参 考 文 档

**课题名称:**

**授课教师：**

目录

[第1章 系统分析与设计 3](#_Toc184567495)

[1.1系统设计说明 3](#_Toc184567496)

[1.2系统应用框架 4](#_Toc184567497)

[1.2.1 SpringBoot 介绍 4](#_Toc184567498)

[1.2.2 Vue. js 介绍 4](#_Toc184567499)

[1.2.3 MYSQL数据库 4](#_Toc184567500)

[1.2.4 B/S结构 5](#_Toc184567501)

[第2章 系统需求分析 6](#_Toc184567502)

[2.1系统可行性分析 6](#_Toc184567503)

[2.1.1技术可行性 6](#_Toc184567504)

[2.1.2经济可行性 6](#_Toc184567505)

[2.1.3操作可行性 6](#_Toc184567506)

[2.2非功能性需求 7](#_Toc184567507)

[2.2.1 安全需求 7](#_Toc184567508)

[2.2.2 性能需求 7](#_Toc184567509)

[2.2.3可用性需求 7](#_Toc184567510)

[2.2.4 可靠性需求 7](#_Toc184567511)

[2.3系统功能需求说明 8](#_Toc184567512)

[第3章 数据库设计 9](#_Toc184567513)

[3.1概念设计 9](#_Toc184567514)

[3.2表结构设计 21](#_Toc184567515)

[3.3数据库ER关系图 25](#_Toc184567516)

[第4章 系统实现 26](#_Toc184567517)

[4.1项目截图 26](#_Toc184567518)

[4.2代码实现 26](#_Toc184567519)

[第5章 系统测试与结果分析 26](#_Toc184567520)

[5.1系统功能测试方法概述 26](#_Toc184567521)

[5.2系统功能测试 27](#_Toc184567522)

[5.3系统兼容性分析 27](#_Toc184567523)

1. 系统分析与设计

## 1.1系统设计说明

本系统采用 SpringBoot+Vue和 Mysql整合来进行开发，系统能够保证数据的安全性，提高可移植性，能够满足用户学习要求。

调研网上同类系统功能，要求本系统包括2种角色：用户和管理员。功能如下：

用户功能

登录注册功能

首页：查看穿搭分享、商品信息；

穿搭分享：可以分页查看用户分享的穿搭，并且可以按照条件筛选，点击查看可以进入详情，可以评论，详情右方推荐按照评论高低排序；

商城：可以分页查询商品，支持多条件筛选、排序查看，点击详情，支持立即购买，可以加入购物车，查看购买用户发布的商品评价；

社区：用户可以分享穿搭心得，可以最新查询（根据发布时间），也可以根据热门查看（点赞量高低），或者可以按照协调过滤算法推荐。同时有收藏榜、评论榜。进入帖子详情，用户可以点赞、评论。

购买：用户可以选择多件商品加入购物车，加入后一起购买，支持购物车数量变更或移除购物车商品。

订单查看：查看订单信息，可以支付订单，或者收货

其它管理：可以查看自己的评论、点赞、维护自己的穿搭分享和帖子信息。

管理员功能：

用户管理、分类管理、商品管理、订单管理、商品评论管理、穿搭信息管理、穿搭评论管理、帖子管理、帖子评论管理、收藏管理。

## 1.2系统应用框架

### 1.2.1 SpringBoot 介绍

SpringBoot是由 Pivotal 团队提供的开源后端框架,设计目的在于简化新 Spring 应用的初始搭建以及开发过程,提供了一种新的编程范式,可以更加快速、便捷地开发 Spring 项目。 在开发过程中,可以专注于应用本身的开发,而无须纠结于 Spring 的配置。SpringBoot基于 Spring4 进行设计,继承了 Spring 框架的优秀基因,具有控制反转和面向切面两大核心,框架内部能有效管理所有的对象创建,并维护着对象之间的依赖关系。 SpringBoot的高内聚低耦合、声明式事务的支持、方便继承框架等特性,使其成为后端服务器应用开发的主流技术。

### 1.2.2 Vue. js 介绍

Vue. js 是于 2014 年在 Github 上公开的一款用于创建用户界面的渐进式 JavaScript 框架,也是用于创建单页应用 SPA的Web应用框架。Vue 改变了前端开发者使用 jQuery 直接对页面上的 DOM 元素进行操作的习惯,通过数据和模板双向绑定的形式更好地组织和简化了 Web 开发。目前，Vue的生态链和周边工具库如 vue – cli，Vuex，Vue - router等不断完善,深受越来越多开发者与企业的青睐。

Vue 的核心思想基于模型视图视图模型( Model-View-ViewModel，MVVM) 原理,使得 Vue 可以对页面上需要使用的数据进行统一管理,并能够很方便地获取数据的更新,实现页面视图与数据模型的交互。 Vue 中提供的一系列属性及指令, 比如: methods, watch, v-if, @click 等,都让开发者省去了获取 DOM、操作 DOM 的过程，通过对数据的控制即可修改视图的渲染。

