



**本科毕业设计（论文）**

基于web的大型商场导购系统的设计与实现

学 号：15

学生姓名：牛明鑫

指导教师：贺 薪 宇

所在学院：计算机科学技术学院

所学专业：软件工程专业

2018 年 6 月

# 摘 要

二十一世纪是经济和科技全面发展的时代，由于经济的发展，越来越多的商场出现在人们生活，小的来说，商场可以是一个小小的超市。大的来说，商场可以包含在一整栋楼。由于商场的规模越来越大，商场内部结构也变得越来越复杂。当人们在大型商场中购物时，很难找到自己想要的商品，并且在离开的时候，常常找不到出口，遇见这种情况很多人，人们会向商场中的人员询问，这样不仅浪费了工作人员的时间，还有可能耽误自己的要事。当商家组织了最近的打折促销活动等不能及时的让购物者得到最新的消息，消费者也不能买到，物美价廉的商品。因此需要开发一个基于大型商场的导购系统，来帮助消费者在商场中的购物更加方便，让商场管理人员管理商品更加方便。本文通过对现代商场的可以更好的定位，让用户在商场中购物更加便捷，因此设计并开发一个商场导购系统。该系统包含6个管理模块，即：商场介绍模块商场介绍模块包括商场的主营业务，各楼层主营业务介绍、商场活动推介模块、最新商品的打折、寻找不同商品的位置、存储商品的信息、管理消费者。商场导购系统是基于mvc三层架构的开发模式，通过这种设计模式把模块，视图，和控制器分开开发，可以达到组件的最大利用。通过mysql存储数据、vue开发前端页面、node进行后台开发。争取让系统成本低、易升级、可扩充。

**关键词**：商场导购系统；node；vue；mysql

# Abstract

The 21st century is an era of comprehensive economic and technological development. Due to the development of economy, more and more shopping malls appear in people's lives. In small ways, shopping malls can be a small supermarket. Big, the mall can be included in a whole building. As the size of the shopping mall becomes larger and larger, the internal structure of the mall becomes more and more complex. When people shopping in the mall, it is hard to find items they want, and at the time of leave, often to find export, met this situation, many people will ask mall in personnel, not only a waste of time for the staff, and may delay your important. When merchants organize the latest discount promotions and so on, they can't get the latest news to the shoppers in a timely manner, and the consumers can't buy the goods at a reasonable price. Therefore, it is necessary to develop a shopping guide system based on large shopping malls to help consumers make shopping more convenient in the shopping mall and make it more convenient for store managers to manage the goods. In this paper, we can better position the modern shopping mall and make shopping more convenient for users in the shopping mall. Therefore, we design and develop a shopping mall system. The system contains six modules, namely: the mall module mall modules including shopping malls, specialising in the main business introduction, market activities to promote the floors module, the latest goods discount, find the location of the different commodity, storage, commodity information, manage the consumer. The shopping guide system is based on the development mode of the MVC three-layer architecture. The module, view and controller can be separately developed by this design mode, which can achieve the maximum utilization of the components. Through mysql storage data, vue development front page, node for background development. Strive to make system cost low, easy to upgrade, can expand.

**Key words:** Shopping guide system; node; vue; mysql

**目 录**

[摘 要 I](#_Toc514678117)

[Abstract II](#_Toc514678118)

[第1章 概述 1](#_Toc514678119)

[1.1 课题开发背景 1](#_Toc514678120)

[1.2 国内外发展状况 1](#_Toc514678121)

[1.3 课题研究的意义 2](#_Toc514678122)

[第2章 方案论证 3](#_Toc514678123)

[2.1 方案选择 3](#_Toc514678124)

[2.1.1 开发语言选择 3](#_Toc514678125)

[2.1.2 数据库选择 3](#_Toc514678126)

[2.1.3 编译器选择 4](#_Toc514678127)

[2.2 可行性分析 4](#_Toc514678128)

[2.2.1 经济可行性 5](#_Toc514678129)

[2.2.2 技术可行性 5](#_Toc514678130)

[2.2.3 操作可行性 5](#_Toc514678131)

[2.2.4 法律可行性 5](#_Toc514678132)

[2.3 方案确定 6](#_Toc514678133)

[2.3 系统开发环境 6](#_Toc514678134)

[第3章 需求分析 7](#_Toc514678135)

[3.1 功能需求分析 7](#_Toc514678136)

[3.2 性能需求分析 7](#_Toc514678137)

[3.3 系统数据流图和数据库需求分析 7](#_Toc514678138)

[3.3.1 系统数据流图 7](#_Toc514678139)

[3.3.2 数据库需求分析 9](#_Toc514678140)

[3.4 数据字典 10](#_Toc514678141)

[3.5 系统E-R图 11](#_Toc514678142)

[3.5 系统用例图 14](#_Toc514678143)

[第4章 设计论述 16](#_Toc514678144)

[4.1 概要设计 16](#_Toc514678145)

[4.1.1 总体功能设计 16](#_Toc514678146)

[4.1.2 用户登录模块流程图 16](#_Toc514678147)

[4.1.3 管理员模块流程图 17](#_Toc514678148)

[4.2 数据库详细设计与实现 18](#_Toc514678149)

[4.2.1 创建数据库 18](#_Toc514678150)

[4.2.2 创建表和字段 19](#_Toc514678151)

[4.3 数据库连接 21](#_Toc514678152)

[4.4 详细设计 22](#_Toc514678153)

[4.4.1 用户登录、注册 22](#_Toc514678154)

[4.4.2 主界面设计 24](#_Toc514678155)

