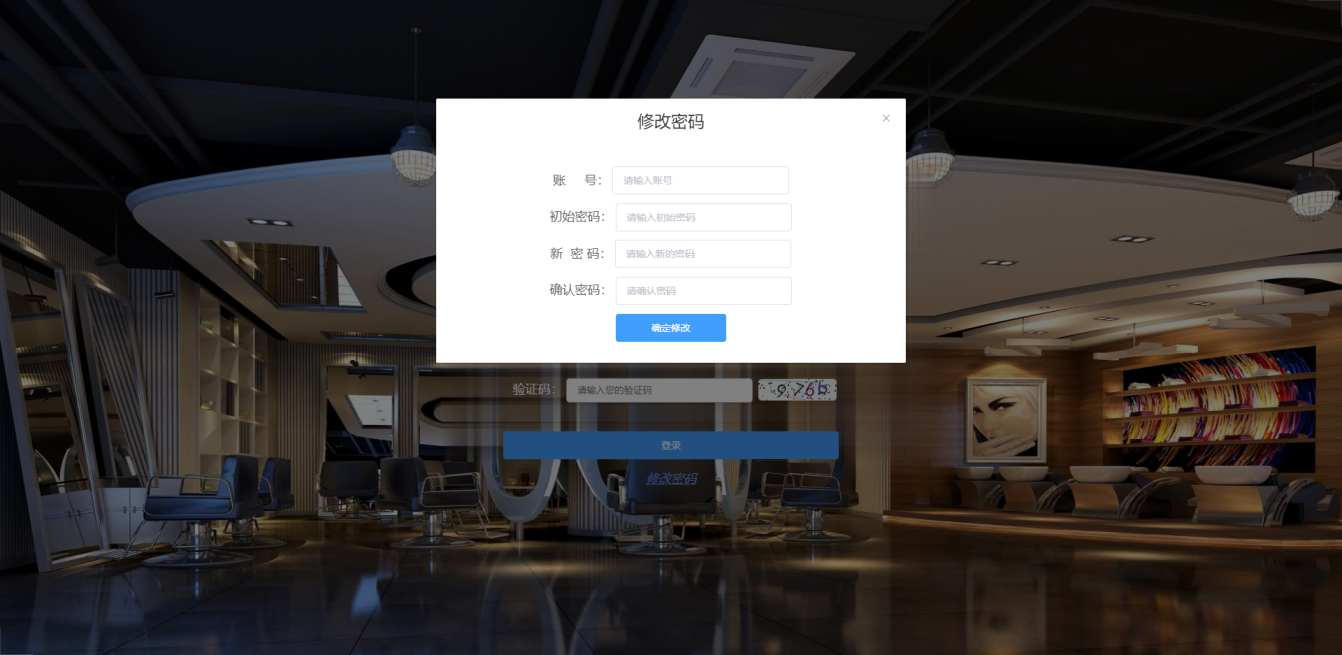


图5.2 修改密码页面

修改密码功能，因为按钮出现在登录页也就意味着用户可能没登录就会点击，所以在修改密码，系统会先让用户输入账号原密码去后台校验是否有这个人，如果有才会允许用户修改密码。之后会让用户填两遍修改后的密码。

### 5.2.3 统计详情模块

统计模块主要分为三个部分：概述、近五个月盈利收入分析、员工业绩分析。概述主要动态的统计了当前年份的总收入金额和盈利情况；近五个月则是动态的具体的统计了近五月的收益情况；员工业绩分析则是统计了店内各个员工当前月份的工作情况。统计详情模块是仅店长可以看到的，店长登入系统后首先看到的就是统计详情页面，这样可以使店长更直观多角度的地了解店内的收益情况。

