**概要设计说明书**

**目录**

[1引言 2](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464958)

[1.1编写目的 2](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464959)

[1.2背景 2](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464960)

[1.3参考资料 2](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464962)

[2总体设计 3](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464963)

[2.1需求规定 3](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464964)

[2.2运行环境 3](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464965)

[2.3基本设计概念和处理流程 4](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464966)

[2.4结构 4](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464967)

[3接口设计 3](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464971)

[3.1用户接口 3](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464972)

[3.2外部接口 3](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464973)

[3.3内部接口 5](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464974)

[4运行设计 5](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464975)

[4.1运行模块组合 6](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464976)

[4.2运行控制 6](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464977)

[4.3运行时间 6](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464978)

[5系统出错处理设计 7](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464983)

[5.1出错信息 7](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464984)

[5.2补救措施 9](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464985)

[5.3系统维护设计 1](file:///D:\绘图\customer-order-master\document\概要设计说明书.doc#_Toc521464986)0

# 1引言

某公司希望为其商品货物建立一个管理系统。消费者的每个订单可以有一种或多种商品。每个订单需要为其开具发票。消费者可以通过多种方式来支付，例如支票、信用卡或者现金。商品被分为几类，当商品货物低于库存警戒线需要从供应商订购时，工作人员需要提出一个采购订单。系统要跟踪记录公司买进、卖出的货物以及其他的费用。

## 1.1编写目的

帮助客户更加清晰地的了解该系统整体框架及功能，为系统开发者开发整个系统提供主要依据，为系统开发以及设计人员的工作提供基础保证。

## 1.3定义

* 用户登录界面友好，可操作性及安全性能较好，能对不同级别的使用者进行限制，以保证系统以及数据库的安全。
* 数据库的可维护性好，数据的录入、删除及更改均能顺利完成，并能实现动态更新。
* 软件对操作系统的兼容性良好，可移植性良好。

## 1.2背景

软件项目名称：客户订购系统

项目任务提出者：软件工程课程设计指导老师

开发者：万鸿达、陈永涛、寇喆、李鹏飞、李欣宇、史绪豪、凌寒、张嘉奇

用户：管理员，客户

## 1.3参考资料

书本资料：

《软件工程开发与技术》第二版 西安电子科技大学出版社 江开耀主编

《软件体系结构原理、方法与实践》第二版 清华大学出版社

《软件需求工程》 毋国庆等编著 机械工业出版社

视频资料：

《Java编程思想》 Bruce Eckel著 机械工业出版社

《MySQL从入门到精通》 <https://www.bilibili.com/>

Java Api 官方文档

Spring、SpringMVC、Mybatis、MybatisPlus、SpringBoot、SpringCloud官方文档

Redis官方文档

Vue、Node.js官方文档

# 2总体设计

## 2.1需求规定

经过需求分析阶段的工作，已经清楚系统必须完成的工作，下面的工作就应该是决定“如何做”的问题。概要设计的基本目的就是“概括地说系统应该如何实现？”通过该阶段的工作将划分出组成系统的物理元素——程序、文件、数据库、文档等，另一方面的主要工作是设计软件的结构，即确定系统都由哪些模块组成及模块之间的相互关系。

## 2.2运行环境

### 2.2.1硬件配置

硬件：PC机8台。

### 2.2.2 软件配置

1.  安装nacos2.2

2.  安装mysql8.0+,jdk1.8

3.  安装redis3.2.100

4.  安装Node.js 18.12.1

5.  安装IDEA编辑器,集成maven3.6.1 +tomcat9

### 2.2.3 软件架构说明

1. 前后端分离项目，采用分布式架构

2. 前端使用vue框架

3. 后端使用SpringCloud微服务框架开发

4.主要技术有：SpringCloud+SpringBoot+MybatisPlus+Redis

## 2.3基本设计概念和处理流程

### 2.3.1设计概念

1.采用面向对象的开发与设计理念。可以进行抽象数据类型为特点的基于对象的程序设计，他的前提是对系统的整体有一个全面的了解，通过他可以保证系统的优良，来保证开发出的系统具有良好的稳定性与实用性。

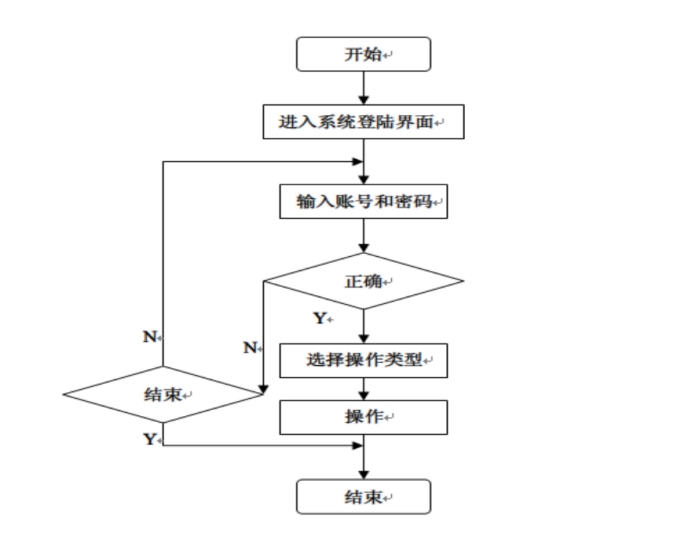
2.采用模块化设计。是将系统分为若干个功能模块，逐个实现系统的功能，有利于软件的开发、测试以及维护等功能，加强了系统的稳定性以及后期的维护过程。