[4.4.3 用户管理管理模块 27](#_Toc514678156)

[4.4.4 商品信息管理/查询模块 29](#_Toc514678157)

[4.4.5 商场管理模块 30](#_Toc514678158)

[4.4.6 商场活动推介模块 31](#_Toc514678159)

[第5章 软件测试与结果分析 33](#_Toc514678160)

[5.1 测试的目的和原则 33](#_Toc514678161)

[5.2 测试计划 33](#_Toc514678162)

[5.3 测试过程 34](#_Toc514678163)

[5.3.1 单元测试 34](#_Toc514678164)

[5.3.2 集成测试 36](#_Toc514678165)

[5.3.3 系统测试 37](#_Toc514678166)

[5.4 测试结论 38](#_Toc514678167)

[结 论 39](#_Toc514678168)

[致 谢 40](#_Toc514678169)

[参考文献 41](#_Toc514678170)

# 第1章 概述

## 1.1 课题开发背景

经济的发展和人们生活水平的提高，大型商场成为人们消费和玩乐的场所，在大型商场中不仅可以购物，而且存在着各种样式的活动。但是活动虽多，并不能及时的把消息传递给消费者。消费者进入大型商场中，也不能及时的找到自己需要的商品，

即便转了很久之后，也不一定找到。因此，开发一个大型商场的导购系统是迫在眉睫的一件事。

二十一世纪是属于科技的时代，科技已经走进人们生活的各个方面，科技改变改变了人们生活方式，给人们带来了各种便利，当人们买单时可以通过，支付宝、微信支付。当人们进入大型商场中，会看到宣传个人品牌和商品占据了比较大的面积和空间，在宣传时，必然会造成资源的浪费、不环保，每次品牌店换位置，就需要更新很多东西，会造成不必要的成本开支等[1]。这种浪费是完全可以避免的，可以通过科技，节省出来人力，物力，财力，从而让商家节省花费。这些可以通过开发一个商场导购系统进行改变。

如今，在商场中进行购物的趋向于年轻化，这一类的消费者易于接受新鲜事物。因此在商场中开发一个导购系统，可以快速的被消费者所接受。虽然一些商场已经有了导购系统，但仍然存在着一定的缺点，例如：仅仅展示商品，却不能让用户根据自己的需要找到该商品。开发商场导购系统，是可以满足用户需求的。

## 1.2 国内外发展状况

计算机网络日益深入人们的生活，Internet已经遍及全世界，现在商场购物还没有完全顺应时代发展的潮流，大型商场没有更好的和科技结合起来。当人们浏览于大型商场中，消费者不能更好的体验新兴的科技，也不能够满足自己心里的需要。商场必须和科技结合起来，才能让消费者在商品种类繁多的商场中购物时，有更好的体验。

从目前情况来看，人们更愿意到商品齐全的大商场进行购物。商场管理人员趋于推广消费者自助购物，这种购物方式，导致了用户在商场中迷失了方向。目前商场中存在的导购系统，可以基本上满足人们的需求。

在国内有导购系统已经运行在各大城市中的各大商场中。例如：在香港、北京、广州、武汉等城市的高端商场已率先应用商场导购信息发布系统，通过商场内的智能互动触摸设备，用户根据自己的需要，来查找商品和各类活动。在国内发展的如此快速，国外也有着同等的发展速度。

## 1.3 课题研究的意义

商场越来越大，人们利用闲暇时间在商场中逛逛，但有的时候，我们也会在商场中迷路，在商场中开发一个导购系统就可以为我们的生活提供很大的便利。一方面互联网技术的成熟，本系统得益于开发。理论上开发一个基于室内的导购系统，是完全可以实现的，实际上个人未接触到太多新兴的技术，只能提供一些相对简单的查询和导购的购台管理系统。伴随这科技时代的发展，我个人的web技术的研究和发展也将迈向一个新的台阶。

从现在看来，我开发的这个商场导购系统。可以为用户提供一些基本便利，让商场管理人员发布商品信息，以及商场打折促销信息，用户可以即时的获取信息，然后根据自己的需要进行选择。

# 第2章 方案论证

## 2.1 方案选择

### 2.1.1 开发语言选择

1．方案一（node,express）

Node.js是一个在JavaScript运行环境，实质是对Chorme V8引擎进行了封装，Node.js对一些特殊用例进行优化，提供替代的API，使得V8在非浏览器下运行的更好，Nodejs作为一个新型的前端框架，后台语言，在单线程、非阻塞IO、V8虚拟机、事件驱动有很大的优势。并且node可以在不新增额外线程的情况下，依然可以对任务进行并发处理[6]。它可以通过事件轮询来进行并发操作Express是一个简介而灵活的node.jsWeb应用框架，提供一系列强大特性来帮助你创建各种web应用。Express不对node.js进行二次抽象，只是扩展了web应用所需要的功能，丰富http工具以及来自connect框架的中间件随取随用，创建强健、友好的API变得快速且简单[7]。

2．方案二（java）

Java是一种简单的，跨平台的，面向对象的，分布式的，解释型的，强大的，安全的，架构中立的，可移植的，性能很优异的，原生适用于多线程，动态面向对象的程序设计语言[8]。作用是作为数据传输的载体和配置系统组件的作用。

3．方案三（ASP.NET）

ASP简单，方便维护，是一个小型页面应用程序的选择，ASP文件是包含在HTML代码所组成的文件中的，修改和测试都比较容易，无须编译或链接就可以解释执行，因为服务器是将以HTML格式执行ASP程序的结果，返回到客户端浏览器，所以用户不会看到由原始代码编写的ASP，以防止ASP程序代码被盗[9]。

