|  |  |
| --- | --- |
| **序号（学号）：** | 071540403 |

**长春光华学院毕 业 设 计**

|  |
| --- |
| 基于JavaEE技术的Unique烘焙店管理  系统的设计与实现 |

|  |  |
| --- | --- |
| **姓 名** | 李婷 |
| **教 学 院** | 电气信息学院 |
| **专 业** | 计算机科学与技术 |
| **班 级** | 计算15404 |
| **指导教师** | 李梦琦 工程师 |
|  | 恽鸿峰 讲师 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2019 | 年 | 06 | 月 | 14 | 日 |

基于JavaEE技术的Unique烘焙店管理系统的设计与实现

[摘要] 近年来,随着人民生活水平的提高，社会对烘焙业的需求量也越来越大，伴随着消费的持续增长，烘焙行业已经成为现代社会中发展最快的行业之一。市场竞争愈演愈烈，要想在这场残酷的游戏中不被踢出局,无论是在经营管理还是人员服务方面都需要进一步加强。

原有的手工记账已经无法满足管理者经营的需求,以往用纸质的方式存储信息,在管理方面存在很大的不便。这就意味着烘焙店在管理上需要摒弃传统的管理手段，采用当前主流的信息化管理方式 ,广泛拓展经营空间,进一步强化管理、降低成本、堵漏节流。本系统是基于B/S模式开发的项目,系统主要提供商品信息管理、人员管理、商品订购、订单处理、商品库存查看等功能模块。人机交互界面友好,为使用人员提供简单、便利的操作界面,数据传入传出流达到可靠、快速、规范的标准,数据精准,制表灵活,具有高度的自适应性,为今后的系统扩充做基石。烘焙店后台管理系统从根本上讲是从人工管理模式到信息化管理手段的一个过度,以信息网络化、智能化的开发理念为主导。烘焙店管理系统是基于J2EE的WEB开发项目,从可实现需求因素考虑,后台基于SpringBoot框架结合MyBatis，通过阿里的Druid 数据源访问MySQL数据库，使用SpringBoot内嵌的Tomcat容器，前台基于titles框架运用传统的JSP技术进行数据展示。本系统能够运行在Windows、Linux操作系统下,充分发挥了JAVA语言跨平台的特性。

[关键词] 烘焙店 信息管理系统 J2EE MySQL

The Design and Implementation of Unique Bakery

Management System Based on JavaEE Technology

**[Abstract]** In recent years, with the improvement of people's living standards, the social demand for baking industry is also growing. In order to survive in the fierce market competition, it is necessary to improve management awareness in the aspects of management and service.

The original manual bookkeeping has been unable to meet the needs of management activities, a large number of paper customer information, product information, order information, in the management of great inconvenience.

This means that bakery management needs to use more effective means of information, widely expand the operating space, further strengthen management, reduce costs, plug leakage and throttle. This system is a network project based on B/S mode. The system mainly provides commodity information management, personnel management, cake ordering, order, inventory management and other functions. The user interface design process fully adds the human nature idea, provides the simple and convenient operation interface for the management personnel, the data input achieves the reliable, the rapid, the standard, the statistics is accurate, the tabulation is flexible, has the high adaptability, will make the cornerstone for the future expansion. Bakery management system is a simplified and intelligent management mode. Bakery management system is a WEB development project based on J2EE. Considering the requirements of realization, the background is based on SpringBoot framework combined with MyBatis, and MySQL database is accessed through ali's Druid data source. Tomcat container embedded in SpringBoot is used. This system can run in Linux, Windows various operating systems, make full use of JAVA cross-platform features.

**[Keywords]** bakery, information management system, J2EE, MySQL

目 录

[第1章 绪论 1](#_Toc11072340)

[1.1 研究背景 1](#_Toc11072341)

[1.2 国内外研究现状 1](#_Toc11072342)

[1.2.1 国外研究现状 1](#_Toc11072343)

[1.2.2 国内研究现状 1](#_Toc11072344)

[1.3 主要研究内容 2](#_Toc11072345)

[1.3.1 主要研究内容 2](#_Toc11072346)

[1.3.2 结构安排 2](#_Toc11072347)

[第2章 系统需求分析 3](#_Toc11072348)

[2.1 可行性分析 3](#_Toc11072349)

[2.1.1 经济可行性 3](#_Toc11072350)

[2.1.2 技术可行性 3](#_Toc11072351)

[2.1.3 运行可行性 3](#_Toc11072352)

[2.2 需求分析 3](#_Toc11072353)

[2.2.1 商家需求分析 4](#_Toc11072354)

[2.2.2 管理员需求分析 5](#_Toc11072355)

[2.2.3 后台服务需求分析 5](#_Toc11072356)

[2.3 系统数据分析 5](#_Toc11072357)

[2.4 数据字典 6](#_Toc11072358)

[第3章 系统总体设计 7](#_Toc11072359)

[3.1 功能设计 7](#_Toc11072360)

[3.1.1 系统功能结构设计 7](#_Toc11072361)

[3.1.2 逻辑分层结构设计 8](#_Toc11072362)

[3.2 系统数据库设计 8](#_Toc11072363)

[第4章 系统详细设计 10](#_Toc11072364)

[4.1 系统运行环境 10](#_Toc11072365)

[4.1.1 系统软件环境 10](#_Toc11072366)

[4.1.2 系统硬件环境 10](#_Toc11072367)

[4.2 数据库详细设计 10](#_Toc11072368)

[4.3 逻辑层详细设计 14](#_Toc11072369)

[4.3.1 系统总体流程 15](#_Toc11072370)

[4.3.2 系统登录模块 16](#_Toc11072371)

[4.3.3 商品管理模块 17](#_Toc11072372)

[4.3.4 订单管理模块 18](#_Toc11072373)

[4.3.5 库存管理模块 19](#_Toc11072374)

[4.3.6 员工管理模块 20](#_Toc11072375)

[第5章 系统测试 22](#_Toc11072376)

[5.1 测试目的 22](#_Toc11072377)

[5.2 测试方法与测试内容 22](#_Toc11072378)

[5.2.1 测试方法 22](#_Toc11072379)

[5.2.2 测试内容 23](#_Toc11072380)

[5.3 测试过程 23](#_Toc11072381)

[5.3.1 员工管理功能测试 23](#_Toc11072382)

[5.3.2 商品管理功能测试 24](#_Toc11072383)

[5.3.3 订单管理功能测试 25](#_Toc11072384)

[5.3.4 库存管理功能测试 25](#_Toc11072385)

[5.4 测试用例 26](#_Toc11072386)

[5.4.1 员工管理测试用例 26](#_Toc11072387)

[5.4.2 商品管理测试用例 30](#_Toc11072388)

[5.4.3 订单管理测试用例 33](#_Toc11072389)

[5.4.4 库存管理测试用例 35](#_Toc11072390)

[5.5 测试结果分析 36](#_Toc11072391)

[第6章 总结与展望 37](#_Toc11072392)

[6.1 总结 37](#_Toc11072393)

[6.2 展望 37](#_Toc11072394)

[谢 辞 38](#_Toc11072395)

[参考文献 39](#_Toc11072396)

[附 录 41](#_Toc11072397)

# 第1章 绪论

## 1.1 研究背景

互联网技术的提升促进了电子商务的发展，网络的应用范围也变得越来越广泛，成为时下最高效的沟通交流方式。科技的发展持续升温使得很多新兴企业也陆续加入信息技术的行列。信息网络的普及给传统的企业发展增加了很大程度的限制[1]。当今早已是信息网络的时代，各行业紧追潮流迫切的发展网络产业。烘焙行业也在网络浪潮的冲击下,陆续摒弃传统手工管理的方式,采用计算机进行信息管理，有效避免了手工管理存在的效率低、性能差等弊端,管理方式得到了有效的提升。随着烘焙业的扩增与企业规模的壮大，原有的手工管理已经无法跟进现代化经营管理的脚步,致使整个烘焙业的发展受到了限制，影响了整体服务水平的提升。计算机技术持续发展，经过几十年的打磨，管理系统不管是在人员使用还是内部性能方面都已经非常完善。发达国家早在很久之前就已经将计算机应用于烘焙店管理中，现如今国内也已经将计算机应用投入系统管理中。

## 1.2 国内外研究现状

### 1.2.1 国外研究现状

随着国外信息技术的飞速发展,烘焙行业规模的不断扩大,烘焙店管理模式也发生了变革。国外很多烘焙店已经采用大型软件公司提供的信息管理系统,国内的管理系统在技术方面相对于国外趋于劣势。国外的软件公司凭借多年的开发经验，在开发过程中以语言的可扩展性和包容性为核心理念，严格遵守开发规范，使得系统具有高可扩、高性能等特性[2]。在以后的扩展及维护时可以有效地保护原有的资源,便捷地调整框架结构,有利于适应需求的更改、系统的升级,既使目前的业务需求得到了满足,又为今后的新增需求留有很大的扩展空间。

### 1.2.2 国内研究现状

经济的发展使得计算机应用得以普及，小到商户大到企业管理计算机应用都是不可或缺的一部分，发展高效的数字化管理方式已成为一种必然的趋势。各行业之间的发展存在着必然的联系，炙手可热的烘焙业的发展成功的引起了国内中小型软件公司的注意，就目前看来烘焙企业的软件开发有着广阔的市场前景，但是从现有烘焙店信息管理系统来看仍旧存在很多的问题和弊端。每一项技术开发设计都有本身自带的价值，由于国内的烘培店还未引起企业决策者的重视，所以使得管理系统仍处于起步阶段，并没有完全达到使用者预期的数据计算精确准、信息输出迅速、具有高度适应性的标准[3]，所以在管理与性能等方面都需要进一步的提高、完善。

## 1.3 主要研究内容

### 1.3.1 主要研究内容

本系统是以“味多美蛋糕店”为实例，了解中小烘焙店本身的具体情况、业务流程、基本需求。烘焙店管理系统相比大型企业管理系统在结构复杂性、系统并发行等方便都简易很多，本系统以中小型烘焙店为研究对象，对其涉猎的业务、烘焙店本身的特点进行多方比较、分析并制定需求开发文档。本课题就是为了适应中小型烘焙店的科学管理和发展，采用面向对象的设计方法、B/S开发模式，通过IntelijIDEA开发工具，结合MySQL数据库，使用SpringBoot内嵌的Tomcat作为WEB服务器，前台以titles作为框架运用传统的JSP技术进行数据展示，整个项目使用Maven进行管理基于JavaEE的MVC设计模式进行构建。

本系统在开发前对烘焙店的管理服务需求进行调研主要业务进行了充分了解后，将主要从客户订单信息维护、职工信息维护、用户信息维护、商品信息维护、库存信息维护等几个方面进行开展。

### 1.3.2 结构安排

本文结构共分为六个章节:

第1章，绪论。主要介绍本课题的研究意义，国内、国外研究现状主要研究内容。

第2章，系统需求分析。阐述了需求分析的概念，起到的作用，形成阶段性的里程碑成果。制定系统总体的开发需求以及明确每一个功能模块的具体需求方案，进行数据库的数据分析和系统的可行性分析。

