目 录

[第1章 绪论 7](#_Toc136757862)

[1.1课题背景及意义 7](#_Toc136757863)

[1.1.1课题背景 7](#_Toc136757864)

[1.1.2课题意义 8](#_Toc136757865)

[1.2选题现状 8](#_Toc136757866)

[1.2.1国外发展现状 8](#_Toc136757867)

[1.2.2国内发展现状 9](#_Toc136757868)

[1.3论文结构及课题内容 9](#_Toc136757869)

[第2章 系统需求分析 11](#_Toc136757870)

[2.1系统用户角色分析 11](#_Toc136757871)

[2.1.1游客 11](#_Toc136757872)

[2.1.2普通用户 11](#_Toc136757873)

[2.1.3超级管理员 11](#_Toc136757874)

[2.1.4客户信息管理员 11](#_Toc136757875)

[2.1.5门户网站管理员 12](#_Toc136757876)

[2.1.6库房信息管理员 12](#_Toc136757877)

[2.2系统功能需求分析 12](#_Toc136757878)

[2.2.1前台门户展示功能 12](#_Toc136757879)

[2.2.2用户中心功能 13](#_Toc136757880)

[2.2.3商品检索功能 13](#_Toc136757881)

[2.2.4订单操作功能 13](#_Toc136757882)

[2.2.5后台管理功能 14](#_Toc136757883)

[2.3系统功能的UML用例图 17](#_Toc136757884)

[2.3.1游客的UML用例图 17](#_Toc136757885)

[2.3.2普通用户的UML用例图 18](#_Toc136757886)

[2.3.3超级管理员的UML用例图 19](#_Toc136757887)

[2.3.4客户信息管理员的UML用例图 19](#_Toc136757888)

[2.3.5门户网站管理员的UML用例图 20](#_Toc136757889)

[2.3.5库房信息管理员的UML用例图 20](#_Toc136757890)

[2.4系统非需求功能分析 21](#_Toc136757891)

[第3章 系统总体设计 22](#_Toc136757892)

[3.1系统软件层次架构设计 22](#_Toc136757893)

[3.2系统功能模块设计 23](#_Toc136757894)

[3.3系统数据库设计 25](#_Toc136757895)

[3.3.1数据库E-R图设计 25](#_Toc136757896)

[3.3.2数据库表设计 25](#_Toc136757897)

[第4章 系统详细设计与实现 29](#_Toc136757898)

[4.1系统工程结构 29](#_Toc136757899)

[4.2系统功能概要设计 30](#_Toc136757900)

[4.2.1注册功能概要设计 30](#_Toc136757901)

[4.2.2登录功能概要设计 31](#_Toc136757902)

[4.2.3商品检索功能概要设计 33](#_Toc136757903)

[4.2.4购物车功能概要设计 34](#_Toc136757904)

[4.2.5订单管理功能概要设计 35](#_Toc136757905)

[4.2.6后台管理功能概要设计 36](#_Toc136757906)

[4.3本章小结 37](#_Toc136757907)

[第5章 系统测试 38](#_Toc136757908)

[5.1测试方案 38](#_Toc136757909)

[5.1.1测试方法 38](#_Toc136757910)

[5.1.2测试环境 38](#_Toc136757911)

[5.2系统功能测试 39](#_Toc136757912)

[5.2.1注册/登录功能测试 39](#_Toc136757913)

[5.2.2购物车功能测试 40](#_Toc136757914)

[5.2.3订单功能测试 41](#_Toc136757915)

[5.2.4商品索引功能测试 41](#_Toc136757916)

[5.2.5后台管理功能测试 41](#_Toc136757917)

[5.3系统压力测试 42](#_Toc136757918)

[5.4本章小结 44](#_Toc136757919)

[第6章 总结与展望 45](#_Toc136757920)

[6.1工作总结 45](#_Toc136757921)

[6.2工作展望 46](#_Toc136757922)

[参考文献 47](#_Toc136757923)

[致 谢 48](#_Toc136757924)

# 绪论

线上自行车销售相对于传统的实体销售更具有成本优势和时空优势，依托于互联网信息传播迅速和数据展示立体的优势，线上自行车销售系统在提高消费者个性化服务体验的同时，也让商家实现了信息化销售。正是基于这些特点，许多自行车厂商纷纷退出自己的购物平台。但随着自行车市场的不断发展，现阶段愈发需要一款能够打通品牌的自行车销售平台，以方便消费者的购物和个体商户的业务拓展。本章主要分为四个部分。1.1节介绍了课题的当前背景及发展意义；1.2节介绍了课题的发展现状；1.3节介绍了课题的主要内容；1.4节介绍了论文的主要组织结构。

## 1.1课题背景及意义

## 1.1.1课题背景

随着互联网和移动技术的迅猛发展，电子商务成为了现代商业活动的重要组成部分。越来越多的消费者倾向于在线上进行购物，这推动了各行业的电子商务发展，包括自行车销售。消费者购物行为在互联网时代发生了巨大变化。越来越多的消费者习惯于在线上搜索产品、比较价格和读取评价，然后进行在线购买。这种变化促使自行车销售商家将目光投向线上销售系统，以满足消费者的需求并提供更好的购物体验。

同时，自行车作为一种运动和交通工具，其市场规模逐年扩大。据2023年中国自行车协会新闻发布会公布的数据来看，2022年，自行车电动自行车规模以上企业营业收入2100亿元，同比增长3% ；实现利润超100亿元，同比增长20%以上；自行车规模以上企业产量同比下降32%，电动自行车规模以上企业产量同比增长 9%[1]，人们越来越重视健康、环保和便捷的出行方式，自行车成为了受欢迎的选择。这种市场需求的增长促使自行车销售商家探索更有效的销售渠道和方式，其中线上销售系统是一种重要的解决方案。随着电子商务技术的不断进步，包括网站开发、移动应用、支付系统、数据分析等方面，自行车销售商家能够建立更强大和便捷的线上销售系统。这些技术的进步为研究和实施自行车销售系统提供了更多的机会和资源。

近年来自行车市场不断增长，竞争也变得更加激烈。传统的线下商店面临着来自在线销售的竞争压力，因此，自行车销售商家需要寻找新的方式来吸引和保留消费者。自行车销售系统的研究可以帮助他们更好地理解市场需求、消费者行为和竞争环境，从而制定有效的销售策略。自行车销售系统通过数据分析和市场调研，可以了解消费者的购买行为、偏好和趋势。研究背景中也包括对数据驱动的决策的研究，以确定如何利用数据来优化销售策略、个性化推荐、定价策略等，提高销售效果和市场竞争力。

## 1.1.2课题意义

本课题使用SSM轻量级开源框架，搭建前后端分离的自行车销售系统。SSM框架是一种Java Web应用开发框架，它由Spring、Spring MVC和MyBatis三个开源框架组合提供了一种简单且高效的方式来实现高效开发。

SSM框架提供了一套成熟的开发模式和工具，使自行车销售系统的开发和维护更加高效。Spring的依赖注入和面向切面编程简化了开发过程，Spring MVC提供了基于MVC的Web开发框架，MyBatis提供了灵活的数据库操作。这些框架的整合使开发者能够专注于业务逻辑的实现，减少了开发周期和工作量。SSM框架在性能和稳定性方面表现出色。Spring的IOC容器管理对象的生命周期和依赖关系，提供了可靠的应用程序结构。MyBatis通过优化的SQL映射和缓存机制提供高效的数据库访问性能。这些特性使得自行车销售系统具有良好的性能和稳定性，能够处理大量的用户请求和数据操作。

SSM框架具有高度的灵活性和可扩展性，可以根据自行车销售系统的需求进行定制和扩展。开发者可以根据业务需求选择使用框架的不同组件，也可以与其他第三方库进行整合。此外，框架本身也提供了丰富的扩展机制，允许开发者根据实际需求进行功能拓展和定制化开发。

基于SSM框架的自行车销售系统能够提供优化的用户体验。通过Spring MVC的Web开发框架，可以实现友好的用户界面和交互体验。MyBatis的数据持久化操作能够高效地获取和存储用户数据，保证系统的响应速度。这些优化措施使得用户能够更流畅地浏览产品、提交订单和进行支付，提升了用户的满意度和购物体验。

自行车销售系统基于SSM框架可以收集和分析用户的购买行为和偏好，获取有价值的市场数据。这些数据可以用于市场调研、用户画像分析和销售预测，为企业提供决策支持和战略指导。通过数据分析，企业可以优化产品推荐、促销活动和库存管理，提高市场竞争力。

## 1.2选题现状

## 1.2.1国外发展现状

国外自行车销售网站注重技术创新，采用各种先进的技术和框架，如Node.js、React、Vue.js等。这些技术能够提供更好的用户体验、更高的性能和更快的网站加载速度。

一些国外自行车销售网站着眼于跨境销售，通过多语言支持和国际物流合作，将产品推向全球市场。他们注重本地化的用户体验，提供多种货币结算和国际运输选项，以吸引国际消费者。

国外自行车销售网站与社交媒体平台进行整合，利用社交媒体的影响力扩大品牌曝光和产品推广。他们通过在社交媒体上发布产品信息、用户评价和活动推广，吸引更多用户访问和购买。

## 1.2.2国内发展现状

国内自行车销售网站主要采用Java技术栈，如SSM框架（Spring、Spring MVC、MyBatis）。这些框架提供了稳定可靠的开发环境和丰富的功能组件，支持高并发、高可用的网站运行。

随着移动互联网的快速发展，国内自行车销售网站注重移动端应用的开发。大部分网站推出了移动App，提供用户便捷的购物体验和个性化推荐功能。同时，响应式网页设计也得到广泛应用，确保在不同终端设备上的良好显示效果。

为了扩大销售渠道和覆盖更广泛的用户群体，国内自行车销售网站与第三方电商平台进行合作，通过开设店铺或入驻平台，将产品销售延伸到多个电商平台上。这样做可以利用电商平台的流量和用户基础，提高产品曝光度和销售量。

国内自行车销售网站与支付平台和物流公司合作，提供多种支付方式和灵活的配送服务。支持支付宝、微信支付等主流支付方式，同时与快递公司合作，提供便捷的配送和售后服务，以提升用户体验。

**结论：**在线上销售平台上，我国存在非常成熟的技术与实例。但当前国内自行车销售平台仍然基于各自品牌独立发展，分散用户及市场，不利于自行车市场的发展与扩张，也为消费者带来很差的消费体验。一个能够整合各品牌的自行车销售网站显得当务之急。

## 1.3论文结构及课题内容

本文共六章，主要内容介绍如下：

第一章 绪论。主要论述了本课题当今的发展背景及意义、课题国内外发展现状和论文结构与课题内容。

第二章 系统的需求分析和技术实现。分别对系统的功能需求和非功能需求进行分析，并论证系统开发的技术可行性。

第三章 系统的概要设计。主要描述了自行车销售系统的软件架构设计、系统的功能模块设计、系统的数据库设计和系统的界面设计。

第四章 系统的详细设计与实现。详细描述了部分子系统主要功能模块的设计与实现，并对主要技术问题进行描述，最后进行了代码展示。 

第五章 系统测试。介绍主要的软件测试方法，对系统各模块进行功能测试和非功能测试，并分析测试结果。

第六章 总结与展望。主要对本文内容做概述性总结，并对本次开发的不足之处进行分析，进行目标展望。

# 系统需求分析

项目需求分析是指识别、收集和分析与项目相关的需求，以便确保项目正常有序的推进。它是项目管理中的一个关键过程，在整个项目周期中起着至关重要的作用。本章的主要内容是对自行车销售系统的功能需求进行全面分析，清晰分析不同角色使用该系统的需求，并用UML用例图进行图示说明。最后对非功能需求也做了总结分析。