### 1.2.3 MYSQL数据库

网站的开发必须配套相应数据库，数据库具有一定的组织结构，能够存放和管理数据信息，在以前数据库的功能仅仅是数据保存和管理操作，但是时代的变迁和发展，现在的数据库演变成了数据处理的方式，数据库从最开始的简单存放数据表格信息到现在的能够存放成千上万数据的大型数据库，期间还是经历了许多的改革。

本次开发的系统设计与实现使用的数据库是MYSQL数据库，该数据库运行速度快，安全性能也很高，而且对使用的平台没有任何的限制，所以被广泛应运到系统的开发中。MySQL是一个开源和多线程的关系管理数据库系统，MySQL是开放源代码的数据库，具有跨平台性，虽然功能未必强大，导致很多人都了解这个数据库的基本应用，在数据库中，总共建立了10几个表，这里面每个表都是相对应的，都各自有各自的联系，数据库意义重大，如果没有数据库的链接，就没办法运行程序，这显然可以看见数据库与程序的重要性，是紧密相连接的。

### 1.2.4 B/S结构

B/S（浏览器/服务器）结构是目前主流的网络化的结构模式，它能够把系统核心功能集中在服务器上面，可以帮助系统开发人员简化操作，便于维护和使用。只需要用户在用户端安装360浏览器、谷歌浏览器、QQ浏览器等当前大众浏览器，在电脑里面安装sqlserver、mysql数据库等数据库。安装好的浏览器与服务器端的数据库进行信息数据的交互。很多专门软件能够做到的事情，采用B/S结构模式也能实现，它能够结合Web浏览器技术，ActiveX技术以及多种脚本语言等技术。帮助程序开发者节约了不少开发成本。目前B/S结构成为程序开发主流结构，它最好的地方就是没有地点限制还不用专门安装软件，笔记本或者电脑能够上网就能访问系统。系统使用B/S进行开发在后期系统维护上面就会很省事，不用什么问题都在服务器上面操作，简单的用户端处理就解决部分问题，开发出来的程序跟用户交互性上面也会增强，还可以实时刷新浏览器进行程序局部的数据信息更新。



第2章 系统需求分析

本章主要的工作是对系统进行需求分析，分析系统可行性、归纳该系统功能需求和对功能需求进行说明。

## **2.1系统可行性分析**

本系统包括2种角色:普通用户和管理员。主要功能为：用户管理、分类管理、商品管理、订单管理、商品评论管理、穿搭信息管理、穿搭评论管理、帖子管理、帖子评论管理、收藏管理

### 2.1.1技术可行性

技术上的可行性分析主要分析技术条件能否顺利完成开发工作，软、硬件能否满足需要。本项目主要结合springboot、JavaScript、MySql、CSS、vue等技术支持，用IntelliJ IDEA进行开发，并以休闲简洁的用户界面布局设计，便于用户理解、操作。根据本系统数据的特点，采用标准MYSQL语句，以便将来的扩展和移植。系统将采用数据库建模工具，根据系统功能模块的设计，构建出整个数据库。在构建数据库时，也会定义好数据库表的约束、关联以及索引。因此本项目的开发平台已成熟可行。

### 2.1.2经济可行性

鉴于计算机技术发展异常迅猛，在硬件软件配置以及开发技术均以可行的情况下开发这样一个系统成本不会很高，但其可以方便购物需求，其必将有比较宽阔的市场，因此该系统在经济可行性上是可行的。

### 2.1.3操作可行性

系统用户界面操作简洁、易用、灵活，风格统一易学。前端方面具有兼容各大主流浏览器的能力，十分方便操作。而且系统合理的利用资源，保证前后台数据操作的效率，以及在数据响应和界面承载方面都要达到不会出现界面混乱、数据报错、触发按钮功能缺失、操作频繁或者快速容易崩溃的问题。总的来说，该系统是很贴合用户的使用习惯的，操作十分容易上手。