### 2.1.2 数据库选择

1．方案一（mysql）

MySQL是一个小型关系型数据库管理系统，开发者为瑞典MySQL AB公司，在2008年1月16号被Sun公司收购。而2009年,SUN又被Oracle收购.对于MySQL的前途，目前MySQL被广泛地应用在各个公司的小型网站上。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，许多中小型网站为了降低网站总体拥有成本而选择了MySQL作为网站数据库[10]。

MYSQL是一个真正的多用户，多线程的结构化查询语言数据服务器，它以客户机/服务器的结构实现，并且还具有以下优势[10]。

速度快；

连接性和安全性；

可移植性；

它支持SQL语言；

容易使用；

成本优势。

2．方案二（MongoDB）

MongoDB是一个基于分布式文件存储的数据库，旨在为WEB应用提供可以扩展的高性能数据存储解决方案。它是介于关系数据库和非关系数据库之间的产品，是非关系数据库中功能最丰富的，最像关系数据库的[11]。

MongoDB具有以下功能特性：

面向集合存储，易存储对象类型的数据。

模式自由。

支持动态查询。

支持复制和故障恢复。

使用高效的二进制数据存储，包括大型对象。

自动处理碎片，以支持云计算层次的扩展性。

文件存储格式为BSON。

可以通过网络访问。

3．方案三（SQL Server）

SQL Sever是Web上最流行的用于储存数据的数据库，它已被广泛应用在电子商务、银行等与数据库相关的行业，SQL Sever提供了大量的web和电子商务功能，例如对XML和Internet标准的普遍支持，通过web便捷安全地访问数据，强健的、灵活的、基于web的和安全的应用程序管理等[12]。而且，因为它方便操作地特性和它的友好的操作页面，非常受广大用户的喜欢。

### 2.1.3 编译器选择

Visual Studio Code 是一个可以运行于mac、windows和linux之上的，现代化跨平台编译器。该编译器，可以满足我使用命令运行程序，以及自动化检查代码，让代码以一种高亮的方式显示。方便查看和修改。

## 2.2 可行性分析

可行性分析是通过对项目的主要内容和配套条件，如市场需求，资源供应等，从技术、经济和工程等方面进行调查研究和分析比较，并且对项目建成以后可能取得的财务、经济效益以及社会环境进行预测[2]。在该项目中主要是通过技术可行性，经济可行性，法律可行性，操作可行性，进行分析。

### 2.2.1 经济可行性

就本系统来说，开发所需要的资金少，系统开发完成后，可以让用户，更容易在商场中找到自己需要的商品，也可以减少商家利用场地打广告的费用，把商场变得现代化，让人们的从繁重的劳动中解放出来，即节省了大量时间，又提高了用户的体验。通过我对本次开发的预算，开发该商场导购系统，可以最大化的经济效益。本次开发主要是个人的花费，最主要的开发该系统可以提高工作效率。

### 2.2.2 技术可行性

在开源的新时代，开发一个基于web的商场导购系统。我们可以有许多工具进行选择，例如我们经常使用的webstorm，HBuilder等都是一些比较理想的前端开发工具，在后台开发工具上，有IDEA，VSCODE等。在数据库选择中有mysql，sqlserver等。

如今前端开发技术已经比较成熟，html，css，js已经广为普及，成为程序员一项比不可少的技能。html是超文本标记语言，是标准通用语言的的下一个应用，主要是用于书写页面，超文本标记语言有简易性，可扩展性，平台无关性，通用性[3]。Css是一种用来表现html或xml（标准通用标记语言的子集）等文件样式的计算机语言，Css即可以修饰静态网页，还可以配合配合各种脚本语言动态地对网页各元素进行格式化[4]。Js是一种动态类型、弱类型、基于原型的语言，内置支持类型，可以给html网页增加动态功能[5]。前端因此也有许多流行的框架，例如vue.js,angular.js,react.js等。该项目就是基于vue.js。vue2.0是一个相对稳定的版本，源码简介，在不操作dom的基础上，通过双向数据绑定，把注意力主要集中在解决问题上。在后端使用了node,express。相对来说，很多大公司以及在使用node了。Node有自己的社区，并且已经很完善，可以很放心的使用。

因此从技术上来看，实现web导购系统是完全可行的。

### 2.2.3 操作可行性

导购系统从操作可行性分析，主要来看，是否方便操作，本系统在开发完成后，在其他电脑上运行是否可以，达到用户的需求。即便出现了问题，也要快速的用现代的技术把他解决，让用户可以尽快的体验到该系统。

除此之外，管理者应该具备一定的计算机知识，只有这样才可以更快的上手。本系统可以在windows，ubuntu，mac下的任意浏览器使用，界面友好，操作便捷。所以在操作上是可行的。

### 2.2.4 法律可行性

本系统的开发和设计过程完全是个人独立完成出来的，虽然在此过程中遇到了不少的问题，但都通过查阅书籍以及寻求老师和同学的帮助解决了问题，所使用的软件是正版授权的。并且所使用的开发技术都可以在相应的官网找到，不存在侵权的问题。从以上来看，web导购系统在法律上是可行的。

## 2.3 方案确定

根据以上几种语言以及数据库进行分析，结合我所掌握的前端框架，以及我在平时学习工作中的事件应用及对他们掌握的程度，最终选择了node作为服务器端开发语言，以express作为后端开发框架，以MySql作为数据库，以vue.js来开发前端，并且在vscode上完成本次开发任务。