点击柱状图，饼状图图例即可筛选掉不想看的指标，即展示其余的指标的统计信息，当鼠标经过数据柱时，会展示当前柱的数据。

统计详情如图5.3所示：

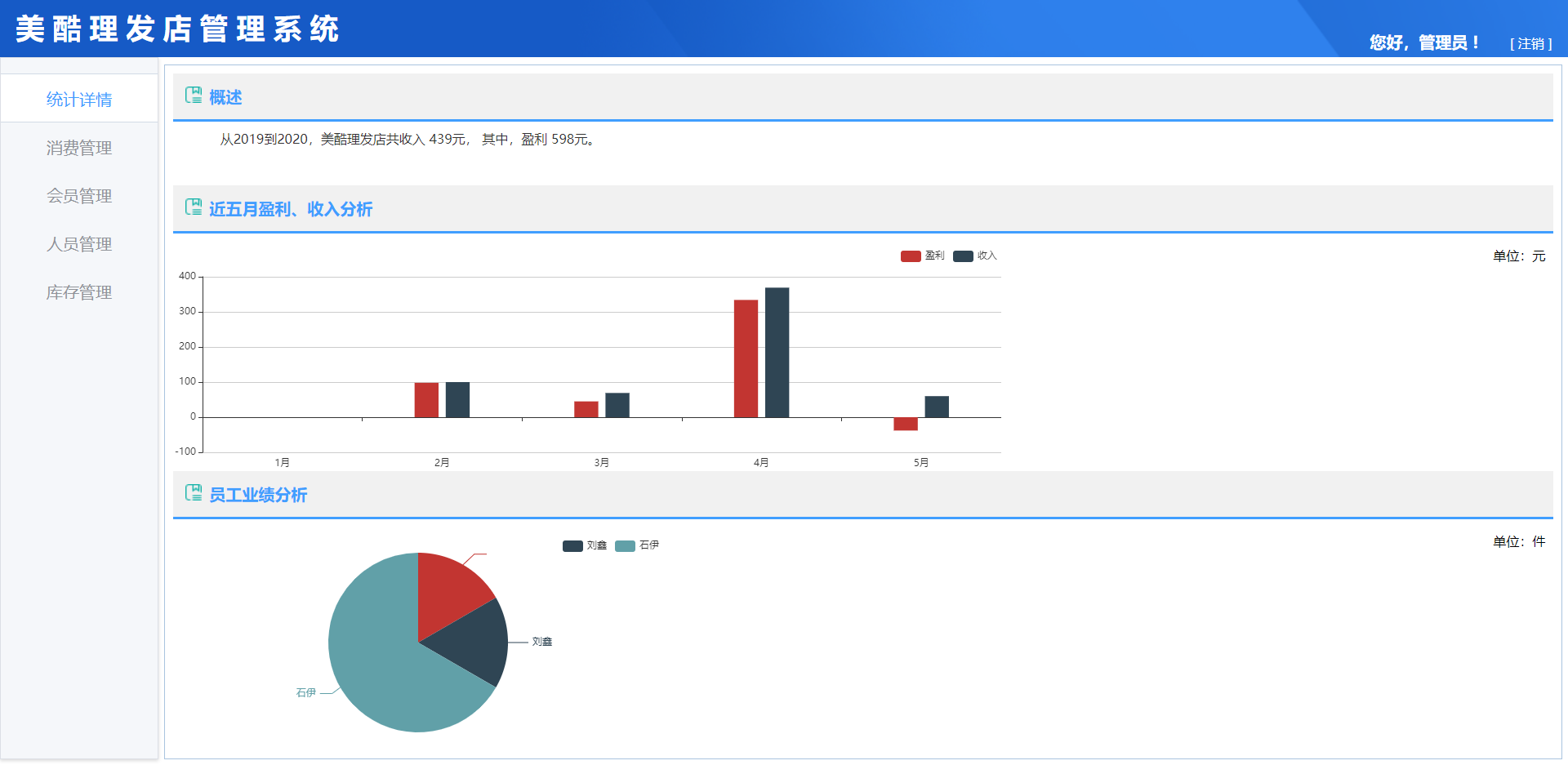


图5.3 统计模块页面

饼状图部分是用echarts实现的，其中最难的部分就是echarts的属性，布局。以下便是饼形图的绑定数据以及属性的设置：

getOptions(bztData) {

    var servicedata = [];

    for (var i = 0; i < bztData.length; i++) {

      var obj = new Object();

      obj.name = bztData[i].name;

      obj.value = bztData[i].cou;

      servicedata[i] = obj;

    }

    let orgOptions = {

      title: {

        text: '员工业绩分析',

        subtext: '美酷',

        left: 'center'

      },

      tooltip: {

        trigger: 'item',

        formatter: '{a} <br/>{b} : {c}件 ({d}%)'