## **2.2非功能性需求**

### 2.2.1 安全需求

严格权限访问控制，用户在经过身份认证后，只能访问其权限范围内的数据，只能进行其权限范围内的操作。所有用户属性信息的传输，均需在接口层数据加密。

### 2.2.2 性能需求

系统从启动到显示首页，时间不超过5秒；

系统内操作的响应时间不能超过3秒，3秒后无响应，需要给出提示：网络繁忙，请稍后再试，且页面可点击重新加载；

程序闲时没有异常的CPU占用，忙时没有异常的峰值占用；

### 2.2.3可用性需求

操作系统支持：当下所有主流系统；

环境贴切：使用用户熟悉的界面和概念；

用户可控：页面出口应明显、退出时需用户手动确认，防止用户误操作；

一致性：保持所有相同界面设计/操作/反馈/概念文案显示的一致性；

人性化帮助：在重要的功能入口处提供相应的帮助/反馈入口。

### 2.2.4 可靠性需求

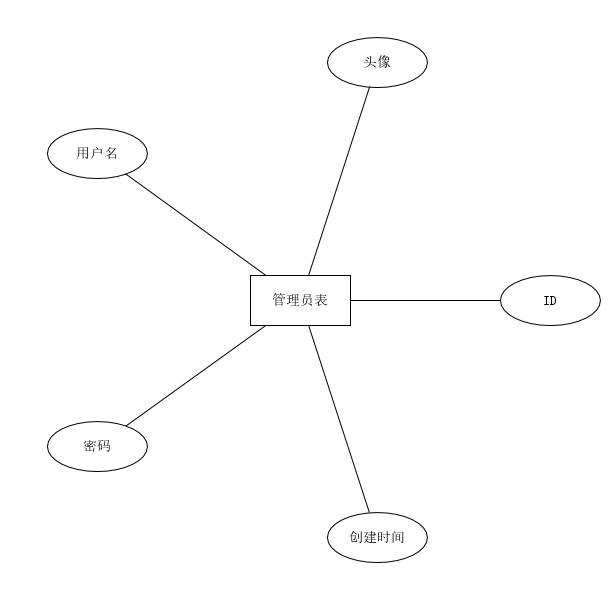
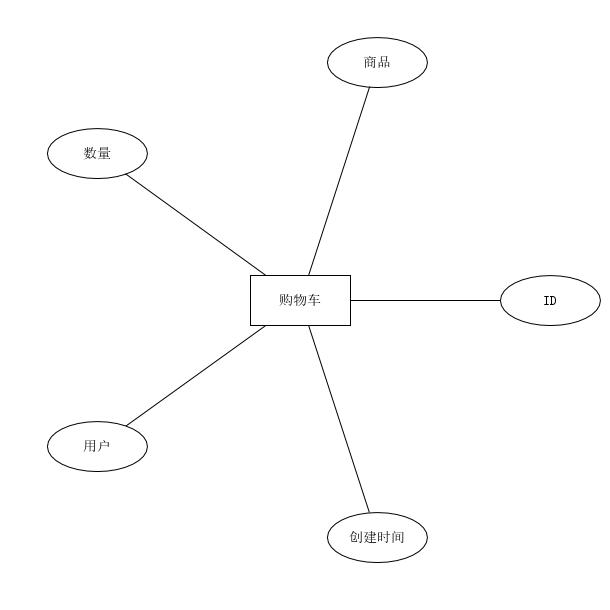
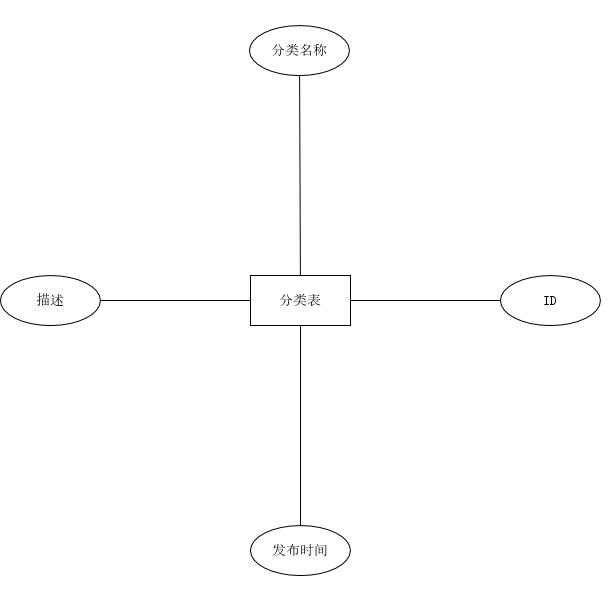
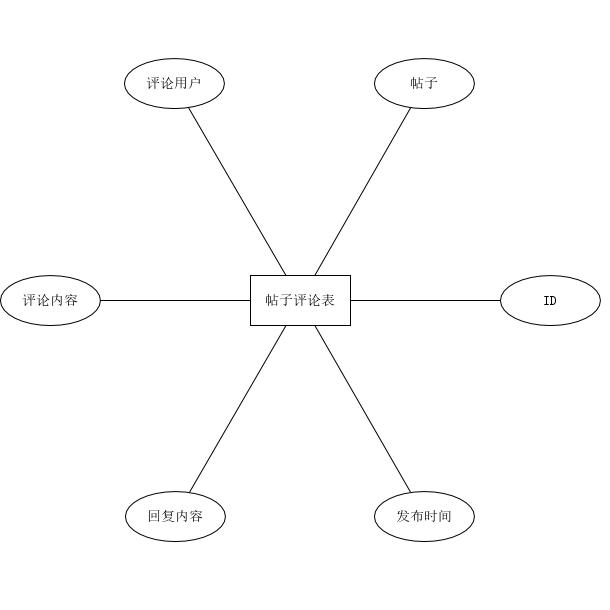
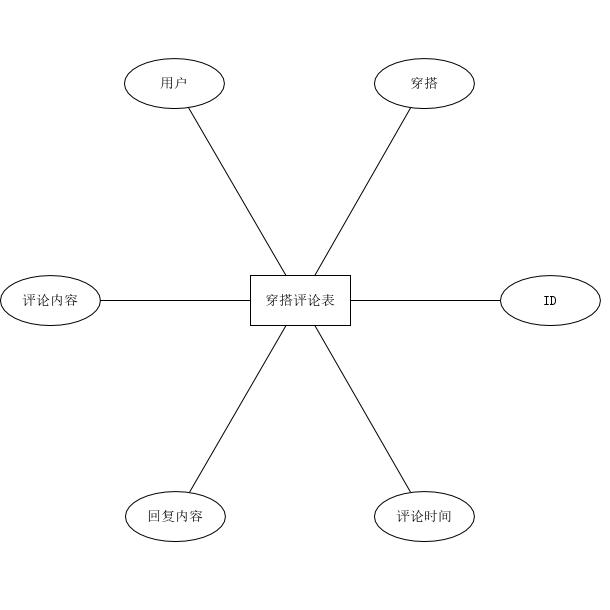
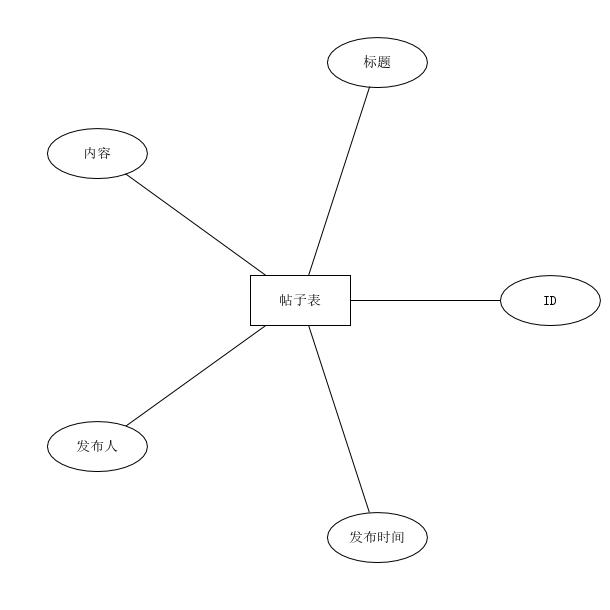
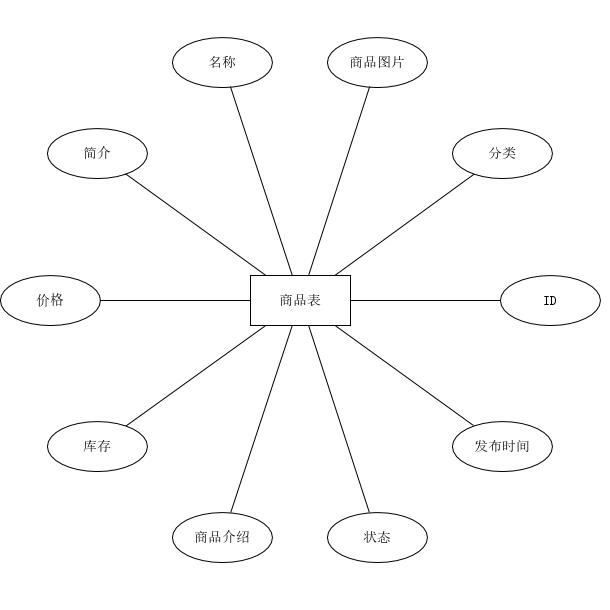