## 2.3 系统开发环境

操作系统：windows 系统

数据库：Mysql

开发工具：Visual Studio Code

后台开发框架：node，express

前台开发框架：vue.js

硬件配置：CPU2.0GHz以上，内存1GB以上。

浏览器：Google/firefox/360安全浏览器

# 第3章 需求分析

## 3.1 功能需求分析

通过可行性分析，确定该系统的使用用户，以及各个模块的功能和意义之后，本系统主要是在商场中方便用户进行购物。主要包括：商场人员信息管理模块、商品信息管理模块、分类商品销售位置查询模块、商品打折促销模块、商场活动推介模块、商场介绍等。下面主要从各个模块进行分析。

1．商场人员信息管理模块：主要给商场中管理人员，提供雇员的详细信息。以及可以给每个雇员进行一系列的操作。主要增删改查，设置权限等。

2．商品信息管理模块：商场中的管理员可以对自己店里的商品进行管理，主要是增删改查商品。并且可以及时的发布一系列最新的消息。商品打折促销等。

3．类商品销售位置查询模块：这一块主要是给每个商场中商家按售卖的商品类型进行分类，然后把各个位置进行存储，用户可以根据自己想要的商品查询到该商品的位置。

4．品打折促销：这类商品主要是雇员发布商品的消息。商品促销打折等。然后用户可以及时的看到该类消息。

5．场活动推介模块：主要是商家与顾客的一场互动，类似于现实中的商家对商品的介绍，宣传来达到吸引用户的目的。

6．场介绍：主要是介绍这个商场中的详细信息，以及各个楼层的详细信息，以及该商场的主要经营的业务，那个方面是强项等。

## 3.2 性能需求分析

本系统界面友好，反应速度快，可以很好的满足用户的体验，操作便利，方便让管理人员很快的上手。在设计上符合大多数人使用习惯。在数据存储方面也比较安全，可靠。可以放心的使用。并且可以极大的提高的办事效率。

## 3.3 系统数据流图和数据库需求分析

### 3.3.1 系统数据流图

1．根据需求分析得到3-1所示的顶层数据流图







图3-1 系统顶级数据流图

2．一级数据流图

对顶层数据流图进行细化，得到如下

（1）商品/商场信息数据流图如图3-2

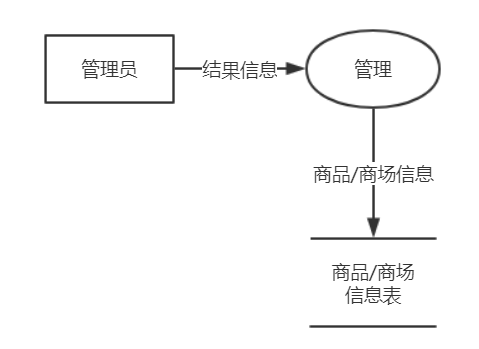


图3-2 商品/商场信息管理数据流图

（2）用户信息管理数据流图，如图3-3

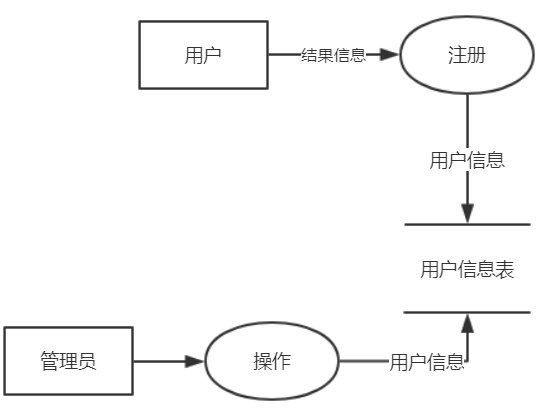


图3-3 用户信息管理流程图

（3）结果管理数据流图，如图3-4

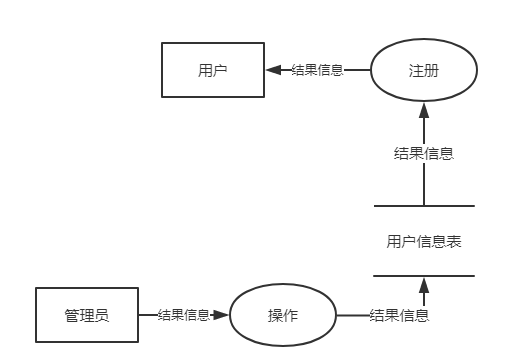


图3-4 结果管理数据流图

### 3.3.2 数据库需求分析

管理员在使用该系统的时候，能够很好的对提供的信息、保存、更新、查询、修改、删除等。这就是要求数据库的结构满足各种信息的输入和输出。收集一些数据、数据结构及数据处理的过程，组成一份详尽的数据字典，为以后的数据库打下坚实的基础。