第3章，系统总体设计。本章主要介绍系统功能结构设计、逻辑分层结构设计以及系统数据库总体的设计。

第4章，系统详细设计。对系统的软件、硬件环境进行分析，基于各个模块，给出系统的逻辑实现流程图，对数据库表结构进行详细设计。

第5章，系统测试。强调了测试在整个开发周期中的重要性，并对系统分模块进行测试。

第6章，总结与展望。对整个系统涉及的主要技术进行概述，总结系统中存在的不足并且会在以后进行优化。

# 第2章 系统需求分析

系统分析是整个软件开发周期中必不可少的一项工序，恰当的分析可以使开发顺利进行，否则将影响整个开发进度。系统分析主要任务就是将用户需求与技术知识相结合划定系统的功能模块，梳理清楚各模块之间的联系[4]。

## 2.1 可行性分析

### 2.1.1 经济可行性

主要是对项目的经济效益进行评价，本系统使用SpringBoot和MyBatis免费开源软件，当下最为主流的框架，技术较为成熟、应用广泛、性能高[5]。并且本系统开发周期短，见效快，在整个开发过程中基本上都是人力成本，所以本系统在经济上是可行的。

### 2.1.2 技术可行性

系统以IntelijIDEA作为开发工具，Java语言进行开发。采用SpringBoot搭建后台框架，titles作为前台框架整合JSP进行数据展示。使用SpringBoot内嵌的tomcat服务器，将MySQL作为本次系统设计的数据库，整个项目使用Maven进行管理，不需手动添加jar包，直接引入依赖系统会自动添加需要的jar。技术的实现在数据分类和建立表关系的过程中就已大致构想出其实现的方法，因此将数据库建立起来是重要的一步[6]。

### 2.1.3 运行可行性

本系统采用的是浏览器/服务器模式，使用者不需安装任何软件只需用自己计算机所带浏览器即可实现访问，对用户的运行内存、存储内存要求较低，所有相关的操作基本都在服务器上实现，因此这个系统的运行是无需担心的。

采用计算机烘焙店管理系统都具有一定的必要性，以少量的人力资源、高效的工作效率、较低的失误率进行管理，将使烘焙点平稳、高效的运营[7]。

## 2.2 需求分析

一份好的需求分析说明书是支撑软件开发的关键，需求分析顾名思义，就是对用户提出的需求进行系统、详细的分析，将所有问题弄清楚之后，梳理所有需要解决的需求，最后制定一份规范的开发需求说明书。需求分析是对软件系统的后期分析，需要进行一系列的活动，包括：分析用户需求、建立 需求原型、分析系统需求和进行需求验证等[8]。

在系统进行开发前要制定好需求分析说明书，并且需求分析的制定一定要做到考虑周到、全面细致，贴合用户提出的业务需求，确保说明书的可行性和合理性。需求分析由专业的需求人员负责，需求人员需要向开发人员明确用户的需求。其实可以把需求分析人员看成用户与开发人员沟通的桥梁，需求人员可以将用户的描述转变为软件的技术描述。一旦用户和开发者之间对系统的需求认知存在差异，便会给开发人员带来恶劣的后果，需求的修改就意味着代码的修改，将严重影响项目的进度，所以需求分析在软件开发中的重要性不可忽视，一份详细精准的需求分析是整个项目的得以顺利开展的关键[9]。

### 2.2.1 商家需求分析

伴随着经济水平的提高，人们的消费水平和生活水准和也逐渐提升，烘焙业在服务行中的热度也逐渐升温，如何在如火如荼的竞争中拔得头筹，已经成为每位烘焙经营者不得不需要思考的问题。经过多年的蜕变，烘焙企业的管理已经渐渐由繁杂的人工管理，逐步向着智能化、规范化管理方式发展。在互联网白热化的今天，要想达到智能化、规范化管理的标准，最有效的方法就是运用软件系统进行管理。如果还用传统管理方式，难免会出现以下问题:

(1)人工计算不如计算机运算准确，容易出现错误。

(2)容易造成数据丢失。

(3)客人具体消费信息很难查询。

(4)无法对以往经营数据进行查询。

传统人的工管理方式已经无法跟进烘焙行业快速发展的脚步，广大烘焙经营者意识到使用计算机软件的重要性，决定在烘焙企业的管理经营上引入烘焙管理系统。根据烘焙行业的特点和实际情况，烘焙管理系统应以烘焙业务需求为主，加强后台管理，注重营业数据。从科学的角度出发，争取为烘焙管理者提供专业、高效的管理模式和数据分析功能。通过对烘焙店的业务进行分析，本系统需要实现以下目标:

(1)操作界面简易、便利。

(2)方便、快捷的订购商品。

(3)准确的查看系统人员、消费人员的信息。

(4)快速查看库存信息、入库、出库记录。

(5)管理员有效身份登录。

(6)查看、修改系统员工的薪资。

(7)系统运行稳定，安全可靠的登陆、注册功能。

根据烘焙业的周边具体情况，开发人员明确开发需求，前期对系统总体功能进行开发，再对系统的的各个功能进行优化、修复缺陷，直到满足用户所有的需求。系统主要由订单管理、库存管理、商品管理、员工管理、用户管理五个方面的需求构成。

### 2.2.2 管理员需求分析

一、超级管理员需求分析

超级管理员拥有高于普通管理员的权限，除了可以执行基本的业务操作以外，主要是维护店铺人员信息。超级管理员可以保证系统在安全的状态下稳定运行。对系统的数据进行备份、查看系统操作及调用日志记录，在有突发情况下数据也不会丢失影响使用。

二、普通管理员需求分析

普通管理员可以由超级管理员进行信息录入也可以自行在注册页面完成注册，在超级管理员进行审核通过后可以具有和超级管理员同等的权限，即超级管理员。普通管理员的主要职责是后台信息管理，普通管理员可以管理后台信息，为客户进行商品的订购，查看订单、订单明细。对库存信息进行操作，查看商品出库记录、入库记录，出库即生成订单，出入库量和商品库存量进行同步。可以查看商品具体信息，亦可进行分类查询，并且对商品增加、修改、上架、下架等操作。

### 2.2.3 后台服务需求分析

后台管理模块主要是对各种信息的维护，包含维护订单信息、库存信息、商品信息等。

(1)人员管理：主要是对客户、系统人员的操作包括增加、删除、修改、查看。

(2)订单管理：主要是对订单的操作，可以查看订单号、客户名等信息，根据订单查看订单明细，包括订购的具体商品名称、数量。

(3)库存管理：该功能是对库存信息的维护，可以直观查看商品的库存量，出库操作与订单表相关联。还可以查看出库、入库记录。商品出库时，会先比对库存量，如果出库量大于库存量，弹框提示库存不足，当前库存为XX。如果满足条件则弹框显示出库成功。

(4)商品管理：主要是对商品进行增加、删除、修改、查看操作。查看商品信息时，加载会显示蒙层框，数据加载完毕蒙层消失。

## 2.3 系统数据分析

对于系统开发来说最基本以及最重要的就是被处理数据的描述和数据采集的要求，数据分析正是为了满足这一需求。数据分析是在系统的开发需求及其他需求明确的情况下进行的，为了设计一个性能良好的数据库系统，明确应用环境对系统的需求是首要的也是至关重要的[10]。

本系统主要的数据字典有以下几种:

(1)员工信息：包括员工编号、员工名称、员工密码、电话号码、员工标识、员工薪资、居住地址等字段。

(2)客户信息：包括客户编号、客户名称、电话号码、用户级别等字段。

(3)订单信息：包括订单编号、总价、出售日期、客户、备注、状态、创建时间、修改时间等字段。

(4)订单明细信息：包括明细编号、订单编号、商品编号、出库数量、用户编号等字段。

(5)商品信息：包括商品编号、商品名称、商品单价、计量单位、商品类别、状态等字段。

(6)出入库明细表：商品编号、入库时间、出库时间、出入标识、入库量、出库量等字段。

## 2.4 数据字典

数据库中表数据需要通过数据字典维护，数据字典定义表如表2.1所示，字典数据表如表2.2所示。

表2.1 字典定义表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 数据类型 | 标识 | 备注 |
| 字典编号 | ID | *varchar2* | *PRIMARY KEY;*  *NOT NULL* |  |
| 字典编码 | CODE\_TYPE | *varchar2* |  |  |
| 字典名称 | CODE\_NAME | *varchar2* |  |  |
| 状态 | STATUS | *INT* |  | 0：不可用；  1:可用； |
| 描述 | DESC | *varchar2* |  |  |

表2.2 字典数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 数据类型 | 标识 | 备注 |
| 编号 | ID | *varchar2* | *PRIMARY KEY;* |  |
| 字典编码 | CODE\_TYPE | *varchar2* |  |  |
| 数据编码 | BASE\_CODE | *varchar2* |  |  |
| 数据名称 | CODE\_NAME | varchar2 |  |  |
| 描述 | REMARK | *varchar2* |  |  |
| 显示顺序 | SORT | *INT* |  |  |

# 第3章 系统总体设计

## 3.1 功能设计

### 3.1.1 系统功能结构设计

J2EE技术架构与传统应用的开发架构相比较做了很大的提升，可以说与传统的技术架构完全不同，J2EE内含很多组件，这样以来使得程序的开发与部署得到了极大的简化，从而提高了程序的安全性、可移植性与再用价值[11]。

J2EE的核心是一组应用程序的技术规范与开发指南，除了含有很多组件之外还包含服务架构和技术层次，他们之间有着共同的标准和规格,这使得不同的平台如果都使用了J2EE架构也能很好的兼容，J2EE技术架构解决了以前程序无法兼容的问题[12]。烘焙管理企业要想提高工作效率、有效的完成人员的分配运用J2EE技术架构搭建管理系统是众架构中的首选。为了满足用户需求本系统主要分为用户信息管理、订单管理、库存管理、商品信息管理几大模块，这几个模块数据相互联系，每个大模块下包含几个子模块。系统功能框图如图3.1所示。

烘焙店管理系统

超级管理员

普通管理员

用户管理模块

员工管理模块

订单管理模块

商品管理模块

库存管理模块

查看入库记录

修改员工信息

删除员工信息

查看员工信息

新增用户信息

用户信息操作

查看出库记录

查看商品库存

订单信息操作

商品信息操作

图3.1 系统功能框图

### 3.1.2 逻辑分层结构设计

在设计一个与业务逻辑相关的方案时一定会进行框架的选择，这时通常会将应用逻辑进行分层，拆分为不同的层与子系统。分层结构使整个方案层次变得清晰，将复杂的代码分解为独立的层次，使用分层结构使得原本复杂的测试代码变得不再复杂，便于理解和维护。

烘焙店管理系统由四层结构组成，采用MVC编程模式进行系统的设计和开发，四层结构分别为持久层、业务逻辑层、表现层、数据访问层。系统逻辑分层图如图3.2所示。

烘焙店管理系统

表示层

业务逻辑层

持久层

数据访问层

图3.2 系统逻辑分层图

## 3.2 系统数据库设计

数据库设计是整个开发周期中必不可少的一步，系统数据库设计特别复杂，它决定着数据库的性能以及系统的性能，是整个项目搭建的关键。数据库的设计并不是简单的表构建，它支撑着系统的运行，最优的数据库设计不可能在一朝一夕完成，需要精益求精，反复探索。

本系统采用MySQL数据库进行系统的开发和构建。数据库设计是系统构建的第一步，也是整个软件开发周期中极其重要的一步[13]，并且是资源管理的核心。数据库设计和软件开发是无法分割的，数据库设计的质量决定了系统的整体性能[14]，在后续的开发过程中也深受数据库的支持，如果数据库设计出现偏差，轻则修改改码，重则系统重建。

数据库逻辑设计是整个数据库设计过程的核心部分，数据库作为整个系统的基石，设计的恰当与否影响着数据库的性能。为了在以后的使用中不出现问题，数据库的设计应该严格遵守设计规范。本系统总体数据库E-R图如图3.3所示。