## 2.1系统用户角色分析

依据实际需求，自行车销售系统的用户角色可分为四种：游客、普通用户、超级管理员和普通管理员，为了保证后台管理的安全性与便捷性，其中普通管理员主要有客户信息管理员、门户网站管理员、库房信息管理员，一共6个角色。

2.1.1游客

游客即进入网站且未进行会员注册的用户，游客作为网站最基本的用户群体，其拥有对网站基本的信息查阅权限，可以浏览网页主界面的轮播图、商品展示区域、商品详情信息，可以使用关键词对商品进行检索查询，但是无法添加商品至购物车、无法购买商品、无法在评论区进行互动等操作。

2.1.2普通用户

普通用户即本网站的注册用户，普通用户作为本网站的主要目标用户群体，其拥有游客用户角色的所有权限，可以浏览、查阅网站信息。除此之外，普通用户可以进行注册及登陆操作、管理个人信息、将商品添加至购物车、对商品进行购买结算、在商品详细页面发送评论等操作。

2.1.3超级管理员

超级管理员即拥有本网站最高权限的管理员，可通过系统后台管理模块，对前台网页内容进行增删更新、还可增删普通管理员的注册信息、管理网站普通用户信息，维护网站正常运行。

2.1.4客户信息管理员

客户信息管理员是本系统普通管理员的一种，可实现增删客户信息、查看客户信息列表、查询具体客户信息，实时更新用户数据，保证用户数据安全有效。

2.1.5门户网站管理员

门户网站管理员是本系统普通管理员的一种，可实现对前台网页商品信息的数据更新、管理货架信息、管理商品分类展示、管理商品详情描述内容、管理商品评价信息及管理网站公告信息，保证网页的正常运行。

2.1.6库房信息管理员

库房信息管理员是本系统普通管理员的一种，可实现对仓库商品数据的更新及查询、对订单信息的更新及查询、对出库信息的更新及查询，保证仓库数据与前台数据相匹配。

## 2.2系统功能需求分析

通过对自行车销售系统的各个子系统进行功能需求分析，大致可分为：前台门户展示、用户中心、商品检索、订单操作、后台管理五大功能。每个功能模块仍需根据进一步要求做详细划分，下面将对这几个功能模块进行功能需求分析，本系统功能结构图如图2.1所示。

2.2.1前台门户展示功能

前台门户展示模块为用户提供了一个便利的购物环境和良好的用户体验，用户可以通过门户网站进行首页浏览、商品分类浏览、商品详细信息浏览、网站公告浏览等，满足了用户对于商品信息获取的基本需求。

1. 首页浏览展示

首页浏览展示要指网站主页的商品概览，主要包括网站导航栏、顶部轮播图展示这两个部分。导航栏模块设置了首页、网站公告、商品分类展示及用户中心的入口；商品概览模块展示了管理员设置的商品内容，为用户提供大致商品信息。

1. 商品分类展示

商品分类展示模块根据商品特点，为用户提供特色商品浏览界面。主要包括推荐商品界面、特价商品界面、全部商品界面；其中在全部商品界面，根据自行车的产品分类，我们也划分了山地车、公路车、全地形公路车、旅行车等不同种类展示；在主界面，根据自行车品牌也区分了不同的模块展示。

1. 商品详情信息展示

商品详情界面为用户展示了具体商品的详细信息。主要包括了商品名称、高清图片、参数描述、价格、上架日期、点击数、销量等数据；同时设置了添加购物车选项、商品评论功能，让用户的购买体验更加舒适。

1. 网站公告展示

商品公告界面为用户展示了本网站的全部文章公告。主要包括了网站促销活动、新款商品数据、商品售后新闻等社区模块，为用户全方位打造一个信息密集化的信息展示界面。

2.2.2用户中心功能

用户可以通过主页右上角的交互按钮进入会员中心，进行注册登录后，即可查看个人信息、管理个人信息、管理个人订单、管理个人收藏内容。

1. 注册

游客可以通过注册功能创建一个新的账户。他们需要提供基本的个人信息，例如姓名、电子邮件地址和密码。一旦成功注册，用户将能够访问网站的其他功能。

1. 登录

已注册的用户可以使用他们的邮箱和密码登录到系统。这允许他们访问他们的个人信息、订单和收藏夹。

1. 管理个人信息

用户可以在个人资料页面上查看和编辑他们的个人信息，例如联系方式、交付地址和支付方式。他们还可以更改他们的用户名和密码。

1. 管理个人订单

用户可以查看他们的所有订单，并跟踪订单的状态。他们可以查看已完成订单的详细信息，例如订单号、产品名称、数量、价格和发货日期。用户还可以取消未完成的订单或重新下单。

1. 管理个人收藏

用户可以在收藏夹中保存他们喜欢的产品。他们可以添加、删除或编辑收藏夹中的商品，并随时查看收藏夹中的产品详情。收藏夹也可以作为购物清单，用户可以从中直接把商品加入购物车并下单购买。

2.2.3商品检索功能

商品检索功能即用户通过主页的搜索栏，实现输入关键词（如品牌、型号、颜色等）来查找自己想要的商品；同时在网站页面的左下角，系统可以通过智能算法根据用户历史购买记录和偏好推荐相关的商品。综合利用以上这两种方式，自行车线上销售系统可以实现一个全面有效的商品检索功能，让用户轻松快速地找到自己需要的商品。

2.2.4订单操作功能

用户通过订单操作功能可以在商品详情界面将其添加至购物车，并在购物车进行结算、提交订单，在订单管理界面可以实现个人订单的一系列操作，最后进行付款、评论。

1. 购物车功能

用户可以通过单击“加入购物车”按钮将商品添加到购物车。该按钮通常位于每个商品的页面上，并显示商品数量和总价。同时，用户可以随时查看购物车中已添加的商品以及它们的数量和总价。购物车通常会在顶部导航栏或侧边栏中显示。用户可以在购物车页面上更改商品数量、删除商品或清空整个购物车。

1. 订单提交功能

用户需要填写收货人姓名、地址、联系电话等信息，以便商家能够准确地配送商品。用户可以选择在线支付或货到付款等支付方式。最后，用户确认收货信息和支付方式后，可以点击“提交订单”按钮完成订单的提交。

1. 订单取消功能

用户可以在订单详细页面中选择要取消的订单，并填写取消原因和退款方式。商家会在订单管理系统中审核订单取消请求，并决定是否同意取消订单。如果商家同意取消订单，退款将直接返回用户支付的账户。

1. 订单支付功能

用户可以在订单页面中选择在线支付或货到付款等支付方式。如果用户选择在线支付，订单页面将跳转至第三方支付网站。用户需要输入收款账户信息并完成支付。支付成功后，用户将获得支付成功的提示。如果支付失败，用户可以重新尝试支付或选择其他支付方式。

1. 订单修改功能

用户可以在订单详细页面中选择要修改的订单，并填写修改原因和修改内容。商家会在订单管理系统中审核订单修改请求，并决定是否同意修改订单。如果商家同意修改订单，订单状态将更新为“已修改”状态，并且相应的商品、价格等信息也将被更新。

2.2.5后台管理功能

后台管理功能主要是管理员在后台用于维护前台网站正常运行的模块，在自行车线上销售系统中，后台管理功能是至关重要的一部分。

1. 管理普通管理员信息

自行车线上销售系统往往需要由多个管理员共同管理，因此后台管理系统需要具有添加、编辑和删除管理员账号等基本功能。管理员可以设定不同的权限级别，以确保各项管理工作的安全性和合理性。

1. 新增管理员信息

系统管理员可以在后台管理系统中添加新管理员账号。该功能需要包括管理员账号信息（如登录账号、密码、姓名、联系方式等），并需要设置对应的权限级别。

1. 管理管理员权限

为了保证后台管理系统的安全性和合理性，系统管理员需要能够针对不同的管理员账号设定不同的权限级别。比如，一些高敏感度的操作需要授权给具有更高权限等级的管理员才能执行，而一些较为简单的操作则可以授权给普通管理员进行执行。

1. 查询普通管理员

后台管理系统还需要提供一个查询普通管理员的功能，以便系统管理员随时查看当前的管理员账号情况。通过该功能，系统管理员可以查看管理员账号列表、编辑管理员信息或者删除管理员账号。

1. 管理客户信息

自行车线上销售系统的客户群体非常广泛，从初学者到专业车手都有可能成为消费者。后台管理系统需要能够对客户进行分类和标记，以便更好地进行针对性的推销和服务。同时，管理员还需要能够查看和编辑客户的基本信息，如姓名、地址、联系方式、订单记录等，从而更好地了解客户需求和行为，优化营销策略。

1. 客户信息管理

自行车线上销售系统需要收集客户的基本信息，如姓名、地址、联系方式等，并将这些信息存储在后台管理系统中。管理员可以通过客户信息管理功能对客户信息进行分类和标记，以便更好地进行针对性的推销和服务。此外，管理员还需要能够查看和编辑客户的订单记录，从而更好地了解客户需求和行为，优化营销策略。

1. 供应商信息管理

除了客户信息外，后台管理系统还需要管理供应商信息。自行车线上销售系统需要与多家供应商合作，获取自行车相关的商品。因此，管理员需要将不同供应商的信息存储在后台管理系统中，并能够进行供应商信息的分类、标记和查询。

1. 客户信息查询

后台管理系统需要提供一个客户信息查询功能，以便管理员随时查看当前的客户情况。通过该功能，系统管理员可以查看客户信息列表、编辑客户信息或者删除客户信息。此外，还可以按照客户类型（例如初学者、专业车手）或者购买历史记录等标准，对客户进行分类和排序，以便更好地进行营销策略制定和销售目标设定。

1. 管理前台信息

自行车线上销售系统的前台页面是吸引客户的重要因素之一，因此后台管理系统需要能够对前台页面进行管理和维护。管理员可以选择上传或删除商品图片，设置商品价格和库存数量，并通过编辑文本框来更新产品描述和说明。此外，管理员还需要能够管理前台页面的各种内容，如广告横幅、页面布局等。

1. 前台展示管理

前台展示管理涵盖了整个自行车线上销售系统的前端页面设计和展示效果。后台管理系统需要提供一些工具和功能，以便管理员可以直接修改、添加或删除前端页面元素，包括主题、布局、颜色、字体和图片等。通过前台展示管理，管理员可以定期更新前端页面内容，以更好地吸引客户。

1. 商品分区管理

为了更好地展示商品，自行车线上销售系统需要将所有商品进行分类，并在前端显示不同的商品分区。后台管理系统需要提供相应工具和功能，使管理员能够轻松的管理这些商品分区、进行分类、编辑或者删除。通过商品分区管理，管理员可以灵活配置自行车线上销售系统的商品分类，以满足不同客户群体对商品的需求。

1. 商品内容管理

除了商品分区外，每个商品还需要提供详细的商品信息，以便客户更好地了解商品属性、特点和优劣势。后台管理系统需要提供相应的工具和功能，以便管理员能够编写、编辑或删除商品详情页面的内容，包括商品描述、规格、价格、图片、库存等。通过商品内容管理，管理员可以更好地管理自行车线上销售系统中的所有商品信息，以满足不同客户群体对商品的需求。