通过系统的各个部分的功能分析，总结出如下需求信息：

1．用户可以分为用户（消费者）和管理员（店员）

2．一个用户可以查看多个商品的消息，一个商品可以被多个用户查看。

3．一个用户可以查看多个楼层的信息，一个楼层可以被多个用户查看。

4．一个用户可以查看多个打折促销产品，一个打折促销产品可以被多个用户查看。

5．一个用户可以查看多个商场活动推介，一个商场活动推介可以被多个用户查看。

通过以上系统功能分析和需求总结设计如下所示的数据项和数据结构。

1．管理员信息表，包括数据项： 用户名，密码，性别，头像，邮箱等。

2．用户表，包括数据项：用户名，密码，性别，头像，邮箱等。

3．商品信息表，包括数据项：商品的名称，商品的简介，商品的价格，商品的图片，位置，发布人，发布时间。

4．商场信息表，包括数据项：商场简介，商场的位置，商场中的各个楼层介绍。

5．打折促销表，包括数据项：发布时间，打折的名称，价格，折扣价，位置

6．商家信息表，包括数据项：商家位置，商家名称，商家满意度。

## 3.4 数据字典

在该系统的数据字典设计过程中，着重对数据字典中的数据结构做了详细的分析和总结。

1．名字： 用户信息表

描述：该系统注册过的消费者信息

定义：用户信息=用户编号+用户昵称+用户密码+用户头像+用户性别+用户邮箱

2．名字： 管理员表

描述：该系统所委派的管理员

定义：管理员信息=管理员编号+管理员昵称+管理员密码+管理员头像+管理员性别+管理员邮箱

3．名字： 商品信息表

描述：管理员所添加的商品信息

定义：商品信息=商品名称+商品售价+商品描述+商品销量+商品评价+发布时间+发布商家

4．名字： 商场信息表

描述：主要介绍该商场的信息

定义：商场信息=商场的名称+商场的主营业务+各个楼层的信息

5．名字： 打折促销信息表

描述：打折促销的商品信息

定义：打折促销信息=打折促销商品名+发布时间+发布楼层+发布人员+发布的商家+促销价格

6．名字： 商家信息表

描述：商家的信息

定义：商家信息=商家的位置+商家的主营业务+商家名称+商家的评价

## 3.5 系统E-R图

E-R图也是实体-联系图，提供表示实体类型属性和联系的方法，用来描述现实世界里的概念模型，是表示概念模型的一种方式，用矩形表示实体型，矩形框内写出实体名；用椭圆形状表示实体的属性，并用无向边将其连接起来，通过菱形表示实体之间的联系，用无向边分别与有关实体型连接起来，同时在无向边旁标上联系的类型（1:1,1：n,1：m）[13]。