拥有

订单

商品

客户

员工

订单明细

1

n

包含

n

包含

1

1

n

管理

n

1

进销记录

生成

n

1

图3.3 数据库总体设计E-R图

# 第4章 系统详细设计

## 4.1 系统运行环境

### 4.1.1 系统软件环境

1.服务器端:

(1)操作系统平台:UNIX, LINUX, Windows

(2)系统开发平台:J2EE, J2SE

(3)Web服务器:Tomcat

(4)数据库:MySQL

(5)开发语言:JSP,JAVA,JS

2.客户端:

(1)操作系统:Windows 7, Windows 10

(2)浏览器:360浏览器，谷歌浏览器

### 4.1.2 系统硬件环境

1.服务器端:4G以上内存

2.网络配置:百兆模式以上

## 4.2 数据库详细设计

结合前面系统数据分析及数据实体E-R图，最终得到数据库表结构。以下给出数据库中有关数据表的设计。

员工信息表bky\_emp中字段的中文名称、数据类型等信息如表4.1所示。

表4.1 员工信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 数据类型 | 标识 | 备注 |
| 序号 | ID | *Int* |  |  |
| 员工编号 | EMP\_NO | *varchar2* | *PRIMARY KEY;*  *NOT NULL* |  |
| 员工名称 | EMP\_NAME | *varchar2* |  |  |
| 员工密码 | PASSWORD | *varchar2* |  | 采用MD5 加密进行处理 |
| 电话号码 | PHONE | *varchar2* |  |  |
| 员工标识 | ROLE\_ID | *int* |  | 1：普通管理员；  2：超级管理员； |

续表4.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 数据类型 | 标识 | 备注 |
| 员工职位 | JOB | *varchar2* |  |  |
| 上级领导 | MGR | *varchar2* |  |  |
| 员工性别 | SEX | *int* |  | 0:男；  1：女； |
| 出生日期 | BIRTH | *Date* |  |  |
| 员工薪资 | SAL | *varchar2* |  | 月工资； |
| 居住地址 | ADDR | *varchar2* |  |  |
| 存档时间 | DONE\_TIME | *Date* |  |  |
| 部门ID | DEP\_NO | *varchar2* |  | 1:销售部；  2:生产部； |
| 状态 | STATUS | *INT* |  | 0:不可用；  1:可用； |
| 创建时间 | CREATE\_TIME | *Date* |  |  |
| 修改时间 | UPDATE\_TIME | *Date* |  |  |
| 描述 | DESC | *varchar2* |  |  |

数据库中用户信息表表名是bky\_user，表中字段的中文名称、英文名称数据类型等信息如表 4.2所示。

表4.2 用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 数据类型 | 标识 | 备注 |
| 序号 | ID | *int* |  |  |
| 客户编号 | USER\_ID | *varchar2* | *PRIMARY KEY;*  *NOT NULL* |  |
| 客户名称 | USER\_NAME | *varchar2* |  |  |
| 电话号码 | USER\_PHONE | *varchar2* |  |  |
| 性别 | SEX | *varchar2* |  |  |
| 存档时间 | DONE\_TIME | *Date* |  |  |
| 状态 | STATUS | *INT* |  | 0:不可用；  1:可用； |
| 创建时间 | CREATE\_TIME | *Date* |  |  |
| 修改时间 | UPDATE\_TIME | *Date* |  |  |
| 备注 | DESC | *varchar2* |  |  |

数据库中订单信息表表名是bky\_order，表中字段的中文名称、英文名称数据类型等信息如表 4.3所示。

表4.3 订单信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 数据类型 | 标识 | 备注 |
| 序号 | ID | *int* |  |  |
| 订单编号 | ORDER\_ID | *varchar2* |  |  |
| 总价 | TOTAL\_OPRICE | *varchar2* |  |  |
| 会员 | IS\_VIP | *varchar2* |  |  |
| 用户电话 | USER\_PHONE | *varchar2* |  |  |

续表4.3:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 数据类型 | 标识 | 备注 |
| 支付方式 | PAY\_METHOD | *varchar2* |  |  |
| 状态 | STATUS | *INT* |  | 0：不可用；  1:可用； |
| 创建时间 | CREATE\_TIME | *Date* |  |  |
| 修改时间 | UPDATE\_TIME | *Date* |  |  |
| 备注 | DESC | *varchar2* |  |  |

数据库中订单信息明细表表名是bky\_order\_dtl，表中字段的中文名称、英文名称数据类型等信息如表 4.4所示。

表4.4 订单明细表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 数据类型 | 标识 | 备注 |
| 序号 | ID | *int* |  |  |
| 明细编号 | ORDER\_DTL\_ID | *varchar2* | *PRIMARY KEY;*  *NOT NULL* |  |
| 订单编号 | ORDER\_ID | *varchar2* |  | 与bky\_order  表ORDER\_ID关联 |
| 商品编号 | PDT\_ID | varchar2 |  | 与bky\_pdt  表PDT\_ID关联 |
| 商品名称 | PDT\_NAME | *varchar2* |  |  |
| 出售数量 | SALE\_NUM | INT |  |  |
| 总价 | TOTAL\_PRICE | *varchar2* |  |  |
| 下单时间 | DONE\_TIME | *DATE* |  |  |
| 客户编号 | USER\_ID | *varchar2* |  | 与bky\_user表USER\_ID关联 |
| 客户名称 | USER\_NAME | *varchar2* |  |  |
| 状态 | STATUS | *INT* |  | 0：不可用；  1:可用； |
| 创建时间 | CREATE\_TIME | *Date* |  |  |
| 修改时间 | UPDATE\_TIME | *Date* |  |  |
| 备注 | DESC | *varchar2* |  |  |

数据库中商品信息表表名是bkt\_pdt，表中字段的中文名称、英文名称数据类型等信息如表 4.5所示。

表4.5 商品信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 数据类型 | 标识 | 备注 |
| 序号 | ID | *int* |  |  |
| 商品编号 | PDT\_ID | *varchar2* | *PRIMARY KEY;*  *NOT NULL* |  |
| 商品名称 | PDT\_NAME | *varchar2* |  |  |
| 商品类别 | PDT\_TYPE | *INT* |  |  |
| 单价 | UNIT\_PRICE | varchar2 |  |  |

续表4.5:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 数据类型 | 标识 | 备注 |
| 计量单位 | MEASER | *varchar2* |  | 1:盒；2:杯；  3:块； |
| 状态 | STATUS | *INT* |  | 0：不可用；  1:可用； |
| 创建时间 | CREATE\_TIME | *Date* |  |  |
| 修改时间 | UPDATE\_TIME | *Date* |  |  |
| 描述 | DESC | *varchar2* |  | 图片 |

数据库中流量统计表表名是bky\_count，表中字段的中文名称、英文名称数据类型等信息如表 4.6所示。

表4.6 流量统计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 数据类型 | 标识 | 备注 |
| 序号 | ID | *int* |  |  |
| 访问次数 | VISIT\_COUNT | *varchar2* | *PRIMARY KEY;*  *NOT NULL* |  |
| 操作时间 | UPDATE\_TIME | *Date* |  |  |
| 描述 | REMARK | *varchar2* |  |  |

数据库中进销表表名是bky\_stock，表中字段的中文名称、英文名称数据类型等信息如表 4.7所示。

表4.7 进销表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | | 数据类型 | | 标识 | | 备注 | |
| 序号 | ID | | *int* | |  | |  | |
| 进销编号 | STOCK\_ID | | *varchar2* | | *PRIMARY KEY;*  *NOT NULL* | |  | |
| 商品编号 | | PDT\_ID | | *varchar2* | | *NOT NULL* | |  |
| 入库时间 | IN\_TIME | | Date | |  | |  | |
| 出库时间 | OUT\_TIME | | *Date* | |  | |  | |
| 出库时间 | OUT\_TIME | | *Date* | |  | |  | |
| 入库量 | IN\_NUM | | *int* | |  | |  | |
| 出库量 | OUT\_UNM | | *int* | |  | |  | |
| 进销标识 | FLAG | | *Int* | |  | | IN:入库  Out:出库 | |
| 修改时间 | UPDATE\_TIME | | *Date* | |  | |  | |
| 状态 | | STATUS | | *INT* | |  | | 0：不可用；  1:可用； |
| 描述 | DESC | | *varchar2* | |  | |  | |

数据库中系统表表名是sys\_menu，表中字段的中文名称、英文名称数据类型等信息如表 4.8所示。

表4.8 系统表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名称 | 英文名称 | 数据类型 | 标识 | 备注 |
| 序号 | ID | *int* |  |  |
| 代码 | DATA\_CODE | *varchar2* |  |  |
| 名称 | DATA\_NAME | *varchar2* |  |  |
| 状态 | STATUS | *int* |  |  |
| 方法名 | FUNCTION\_NAME | *varchar2* |  |  |
| 图标 | CSS\_ICON | *varchar2* |  |  |
| 排序 | SORT | *int* |  |  |
| 描述 | REMARK | *varchar2* |  |  |

## 4.3 逻辑层详细设计

进入系统前前台会对登陆信息进行非空、数据格式等进行校验，后台进行判断该用户是否存在，存在则放行。如图4.1所示为登录系统的界面图。

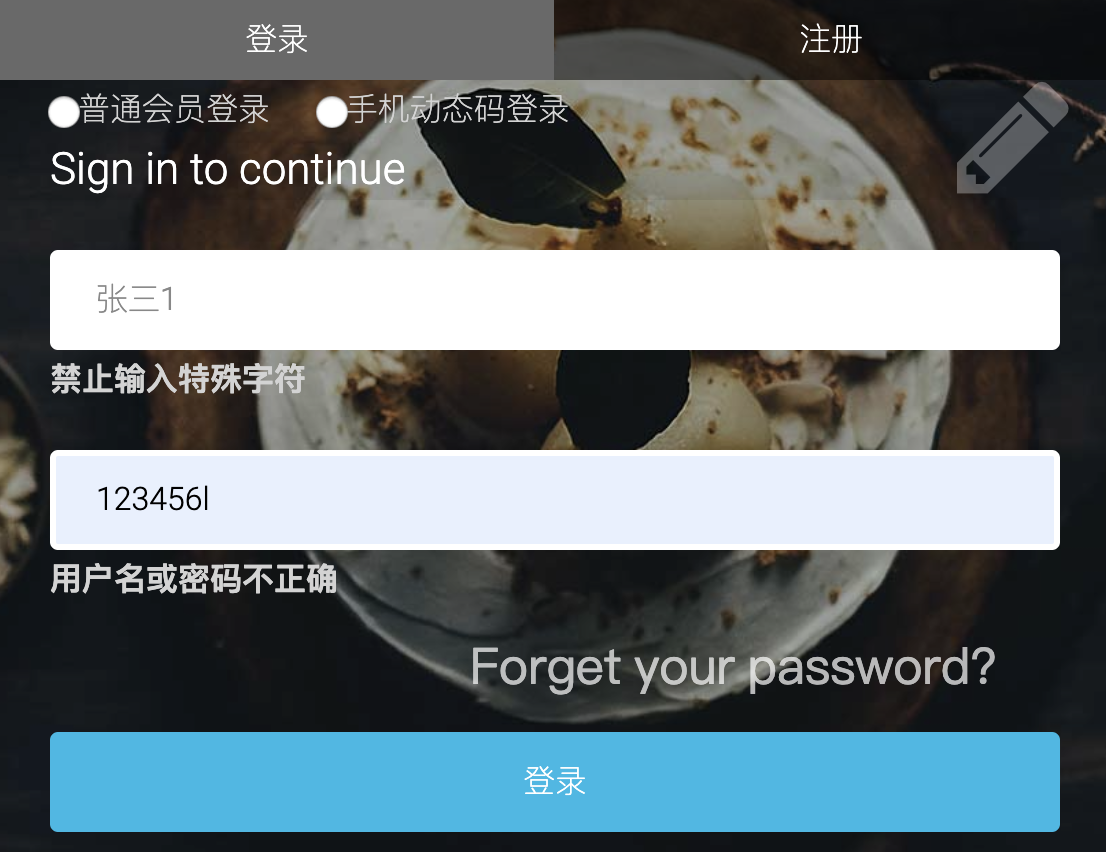


图4.1 系统登录界面

用户进行账号注册时，如果注册账号密码在数据库中已经存在则提示“请更换账号或密码”，用户提交注册信息后密码会经过“MD5”进行加密存到数据库中，若有人非法盗取密码获得的也是密文。注册成功后会提示用户注册成功，跳转到登陆界面。注册成功界面如图4.2所示。