1. 管理库房信息

自行车线上销售系统的商品都需要存储在库房中，因此后台管理系统需要能够管理库房信息。管理员可以查看库房的状态，包括库存数量、入库时间、出库时间等，从而及时地调整商品的库存量。同时，管理员还可以进行库存盘点、调拨、报废等操作，以确保库房管理的准确性和及时性。

1. 入库信息管理

入库信息管理是指管理自行车线上销售系统的库房中所有入库商品的相关信息。该模块需要提供相应的功能和工具，以便管理员能够轻松地管理入库信息，包括入库日期、入库数量、入库单价、供应商信息等。管理员还可以根据需求查询、编辑或删除入库信息。

1. 订单信息管理

订单信息管理是指管理自行车线上销售系统内所有客户订单的相关信息。该模块需要提供相应的功能和工具，以便管理员能够轻松地管理订单信息，包括订单编号、订单日期、订单数量、订单状态、客户信息、收货地址等。管理员还可以根据需求查询、编辑或删除订单信息。

1. 出库信息管理

出库信息管理是指管理自行车线上销售系统库房中所有出库商品的相关信息。该模块需要提供相应的功能和工具，以便管理员能够轻松地管理出库信息，包括出库日期、出库数量、出库单价、客户信息等。管理员还可以根据需求查询、编辑或删除出库信息。

## 2.3系统功能的UML用例图

本章节通过UML用例图，直观的理解自行车线上销售系统的用户角色分析和系统功能分析。用例图是一种直观的图形表示方法，能够清晰地表达系统与用户之间的交互过程，易于理解和使用。

### 2.3.1游客的UML用例图

游客即首次进入系统网站并未注册、登录的用户。该角色可在前台门户子系统进行首页浏览、商品分类查看；可在搜索子系统进行商品检索；可在用户中心子系统进行会员注册，并成为普通用户。游客的UML用例图如图2.1所示。

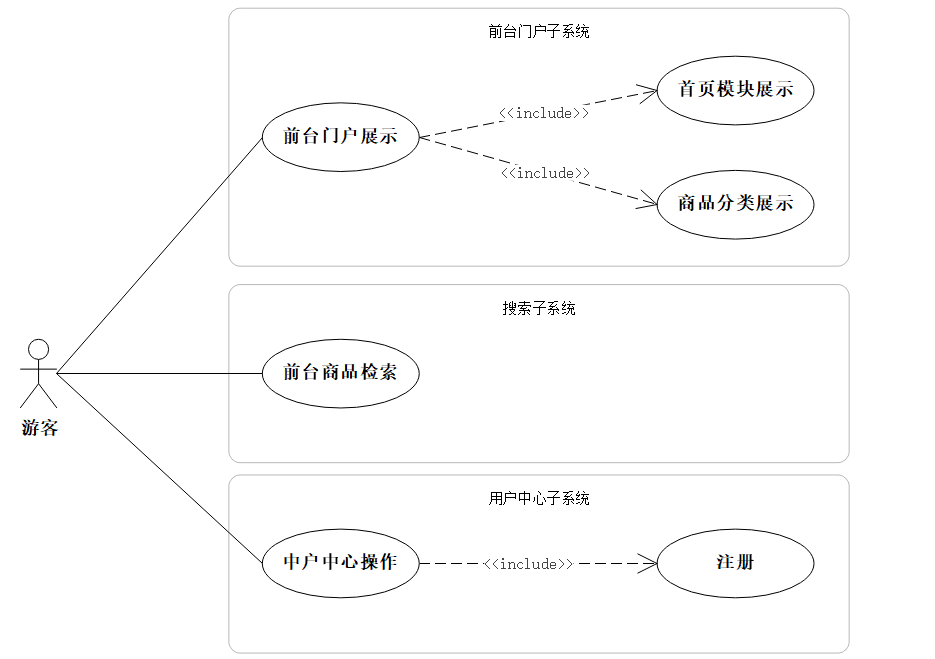


图2..1 游客用例图

### 2.3.2普通用户的UML用例图

普通用户通过用户中心子系统进行登录后，即可对个人信息进行修改、管理；可以在前台门户子系统进行商品浏览和商品分类查看；也可在搜索子系统进行商品检索；并且能在订单管理子系统进行订单管理。普通用户UML用例图如图2.2所示。

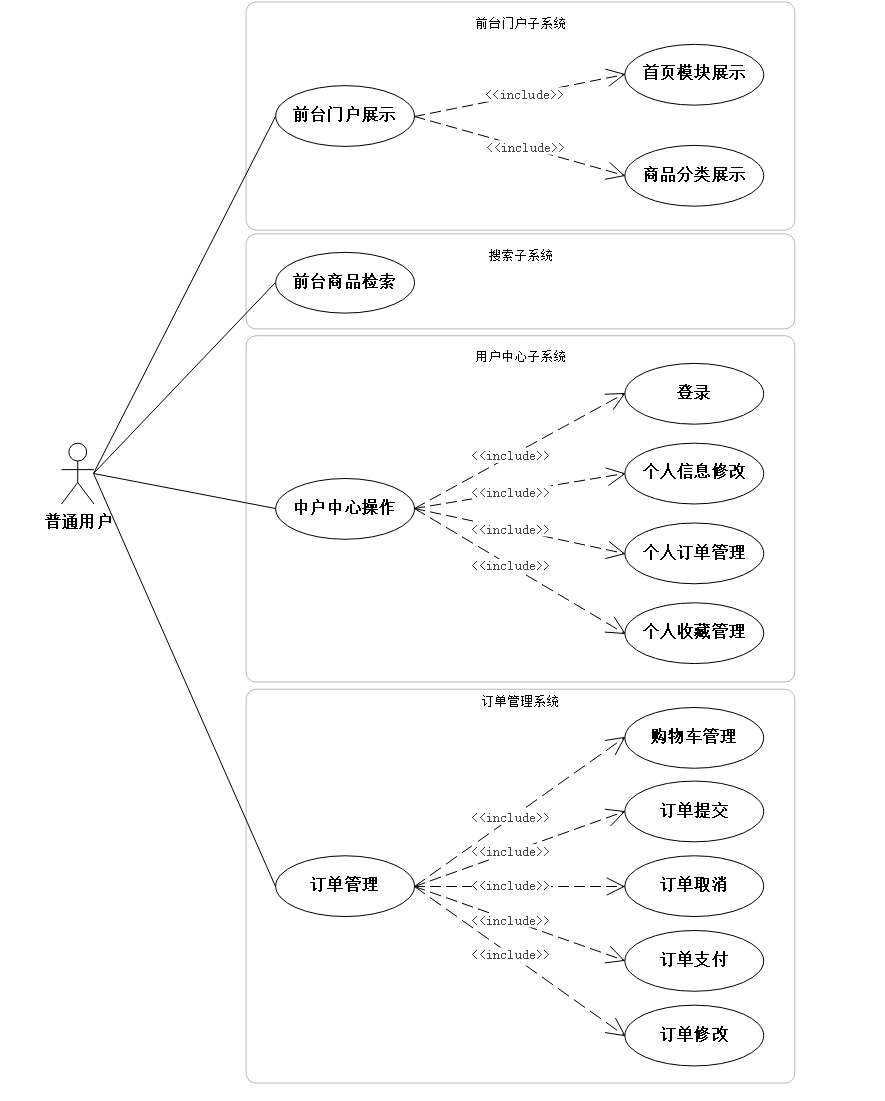


图2..2 普通用户用例图

### 2.3.3超级管理员的UML用例图

超级管理员主要在后台对普通管理员进行权限管理，在后台管理子系统中拥有最高权限。可以增删普通管理员条目，并对具体管理员的操作权限名称即相应字段进行操作管理。超级管理员的UML用例图如图2.3所示。

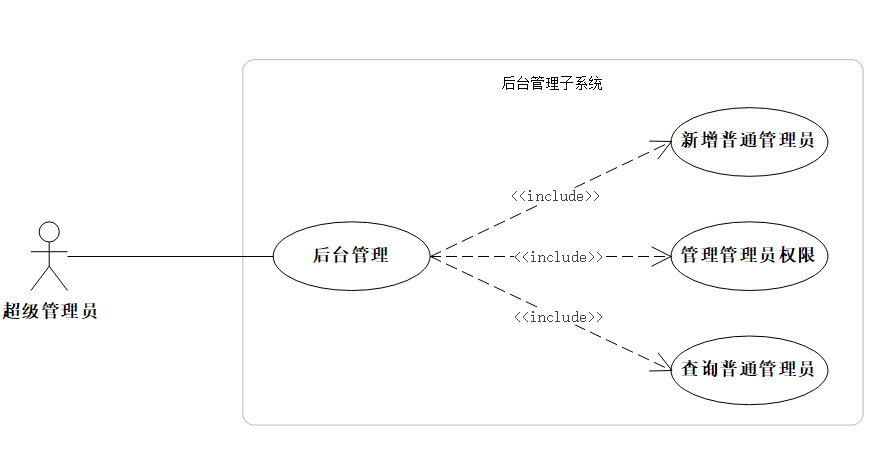


图2.3超级管理员用例图

### 2.3.4客户信息管理员的UML用例图

客户信息管理员拥有后台管理子系统中客户信息模块的管理权限。可以对自行车线上销售系统收集的客户、供应商基本信息进行管理，同时也能够对该信息进行查询检索，以便后台分类标记。客户信息管理员的UML用例图如图2.4所示。



图2.4客户信息管理员用例图

### 2.3.5门户网站管理员的UML用例图

门户网站管理员拥有后台管理子系统的前台门户模块的管理权限。可以使用该模块提供的一些工具和功能进行接修改、添加或删除前端页面元素，更新前台展示内容。同时也需要对新产品进行分类、编辑描述信息、前台展示等操作。通过商品内容管理，管理员可以更好地管理自行车线上销售系统中的所有商品信息，以满足不同客户群体对商品的需求。门户网站管理员的UML用例图如图2.5所示。

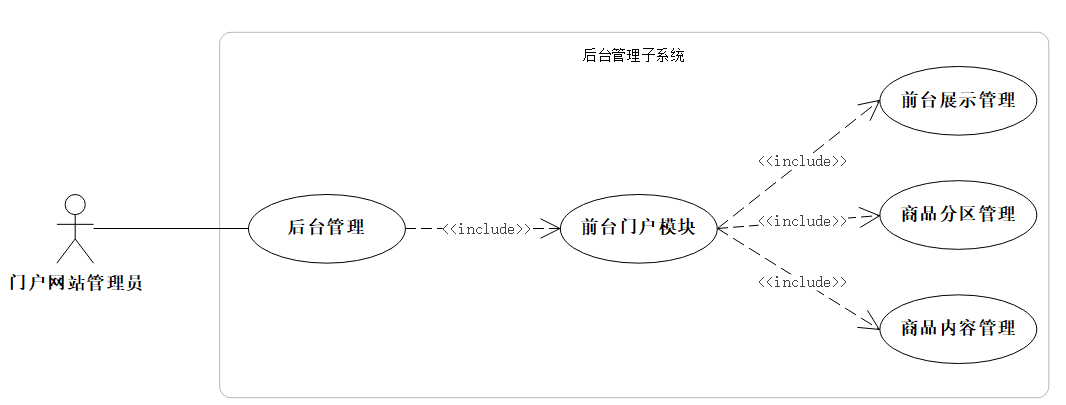


图2.5门户网站管理员用例图

### 2.3.5库房信息管理员的UML用例图

库房信息管理员拥有后台管理子系统的库房信息模块的管理权限。该管理员需随时更新后台出、入库产品，并根据需求查询、编辑或删除出、入库信息；并及时同步前台订单信息与仓库信息，跟踪订单状态，管理订单信息。该管理员能够更好地建立一个高效、安全的库房管理系统，从而提升自行车线上销售系统的整体运作效率和客户满意度。库房信息管理员的UML用例图如图2.6所示。