1．系统E-R图，如图3-5所示

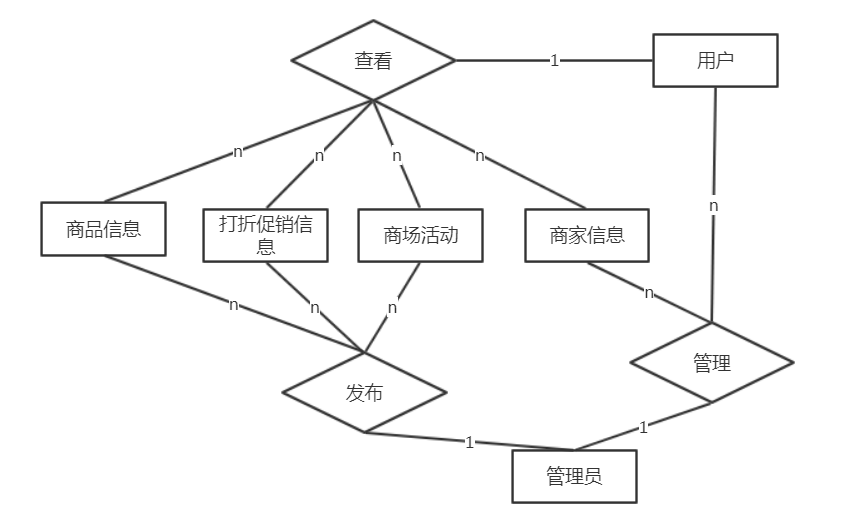


图3-5 系统E-R图

2．管理员的实体属性图，如图3-6所示

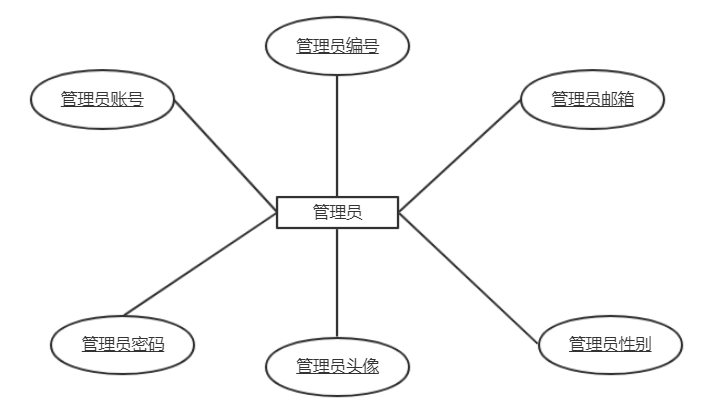


图3-6 管理员实体图

2．用户的实体属性图，如图3-7所示

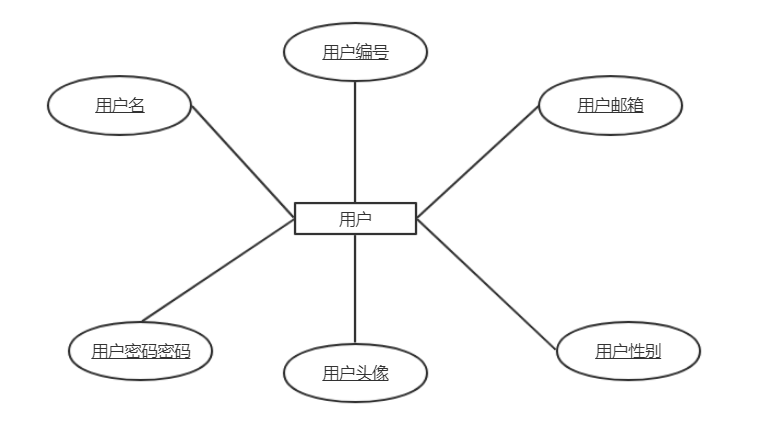
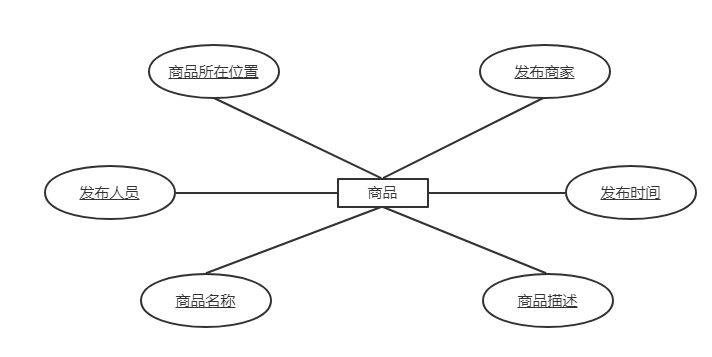


图3-7 户实体图

3．商品实体属性图，如图3-8所示



3-8 商品实体图

4．商场实体属性图，如图3-9所示

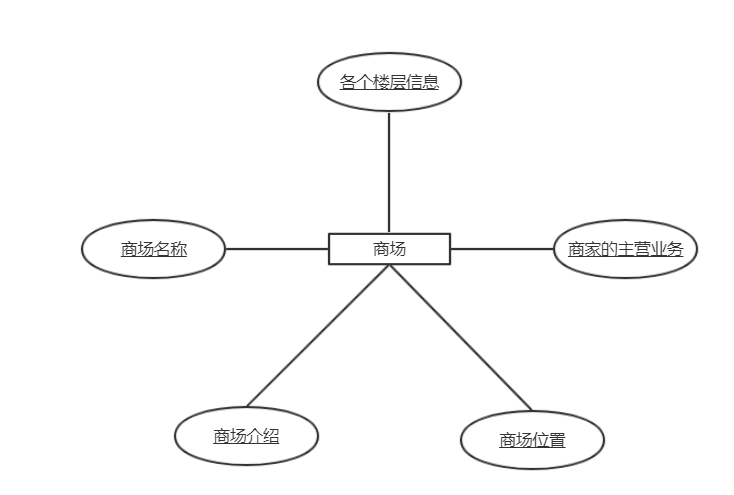


图3-9 商场实体图

5．商家实体属性图，如图3-10所示

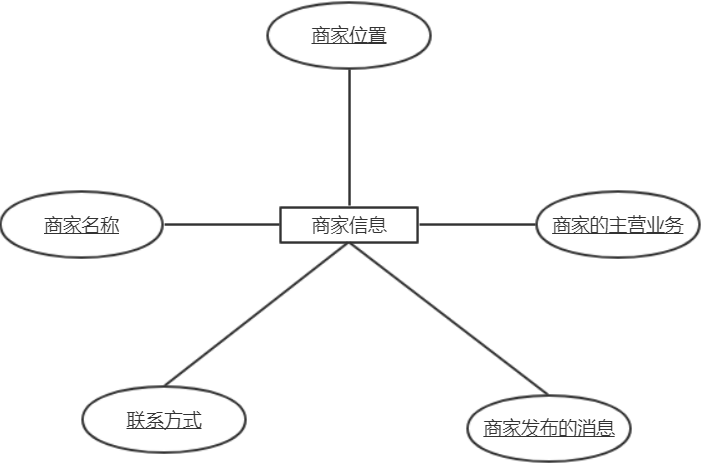


图3-10 商家实体图

6．打折促销实体属性图，如图3-11所示

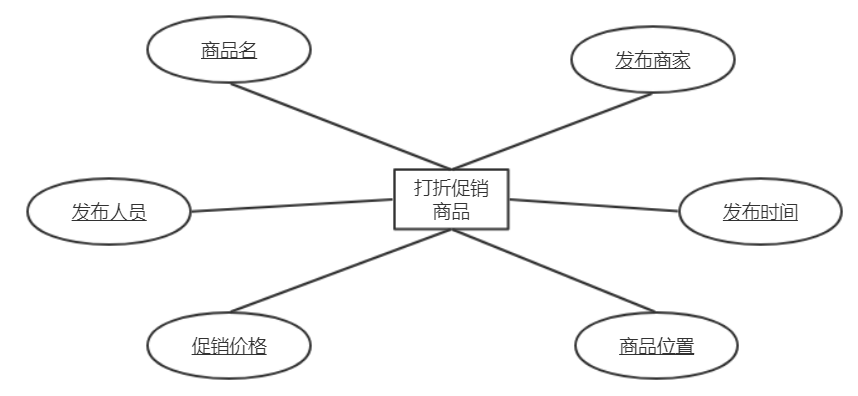


图3-11 打折促销实体图

## 3.5 系统用例图

用例图是由use case（用例）和actor（角色）和系统边界组成的。表示系统做的那些事情，可以帮助你分析系统有那些功能，以及让你清楚的知道系统的内外部（角色）的交互[13]。