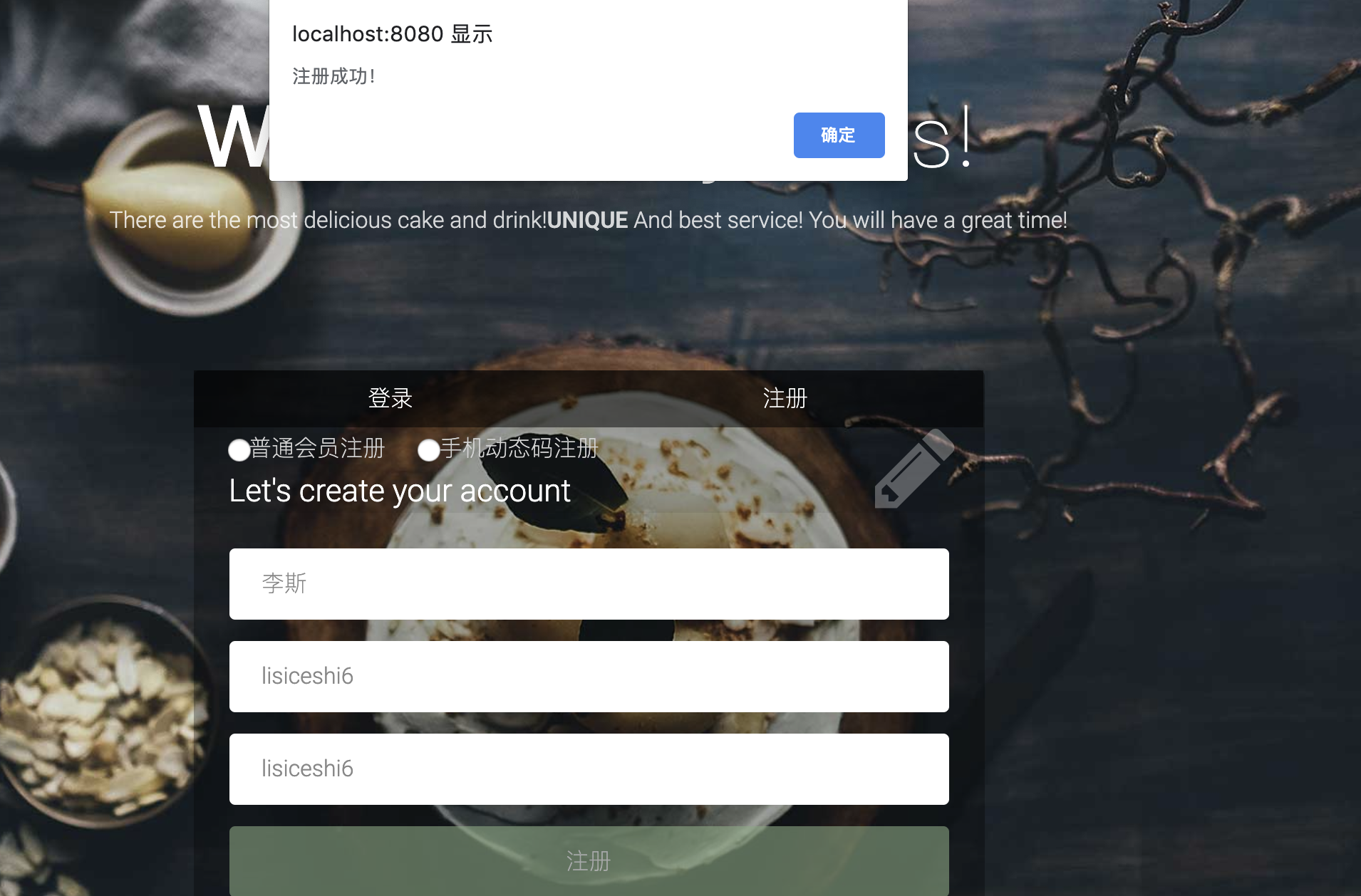


图4.2 管理员注册页面

### 4.3.1 系统总体流程

结合需求分许、根据系统总体的功能设计出了系统总体功能流程图，如图4.3所示。

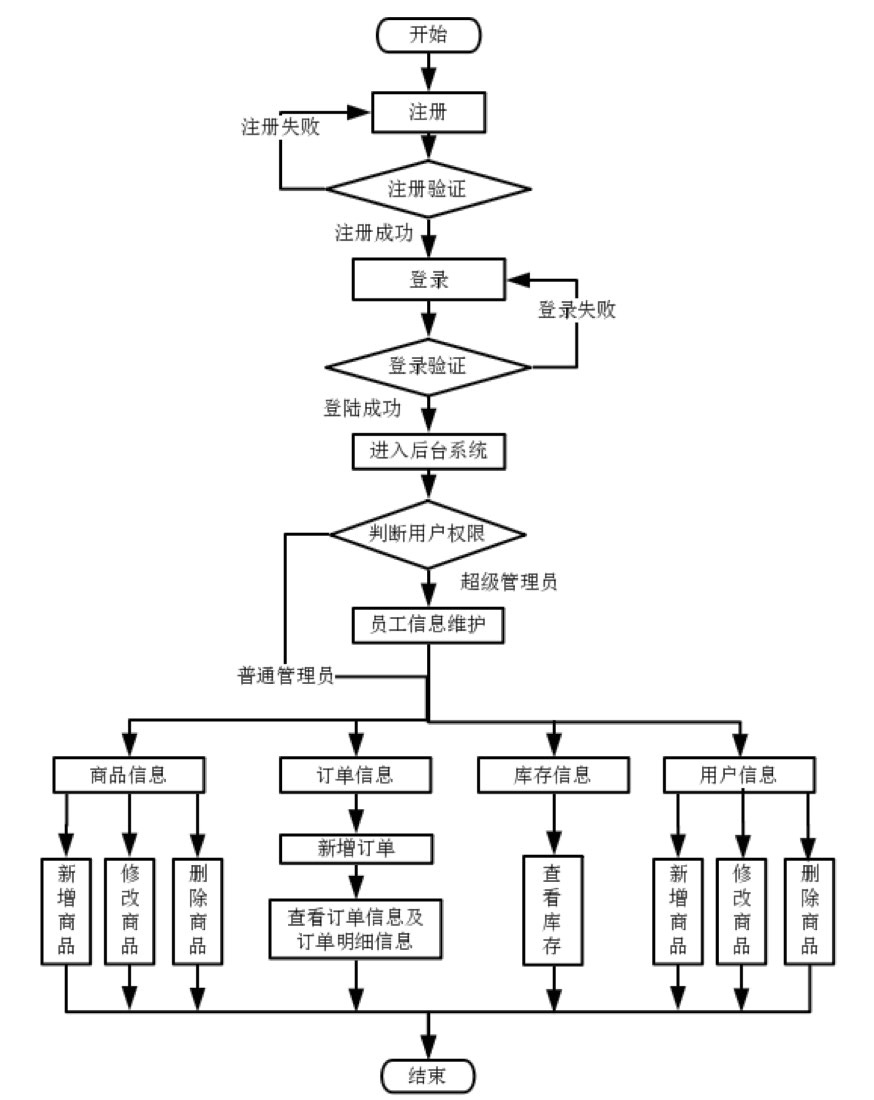


图4.3 系统总体功能流程图

### 4.3.2 系统登录模块

登录方式分为用户名、密码登录和手机动态码登录两种，以用户名、密码方式登录为例进行说明。

管理员在登陆界面录入账号密码后，点击登陆按钮Ajax会将输入的信息传送到后台进行用户身份判定，符合则进入系统，不符合则提示输入正确信息。如图4.4所示为登录系统流程图。

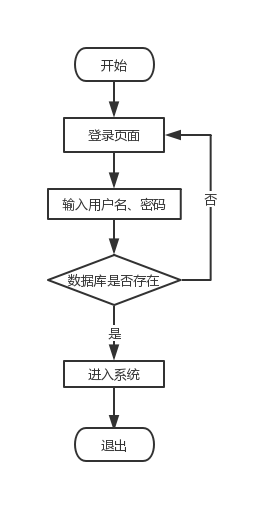


图4.4 登录流程图

### 4.3.3 商品管理模块

用户进入系统后，可以选择“商品管理”菜单，首页分页查询展示商品信息，用户可以点击增加、修改、上架等操作按钮进行相应操作，确认操作后数据会保存到数据库，更新记录。商品管理模块流程如图4.5所示。

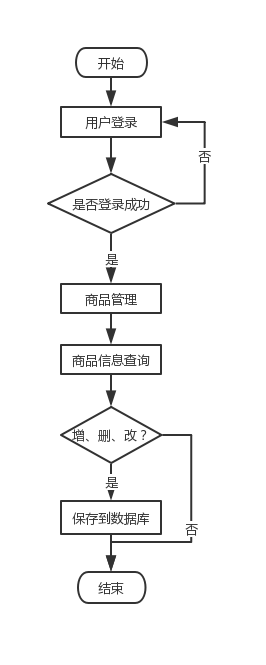


图4.5 商品管理流程图

### 4.3.4 订单管理模块

用户进入订单管理模块可以在表单输入查询条件通过精确、模糊查询所有订单记录，点击单笔订单可以根据订单号查看该订单下的所有订单明细信息。订单只允许单条数据操作，也就是每次只可以删除、修改一条订单记录。删除当前订单，该订单下的所有订单明细记录也会随之删除。“订单明细记录”支持批量删除，删除某一订单下的部分订单明细记录，订单记录不会改变，若删除所有订单明细记录，该订单也会删除。订单记录可以查看出订单金额、订单日期等信息。订单管理模块流程如图4.6所示。