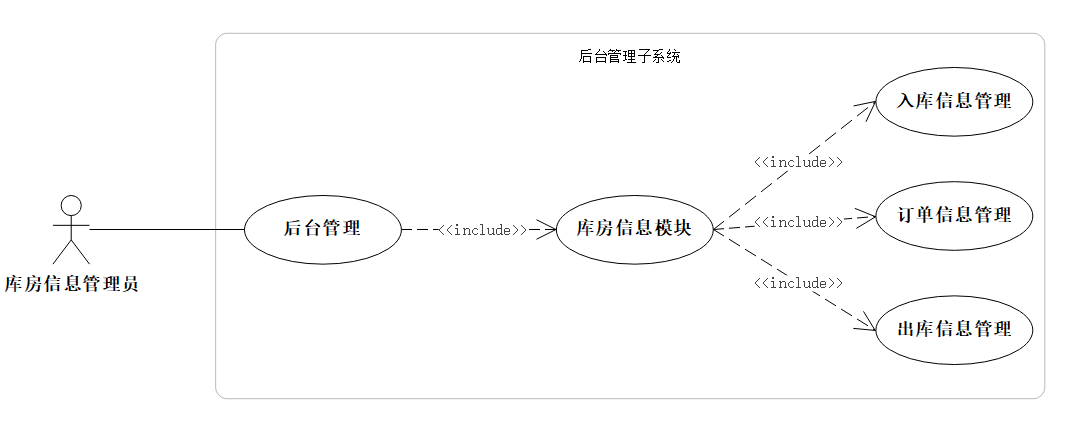


图2.6库房信息管理员用例图

## 2.4系统非需求功能分析

本小结主要列述了构建高质量系统所需要考虑的不同方面的需求，包括可靠性和稳定性、安全性、性能、可扩展性、易用性和可维护性。其中，可靠性和稳定性、安全性、性能、可扩展性、易用性和可维护性被视为构建高质量系统的关键要素。同时，数据一致性是确保数据质量的基础，事务控制技术和备份与恢复策略等都是保证数据可靠性的重要手段。

**系统的可靠性和稳定性需求：**系统应具有高可靠性和稳定性，能够保证服务可用性。系统应该考虑到服务器故障、网络中断等可能出现的问题，并且能够及时发现并处理这些问题，确保系统始终稳定运行。

**系统的安全性需求：**系统应具有高度的安全性，能够防止未经授权的用户对系统进行访问或者恶意攻击。系统需要采取一系列安全措施，如密码加密、数据备份、防火墙等，以确保用户信息的安全性和机密性。

**系统的性能需求：**系统应该具有良好的性能，包括响应速度、吞吐量、并发量等。系统需要优化数据库的设计，提高查询效率，使用缓存技术等提升系统的性能。

**系统的可扩展性需求：**系统应该具有良好的可扩展性，可以根据业务需要灵活地增加或减少系统的功能模块。系统设计应该考虑到未来的扩展性，并且能够支持快速的部署和集成新的功能。

**系统的易用性需求：**系统应该具有良好的用户界面设计，方便用户快速上手并使用系统。系统应该提供友好的操作界面和帮助文档，以降低用户的学习成本和使用难度。

**系统的可维护性需求：**系统应该具有良好的可维护性，能够方便地进行升级和维护。系统需要采用模块化的设计思路，将各个模块分开，方便单独维护和修改。同时，系统应该具备完善的日志记录和错误处理机制，方便管理员快速定位问题并解决。

**系统的可靠性需求：**系统应该保证数据的一致性，避免数据重复或者丢失。系统应该采用事务控制技术、备份与恢复策略等手段来保证数据的可靠性。

# 第3章 系统总体设计

总体设计是指在软件开发过程中，针对项目的整体需求和目标，对系统进行整体规划和设计的过程。项目总体设计帮助确定系统的整体架构，包括系统的层次结构、模块划分和组件的选择。通过对系统的结构进行规划和设计，可以确保系统的可扩展性、可维护性与可重用性，减少后期的重构和改动成本。本章分为三大部分，分别是系统软件层次架构设计、系统功能模块设计以及系统数据库结构设计。

## 3.1系统软件层次架构设计

自行车线上销售系统的软件架构设计需要综合考虑性能、安全性、可扩展性、可靠性和用户体验等方面，以提供稳定、安全、高效和用户友好的购物体验。本系统软件架构采用了满足上述条件的SSM框架设计模式和面向服务的分布式集群架构。

SSM是一种将Spring、Spring MVC和Mybatis三个框架结合起来的Java Web应用开发框架。该框架具有高效性、松耦合、灵活性、易集成、成熟稳定等优点，松耦合的设计使得系统模块可以独立开发、测试和维护，提高了代码的可维护性和可测试性。本系统基于SSM框架设计，将系统软件逻辑架构分为展示层、业务层、数据层。

**展示层：**展示层是与用户进行交互的界面层，负责呈现数据给用户，并接收用户的输入。它通常包括用户界面、用户体验设计和用户交互逻辑。此层使用Spring MVC框架，在Spring MVC模式中，视图（View）位于展示层，负责呈现数据呈现给用户并接收用户输入，它代表应用程序用户界面的可视化表示，并与数据的表示和格式有关。View组件不包含任何应用程序逻辑，不应直接操作数据。

**业务层：**业务层是应用程序的核心层，包含了应用程序的业务逻辑和规则。它负责处理数据的验证、处理和转换，以及应用程序的各种业务流程和规则的实现。此层使用Spring框架，Spring是一个轻量级的IoC（Inversion of Control）和AOP（Aspect-Oriented Programming）容器。IoC容器负责管理和组织应用程序中的对象，通过依赖注入（Dependency Injection）将对象之间的依赖关系解耦。AOP提供了一种将横切关注点（如日志、事务管理等）与业务逻辑分离的方式。

**数据层：**数据层是与数据存储和访问相关的层次，负责数据的读取、写入和管理。它涉及数据库或其他数据存储系统的访问和操作。此层使用Mybatis框架，MyBatis是一种Java持久化框架，提供了对关系型数据库的数据访问和操作它通过使用SQL映射和对象关系映射（ORM）的方式，将Java对象与数据库表之间进行映射。MyBatis可以通过XML配置文件或注解的方式编写和管理SQL语句，并提供了对数据库的增删改查操作的支持。

这三个层次之间的关系通常是分层和解耦的，以实现应用程序的可维护性、可扩展性和可重用性。展示层负责与用户进行交互，业务层负责处理核心业务逻辑，数据层负责数据的存储和访问。它们之间的交互通常通过接口或服务进行，以保持层与层之间的松耦合。系统总体软件层架构图如图3.1所示。

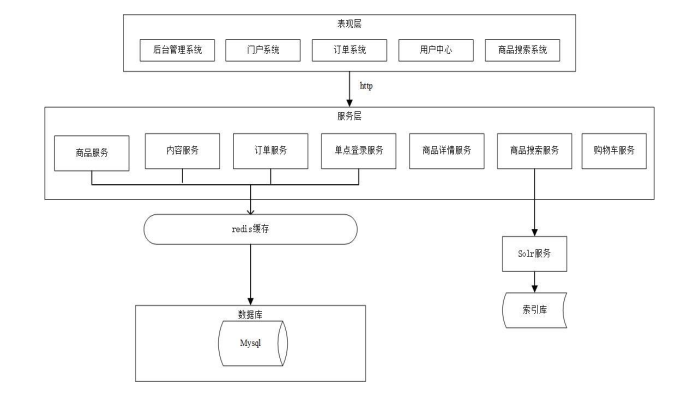


图3..1系统软件架构图

## 3.2系统功能模块设计

通过对系统的实际功能需求分析，以及在技术层面的系统软件架构设计的基础上，本节对系统实现做进一步细化设计。整个系统分为网站前台和网站后台整体划分，前台包括门户展示模块、用户中心模块、商品检索模块、订单操作模块；后台包括超级管理模块、客户信息模块、前台门户模块、库房信息模块；一共划分八个设计模块，具体模块功能如下：

**门户展示模块：**展示自行车产品的首页和各类别产品列表，包括产品图片、价格、描述等信息，以吸引用户并提供浏览和搜索产品的功能。

**用户中心模块：**用户注册、登录、个人信息管理等功能，允许用户创建账户、查看和编辑个人信息，以及管理订单和收货地址。

**商品检索模块：**提供各种条件和筛选选项，以帮助用户快速检索和筛选符合其需求的自行车产品。

**订单操作模块：**包括用户下单、购物车管理、订单支付、订单状态跟踪等功能，确保用户能够方便地完成购买流程并了解订单状态。超级管理模块：管理员登录、权限管理、系统设置等功能，用于管理整个系统的各个模块和功能，控制访问权限以及对系统进行配置和管理。

**客户信息模块：**管理用户注册的客户信息，包括个人信息、订单记录等，允许管理员进行查看、编辑和删除操作。

**前台门户模块：**管理前台网站的展示内容，包括首页轮播图、推荐商品、广告位等，允许管理员对展示内容进行编辑和调整。

**库房信息模块：**管理库存和商品信息，包括商品分类、库存管理、价格管理等，允许管理员进行商品的上架、下架和库存调整等操作。

这些功能模块的设计和实现，可以满足自行车线上销售系统的整体需求，从前台用户体验到后台管理功能的支持，提供了一个完整的在线销售平台。通过模块图能够更加直观的分析系统模块设计，系统功能模块设计图如图4.2所示。