      },

      legend: {

        orient: 'horizontal',

        left: 'right',

        top: 100,

        data: bztData.name,

      },

      series: [

        {

          name: '本月员工业绩情況',

          type: 'pie',

          radius: '55%',

          center: ['50%', '60%'],

          data: servicedata,

          emphasis: {

            itemStyle: {

              shadowBlur: 10,

              shadowOffsetX: 0,

              shadowColor: 'rgba(0, 0, 0, 0.5)'

            }

          }

        }

      ]

    };

    return orgOptions

  },

### 5.2.4 消费管理模块

消费管理模块主要针对的是会员的消费情况进行展示。首先点击消费管理映入眼帘的是所有会员与非会员的消费记录。在本页右上角有两个按钮，分别是折扣修改和添加消费记录。

消费管理页面如图5.4所示：

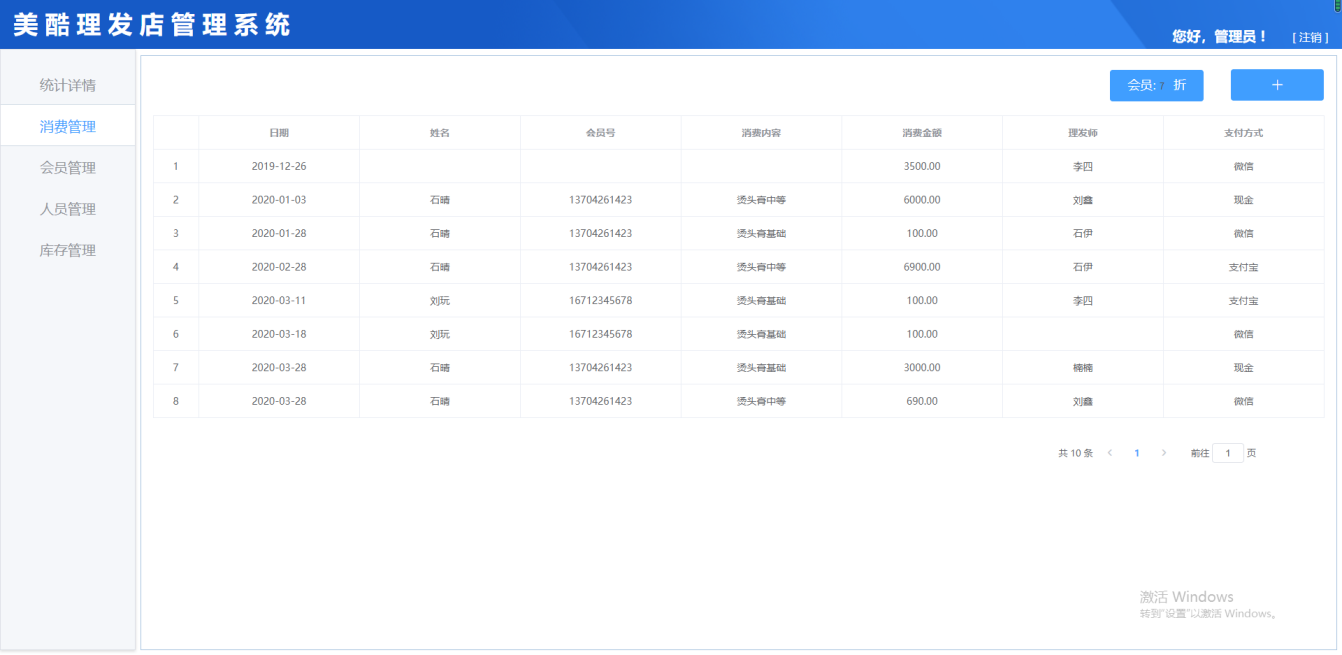


图5.4 消费管理页面

页面展示的数据核心代码如下所示：

getData() {

axios

.post("/axios/api/getXfxx")

.then(response => {

if (response.data.code == "200") {

this.xfjlList = response.data.data;

for (var i in this.xfjlList) {

if(this.xfjlList[i].id == '0000'){

this.xfjlList.splice(i,1)

}

// 处理时间类型数据

this.xfjlList[i].data = util.getTimeY(this.xfjlList[i].data);

}

}

})

.catch(error => {

console.log(error);