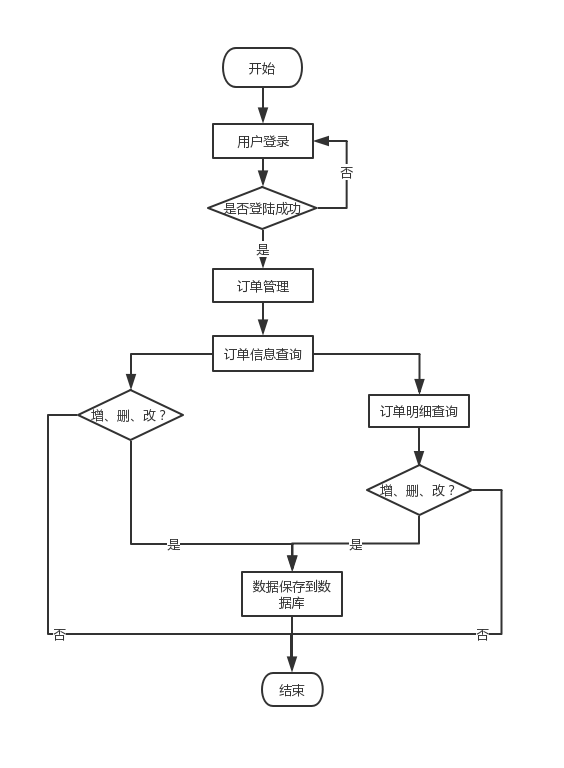


图4.6 订单管理流程图

### 4.3.5 库存管理模块

库存管理模块下有商品库存、出库记录、入库记录三个子模块。商品库存模块可以查看某商品当前库存量，该模块不支持除了查看以外其他的操作。商品入库、商品出库模块可以查看所有出库、入库记录，在数据出现差错时可以根据记录进行确认。管理员可以对出、入库记录查询模块进行修改、删除操作，所操作的数据会实时在商品库存信息模块中发生改变，三个模块相互关联。库存管理模块操作流程如图4.7所示。

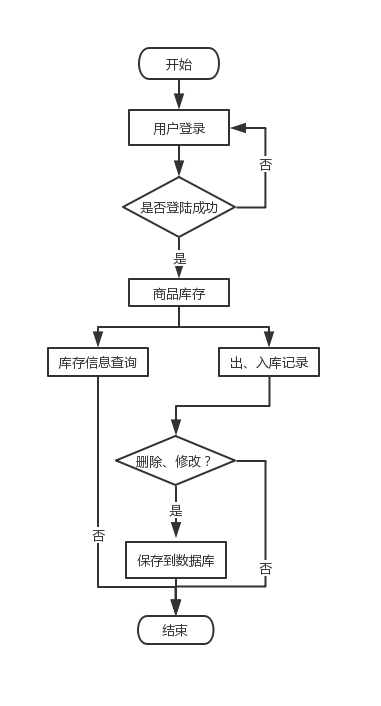


图4.7 商品库存流程图

### 4.3.6 员工管理模块

用户登录时会在后台校验用户登录权限，当用户为“超级管理员”时，进入系统后会有权限操作“员工管理模块”。员工列表使用分页查询进行数据展示，用户可以通过查询条件定位自己需要的信息，并且可以点击列表上方的按钮组，对数据进行增加、删除、删改等操作，用户确认操作后数据库会保存最新数据。

员工管理模块流程如图4.8所示。

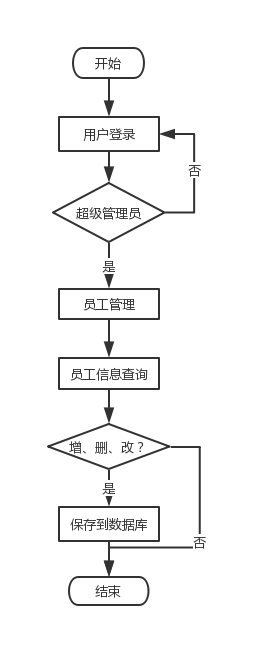


图4.8 员工管理模块流程图

第5章 系统测试

## 5.1 测试目的

软件测试是系统软件周期中一项必不可少的工序[15]，测试的目的就是发现软件中存在的漏洞并修改，提高软件的可靠性，对后期软件的稳定运行提供保障。在目前形式化方法和程序正确性证明技术还无望成为实用性方法的情况下，软件测试在将来一段时间内仍然是软件可靠性保证的有效方法。不足的测试势必使软件带着一些未揭露的隐藏错误投入运行，这将意味着更大的危险让用户承担[16]，为后期的系统维护加大了难度。

## 5.2 测试方法与测试内容

### 5.2.1 测试方法

一、功能测试

在本次毕业设计中，选择了黑盒测试来对项目进行功能测试。可以选择合适的测试数据以及系统的接口，二者相互结合，便可以得到我们理想中的输出结果，得到分析结果之后，通过将输出结果与预期的结果进行对比，便可以知道我们某一部分的功能是否正确。本次测试主要采用了以下的几种方法：

1.挑选出临界值的数据、具有代表性的数据，来验证当前要测试的部分功能是否可以按照预想的结果操作.

2.将无效的测试数据筛选出来，来验证系统面对有误的命令或者应对异常的情况的时候是否可以按照预想的结果进行[17].

3.系统中的页面跳转功能也是要检测的最重要的一部分，查看每个页面中的链接是否可以正确的进行跳转以及传值等功能。

4.测试互相关联的部门，测试不同的功能之间是否具有完整性以及相关性。

二、性能测试

性能测试主要就是测试系统页面的响应速度、数据库查询速度、内存占用率、CPU、流量、等功能[18]，对于这些功能来进行性能测试。在本次毕业设计中主要用以下方法来进行测试：

1.测试系统可以接收数据时，可以承受多大的极限，测试在超负荷情况下的系统运行情况。

2.统计出系统出现故障的平均周期，十到二十天连续运行系统并且进行数据的分析。

3.网络环境的改变有时也会影响系统的运行，所以我们要考虑不同的网络环境下，系统的运行状况。

### 5.2.2 测试内容

1.根据需求分析说明书划分测试范围，明确测试的功能，通过测试验证用户提出的需求是否己经在体现在系统中。

2.分模块测试软件各个功能模块，包括操作界面外观是否和UI一致，数据数据输出流是否准确、迅速，页面响应速度、按钮灵活度是否达到标准[19]。

3.整体测试，系统按模块进行测试以后，将所有模块作为一个整体进行测试，测试各模块之间的连通性，检验实现的效果是否和用户要求的相符[20]。

## 5.3 测试过程

### 5.3.1 员工管理功能测试

员工管理模块测试名称及测试内容如表5.1所示。

表5.1 员工管理测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试名称 | 测试内容 | 测试结果 | 预期结果 |
| 1 | 分页查询员工 | 在员工列表首页搜索框内录入员工名称、电话、标识等信息，点击“查询”按钮进行分页查询，如果员工列表无数据代表员工不存在，成功则显示员工信息。 | 查询成功，员工列表显示数据。 | 查询成功，员工列表显示数据。 |
| 2 | 新增员工 | 通过点击“添加”按钮出发新增员工弹框，完成信息录入通过表单验证，点击“保存”按钮完成新增。如果新增成功提示：“新增成功”，失败则提示“新增失败！”。 | 弹框提示“新增成功”！ | 弹框提示“新增成功”！ |
| 3 | 修改员工信息 | 选择一条要操作的员工记录，点击修改按钮，在弹框中修改要修改的字段，通过表单验证，点击“保存”按钮，如果修改成功提示：“修改成功”，失败则提示“修改失败！”。 | 弹框提示“修改成功”！ | 弹框提示“修改失败”！ |
|  |  |  |  |  |
|  |  |  |  |  |

续表5.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试名称 | 测试内容 | 测试结果 | 预期结果 |
| 4 | 删除员工 | 选择一条要删除的记录，进行删除操作，成功提示“删除成功”！且该记录不再显示在列表中。失败则显示，“删除失败”，记录无变化。 | 弹框提示“删除成功”，数据列表中查询不该记录。 | 弹框提示“删除成功”，数据列表中查询不该记录。 |

### 5.3.2 商品管理功能测试

商品管理测试名称及测试内容如表5.2所示。

表5.2 商品管理测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试名称 | 测试内容 | 测试结果 | 预期结果 |
| 1 | 分页查询商品 | 管理员在商品管理页面通过商品名称、商品类别查询商品。 | 查询成功，商品列表显示所查商品。 | 查询商品成功显示在商品列表。 |
| 2 | 新增商品 | 通过点击“添加”按钮出发新增员工弹框，完成信息录入通过表单验证，点击“保存”按钮完成新增。如果新增成功提示：“新增成功”，失败则提示“新增失败！”。 | 弹框提示“新增成功”！ | 新增成功，记录数增加一条。 |
| 3 | 商品图片上传 | 在新增商品的模态框选则商品图片，点击预览图片，显示图片。 | 图片显示在页面，上传成功。 | 图片上传成功。 |
| 4 | 修改商品 | 选择一条要修改的记录，更改原有信息，点击保存成功提示“修改成功”！失败则显示，“修改失败”。 | 弹框提示“修改成功”。 | 商品修改成功 |
| 5 | 商品上架 | 选择一条状态为“待上架”的上架，点击上架按钮，成功显示提示信息“上架成功”，该条记录状态变为“已上架”。 | 提示“上架成功”，且该商品状态变为“已上架”。 | 商品成功上架 |
| 6 | 商品下架 | 选择一条状态为非“已上架”的上架，点击下架按钮，成功显示提示信息“下架成功”，失败显示“只有已上架的商品才能完成下架操作！”。 | 提示“只有已上架的商品才能完成下架操作！”。 | 商品下架失败 |
| 7 | 删除商品 | 选择一条商品点击删除按钮，选择确定删除。 | 删除成功，商品从列表中移除。 | 商品删除成功 |
| 8 | 商品入库 | 选择想要进行入库操作的商品对应的入库按钮，输入入库量等信息，点击确定按钮。成功提示“入库成功”，并且在入库记录中可以查到该记录。 | 提示“入库成功”，在入库记录管理中，可以查看该记录。 | 入库成功。 |

### 5.3.3 订单管理功能测试

订单管理测试名称及测试内容如表5.3所示。

表5.3 订单管理测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试名称 | 测试内容 | 测试结果 | 预期结果 |
| 1 | 分页查询订单 | 在订单列表首页搜索框内录入员工名称、电话，点击“查询”按钮进行分页查询，查询该客户下所有订单，如果无数据代表该客户没有订购过商品。 | 查询成功，订单列表显示某客户下全部订单。 | 查询成功。 |
| 2 | 修改订单 | 点击将要修改订单的备注字段，输入想要修改的信息，点击“对号”，备注信息修改成功。 | 备注信息显示更改后的信息。 | 修改成功。 |
| 3 | 查看订单明细 | 点击订单前对应的加号，查看该订单下对应的订单明细信息。 | 订单明细信息展示在该订单下。 | 查询成功。 |
| 4 | 删除订单 | 点击要删除的订单所对应的删除按钮，提示“删除成功”，订单从订单列表消失，该订单下对应的所有订单明细也被删除。 | 订单及订单明细全部删除。 | 删除成功。 |
| 5 | 删除订单明细 | 选择某订单下的一条订单明细执行删除操作，删除成功该明细记录消失，并且订单总价为正确值。 | 删除成功，订单总价也自动更改为正确价格。 | 删除成功。 |
| 6 | 修改订单明细商品数量 | 点击某条订单明细的商品数量字段，在弹出的输入框中更该数量，提交更改后订单总价自动改变为准确值。 | 数量修改成功，订单总价自动更改。 | 商品数量及丁达总价修改成功。 |
| 7 | 新增订单 | 管理员进入新增订单页面，录入商品数量、名称、单价等信息后点击添加按钮，成功会在新增列表中显示刚刚录入的数据，可以多次执行此操纵，则每次添加的记录都会展示在列表中。最后选择付款方式等进行提交。 | 订单新增成功，在订单列表可以查到生成的订单及对应的订单明细。 | 订单及订单明细新增成功。 |