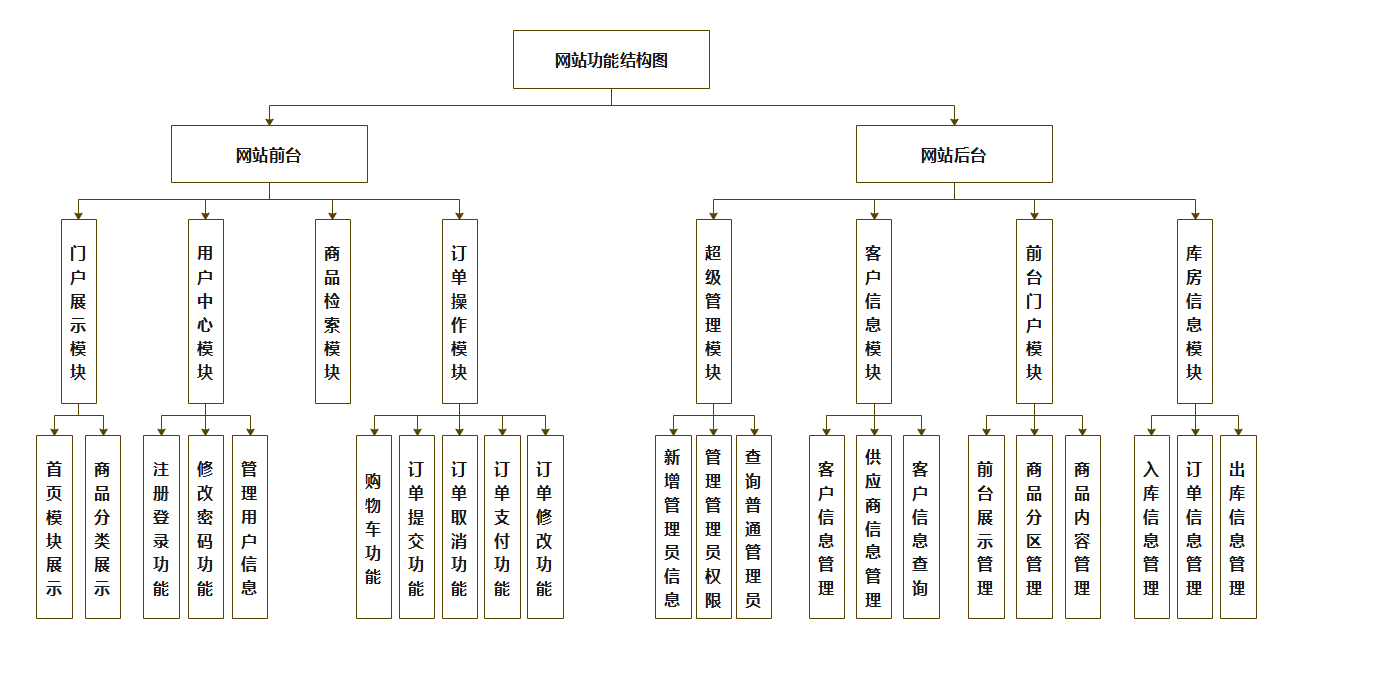


图3..2系统功能模块图

## 3.3系统数据库设计

系统数据库设计是指在构建一个系统时，设计和规划系统所需的数据库结构、表、字段以及其关系的过程。它涉及到定义数据库的逻辑结构、数据模型、数据存储方式、表结构、索引、关系和约束等方面的设计决策。系统数据库设计是根据系统需求，设计和规划系统所需的数据库结构和关系的过程。它是系统开发中重要的一部分，直接影响系统的功能实现和数据管理效果。

### 3.3.1数据库E-R图设计

E-R图（Entity-Relationship Diagram）是一种用于数据库设计的图形化表示方法，用于可视化和描述实体（Entity）、关系（Relationship）和属性（Attribute）之间的关系。它提供了一种直观的方式来展示数据库中实体之间的联系和属性的特征。本系统当中表种类繁多，本节仅列举重要的七张表之间的关系，如下图3.3所示，自行车销售系统的实体总联系图，直观展示了表与表之间的逻辑关系。

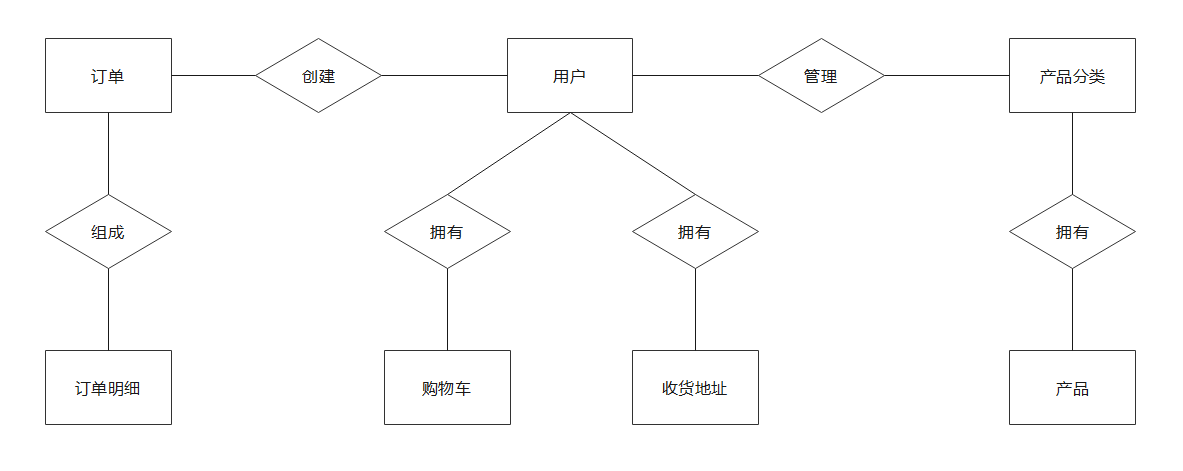


图3..3数据库E-R图

### 3.3.2数据库表设计

本系统采用了MySQL数据库，并使用数据库管理工具Navicat for MySQL进行数据表的具体实现。通过对以上数据库E-R图的设计和分析，本节将对以下七个表进行具体分析说明。

1. 表3.1为ORDER（订单表）：order\_id为订单号，是本表的唯一索引，保证订单号的非重复性；

表3.1订单表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 默认值 | 备注 |
| id | int | 否 | 无 | 订单id |
| order\_id | long | 否 | 无 | 订单号 |
| user\_id | int | 否 | 无 | 用户id |
| shipping\_id | int | 否 | 无 | 地址id |
| payment | decimal | 否 | 无 | 付款金额（元） |
| status | int | 否 | 10 | 订单状态：0-已取消10-未付款20-已付款30-已发货40-交易成功50交易关闭 |
| payment\_time | datetime | 否 | 无 | 支付时间 |
| send\_time | datetime | 否 | 无 | 发货时间 |
| end\_time | datetime | 否 | 无 | 交易完成时间 |
| close\_time | datetime | 否 | 无 | 交易关闭时间 |
| create\_time | datetime | 是 | 无 | 创建时间 |
| update\_time | datetime | 是 | 无 | 更新时间 |

1. 表3.2为ORDER\_iTEM（订单明细表）。

表3.2订单明细表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 默认值 | 备注 |
| id | int | 否 | 无 | 订单子表id |
| user\_id | int | 否 | 无 | 用户id |
| order\_id | long | 否 | 无 | 订单id |
| product\_id | int | 否 | 无 | 商品id |
| product\_name | varchar | 否 | 无 | 商品名称 |
| product\_img | varchar | 否 | 无 | 商品图片地址 |
| order\_price | decimal | 否 | 无 | 订单总价 |
| quantity | int | 否 | 无 | 商品数量 |
| create\_time | datetime | 是 | 无 | 创建时间 |
| update\_time | datetime | 是 | 无 | 更新时间 |

1. 表3.3为USER（用户表）。

表3.3用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 默认值 | 备注 |
| id | int | 否 | 无 | 用户表id |
| user\_name | varchar | 否 | 无 | 用户名 |
| password | varchar | 否 | 无 | 用户密码，MD5加密 |
| email | varchar | 否 | 无 | 邮箱 |
| phone | varchar | 否 | 无 | 电话 |
| role | varchar | 否 | 无 | 角色：0-管理员，1-普通用户 |
| create\_time | datetime | 是 | 无 | 创建时间 |
| update\_time | datetime | 是 | 无 | 最后一次更新时间 |

1. 表3.4为CATEGORY（分类表）

表3.4分类表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 默认值 | 备注 |
| id | int | 否 | 无 | 类别id |
| name | varchar | 否 | 无 | 类别名称 |
| status | int | 否 | 1 | 类别状态：1-正常2-已注销 |
| sort\_order | int | 否 | 无 | 排序编号 |
| create\_time | datetime | 是 | 无 | 创建时间 |
| update\_time | datetime | 是 | 无 | 更新时间 |

1. 表3.5为PRODICT（产品表）

表3.5产品表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 默认值 | 备注 |
| id | int | 否 | 无 | 商品id |
| category\_id | int | 是 | 0 | 分类id，外键-CATEGORY表主键 |
| name | varchar | 否 | 无 | 商品名称 |
| product\_img | varchar | 否 | 无 | 商品图片url地址 |
| detail | varchar | 是 | 无 | 商品详情 |
| price | decimal | 否 | 无 | 商品价格（元） |
| stock | int | 否 | 无 | 库存数量 |
| status | int | 否 | 1 | 商品状态：1-在售2-下架3-注销 |
| create\_time | datetime | 是 | 无 | 创建时间 |
| update\_time | datetime | 是 | 无 | 更新时间 |

1. 表3.6为CART（购物车表）

表3.6购物车表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 默认值 | 备注 |
| id | int | 否 | 无 | 购物车表id，自增主键 |
| user\_id | int | 否 | 无 | 用户id |
| product\_id | int | 否 | 无 | 商品id |
| quantity | int | 否 | 无 | 商品数量 |
| checked | int | 否 | 1 | 是否选择：1-已勾选0-未勾选 |
| create\_time | datetime | 是 | 无 | 创建时间 |
| update\_time | datetime | 是 | 无 | 更新时间 |

1. 表3.7为SHIPPING（收货地址表）

表3.7收货地址表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 默认值 | 备注 |
| id | int | 否 | 无 | 收货表id |
| user\_id | int | 否 | 无 | 用户id |
| receiver\_name | varchar | 否 | 无 | 收货姓名 |
| receiver\_phone | varchar | 否 | 无 | 收货电话 |
| receiver\_province | varchar | 否 | 无 | 收货身份 |
| receiver\_city | varchar | 否 | 无 | 收货城市 |
| receiver\_district | varchar | 否 | 无 | 收货区/县 |
| receiver\_address | varchar | 否 | 无 | 详细地址 |
| receiver\_zip | varchar | 否 | 无 | 邮编 |
| create\_time | datetime | 是 | 无 | 创建时间 |
| update\_time | datetime | 是 | 无 | 更新时间 |

# 第4章 系统详细设计与实现

系统详细设计与实现是将系统设计的概念和抽象转化为具体的代码实现的过程，它具体描述了系统的架构、模块、组件、函数、接口等的详细设计，并将设计转化为可执行的代码实现。系统详细设计与实现的主要目标是根据系统设计的概念和要求，将系统划分为更小的模块和组件，并定义它们之间的接口和交互方式，以实现系统的功能和需求。本章将从系统工程结构、系统主要功能设计两个角度进行系统具体模块设计与实现。

## 4.1系统工程结构

系统工程结构设计是指在软件开发过程中，将整个系统按照一定的结构和组织方式进行设计和构建的过程。它涉及到对系统的各个组成部分进行划分、组织和连接，以满足系统需求、可维护性、可扩展性和可重用性等方面的要求。

本项目使用了Maven项目搭建工程，Maven是一个用于构建和管理Java项目的开源工具。它提供了一个标准化的项目结构、构建流程和依赖管理系统，可以简化项目的构建和维护过程。

Maven基于项目对象模型（Project Object Model，POM）来管理项目。POM是一个XML文件，用于描述项目的结构、依赖关系、构建配置等信息。通过定义POM文件，Maven可以自动下载项目所需的依赖库，执行项目的编译、测试、打包和部署等任务。其主要功能包括：

1. 依赖管理：Maven可以自动下载和管理项目的依赖库，简化了依赖管理的过程。它使用中央仓库和其他远程仓库来存储和获取依赖库。
2. 构建工具：Maven提供了一套命令和插件，用于执行项目的编译、测试、打包和部署等构建任务。通过定义POM文件中的构建配置，可以自定义构建过程。
3. 标准化项目结构：Maven推荐一种标准的项目结构，使得不同项目的结构更加统一和易于理解。这有助于开发人员更好地组织和管理项目。
4. 生命周期和插件机制：Maven定义了一系列的构建生命周期和阶段，每个阶段可以关联相应的插件执行特定的任务。这种机制可以使构建过程更加灵活和可扩展。