});

},

首先第一个按钮默认展示当前会员的折扣，当用户点击数字之后会变成可输入模式，即用户可以自由输入某个时间段的会员折扣，而当焦点失去之后会自动将用户修改的折扣存到数据库中。并且提示修改折扣成功。

第二个按钮是添加消费记录按钮。店员可以先根据顾客提供的电话号查询该顾客是否拥有本店会员，如果没有会员则系统会提示暂无此会员，此时店员可以引导顾客到会员管理页面办理会员；如果有会员则弹出框会对应展示会员信息，接下来就是选择消费的项目，服务的理发师名称，以及支付方式。弹出框底部如果是会员则会展示折扣价格，非会员则展示消费金额。点击确定之后就会关掉弹出框，消费记录就会添加一条记录，并且后台会对会员积分、会员卡余额和商品库存进行修改。

添加消费页面记录如图5.5所示：



图5.5 添加消费记录页面

### 5.2.5 会员管理模块

会员管理主要是对会员的操作。进入该模块首先看到的是会员信息。在左上角是一个查询功能，可以根据姓名、会员号、办理人姓名来进行模糊查询，默认姓名查询。

在每一条信息后面有一个操作栏，操作栏的功能包括充值、修改会员信息、删除会员、查看会员充值记录。充值和修改会员信息会弹出输入框提示充值金额和需要修改的会员信息，默认会员积分、账户余额和办理人是不可以修改的。删除会员会二遍提示是否删除该会员确定的话才会删除。

右上角是一个添加会员功能，在结算时如果是非会员店员可以引导顾客办理会员，即在会员管理页面右上角按钮点击添加。添加新会员默认积分等级都为零，办理日期为当日且不可修改。