本系统的用例图，如图3-12所示

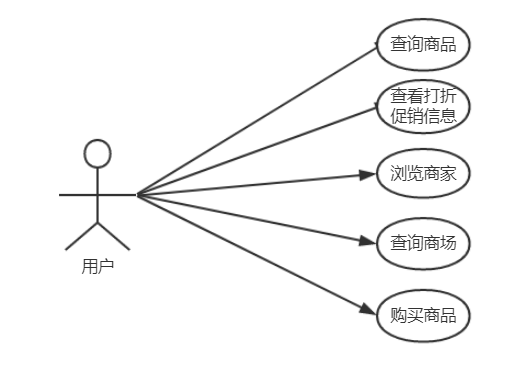


图3-12 系统用例图

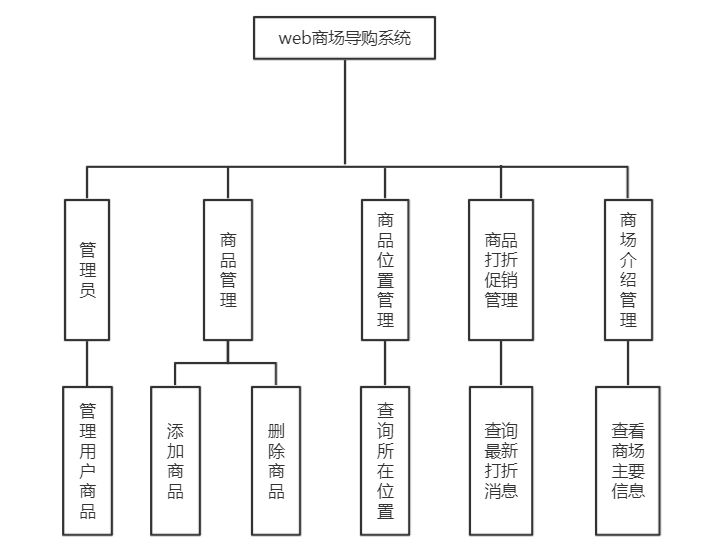
# 第4章 设计论述

## 4.1 概要设计

### 4.1.1 总体功能设计

本商场导购系统主要目的为用户和商家提供方便、快捷的服务，方便管理人员对雇员、商品、商品打折促销、商场活动推广、商品查询等进行管理。本系统具体的功能如下。

整个web商场导购系统主要围绕跟商场相关的各个方面信息以及使用的权限不同总共有以下相对独立的模块。主要从商场，商品，以及人员进行管理。如图4-1所示。

图4-1 web商场导购系统功能结构图

### 4.1.2 用户登录模块流程图

当用户使用本系统的时候，首先可以已访客的身份进行查询浏览商品信息，并且可以找到该商品的位置。也可以注册成为该商场的用户。商场人员只有管理员可以添加。注册完成后，用户登录时，可以进行如下操作。如图4-2所示。

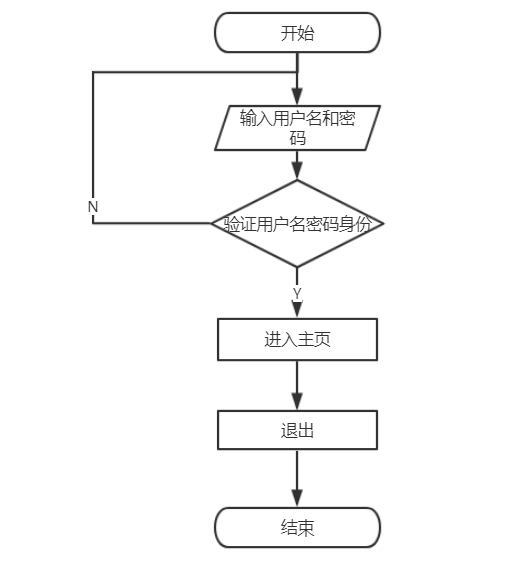


图4-2 登录模块流程图

### 4.1.3 管理员模块流程图

管理在登录的时候和普通用户登录的界面是一样的，只不过系统会先进行检测你是否为管理员。如果为管理员则拥有权限，可以发布最新的商品消息，也可以对用户进行管理，在添加好要发布的消息，系统会检测是否存在错误。信息信息只有在无误的情况在，在可以保存到数据库，然后进行发布。但是要注意一点每个商家，仅可以修改本人发布的消息，对他人发布的信息没有权限修改和删除。如图4-3所示。

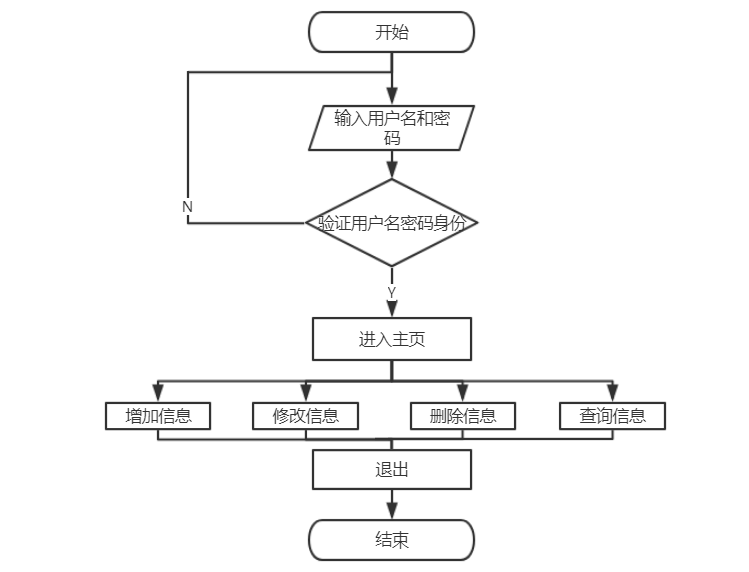


图4-3 管理员模块流程图

## 4.2 数据库详细设计与实现

本系统采用了被广泛使用的数据库管理系统MySql，数据库名称为“market”。数据库“market包含以下”6个表：用户表，商品表，管理员表，打折促销表，商场表，商家表。