综上所述，本系统工程结构图如图4.1所示。

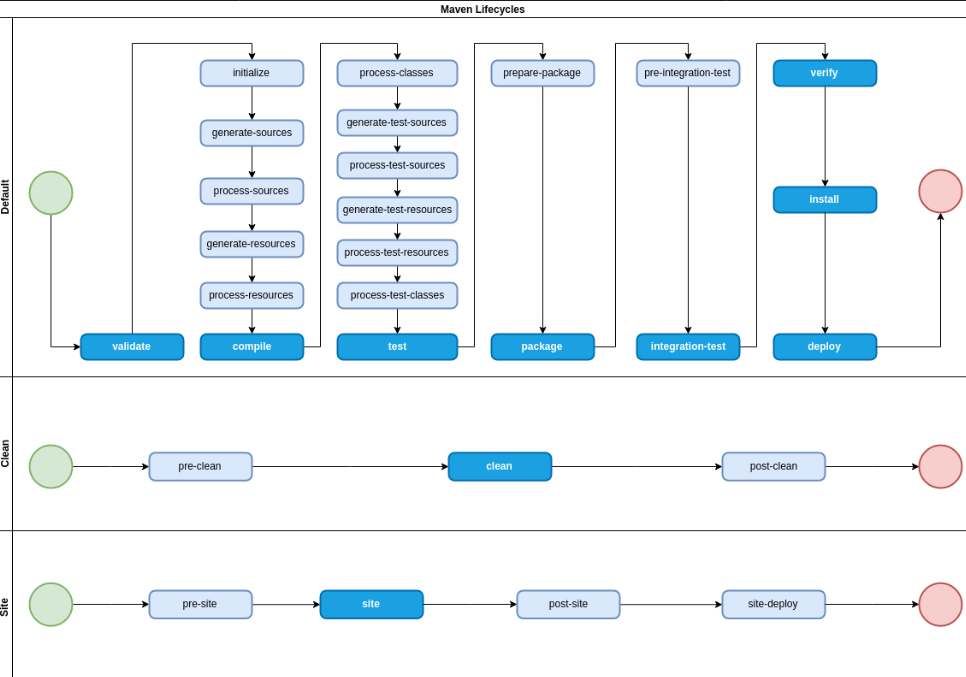


图 4.1maven工程结构图

## 4.2系统功能概要设计

### 4.2.1注册功能概要设计

注册功能需实现以下8点操作，具体注册时序图如图4.2所示：

1. 接收请求：接收后端服务器接收来自前端的注册请求；
2. 参数验证：对接收到的注册请求参数进行验证，包括用户名、密码、电子邮件等。验证可以使用表单验证库或自定义验证逻辑，确保用户输入的数据符合要求；
3. 数据处理：将验证通过的用户注册信息进行处理。可以创建一个用户模型或实体类，将用户输入的数据存储在模型的属性中；
4. 密码加密：为了保护用户的密码安全，通常需要对密码进行加密处理。可以使用加密算法（如哈希算法）对用户输入的密码进行加密，并将加密后的密码存储在数据库中；
5. 数据库操作：将用户注册信息存储到数据库中。使用数据库操作语言（如SQL）执行插入操作，将用户信息插入到用户表中。确保数据库连接正常，并处理数据库操作可能出现的异常；
6. 注册结果返回：根据注册操作的结果，返回相应的响应给前端。可以定义一组状态码或响应信息，表示注册成功或失败的情况。在注册成功的情况下，可以返回用户注册成功的提示信息或用户对象的部分信息；
7. 异常处理：需要考虑注册过程中可能出现的异常情况，例如数据库连接失败、注册信息重复等。对于这些异常，可以捕获并返回相应的错误信息给前端，帮助用户解决问题或重新尝试注册；
8. 日志记录：为了追踪注册操作和方便日后的故障排查，可以在注册功能中添加日志记录，记录关键操作、错误信息等重要信息。

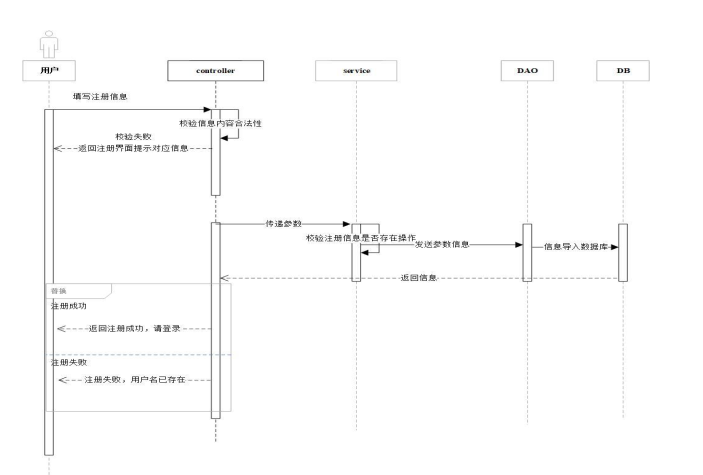


图4..2注册时序图

### 4.2.2登录功能概要设计

登录功能需实现以下8点操作，具体注册时序图如图4.3所示：

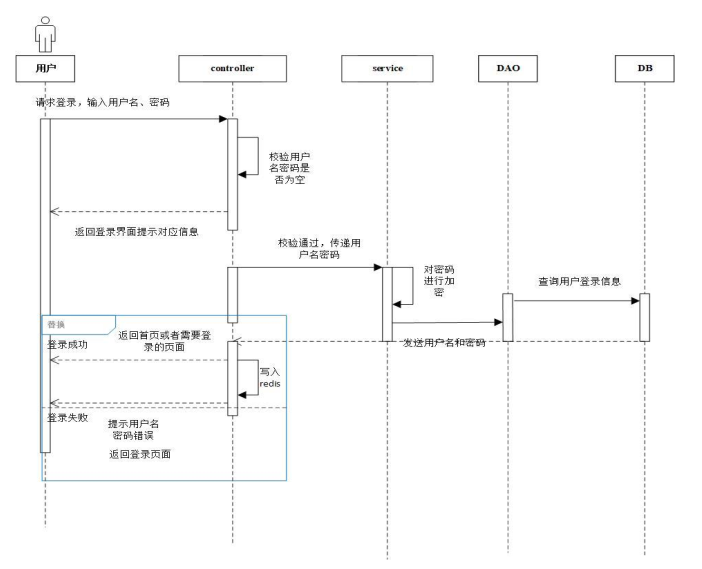


图4..3登录时序图

1. 接收请求：后端服务器接收来自前端的登录请求。可以使用框架或库提供的路由功能，将请求映射到相应的处理方法。
2. 参数验证：对接收到的登录请求参数进行验证，包括用户名、密码等。验证可以使用表单验证库或自定义验证逻辑，确保用户输入的数据符合要求。
3. 用户验证：使用用户名和密码进行用户验证。可以查询数据库中的用户表，检查用户名和密码是否匹配。如果使用密码加密算法进行了密码加密，在验证过程中需要对用户输入的密码进行相同的加密处理。
4. 认证与授权：在验证成功后，生成一个令牌（token）作为用户的身份凭证，并将令牌返回给前端。令牌可以使用JSON Web Token（JWT）或其他认证机制生成，用于后续的用户身份验证和授权。
5. 会话管理：可以使用会话管理技术，如使用会话标识符（session ID）将登录状态与用户关联起来，以便后续的请求可以进行身份验证和授权。
6. 返回响应：根据登录验证的结果，返回相应的响应给前端。可以定义一组状态码或响应信息，表示登录成功或失败的情况。在登录成功的情况下，可以返回用户登录成功的提示信息或用户对象的部分信息。
7. 异常处理：考虑登录过程中可能出现的异常情况，例如数据库连接失败、用户名或密码错误等。对于这些异常，可以捕获并返回相应的错误信息给前端，帮助用户解决问题或重新尝试登录。
8. 日志记录：为了追踪登录操作和方便日后的故障排查，可以在登录功能中添加日志记录，记录关键操作、错误信息等重要信息。

### 4.2.3商品检索功能概要设计

商品检索功能需实现以下8点操作，具体注册时序图如图4.4所示：

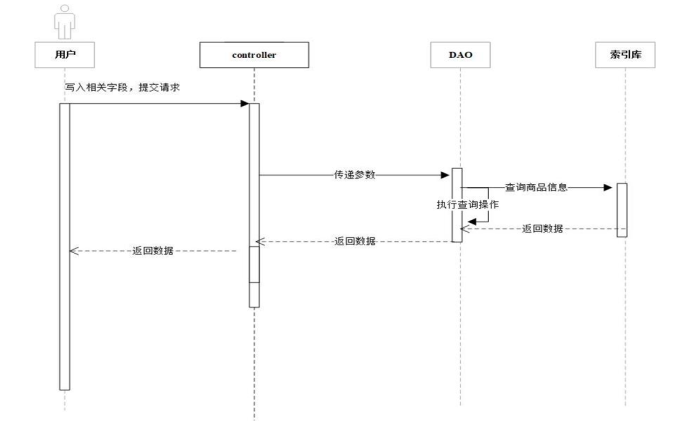


图4..4商品检索时序图

1. 接收请求：后端服务器接收来自前端的商品检索请求。可以使用框架或库提供的路由功能，将请求映射到相应的处理方法。
2. 参数解析：解析检索请求中的参数，包括关键字、筛选条件等。可以通过请求的URL参数、查询字符串或请求体中的数据来获取参数。
3. 查询数据库：根据检索参数在数据库中执行相应的查询操作。可以使用数据库查询语言（如SQL）编写查询语句，根据关键字和筛选条件检索匹配的商品信息。
4. 数据处理：对查询结果进行必要的数据处理，如过滤、排序、分页等。可以根据需求使用后端框架或库提供的功能，对数据进行处理和加工。
5. 返回响应：将处理后的商品检索结果返回给前端。可以以JSON格式返回结果，包括商品的名称、价格、图片等相关信息。根据具体需求，可以返回完整的商品列表或进行分页处理。
6. 异常处理：考虑商品检索过程中可能出现的异常情况，如数据库连接失败、查询错误等。对于这些异常，可以捕获并返回相应的错误信息给前端，帮助用户解决问题或重新尝试商品检索。
7. 日志记录：为了追踪商品检索操作和方便日后的故障排查，可以在商品检索功能中添加日志记录，记录关键操作、错误信息等重要信息。