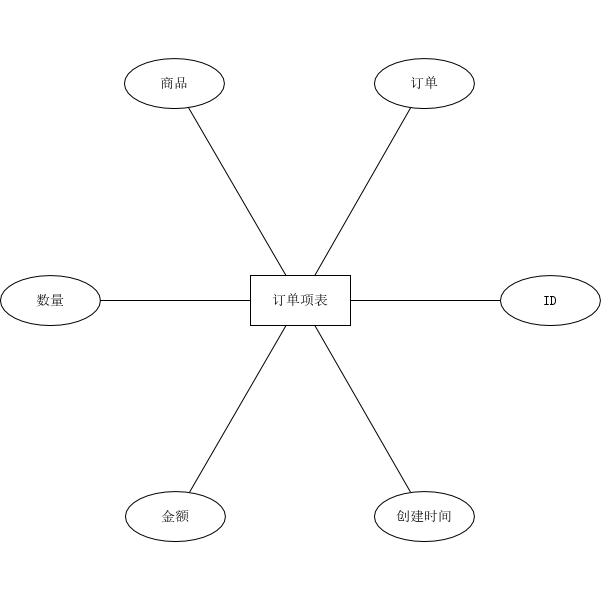
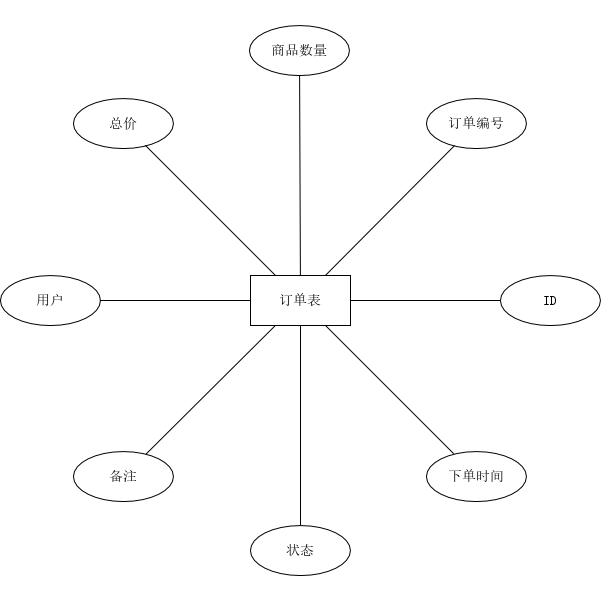
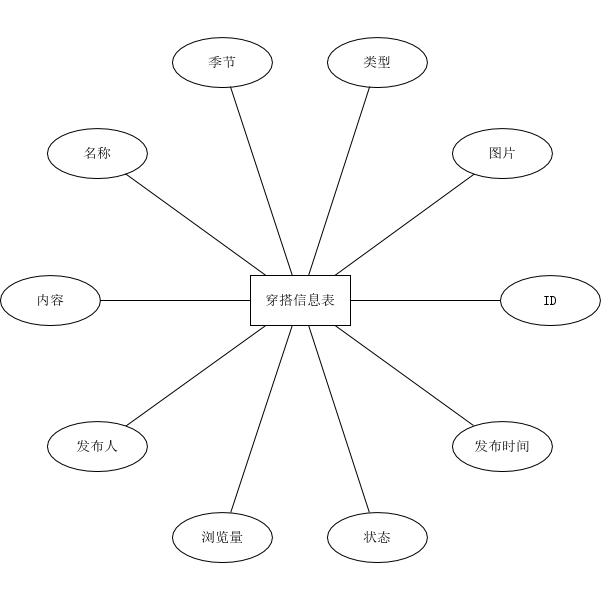
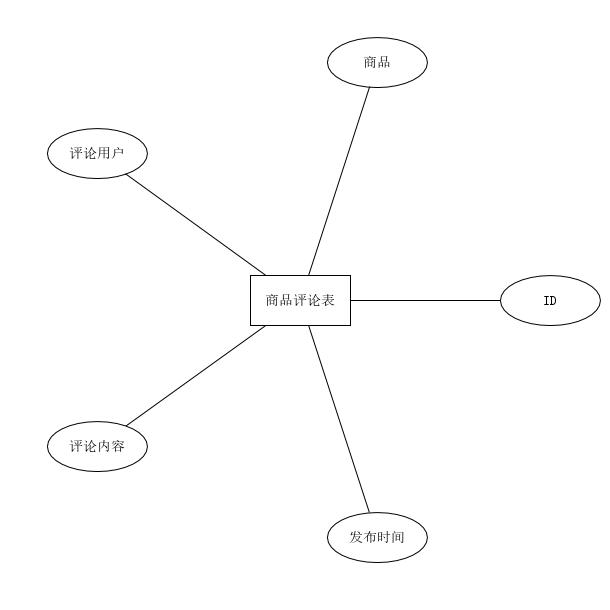
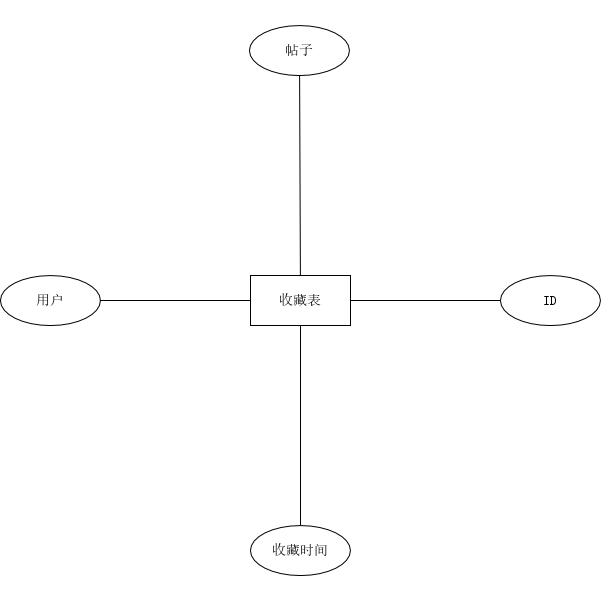
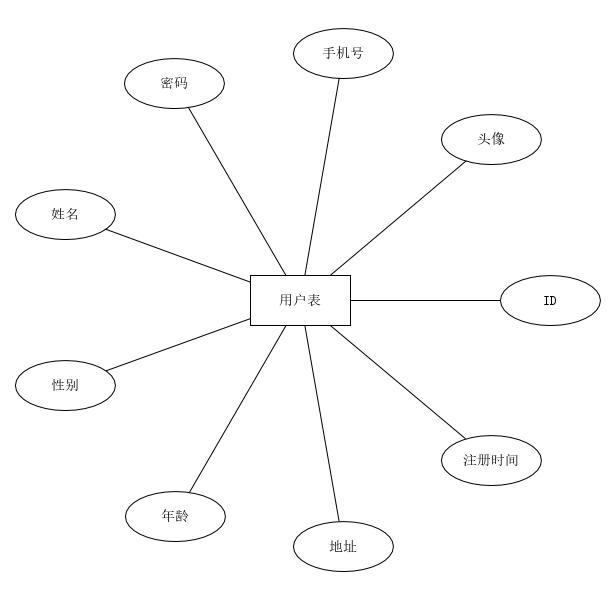
对输入有提示，数据有检查，防止数据异常。能处理系统运行过程中出现的各种异常情况，如：人为操作错误、输入非法数据等，系统能正确的处理，恰当的回避。