### 5.3.4 库存管理功能测试

库存管理测试名称及测试内容如表5.4所示。

表5.4 库存管理测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试名称 | 测试内容 | 测试结果 | 预期结果 |
| 1 | 分页查询商品库存 | 在库存列表首页搜索框内商品名称，点击“查询”按钮进行分页查询。 | 商品库存信息显示在列表中。 | 可以查出数据。 |
| 2 | 分页查询入库记录 | 管理员进入入库记录模块中，根据商品名称及起始日期检索。 | 入库记录展示在列表中。 | 可以查出数据。 |

续表5.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试名称 | 测试内容 | 测试结果 | 预期结果 |
| 3 | 修改入库记录 | 点击要修改的记录所对应的入库量，在弹出的输入框中填写要修改的数量，点击保存。 | 弹框提示“修改成功”，且数量变为更改后的数量。 | 弹框提示“修改成功”，且数量变为更改后的数量。 |
| 4 | 删除入库记录 | 选择一条要删除的记录，进行删除操作，成功提示“删除成功”！且该记录不再显示在列表中。失败则显示，“删除失败”，记录无变化。 | 弹框提示“删除成功”，数据列表中查询不该记录。 | 弹框提示“删除成功”，数据列表中查询不该记录。 |
| 5 | 查询出库记录 | 管理员进入出库记录模块中，根据商品名称及起始日期查询某商品所有出库记录。 | 该商品对应的所有出库记录展示在列表中。 | 数据查询成功。 |

## 5.4 测试用例

### 5.4.1 员工管理测试用例

1.新增员工功能，新增成功时的测试内容、数据输入及操作步骤如表5.5所示。

表5.5 新增员工成功测试用例

|  |  |  |
| --- | --- | --- |
| 测试标题 | 测试输入 | 操作步骤 |
| 输入员工姓名 | 张三 | 点击员工姓名输入框，输入：张三 |
| 选择性别 | 选择男 | 点击下拉框：选择男 |
| 输入电话 | 17389287365 | 点击电话输入框，输入：17389287365 |
| 输入地址 | 北京 | 点击地址输入框，输入：北京 |
| 选择员工级别 | 普通管理员 | 点击员工级别下拉菜单选择：普通管理员 |
| 输入员工薪资 | 2000 | 点击薪资输入框，输入：2000 |
| 选择员工职位 | 店长 | 点击员工职位下拉菜单选择：店长 |



图5.1 新增员工信息

管理员进入系统后选择员工管理模块，点击“添加”功能按钮会触发新增用户的弹框，表单进行了数据验证，管理员按照表5.5输入数据完成验证，点击“保存”按钮后，会触发提示弹框并显示“保存成功！”。新增员工模态框如图5.1，新增员工成功界面如图5.2。



图5.2 新增员工成功

2.新增员工功能，新增失败时的测试内容、数据输入及操作步骤如表5.6所示。

表5.6 新增员工失败测试用例

|  |  |  |
| --- | --- | --- |
| 测试标题 | 测试输入 | 操作步骤 |
| 输入员工姓名 | 李四 | 点击员工姓名输入框，输入：张三 |
| 选择性别 | 选择性别男或女 | 点击下拉框：选择男 |
| 输入电话 | 17389287365 | 点击电话输入框，输入：17389287365 |
| 输入地址 | 北京 | 点击地址输入框，输入：北京 |
| 输入员工级别 | 普通管理员 | 点击员工级别下拉菜单选择：普通管理员 |
| 输入员工薪资 | 2000 | 点击薪资输入框，输入：2000 |
| 选择员工职位 | 店长 | 点击员工职位下拉菜单选择：店长 |

管理员在新增用户表单按照表5.6提示输入数据，在数据库查询一条已经存在的用户，将该用户的电话号码输入到“电话号码”输入框，完成所有输入框数据校验后，点击“保存”按钮，会有弹框提示“该用户已存在！”，新增员工失败界面如图5.3。

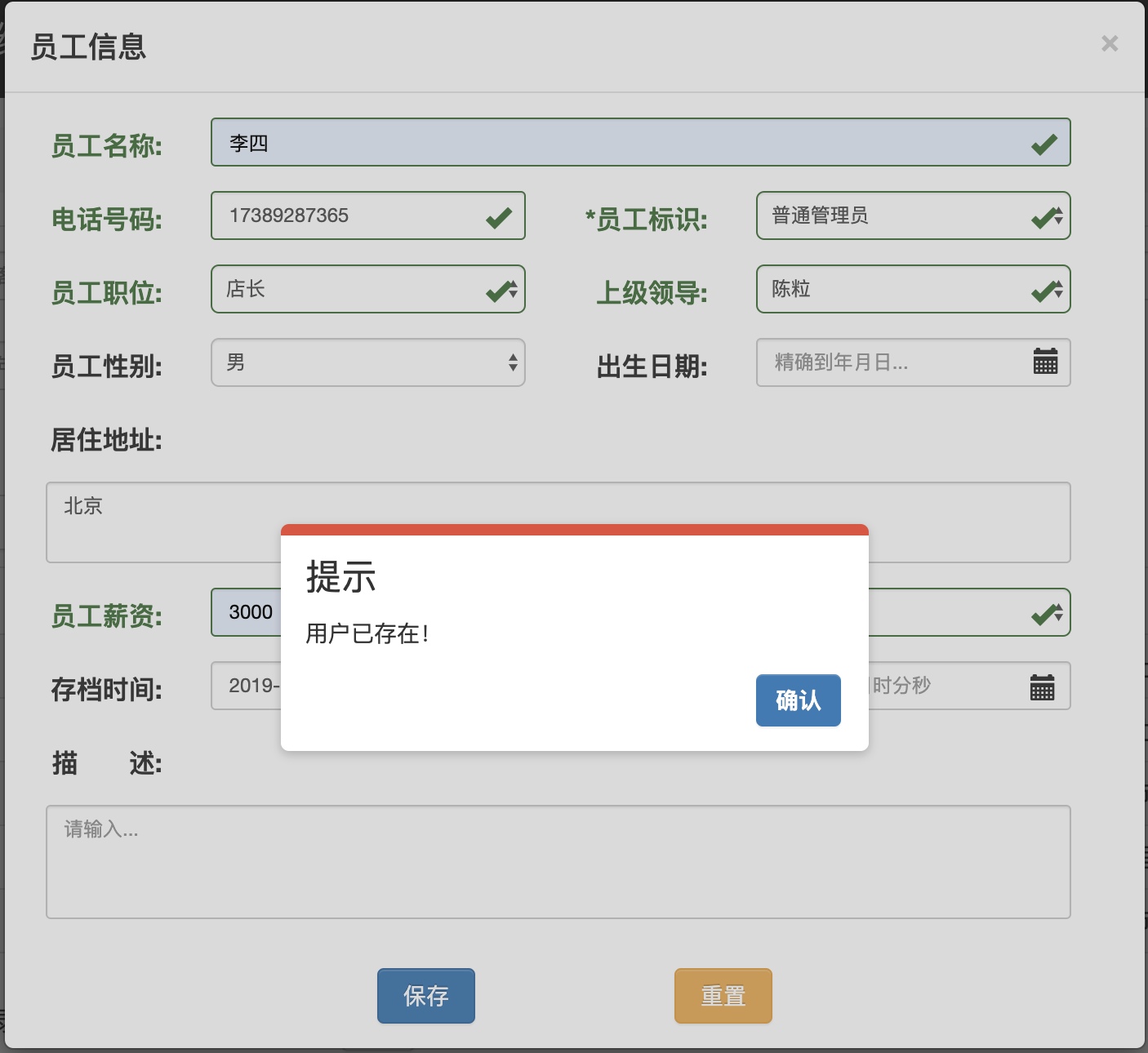


图5.3 新增员工失败

3.查询员工信息，查询员工的测试内容、数据输入、操作步骤如表5.7所示。

表5.7 查询员工用例

|  |  |  |
| --- | --- | --- |
| 测试标题 | 测试输入 | 操作步骤 |
| 输入员工姓名 | Jor | 点击员工姓名输入框，输入：张三 |
| 查询 |  | 点击查询按钮 |

管理员按照表5.6的操作步骤输入测试数据，点击查询按钮，迷糊查询将名字中含有“Jor”的人展示在列表中。查询成功界面如图5.4.



图5.4 查询员工

4.修改员工信息，成功修改员工信息时的测试内ß容、数据输入及操作步骤如表5.8所示。

表5.8 修改员工信息测试用例

|  |  |  |
| --- | --- | --- |
| 测试标题 | 测试输入 | 操作步骤 |
| 选择员工 |  | 选择一条员工记录 |
| 按钮是否生效 |  | 点击修改按钮 |
| 输入地址 | 北京市海淀区中关村 | 在地址输入框输入：北京市海淀区中关村 |
| 输入员工薪资 | 4000 | 点击薪资输入框，输入：4000 |
| 选择员工职位 | 店长 | 点击员工职位下拉菜单选择：店长 |

管理员在员工管理界面按照表5.8的操作步骤完成数据测试，点击保存按钮，会有信息提示“修改成功”，点击查看修改的记录，已经变为修改后的数据。修改界面如图5.5所示。