添加页面如图5.6所示：



图5.6 添加会员页面

### 5.2.6 人员管理模块

人员管理即是对店员的管理，这一模块仅对店长可见，只有店长可以查看员工的信息。本模块实现的是主要是对店员的增加，修改店员信息，删除店员功能。

人员管理如图5.7所示：

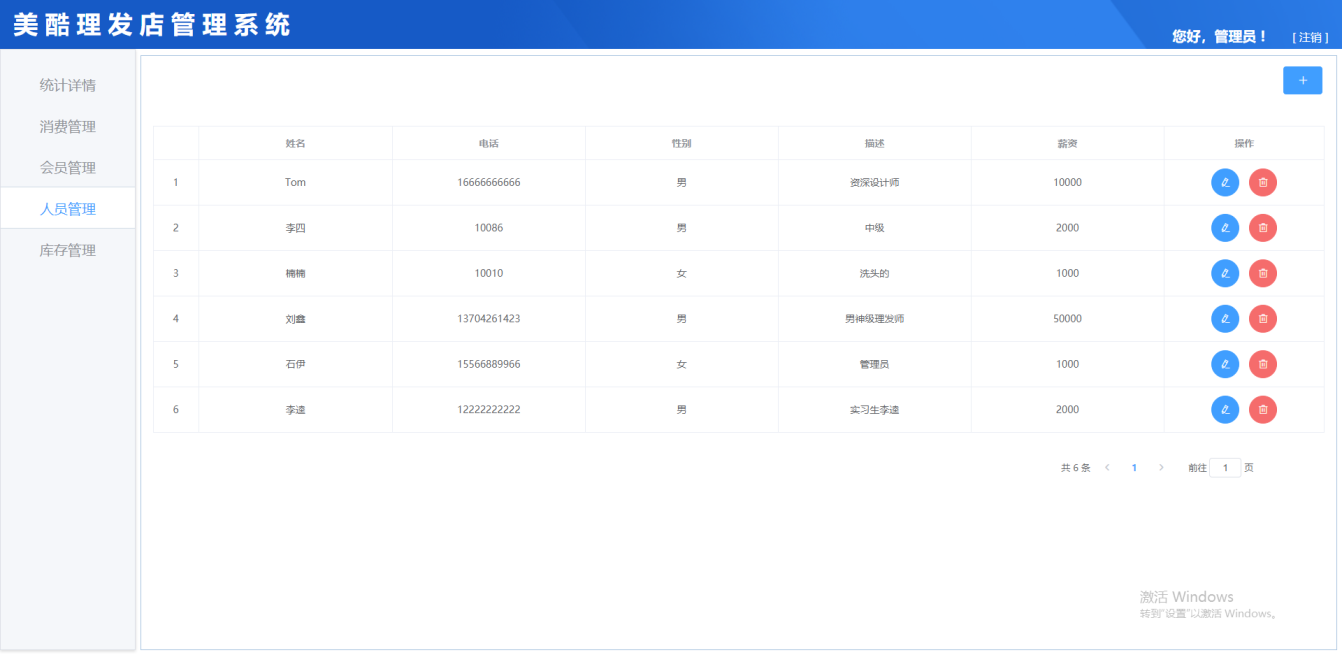


图5.7 人员管理页面

此模块比较核心内容就是后台的处理，以下为部分代码：

@RestController  
@RequestMapping(value = {"api"}, produces = "application/json;charset=UTF-8")  
@Api(tags = {"人员管理"})  
public class UserController {  
  
 @Autowired  
 private UserService userService;  
 private static final Logger logger = LoggerFactory.getLogger(UserController.class);  
  
 /\*\*  
 \* 新增员工账号  
 \*/  
 @PostMapping(path = "/insertYg")  
 @ApiOperation(value = "新增员工账号", notes = "新增员工账号")  
 public ResultData insertYg(@RequestBody User user){  
 try {  
 userService.insertYg(user);  
 } catch (Exception e) {  
 logger.error("新增员工出错", e);  
 return ResultData.error(CodeMsg.SERVER\_ERROR);  
 }  
 return ResultData.success(CodeMsg.SUCCESS);  
  