### 4.2.4购物车功能概要设计

购物车功能需实现以下10点操作，具体注册时序图如图4.5所示：

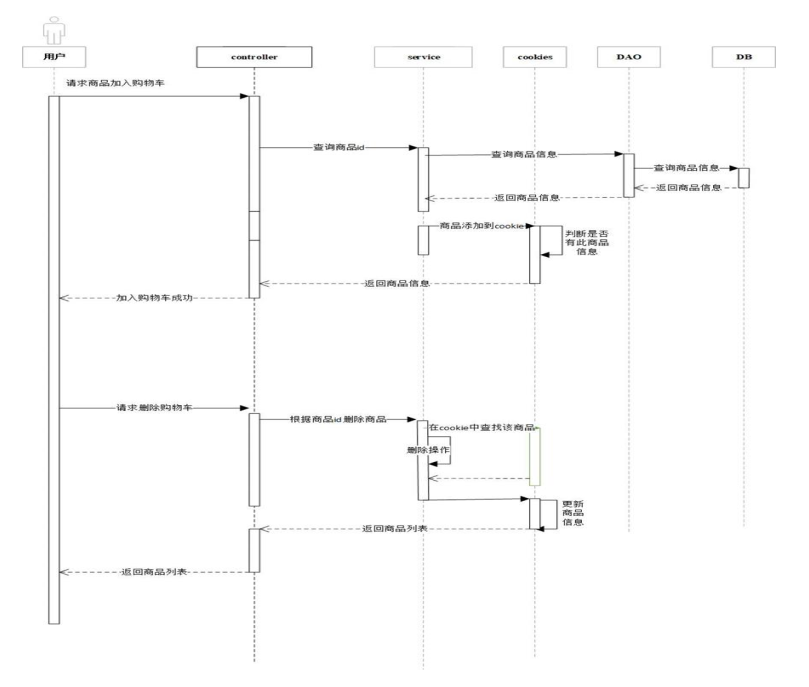


图4..5购物车功能时序图

1. 用户登录：确保用户已经登录，以便购物车功能可以与用户关联。可以使用会话管理技术或令牌验证来验证用户的身份。
2. 添加商品：接收来自前端的添加商品请求，包括商品ID和数量等参数。将商品添加到用户的购物车中。购物车可以使用数据库表或缓存来存储用户的购物车信息。
3. 修改商品数量：接收来自前端的修改商品数量请求，包括商品ID和新的数量等参数。更新购物车中指定商品的数量。
4. 删除商品：接收来自前端的删除商品请求，包括商品ID等参数。从购物车中移除指定的商品。
5. 查看购物车：接收来自前端的查看购物车请求。根据用户的登录状态，查询用户的购物车信息并返回给前端。购物车信息可以包括商品列表、数量、价格等相关信息。
6. 清空购物车：接收来自前端的清空购物车请求。删除用户购物车中的所有商品。
7. 计算总价：根据购物车中的商品数量和价格等信息，计算购物车中商品的总价。可以在后端进行计算，并将结果返回给前端。
8. 结算功能：接收来自前端的结算请求。根据用户购物车中的商品信息生成订单，并进行相应的处理，如库存减少、生成订单号等。
9. 异常处理：考虑购物车功能过程中可能出现的异常情况，如数据库连接失败、商品不存在等。对于这些异常，可以捕获并返回相应的错误信息给前端，帮助用户解决问题或重新尝试购物车操作。
10. 日志记录：为了追踪购物车操作和方便日后的故障排查，可以在购物车功能中添加日志记录，记录关键操作、错误信息等重要信息。

### 4.2.5订单管理功能概要设计

订单管理功能需实现以下8点操作，具体注册时序图如图4.6所示：

1. 接收请求：后端服务器接收来自前端的订单管理请求。可以使用框架或库提供的路由功能，将请求映射到相应的处理方法。
2. 订单查询：接收来自前端的订单查询请求，可以根据不同的查询条件（如订单状态、时间范围、用户ID等）查询数据库中符合条件的订单信息。
3. 订单详情：接收来自前端的订单详情请求，根据订单ID查询数据库中对应的订单详细信息，包括订单号、商品信息、数量、价格等。
4. 订单状态更新：接收来自前端的订单状态更新请求，根据订单ID更新数据库中对应订单的状态，如确认订单、发货、完成等。
5. 订单统计：根据不同的统计维度（如时间范围、商品类别等），查询数据库中的订单数据并进行统计分析，如销售额、订单数量等。
6. 订单删除：接收来自前端的订单删除请求，根据订单ID从数据库中删除对应的订单记录。
7. 异常处理：考虑订单管理过程中可能出现的异常情况，如数据库连接失败、订单不存在等。对于这些异常，可以捕获并返回相应的错误信息给前端，帮助用户解决问题或重新尝试订单管理操作。
8. 日志记录：为了追踪订单管理操作和方便日后的故障排查，可以在订单管理功能中添加日志记录，记录关键操作、错误信息等重要信息。

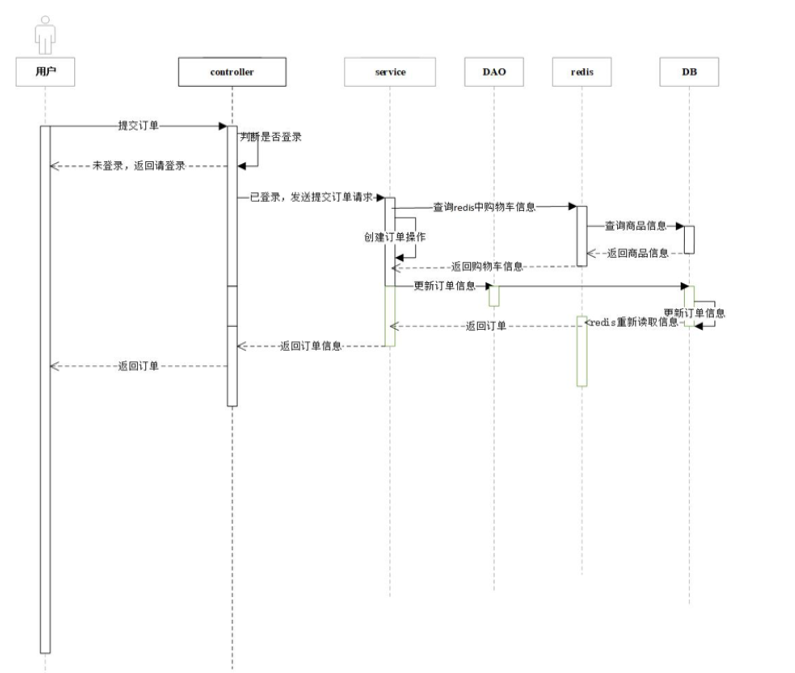


图4..6订单管理时序图

### 4.2.6后台管理功能概要设计

后台管理功能需实现以下6点操作，具体注册时序图如图4.7所示：

1. 用户认证和权限管理：实现用户登录认证功能，验证用户的身份，并根据用户的角色和权限限制其对后台管理功能的访问。可以使用身份验证技术（如会话管理或令牌验证）和权限管理框架来实现。
2. 数据管理：实现对后台管理功能所涉及的数据进行增删改查操作。包括对用户数据、商品数据、订单数据等进行管理，可以使用数据库操作语言（如SQL）编写相应的数据库操作语句。
3. 页面展示和交互：根据后台管理功能的需求，设计和实现相应的后台管理页面，包括数据列表展示、编辑表单、搜索功能等。可以使用前端框架和模板引擎来构建动态页面，并通过前后端交互实现数据的展示和操作。
4. 数据验证和处理：对后台管理功能涉及的数据进行验证和处理，确保数据的合法性和一致性。可以使用后端框架提供的验证机制或自定义验证逻辑，对输入的数据进行验证，并进行相应的处理，如数据格式化、数据清洗等。
5. 异常处理：考虑后台管理功能过程中可能出现的异常情况，如数据库连接失败、权限验证失败等。对于这些异常，可以捕获并返回相应的错误信息给前端，帮助用户解决问题或重新尝试后台管理操作。
6. 日志记录：为了追踪后台管理操作和方便日后的故障排查，可以在后台管理功能中添加日志记录，记录关键操作、错误信息等重要信息。

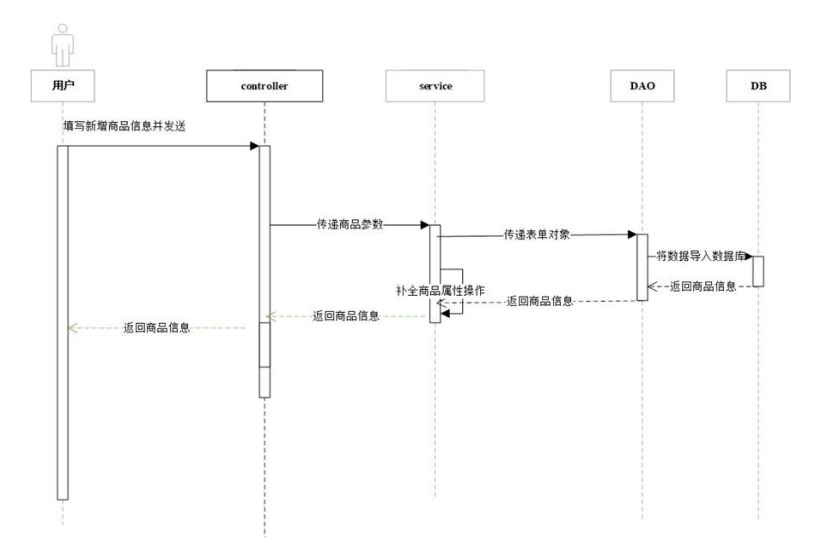


图4..7后台管理功能时序图

## 4.3本章小结

本章系统详细设计与实现对系统的功能模块进行了概要设计，通过定义功能模块的输入、输出、流程逻辑等方面的设计，确保了系统的功能实现和交互流程的合理性。同时，通过系统工程结构的设计，建立了清晰的系统组织架构，使得系统具备良好的可维护性、可扩展性和可重用性。这些设计为后续的系统具体实现提供了指导和框架。

# 第5章 系统测试

系统测试用于验证系统是否符合预期的功能、性能和质量要求。通过测试用例的执行，可以确保系统按照预期进行操作，功能符合用户需求。通过性能测试，可以发现系统的瓶颈和性能问题，并进行优化和调整。本章针对自行车销售系统进行测试和分析，主要通过黑盒测试和界面展示图来对系统功能进行测试，以及使用测试工具Jmester对系统进行接口压力测试。

## 5.1测试方案

系统测试方案是指在进行系统测试时制定的计划和策略，用于确保系统的功能、性能和质量达到预期要求。

### 5.1.1测试方法

针对系统功能测试部分，采用黑盒测试的方法，通过执行各种测试用例，验证系统的功能是否按照预期工作。功能测试可以对需求规格进行验证，确保需求规格中定义的功能被正确实现。通过与需求规格的对比，可以发现是否存在遗漏的功能或与需求不符的功能，从而保证软件系统与用户需求的一致性。

针对系统接口压力测试部分，采用测试工具进行模拟访问网页请求，通过模拟大量用户同时访问系统、大量数据处理、复杂业务逻辑等场景，可以测试系统的响应时间、吞吐量、并发用户数等性能指标，确定系统的性能水平。

### 5.1.2测试环境

本次测试环境如下表5.1所示。

表 5.1 测试环境信息表

|  |  |
| --- | --- |
| **软硬件名称** | **版本** |
| 操作系统 | Windows 10 |
| CPU | Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz |
| 数据库 | MySQL 5.5.36 |
| Java | Java 1.8 |
| 测试工具 | apache-jmeter-5.5 |

## 5.2系统功能测试

### 5.2.1注册/登录功能测试

注册和登录功能是系统中的核心功能之一，进行功能测试是确保其正确性和可靠性的关键步骤。该模块主要测试游客对前台界面的浏览是否正常、注册信息有误时的健壮性测试，以及对普通用户登录时的信息有误时是否有及时反馈。