3.简单方便的系统界面。设计简单友好的系统界面，方便用户较快的适应系统的操作。

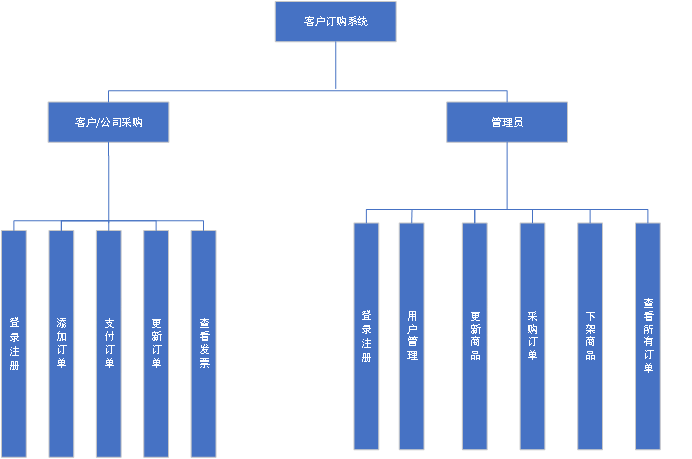
4.速度优先原则。在开发过程中由于实际的需要，所开发出来的系统要尽量做到简洁，一方面利于测试人员的工作，最主要的是要是整个系统所占的资源达到最少，运行速度尽可能的达到最快。

5.设计既要突出重点，又要细致周到。

### 2.3.2管理系统流程图



## 2.4系统结构图



## 3.用户接口

说明将向用户提供的命令和它们的语法结构，以及软件的回答信息。

## 3.2外部接口

硬件环境：内存: 512M以上;硬盘:至少80G以上:CD-ROM: 32倍速以上;

软件环境： 操作系统：Win 10数据库：MySQL 2.7 、JDBC

## 3.3内部接口

登录模块：验证用户身份，进入个人账户；

查询客户信息模块：显示客户的信息；

修改客户信息模块：提示操作是否完成，显示更新后的信息；

删除客户信息模块：提示操作是否完成，显示更新后的信息；

查询订单信息模块：显示采购信息；

更新订单信息模块：提示操作是否完成，显示更新后的信息；

添加订单信息模块：提示操作是否完成，显示更新后的信息；

删除订单信息模块：提示操作是否完成，显示更新后的信息；

支付订单信息模块：提示操作是否完成，显示更新后的信息；

查看发票模块：提示操作是否完成，显示更新后的信息；

添加商品模块：提示操作是否完成；

下架商品模块：提示操作是否完成，显示更新后的信息；

修改商品模块：提示操作是否完成，显示更新后的信息；

查询商品模块：提示操作是否完成，显示更新后的信息；

查询所有订单：提示操作是否完成，显示更新后的信息；

# 4运行设计

## 4.1运行模块组合

客户：浏览商品列表，查看商品库存、加入购物车、提交订单、付款、查看发票；

系统管理员：修改商品库存、添加、删除商品种类、处理订单、采购订单商品；

## 4.2运行控制

用户密码错误不能进入系统，各模块均设计提示信息。客户填写订单并提交后，反馈给客户相关确认信息。客户进行信息确认并支付费用后，系统确认支付信息，并反馈给客户。

（1）客户登录注册时序图



1. 客户更新订单



1. 客户添加订单时序图



1. 客户支付订单时序图



1. 客户查看发票时序图



（6）管理员更新商品时序图



（7）管理员下架商品时序图



（8）管理员采购订单时序图



## 4.3运行时间

登录时间小于3秒，要求每种运行模块组合将占用资源运行时间小于2秒，网页反应速度小于3秒。

# 5系统出错处理设计

## 5.1出错信息

用一览表的方式说朗每种可能的出错或故障情况出现时，系统输出信息的形式、含意及处理方法。

|  |  |  |
| --- | --- | --- |
| 出错情况： | 输出信息： | 处理方法： |
| 用户名格式不正确 | 系统提示错误信息 | 用户重新输入用户名 |
| 注册密码输入格式不正确 | 系统提示错误信息 | 重新按照要求输入用户密码 |
| 确认密码输入不正确 | 提示与密码不一致 | 输入与密码一致的确认密码 |
| 系统出错 | 找不到页面 | 刷新页面并重启服务器 |

## 5.2补救措施

说明故障出现后可能采取的变通措施，包括：

1. 快速定位故障服务：使用监控工具（例如，链路跟踪和日志聚合工具）来帮助快速定位故障服务。
2. 发布故障服务的替代版本：如果故障服务无法立即修复，可以发布替代版本来满足业务需求。
3. 将故障服务从服务注册中心中摘除：如果故障服务无法提供有效服务，应将其从服务注册中心中摘除，以防止其影响其他服务。
4. 恢复故障服务：修复故障服务，并将其重新发布到生产环境中。
5. 进行故障排除：分析故障原因，并采取预防措施，以防止类似故障再次发生。
6. 在处理微服务故障时，还可以考虑使用自动化工具（例如，自动重启和自动回滚工具）来帮助解决问题。这些工具可以使得处理故障的过程更加快速和简单。
7. 当系统出现故障后，合理合情提示用户并尽可能给出解决方案。
8. 在系统设置异常处理机制及时处理系统异常，并记录在日志文件中，方便开发人员后序处理。
9. 故障无法处理时，重启系统。

## 5.3系统维护设计

1.定期检查数据库内容，将数据库内容备份，将代码上传到gitee仓库当中以防电脑故障代码丢失；

2.前端界面以客户的需求和当下潮流为出发点，后序进行升级、优化界面；

3.可根据用户提出的新功能、进行系统升级。