 }

}

### 5.2.7 库存管理模块

库存管理是店员店长共同都可见的，本模块是对库存进行管理，包括日常进货，对商品进行价格记录等等。

库存管理页面如图5.8所示：

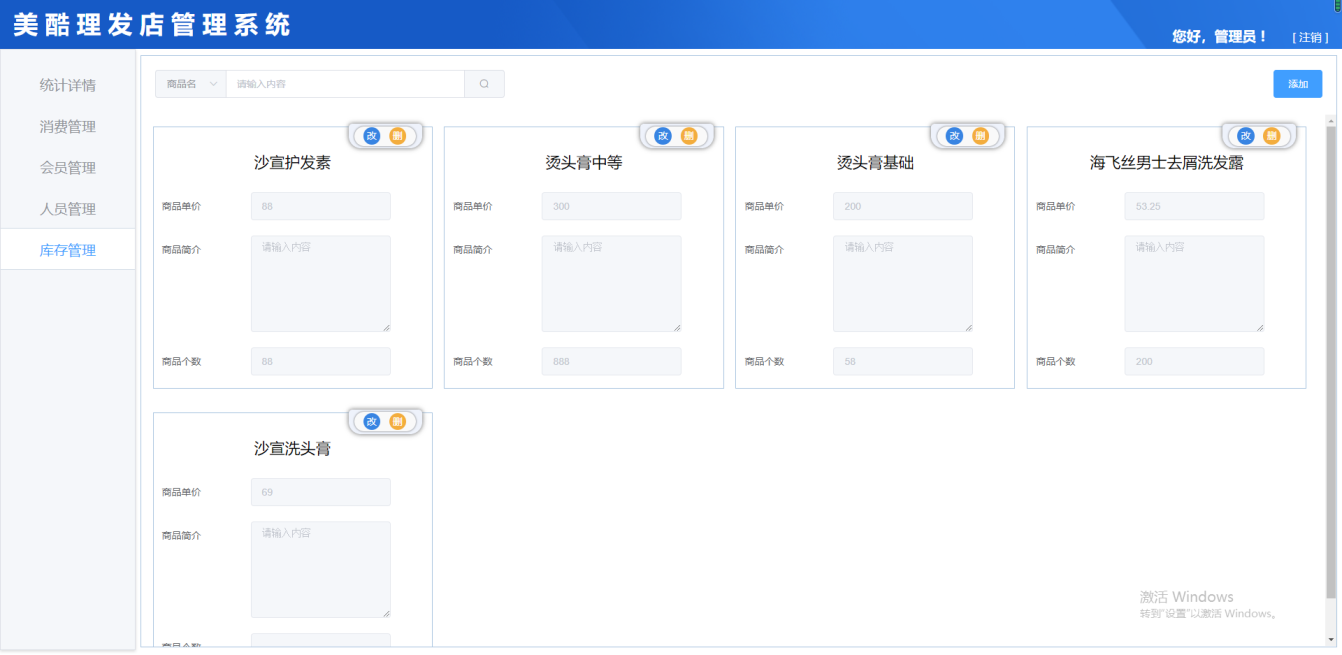


图5.8 库存管理页面

左上角实现了模糊查询，可以根据商品名进行模糊查询。右上角则是添加商品按钮，实现对新产品的入库功能。

页面上黄色的“删”字按钮是将商品下架功能。下架之后将不可以再看到这个商品。蓝色“改”字按钮是对商品的介绍以及商品售价进行修改，点击之后弹出输入框，直接修改然后确定即可。

修改如图5.9所示：



图5.9 修改商品信息弹出框

# 6 系统测试

## 6.1 系统测试

系统测试顾名思义是测试现有的系统是否已经实现了需求规划的功能，页面布局是否与最初始的视觉设计的一致，性能方面是否达标，在正式的真实环境下是否能按照预想的正常运行，本系统采用的是白盒测试、黑盒测试、安全性测试来完成整体的测试。白盒测试即用代码来测试系统功能的实现，通俗地讲就是把系统透明化，运用白盒测试测试系统在运行过程中可能会发生的问题。对每个部分都需要进行测试，最后需要对系统进行整合进行整体测试。而黑盒测试则是不需要代码，就像将系统放进黑盒子里面，只关注结果，需要实际去点击每一个功能，看是否会有异常。安全性测试即是对系统的安全性进行测试，包括是否会对用户信息有加密处理。

具体黑盒测试如下所示：

1. 登录：测试密码错误时提示是否友好；是否可以拦截用户的操作；登录页面是否符合设计；登陆成功后是否拥有对应的权限；登录成功后是否将用户信息展示在系统的右上角；密码框是否采用的加密框；密码所支持的格式等。
2. 修改密码：测试是否校验是否存在该用户；测试是否有再次确认的输入框；测试页面的易用性，是否与主系统页面风格一致；前后台交互过程中，对用户的信息是否加密；修改之后是否回到登录页面；修改密码是否成功。
3. 注销：按钮是否生效；能否回到登录页面；再次登录能否正常；切换登录人是否可以正常赋予权限。
4. 功能模块：主页面展示风格是否一致；页面布局、色彩、字体大小是否合理；点击功能是否有明显的报错；展示数据是否可以正常展示；删除某个记录对应数据库是否是删除的某条记录；添加记录时文本框的最值前框是否合理；模糊查询是否准确；左侧导航栏是否可以正常切换，右侧是否可以正常展示对应模块内容；统计页面是否动态展示当前年份的收入情况；添加消费记录时总金额显示是否根据是否会员联动；总金额是否根据选择的项目不同自动增加减；对应会员的积分、账户余额、库存是否有对应的修改。

安全性测试包括：

检验是否对用户信息加密处理；密码框是否是不支持复制粘贴的；用户信息在储存的过程中是否安全；在未登陆的情况下，直接访问系统主页面是否可以拦截，并返回到登录页面；对爆破破解系统是否做了对应处理。