## **2.3系统功能需求说明**

根据系统总体业务流程，结合系统操作的分级、分角色及系统操作的便捷性，从系统操作功能上，将系统划分成几个功能模块，各个角色单独彼此分离，保证系统独立性，系统功能全面。

第3章 数据库设计

## 3.1概念设计



## 3.2表结构设计

admin(管理员表)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为空 | 主键 | 自增 | 默认值 | 备注 |
| id | int(11) | 否 | 是 | 是 |  | ID |
| image | varchar(100) | 是 | 否 | 否 |  | 头像 |
| username | varchar(20) | 是 | 否 | 否 |  | 用户名 |
| password | varchar(20) | 是 | 否 | 否 |  | 密码 |
| create\_time | datetime | 是 | 否 | 否 |  | 创建时间 |

cart(购物车)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为空 | 主键 | 自增 | 默认值 | 备注 |
| id | int(11) | 否 | 是 | 是 |  | ID |
| gid | int(11) | 是 | 否 | 否 |  | 商品 |
| num | int(11) | 是 | 否 | 否 |  | 数量 |
| uid | int(11) | 是 | 否 | 否 |  | 用户 |
| create\_time | datetime | 是 | 否 | 否 |  | 创建时间 |

category(分类表)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为空 | 主键 | 自增 | 默认值 | 备注 |
| id | int(11) | 否 | 是 | 是 |  | ID |
| name | varchar(50) | 是 | 否 | 否 |  | 分类名称 |
| remark | varchar(100) | 是 | 否 | 否 |  | 描述 |
| create\_time | datetime | 是 | 否 | 否 |  | 发布时间 |

comment(帖子评论表)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为空 | 主键 | 自增 | 默认值 | 备注 |
| id | int(11) | 否 | 是 | 是 |  | ID |
| fid | int(11) | 是 | 否 | 否 |  | 帖子 |
| uid | int(11) | 是 | 否 | 否 |  | 评论用户 |
| content | varchar(500) | 是 | 否 | 否 |  | 评论内容 |
| reply | varchar(500) | 是 | 否 | 否 |  | 回复内容 |
| create\_time | datetime | 是 | 否 | 否 |  | 发布时间 |

discuss(穿搭评论表)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为空 | 主键 | 自增 | 默认值 | 备注 |
| id | int(11) | 否 | 是 | 是 |  | ID |
| oid | int(11) | 是 | 否 | 否 |  | 穿搭 |
| uid | int(11) | 是 | 否 | 否 |  | 用户 |
| content | varchar(500) | 是 | 否 | 否 |  | 评论内容 |
| reply | varchar(500) | 是 | 否 | 否 |  | 回复内容 |
| create\_time | datetime | 是 | 否 | 否 |  | 评论时间 |

forum(帖子表)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为空 | 主键 | 自增 | 默认值 | 备注 |
| id | int(11) | 否 | 是 | 是 |  | ID |
| title | varchar(100) | 是 | 否 | 否 |  | 标题 |
| content | varchar(2000) | 是 | 否 | 否 |  | 内容 |
| uid | int(11) | 是 | 否 | 否 |  | 发布人 |
| create\_time | datetime | 是 | 否 | 否 |  | 发布时间 |

goods(商品表)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为空 | 主键 | 自增 | 默认值 | 备注 |
| id | int(11) | 否 | 是 | 是 |  | ID |
| cid | int(11) | 是 | 否 | 否 |  | 分类 |
| image | varchar(100) | 是 | 否 | 否 |  | 商品图片 |
| name | varchar(30) | 是 | 否 | 否 |  | 名称 |
| remark | varchar(30) | 是 | 否 | 否 |  | 简介 |
| money | varchar(30) | 是 | 否 | 否 |  | 价格 |
| num | int(11) | 是 | 否 | 否 |  | 库存 |
| content | varchar(2000) | 是 | 否 | 否 |  | 商品介绍 |
| status | varchar(10) | 是 | 否 | 否 |  | 状态 |
| create\_time | datetime | 是 | 否 | 否 |  | 发布时间 |

order\_item(订单项表)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为空 | 主键 | 自增 | 默认值 | 备注 |
| id | int(11) | 否 | 是 | 是 |  | ID |
| oid | int(11) | 是 | 否 | 否 |  | 订单 |
| gid | int(11) | 是 | 否 | 否 |  | 商品 |
| num | int(11) | 是 | 否 | 否 |  | 数量 |
| money | varchar(20) | 是 | 否 | 否 |  | 金额 |
| create\_time | datetime | 是 | 否 | 否 |  | 创建时间 |

orders(订单表)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为空 | 主键 | 自增 | 默认值 | 备注 |
| id | int(11) | 否 | 是 | 是 |  | ID |
| no | varchar(50) | 是 | 否 | 否 |  | 订单编号 |
| num | int(11) | 是 | 否 | 否 |  | 商品数量 |
| total | varchar(50) | 是 | 否 | 否 |  | 总价 |
| uid | int(11) | 是 | 否 | 否 |  | 用户 |
| remark | varchar(200) | 是 | 否 | 否 |  | 备注 |
| status | varchar(20) | 是 | 否 | 否 |  | 状态 |
| create\_time | datetime | 是 | 否 | 否 |  | 下单时间 |