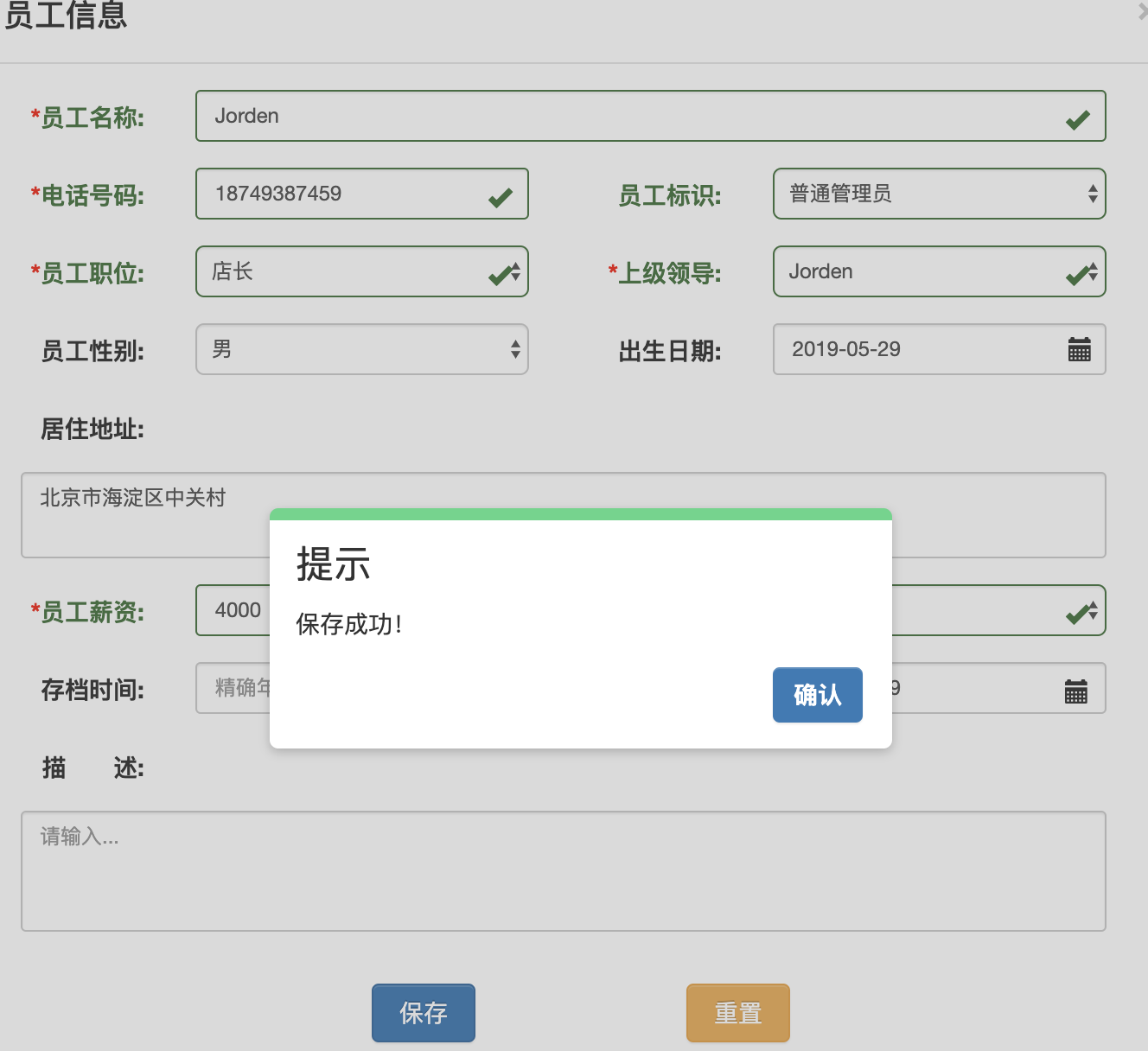


图5.5 修改员工

5.删除员工，成功修改员工信息时的测试内容及操作步骤如表5.9所示。

表5.9 删除员工测试用例

|  |  |  |
| --- | --- | --- |
| 测试标题 | 测试输入 | 操作步骤 |
| 选择员工 |  | 选择一条员工记录 |
| 按钮是否生效 |  | 点击删除按钮 |
| 确定删除 |  | 点击确定删除按钮 |

按照表5.9的操作步骤进行删除员工操作，删除成功后在员工列表中查询不到该员工记录。删除员工界面如图5.6所示。

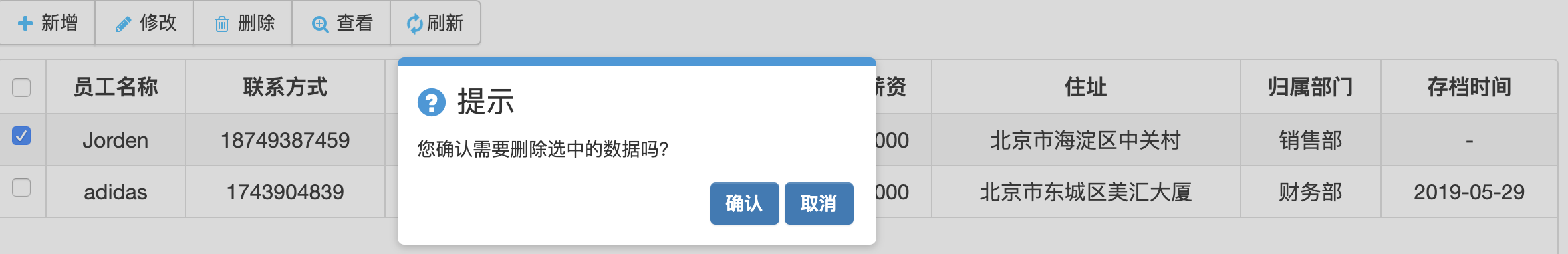


图5.6 删除员工

### 5.4.2 商品管理测试用例

1.查询商品，查询商品的测试标题、测试输入及操作步骤如表5.10所示。

表5.10 查询商品测试用例

|  |  |  |
| --- | --- | --- |
| 测试标题 | 测试输入 | 操作步骤 |
| 输入商品名称 | 输入商品名称：全麦 | 点击商品名称输入框输入：全麦 |
| 选择商品类别 | 选择商品类别：面包 | 点击商品类别下拉框选择：面包 |
| 查询 |  | 点击查询按钮 |

按照表5.10的操作步骤查询商品，查询到的商品信息会展示在列表中，如图5.7所示，数据库中符合的商品只有两条。



图5.7 查询商品

2.添加商品，添加商品的测试内容、测试输入及操作步骤如表5.11所示

表5.11 添加商品测试用例

|  |  |  |
| --- | --- | --- |
| 测试标题 | 测试输入 | 操作步骤 |
| 输入商品名称 | 输入商品名称：巧克力慕斯蛋糕 | 点击商品名称输入框输入：巧克力慕斯蛋糕 |
| 选择商品类别 | 选择商品类别：蛋糕 | 点击商品类别下拉框选择：蛋糕 |
| 选择计量单位 | 选择计量单位：块 | 点击计量单位下拉框选择：块 |
| 输入商品单价 | 输入商品单价：24 | 点击商品单价输入框输入：24 |
| 选择图片 | 选择描述商品的图片 | 点击选择图片按钮，选择图片后点击预览按钮 |
| 提交 |  | 点击保存按钮 |

按照表5.11的操作步骤新增商品，新增的商品信息会展示在列表中，如图5.8所示。



图5.8 新增商品

3.删除商品功能，删除商品时的测试标题、测试输入及操作步骤如表5.12所示。

表5.12 删除商品测试用例

|  |  |  |
| --- | --- | --- |
| 测试标题 | 测试输入 | 操作步骤 |
| 选择商品 |  | 选择一条商品记录 |
| 按钮是否生效 |  | 点击删除按钮 |
| 确定删除 |  | 点击确定删除按钮 |

管理员进入商品管理模块，按照如表5.12操作用例进行操作。如图5.9可以看到删除前的商品总共有8条记录； 操作完成后会看到如图5.10删除商品成功界面，商品名称为“月半蕉”的商品已经在列表中删除，商品总记录数变为7，但是该操作并不是将商品彻底从数据库删除，而是将商品的状态修改为不可用的状态。



图5.9 删除商品图



图5.10 成功删除商品

4.商品入库功能，商品入库的测试内容、测试方式及操作步骤如表5.13 所示。

表5.13 商品入库测试用例

|  |  |  |
| --- | --- | --- |
| 测试标题 | 测试输入 | 操作步骤 |
| 选择商品 |  | 点击商品名称为“绿带子”对应的“商品入库”按钮 |
| 输入入库数量 | 输入入库数量：5 | 点击入库数量输入框，输入5 |
| 选择入库时间 | 选择入库时间 | 点击入库时间输入框，选择当前时间 |
| 保存 |  | 点击保存按钮 |

按照如果5.13的操作步骤执行商品入库操作，入库输入信息界面如图5.11所示，入库成功的商品可以在入库记录中查询到，如图5.12所示。



图5.11 商品入库



图5.12 商品入库成功

### 5.4.3 订单管理测试用例

1.新增订单，新增订单的测试内容、测试输入及操作步骤如表5.14所示。

表5.14 新增订单测试用例

|  |  |  |
| --- | --- | --- |
| 测试标题 | 测试输入 | 操作步骤 |
| 输入商品名称 | 输入商品名称：绿带子 | 点击商品名称输入框选择：绿带子 |
| 输入商品数量 | 输入商品数量：2 | 点击商品数量输入框输入：2 |
| 添加订单明细 |  | 点击保存并录入下一条按钮 |
| 输入商品名称 | 输入商品名称：芋之心 | 点击商品名称输入框选择：芋之心 |
| 输入商品数量 | 输入商品数量：1 | 点击商品数量输入框输入：1 |
| 添加订单明细 |  | 点击保存并录入下一条按钮 |
| 选择支付方式 | 选择支付方式：微信 | 点击支付方式下拉框选择微信 |
| 提交订单 | 提交订单 | 点击提交订单按钮 |

按照如表5.14的操作步骤订购商品，新增订单界面如图5.13所示。



图5.13 新增订单

2.查询订单，查询订单的测试标题、测试输入及操作步骤如表5.15所示。

表5.15 查询订单测试用例

|  |  |  |
| --- | --- | --- |
| 测试标题 | 测试输入 | 操作步骤 |
| 输入客户电话 | 输入客户电话 | 点击客户电话输入框输入：3457564568 |
| 选择支付方式 | 选择支付方式：支付宝 | 点击支付方式下拉框选择：支付宝 |

按照如表5.15的操作步骤输入测试数据，新增订单界点击查询按钮，查询订单页面如图5.14所示。



图5.14 查询订单

3.修改订单明细功能，修改订单时的测试内容、数据输入及操作步骤如表5.16所示。

表5.16 修改订单明细测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 测试标题 | 测试输入 | 操作步骤 | |
| 输入商品数量 | 2 | 点击订单明细列表的原有“商品数量”，在输入框修改商品数量为“2”，点击确定按钮。 |

操作员进入“订单管理”列表，点击订单前方的展开符号“+”可以看到该订单下挂载的订单明细，操作员只可以对明细列表的商品数量进行修改，若想做其他操作需要删除订单重新下单。操作员按照表6.4的操作步骤进行操作，在商品数量输入框将原来的数量修改为“2”，点击确认，商品数量变为：2；商品明细总价变为：10；订单总价变为：60；与预期效果一样，测试通过。修改信息界面如图5.15，修改成功界面如图5.16。

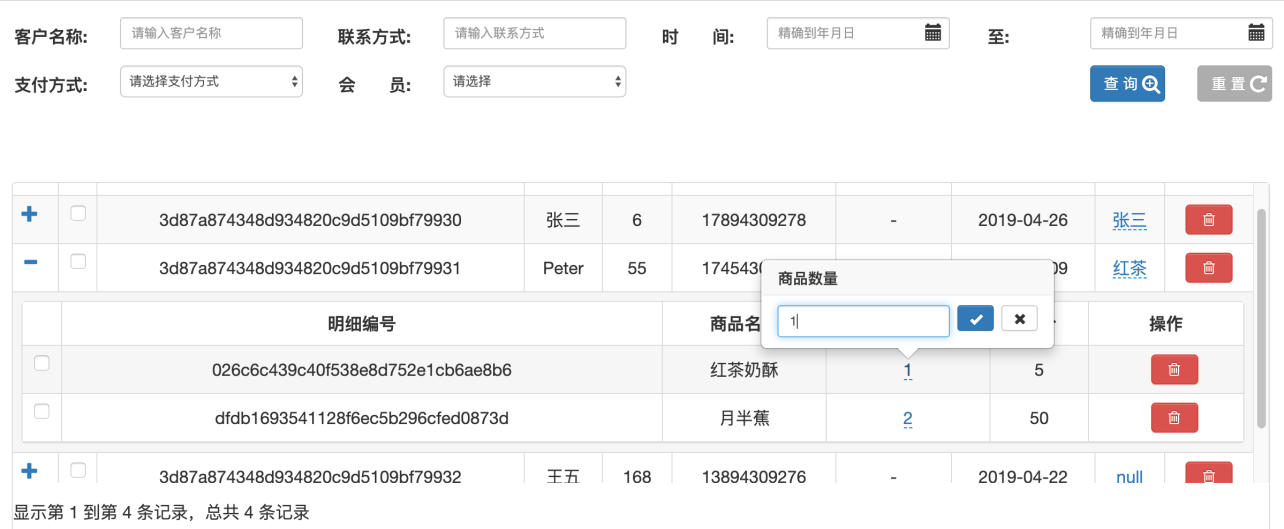


图5.15 修改订单明细商品数量



图5.16 成功修改订单明细商品数量

### 5.4.4 库存管理测试用例

1.查询商品库存的测试标题、测试输入及操作步骤如表5.17所示。

表5.17 查询商品库存测试用例

|  |  |  |
| --- | --- | --- |
| 测试标题 | 测试输入 | 操作步骤 |
| 输入商品名称 | 输入商品名称：茶物语 | 点击商品名称输入框输入：全麦 |
| 查询 |  | 点击查询按钮 |