## 6.2 测试发现的问题与改进

经过全方面的测试主要发现以下问题：

1. 登录页面验证码样式与整体不符，需要与整体对齐。并且页面尺寸需要调整，避免出现滚动条。如图6.1所示：



图6.1 登录页缺陷

1. 统计页面数据分割失误，Y轴数据展示不全。如图6.2所示：

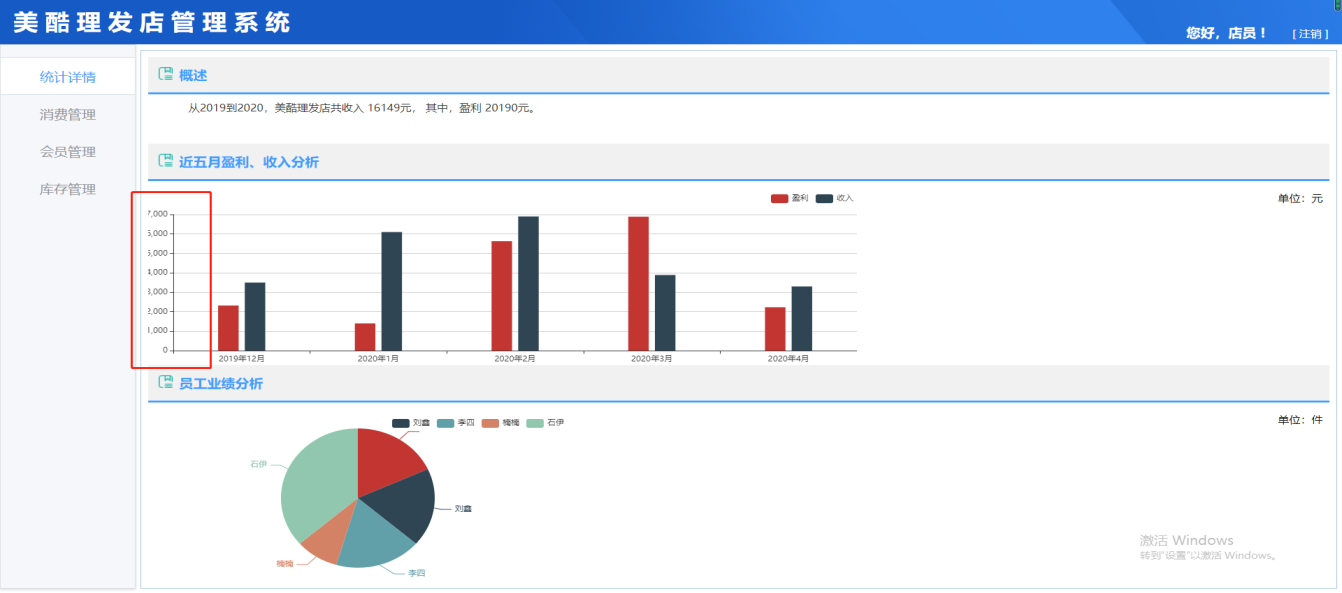


图6.2 统计页缺陷

1. 在添加消费记录时，点击消费项目时正常，但是点击理发师的时候再次点开下拉框时，会出现默认全选状态。预期结果只选中一个理发师，一个理发师名字为蓝色样式。如图6.3所示：