outfit(穿搭信息表)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为空 | 主键 | 自增 | 默认值 | 备注 |
| id | int(11) | 否 | 是 | 是 |  | ID |
| image | varchar(100) | 是 | 否 | 否 |  | 图片 |
| type | varchar(20) | 是 | 否 | 否 |  | 类型 |
| season | varchar(20) | 是 | 否 | 否 |  | 季节 |
| name | varchar(50) | 是 | 否 | 否 |  | 名称 |
| content | text | 是 | 否 | 否 |  | 内容 |
| uid | int(11) | 是 | 否 | 否 |  | 发布人 |
| num | int(11) | 是 | 否 | 否 |  | 浏览量 |
| status | varchar(10) | 是 | 否 | 否 |  | 状态 |
| create\_time | datetime | 是 | 否 | 否 |  | 发布时间 |

review(商品评论表)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为空 | 主键 | 自增 | 默认值 | 备注 |
| id | int(11) | 否 | 是 | 是 |  | ID |
| gid | int(11) | 是 | 否 | 否 |  | 商品 |
| uid | int(11) | 是 | 否 | 否 |  | 评论用户 |
| content | varchar(500) | 是 | 否 | 否 |  | 评论内容 |
| create\_time | datetime | 是 | 否 | 否 |  | 发布时间 |

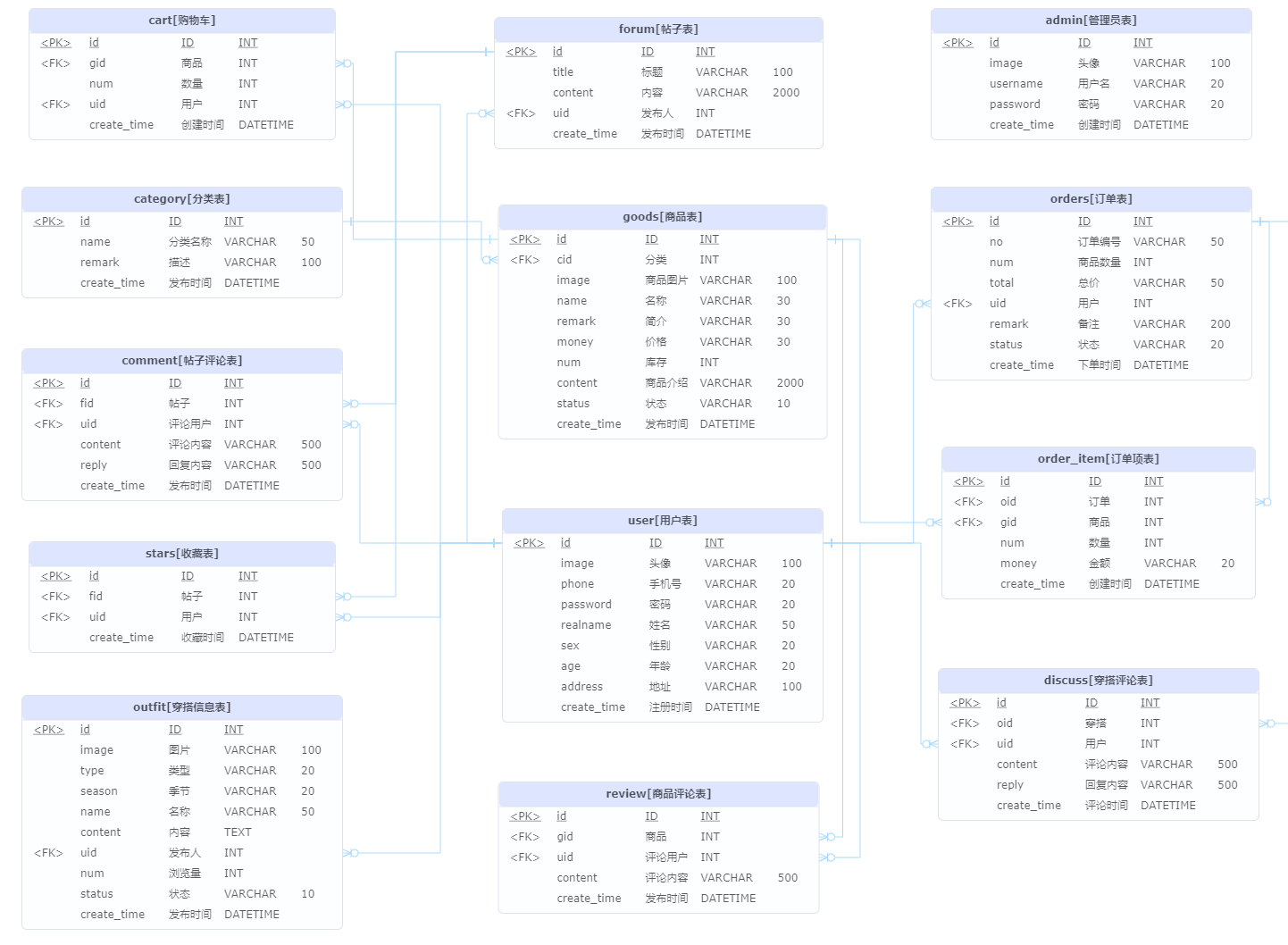
stars(收藏表)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为空 | 主键 | 自增 | 默认值 | 备注 |
| id | int(11) | 否 | 是 | 是 |  | ID |
| fid | int(11) | 是 | 否 | 否 |  | 帖子 |
| uid | int(11) | 是 | 否 | 否 |  | 用户 |
| create\_time | datetime | 是 | 否 | 否 |  | 收藏时间 |