1. 游客访问主页测试

如图5.1所示，系统前台界面对游客正常开放权限。

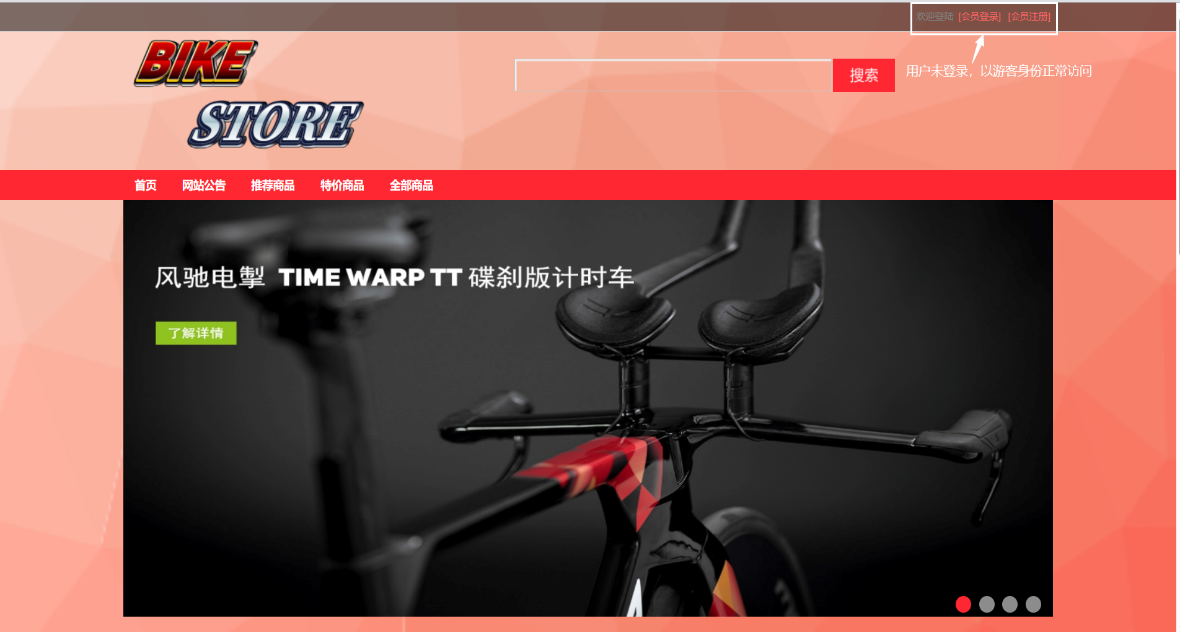


图5..1前台主页面

1. 游客注册测试

分别采用三种用例测试不同情况下的注册功能是否有效。注册测试用例如下表5-2所示。

表5.2 注册测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **测试用例** | **预期结果** | **实际结果** | **结论** |
| 1 | 填写不完整注册信息 | 提示信息补全 | 提示信息补全 | 正常 |
| 2 | 填写已注册用户信息 | 提示用户已注册 | 提示用户已注册 | 正常 |
| 3 | 填写未注册用户信息 | 跳转至登录界面 | 跳转至登录界面 | 正常 |

用例1测试图如下图5.2所示。



图5..2注册信息缺少图

1. 用户登录测试用例如表5.3所示。

表5.3 登录测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **测试用例** | **预期结果** | **实际结果** | **结论** |
| 1 | 填写不完整登录信息 | 提示信息补全 | 提示信息补全 | 正常 |
| 2 | 填写错误用户信息 | 提示信息错误 | 提示信息错误 | 正常 |
| 3 | 填写正确用户信息 | 跳转至主界面 | 跳转主界面 | 正常 |

### 5.2.2购物车功能测试

购物车测试用例如表5.4所示。

表5.4 购物车功能测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **测试用例** | **预期结果** | **实际结果** | **结论** |
| 1 | 添加商品至购物车 | 提示信息补全 | 提示信息补全 | 正常 |
| 2 | 去购物车结算 | 提示信息错误 | 提示信息错误 | 正常 |
| 3 | 继续购物 | 跳转至商品详情界面 | 跳转至商品详情界面 | 正常 |
| 4 | 删除购物车商品 | 商品被删除 | 商品被删除 | 正常 |
| 5 | 增减购物车商品数量 | 商品数量变化 | 商品数量变化 | 正常 |

### 5.2.3订单功能测试

订单测试用例如表5.5所示。

表5.5 订单功能测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **测试用例** | **预期结果** | **实际结果** | **结论** |
| 1 | 未登录去结算 | 强制跳转至登录界面，随后转到订单确认界面 | 强制跳转至登录界面，随后转到订单确认界面 | 正常 |
| 2 | 提交订单 | 进入付款界面 | 进入付款界面 | 正常 |

### 5.2.4商品索引功能测试

商品索引测试用例如表5.6所示。

表5.6 商品索引功能测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **测试用例** | **预期结果** | **实际结果** | **结论** |
| 1 | 商品搜索 | 返回包含搜索的相关字段的商品 | 返回包含搜索的相关字段的商品 | 正常 |

### 5.2.5后台管理功能测试

后台管理测试用例如表5.7所示。

表5.7 后台管理功能测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **测试用例** | **预期结果** | **实际结果** | **结论** |
| 1 | 新增商品 | 显示新增商品成功 | 显示新增商品成功 | 正常 |
| 2 | 查询商品 | 分页展示商品列表 | 分页展示商品列表 | 正常 |
| 3 | 修改商品信息 | 显示修改信息成功 | 显示修改信息成功 | 正常 |
| 4 | 修改商品类别 | 显示修改商品成功 | 显示修改商品成功 | 正常 |
| 5 | 内容管理 | 跳转至内容管理界面 | 跳转至内容管理界面 | 正常 |

## 5.3系统压力测试

系统压力测试可以帮助评估系统在高负载情况下的性能表现。通过模拟大量用户同时访问系统、大量数据处理、复杂业务逻辑等场景，可以测试系统的响应时间、吞吐量、并发用户数等性能指标，确定系统的性能水平。系统压力测试对于评估系统性能、发现瓶颈、验证稳定性、优化性能和提供决策依据具有重要的意义。它可以帮助保证系统在高负载情况下的稳定性和可靠性，提升用户体验，并为系统的持续改进和优化提供指导。

本次测试采用测试工具JMester进行系统性能测试。JMeter 是一个功能强大且广泛使用的测试工具，可用于对应用程序和系统进行性能测试和负载测试。JMeter 可以模拟多种类型的负载，包括并发用户、网络负载和服务器负载，以评估系统在不同负载条件下的性能和稳定性。

1. 系统并发测试

系统并发测试是一种测试方法，用于评估系统在同时处理多个并发用户或请求时的性能和稳定性。通过模拟真实的并发负载，可以确定系统在高负载条件下的响应时间、吞吐量、资源利用率和系统的稳定性。

本次测试实现了通过访问首页并发两在5000个虚拟用户和访问登录并发量情况下能，系统是否能够在规定响应时间返回服务请求。

测试设置如图6.1所示。

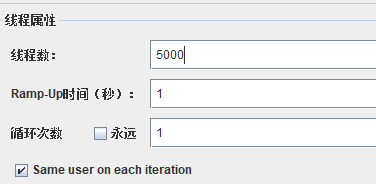


图5..3 JMester测试设置

1. 线程数：表示虚拟用户数。设置多少个线程数也就是设置多少虚拟用户数，当前设置5000个虚拟用户。
2. Ramp-Up时间(秒)：设置虚拟用户数全部启动的时长。当前设置即平均每秒启动5000个线程。
3. 循环次数：每个线程发送请求的个数。若勾选“永远”, 则所有线程会一直发送请求,直到手动点击工具栏上的停止按钮,或者设置的线程时间结束。当前设置为每个线程请求1次。
4. 系统压力测试报告图

本次测试结果如图5.4所示。本次测试所有线程访问的平均响应时间为1439ms，所有响应时间中位数为1508ms，其中90%的线程相应时间能达到1902ms以下，90%的线程相应时间能达到1920ms以下，99%的线程相应时间能达到1926ms以下，所有线程最低响应时间为4ms，最高响应时间为1930ms。从测试结果可以看出系统在500并发量时压力不大，系统完全能够进行服务响应。

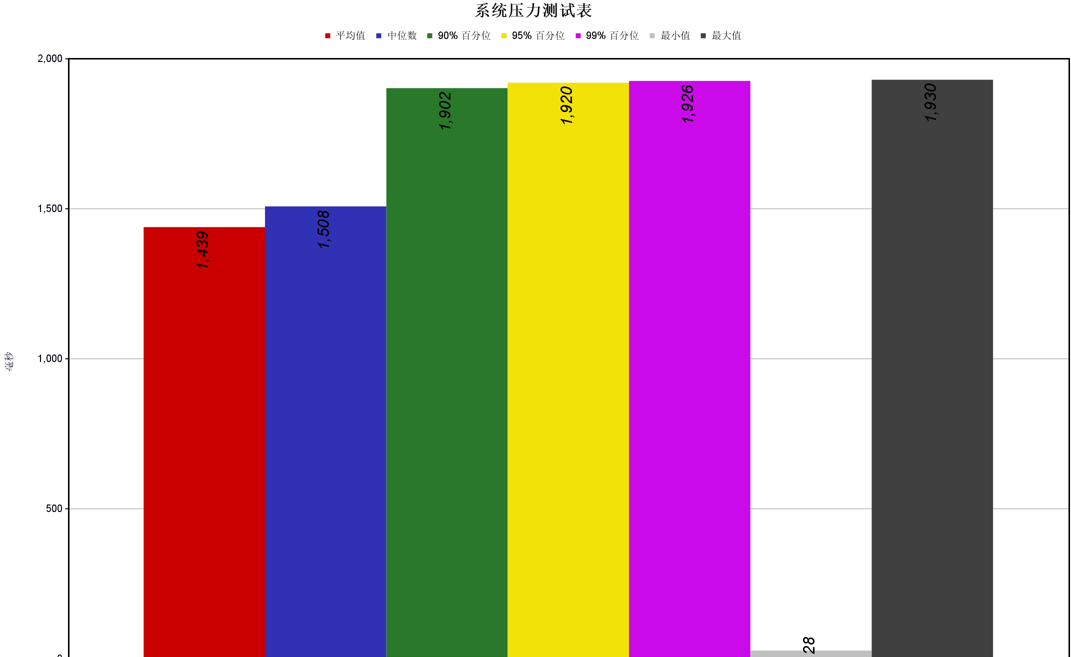


图 5..4系统压力测试结果图

1. 系统CPU和内存占用率

监测系统的CPU和内存占用率对于系统性能和稳定性的评估非常重要。如果CPU占用率过高，可能导致系统响应变慢或出现延迟，甚至系统崩溃。类似地，高内存占用率可能导致系统性能下降，甚至引发内存溢出错误。

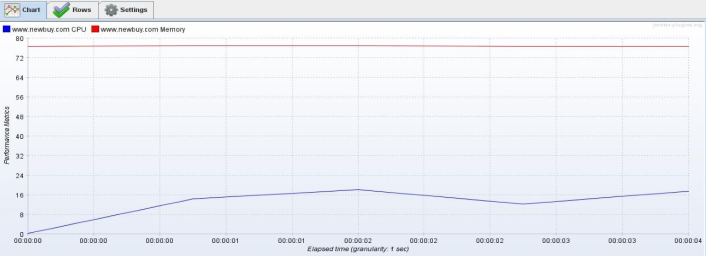


图 5..5CPU和内存占用率

本次测试结果如图5.5所示。从图中可以分析系统在5000并发量测试的CPU和内存占用率的实时数据。CPU占用率随时间推迟逐渐偏高，数据在1%-25%，数据情况良好。每次占用率未达到100%，表示内存运行正常，并未发生内存泄漏。

## 5.4本章小结

本章系统测试对系统进行了全面的测试，包括功能测试和压力测试。通过制定合理的测试方案和执行测试过程，可以及时发现和解决系统中存在的问题和缺陷，保证系统的质量和性能。测试是确保系统可靠性和稳定性的关键步骤，通过系统测试可以有效评估系统的功能和性能，并提供参考意见和改进方案。

参考文献

1. 中高端产品受追捧 2022年自行车电动自行车行业利润同比增长超20%——中国自行车协会召开2023年首场新闻发布会[J].中国自车,2023,No.505(02):2.

# 致 谢