按照表5.17的操作步骤查询商品库存，查询到的商品信息会展示在列表中，如图5.17所示。

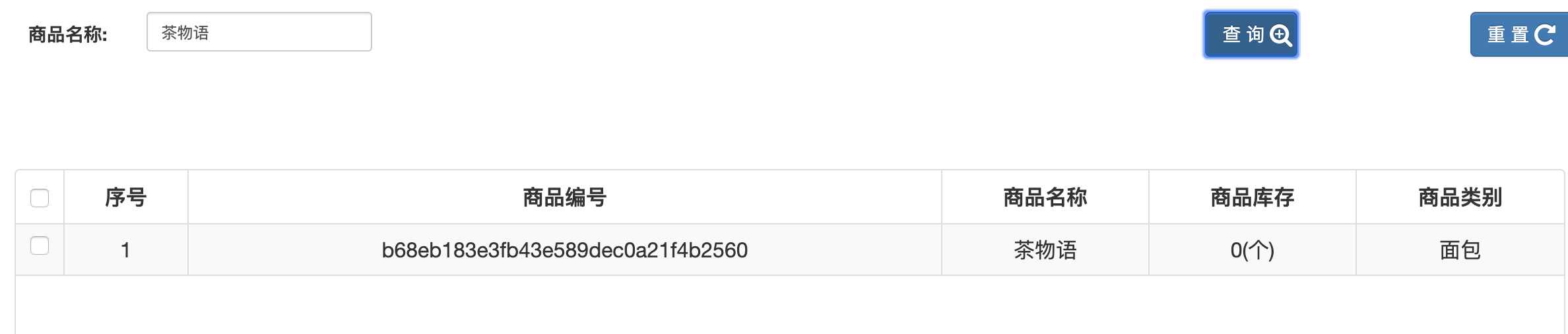


图5.17 商品库存

2.查询商品出库记录，查询商品出库记录的测试输入及操作步骤如表5.18所示。

表5.18 查询商品出库记录测试用例

|  |  |  |
| --- | --- | --- |
| 测试标题 | 测试输入 | 操作步骤 |
| 输入商品名称 | 输入商品名称：绿带子 | 点击商品名称输入框输入：绿带子 |
| 查询 |  | 点击查询按钮 |

按照表5.18的操作步骤查询商品库存，会查到该商品所有的出库记录，查询到的商品信息会展示在列表中，如图5.18所示。



图5.18 商品出库记录

## 5.5 测试结果分析

本次系统测试采用分模块测试的方式，根据系统运行效果及测试结果来看，系统运行正常界面响应速度快、数据传输准确迅速，需求得到良好的实现。数据库数据处理得当、精准，达到要求。

第6章 总结与展望

## 6.1 总结

该系统以IntelijIDEA为开发平台应用于Java语言、MySQL数据库等技术来编写设计，基本上实现了前期市场调研需求式样书上的需求，前端网站页面的设计简单爽目且易于操作管理，后端开发代码采用优化之后的Sql语句实现增加、删除、更新的功能，业务逻辑处理部分的代码简洁实现高内聚低耦合，降低数据冗余。

蛋糕店管理系统的设计提高店内蛋糕生产率和售卖率，改善管理人员传统的手工记账管理方式，结合了实际当中销售的步骤，实时更新系统中蛋糕的种类和数量，通过订单明细记录售出蛋糕的过程，提供给客户优质的服务。系统整体完成之后，我针对每个模块的每个功能进行了测试，测试过程中遇到的问题及时修改并优化。

由于开发周期时间有限，系统还有许多方面需要完善，后期我会继续实现该系统的设计，将自己大学所学的理论知识做到学以致用。

## 6.2 展望

一个完善并且合理的管理系统，是需要时间和全身心的投入，要不断利用新的知识扩充系统优化功能，即便我现在所开发的系统已经初步完成了最初的需求设计，但仍有许多的不足，和以下存在的问题：

1.在客户端系统上也没有提供完善的帮助系统，当操作人员遇到困难时，不能对于常见问题得到及时的解答。

2.系统还没有实现“客户”登录功能，待该功实现后客户可以登录系统，在线选购商品通过支付宝、微信等方式自行支付。

谢 辞

行文至此，蓦然回首，时光匆匆，转瞬即逝。一年以来的毕业设计终于完美收官，从项目的搭建到开发，设计的整理和修改直至完成编写的整个过程，在遇到问题时，会和身边的同学一起探讨解决问题，优化代码，后期一轮又一轮的论文修改，老师都会仔细的给每一个错误加上批注，现在向她们真诚的帮助表达诚挚的谢意。

大学的四年时光带给了我很多的美好，很想回到军训时的生活，初入大学略带懵懂无知，却又不敢横冲直撞，和室友操场上嬉笑打闹，一动不动的站军姿，还记得被教练夸赞优秀。感谢母校的悉心培养和给予我优质的教学环境，专业老师风趣的讲课方式，让枯燥乏味的理论知识印刻在脑海里，感谢身边的那位朋友能够感知我的难过陪我一起欢笑，感谢给予我最大帮助的指导老师，面对面的告诉我论文出现的问题，在开发系统过程中也给了我很多宝贵的意见和想法。很幸运还能够在即将毕业遇到这么负责、专业和细心的老师，在此向指导老师表示我最衷心的感谢!

曾益吾所不能，求学之路，道阻且长，困顿迷茫有之，欢呼雀跃亦有之，幸有老师之悉心教导。

参考文献

[1]苏选良.管理信息系统.北京：电子工业出版社，2013年,198-200.

[2]李刚.疯狂Java讲义.（第2版），北京：电子工业出版社，2014年,120-121.

[3]赵景晖.Java 程序设计.北京：北京机械工业出版社，2005年，230-231.

[4]明日科技.Java从入门到精通.（第3版），北京：清华大学出版社，2014年，260-263.

[5]车传文.J2EE轻量级框架在预算管理系统中的应用研究[D].大连：大连海事大学，2007年，410-411.

[6]刘苇.基于J2EE的多种持久化机制自适应框架的研究[D].河南：河南科技大学，2008年，321-322.

[7]叶长春.基于MVC的Struts框架的应用研究[D].武汉：武汉理工大学，2008年，210-212.

[8]Roumeliotis-Curator R.Enterprise Java Development[M]．O'Reilly Media，Inc．2015年，214-216.

[9]朱育发.jQuery与jQuery Mobile开发完全技术宝典.北京：中国铁道出版社，2014年，354-357.

[10]汤东，张富银．JQUERY入门实战.四川：西南财经大学出版社，2015年，32-36.

[11]高云.深入解析jQuery架构设计与实现原理.机械工业出版社，2014年，261-263.

[12]雪松.J2EE网络编程标准教程.（第一版），上海：上海科学普及出版社，2004年，323-335.

[13]计磊.精通J2EE-Eclipse, Struts, Hibernate及Spring整合应用案例.北京：人民邮电出版社，2006年，212-215.

[14]展涛.基于J2EE技术的高校OA系统研究设计[D].青岛：中国海洋大学，2008年，321-323.

[15]冯新扬,范颖,崔凯.利用设计模式改进分层架构[J].北京：计算机工程与设计,2007年，210-212.

[16]陶宏才.数据库原理及设计.北京：清华大学出版社，2004年，200-203.

[17]Farcic V，Garcia A．Test-Driven Java Development[M]．Packt Publishing，2015年，230-250．

[18]Jackson W．Setting Up a Java 8 Game Development Environment[M]．Beginning Java 8 Games Development. Apress，2014年，1-4．

[19]Roumeliotis-Curator R.Enterprise Java Development[M]．O'Reilly Media，Inc．2015年，234-235．

[20]Ferrel．Distributing web applications across a pre-existing web：US，US8943035[J]．2015年，254-259．

# 附 录

application.properties配置文件：

|  |
| --- |
| Spring.mvc.view.prefix=/WEB-INF/views/  spring.mvc.view.suffix=.jsp  spring.devtools.restart.enabled=true  #打开该路径的访问权限  spring.mvc.static-path-pattern=/static/\*\*  #file upload  #single file  spring.servlet.multipart.max-file-size=1048576  spring.servlet.multipart.max-request-size=10485760  # set codding  spring.http.encoding.charset=utf-8  spring.http.encoding.force=true  spring.http.encoding.enabled=true  # format  spring.mvc.date-format=yyyy-MM-dd |

application.yml配置文件：

|  |
| --- |
| spring:  datasource:  username: root  password: 960218lee  driver-class-name: com.p6spy.engine.spy.P6SpyDriver  url:jdbc:p6spy:mysql://localhost:3306/bakery?serverTimezone=GMT%2B8  type: com.alibaba.druid.pool.DruidDataSource  maxIdle: 10  maxWait: 60000  minIdle: 5  initialSize: 5  maxActive: 20  timeBetweenEvictionRunsMillis: 60000  minEvictableIdleTimeMillis: 30000  validationQuery: select 1 from dual  testWhileIdle: true  testOnBorrow: false  testOnReturn: false  poolPreparedStatement: true  maxPoolPreparedStatementPerConnectionSize: 20  useGlobalDataSourceStat: true  # mybatis config  mybatis:  config-location: classpath:sqlMapconfig.xml  mapper-locations: classpath:mapping/\*.xml  type-aliases-package: com.bakery.application.entity  # pageHealper config  pagehelper:  helper-dialect: mysql  reasonable: true  support-methods-arguments: true  params: count=countSql |

MVCConfig配置类：

|  |
| --- |
| @Configuration  public class MVCConfig implements WebMvcConfigurer {  @Autowired  LoginInterceptor loginInterceptor;  //注册视图控制器  @Override  public void addViewControllers(ViewControllerRegistry registry) {  registry.addViewController("oms").setViewName("login");  }  //注册消息转化器  @Override  public void configureMessageConverters(List<HttpMessageConverter<?>> converters) {  FastJsonHttpMessageConverter fc = new FastJsonHttpMessageConverter();  FastJsonConfig fastJsonConfig = new FastJsonConfig();  fastJsonConfig.setSerializerFeatures(SerializerFeature.PrettyFormat);  fc.setFastJsonConfig(fastJsonConfig);  converters.add(fc);  }  //资源映射路径  @Override  public void addResourceHandlers(ResourceHandlerRegistry registry) {  registry.addResourceHandler("/image/\*\*").addResourceLocations("file:/Users/liting/文档/upload/");  }  //注册拦截器  @Override  public void addInterceptors(InterceptorRegistry registry) {  registry.addInterceptor(loginInterceptor).addPathPatterns("/\*\*") .excludePathPatterns("/error400Page","/error401Page","/error404Page","/error403Page","/error405Page","/error500Page","/oms","/login","/regisOms","/loginValidate","/getValidNumber","/validateNumber","/regisValidate","validateRegisNum","getRegisValidNum","pwdGetValidNum","repwdValidateNum","/\*\*/\*.css","/\*\*/\*.js", "/\*\*/\*.png", "/\*\*/\*.jpg", "/\*\*/\*.jpeg");  }  } |