user(用户表)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为空 | 主键 | 自增 | 默认值 | 备注 |
| id | int(11) | 否 | 是 | 是 |  | ID |
| image | varchar(100) | 是 | 否 | 否 |  | 头像 |
| phone | varchar(20) | 是 | 否 | 否 |  | 手机号 |
| password | varchar(20) | 是 | 否 | 否 |  | 密码 |
| realname | varchar(50) | 是 | 否 | 否 |  | 姓名 |
| sex | varchar(20) | 是 | 否 | 否 |  | 性别 |
| age | varchar(20) | 是 | 否 | 否 |  | 年龄 |
| address | varchar(100) | 是 | 否 | 否 |  | 地址 |
| create\_time | datetime | 是 | 否 | 否 |  | 注册时间 |

## 3.3数据库ER关系图



第4章 系统实现

## 4.1项目截图

关于项目介绍和详细功能的演示视频，请直接查看演示视频

## 4.2代码实现

由于代码过多，自行根据需要粘贴即可，核心代码看讲解视频。

# 第5章 系统测试与结果分析

## 5.1系统功能测试方法概述

软件测试的方法和技术是多种多样的。对于疫情防控系统采用的测试方法，使用了软件测试中常用的白盒测试和黑盒测试。

(1)黑盒测试

黑盒测试也称[功能测试](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD%E6%B5%8B%E8%AF%95/10921202)，它是通过测试来检测每个功能是否都能正常使用。在测试中，把[程序](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F/71525)看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在[程序接口](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E6%8E%A5%E5%8F%A3/150383)进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试着眼于[程序](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F/71525)外部结构，不考虑内部[逻辑结构](https://baike.baidu.com/item/%E9%80%BB%E8%BE%91%E7%BB%93%E6%9E%84/9663235)，主要针对[软件界面](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E7%95%8C%E9%9D%A2/792349)和软件功能进行测试。黑盒测试是以用户的角度，从输入数据与输出数据的对应关系出发进行测试的。很明显，如果外部特性本身设计有问题或规格说明的规定有误，用黑盒测试方法是发现不了的。

(2)白盒测试

白盒测试又称结构测试、透明盒测试、逻辑驱动测试或基于[代码](https://baike.baidu.com/item/%E4%BB%A3%E7%A0%81)的测试。白盒测试是一种[测试用例设计](https://baike.baidu.com/item/%E6%B5%8B%E8%AF%95%E7%94%A8%E4%BE%8B%E8%AE%BE%E8%AE%A1/9391582)方法，盒子指的是被测试的[软件](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6/12053)，白盒指的是盒子是可视的，即清楚盒子内部的东西以及里面是如何运作的。“白盒”法全面了解程序内部逻辑结构、对所有逻辑路径进行测试。“白盒”法是穷举路径测试。在使用这一方案时，测试者必须检查程序的内部结构，从检查程序的逻辑着手，得出测试数据。贯穿程序的独立路径数是天文数字。白盒测试的主要方法有逻辑驱动、基路测试等，主要用于软件验证。

## 5.2系统功能测试

本系统根据实际情况选用黑盒测试用例进行测试，以下是测试用例：

（1）用户登录：用户进行登录操作，登录之前需要验证用户名和密码是否正确。登录时，同时输入用户名、密码、如果验证成功，跳转到主页面，在界面右方显示当前登录用户的用户名和角色；如果输入的用户名、密码有一个或多个错误，会提示相应的错误。

用户登录用例测试表如下表6-1所示。

表6-1 用户登录用例测试表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模块 | ID | 测试用例 | 预期结果 | 实际结果 | 评价 |
| 用户登录 | 1 | 用户名：zhangym  密码：123456 | 登录成功 | 登录成功 | 实现 |
| 用户登录 | 2 | 用户名：zhangym  密码12345 | 提示密码错误 | 提示账号错误 | 实现 |

其他测试用例可根据实际情况编写。

## 5.3系统兼容性分析

对于B/S系统的兼容性测试，客户端的兼容性测试主要集中在浏览器上，要针对不同的浏览器及其不同版本进行兼容性测试，例如，需要针对IE7.0和IE.8.0、Firefox3.0和Firefox3.1、Chrome1.0等完成相应的兼容性测试。许多B/S系统，虽然不需要安装客户端软件，但实际上会通过浏览器下载ActiveX控件或Firefox/Chrome插件（Plugin），这就存在控件或插件的版本问题，需要进行相对应的兼容性测试。而对于应用服务器的兼容性测试，需要考虑Web服务器和应用服务器的兼容性测试，有时还需要考虑Web服务器、应用服务器和数据库系统的兼容性测试。本系统在多个不同浏览器及其不同版本都几乎能得以兼容，能正常工作。