图6.3 消费管理页面缺陷

1. 库存管理页面，点击头部筛选进行模糊查询时方法不生效。如图6.4所示：

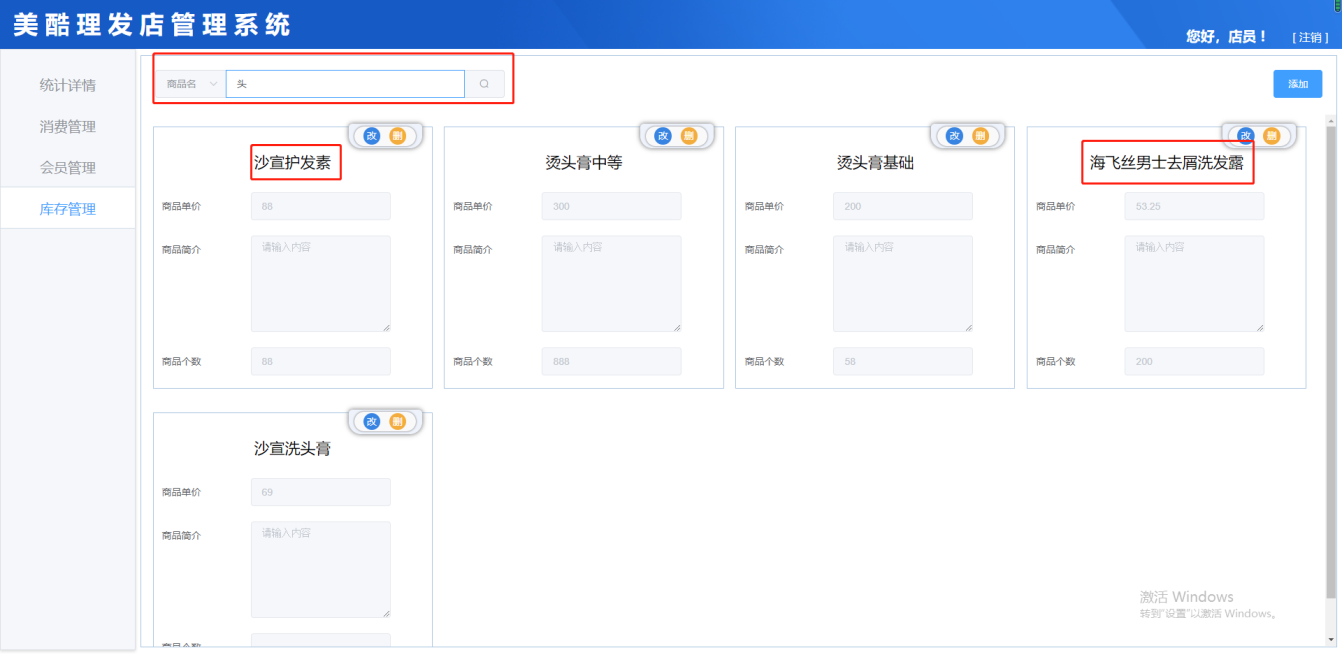


图6.4 库存管理页面缺陷

经过全面的测试未发现致命性错误，功能未实现等。修改之后针对问题点进行复测，之后准备发布包与相关脚本以及实施文档，准备一套接近客户环境的服务器，按照实施文档将程序部署到服务器上，完成整个项目的搭建从无到有的过程，之后将全部测试流程重新执行一遍，查看是否发现问题。存在问题则进行迭代修改，最后将所有问题暴露之后并且修改好之后，进行最后一次收敛测试。

本系统整个流程已完成，收敛测试时并未发现其他问题。

# 结 论

本系统的选题来源于高速发展的网络时代，针对于线上理发店的管理进行了系统的研究。就目前美发行业的压力进行了论述，采用当今流行的技术实现了轻量级的美发管理系统。减轻了理发店店员在每天疲惫的工作后的收银压力，使他们仅用很短的时间就可以实现订单的添加，提高收银效率，简洁大方的画面也可以给店员美好心情做一个铺垫；同时店长在管理店铺时省去很多的统计时间，可以在页面直观的看到近一段时间以来店铺收益情况以及店员的业绩，从而实现真正的网络化、便捷化。

在视觉方面，设计简洁，主要的信息突出，有重点的前提下也不失简洁的风格，用户更容易接受。

在功能实现方面，不仅仅满足于店员，更满足店长的统计需求。并且操作简单。

总体来说，本系统达到了预期效果，实现了高效工作。其次，本系统作为第一版本，后续版本可能会添加预约管理功能，真正实现客户店员店长三位一体化，更具体的解决预约问题。

# 参 考 文 献

1. 袁绪峰.基于Spring框架的AOP编程[J].计算机与现代化,2006(1):118-120.
2. 王永和.张劲松.邓安明.et al.Spring Boot研究和应用[J].信息通信,2016(10):91-94.
3. 王鹏强. 基于vue的MVVM框架的研究与分析[J]. 电脑知识与技术, 15(11):103-104+106.
4. 杨新艳.于伟涛.基于Maven的轻量级Java软件开发研究[J].科技传播,007(17):134-135.
5. 萨师煊.数据库系统概论（第四版），高等教育出版社.2016.5.
6. 刘斌著.精通Web 整合开发.电子工业出版社.2015.5.
7. 刘勇军.Java Web核心编程技术.电子工业出版社.2016..
8. S. Vidal,A. Bergel,J.A. Díaz-Pace,C. Marcos. Over-exposed classes in Java: An empirical study[J]. Computer Languages, Systems &amp; Structures.2016.

# 附 录

1. 系统源程序：

源码参见光盘

1. 系统使用说明书：
2. 运行环境：
3. 硬件平台：
4. CPU：酷睿i3.3.0GHZ
5. 内存：4G以上
6. 硬盘：320GB
7. 软件平台：
8. 操作系统：Windows7以上
9. 开发工具：Eclipse
10. 数据库：Oracle
11. 关于开发：

在Windows7以上系统安装，并安装好数据库和Tomcat。

1. 关于登录：

拥有初始密码，输入正确的用户名和密码，后续可自行更改密码。