### 4.2.1 创建数据库

本系统通过node运行，采用Navicat来简化数据库的管理以及可以降低系统的管理成本。Navicat是图形化用户界面，可以让你安全简单的创建，组织，访问并且使用已经存在的信息。在打开Navicat之后创建一个新数据库“market”，如图4-4所示。

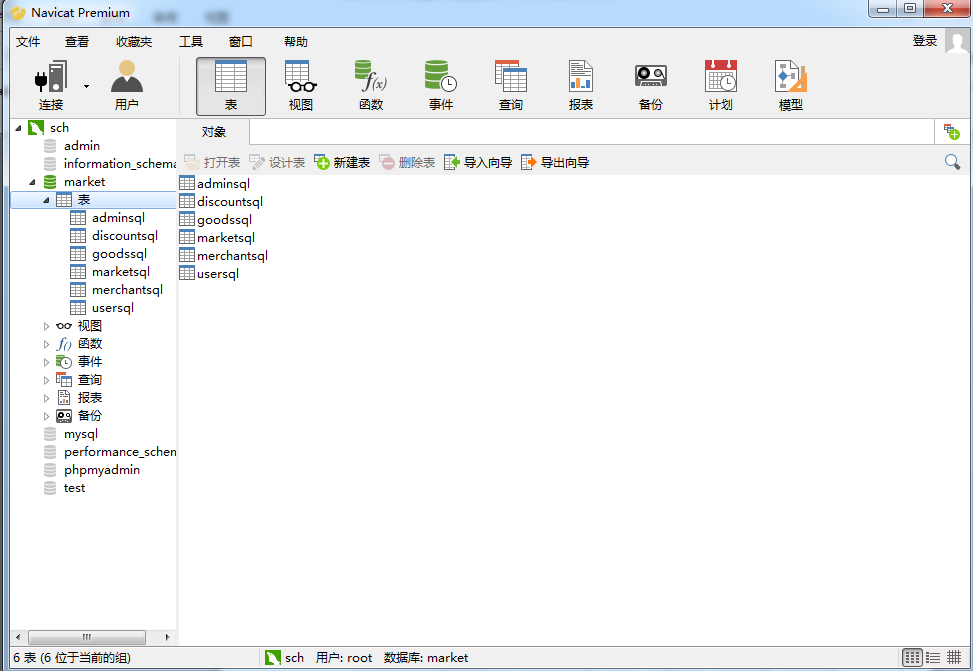


图4-4 Navicat界面图

### 4.2.2 创建表和字段

根据第3章对数据库的概念模型以及逻辑模型设计的理解，结合分析实体属性图和E-R图可以规划出web商场导购系统的具体数据库表，详细设计如下所示：

1．用户信息表(usersql)

用户信息表主要用来存储注册用户的基本信息。主要包括：用户编号，用户账号，用户密码，用户头像，用户性别，用户邮箱其中用户编号作为主键。该数据库设计表如4-5所示。

表4-5 用户信息表(usersql)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **数据类型** | **长度** | **是否为空** | **是否为主键** | **字段说明** |
| 1 | uid | 自动编号 | 4 | 否 | 是 | 用户编号 |
| 2 | uname | 字符 | 30 | 否 | 否 | 用户名 |
| 3 | upwd | 字符 | 30 | 否 | 否 | 用户密码 |
| 4 | uimg | 字符 | 100 | 否 | 否 | 用户头像 |
| 5 | usex | 整型 | 1 | 否 | 否 | 用户性别 |
| 6 | uemail | 字符 | 50 | 否 | 否 | 用户邮箱 |

2．管理员信息表(adminsql)

管理员信息表主要存放商家入住管理员的基本信息。主要包括管理员编号，管理员账号，管理员密码，管理员头像，管理员性别，管理员邮箱，其中管理员编号作为主键。该数据库设计表如4-6所示。

表4-6 管理员信息表(adminsql)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **数据类型** | **长度** | **是否为空** | **是否为主键** | **字段说明** |
| 1 | aid | 自动编号 | 4 | 否 | 是 | 管理员编号 |
| 2 | aname | 字符 | 30 | 否 | 否 | 管理员账号 |
| 3 | apwd | 字符 | 30 | 否 | 否 | 管理员密码 |
| 4 | aimg | 字符 | 100 | 否 | 否 | 管理员头像 |
| 5 | asex | 整型 | 1 | 否 | 否 | 管理员性别 |
| 6 | aemail | 字符 | 50 | 否 | 否 | 管理员邮箱 |

3．商品信息表(goodssql)

商品信息表主要存放平台管理员发布的关于商品的各项信息。主要包括商品编号，商品名称，商品描述，商品销量，商品评价，发布时间，发布商家，商品位置。其中商品编号，作为主键。该数据库设计表如4-7所示。

表4-7 商品信息表(goodssql)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **数据类型** | **长度** | **是否为空** | **是否为主键** | **字段说明** |
| 1 | gid | 自动编号 | 4 | 否 | 是 | 商品编号 |
| 2 | gname | 文本 |  | 否 | 否 | 商品名称 |
| 3 | gdescribe | 文本 |  | 否 | 否 | 商品描述 |
| 4 | gdate | 日期 |  | 否 | 否 | 发布时间 |
| 5 | gcomsign | 字符 | 255 | 否 | 否 | 发布商家 |
| 6 | glocation | 字符 | 255 | 否 | 否 | 商品位置 |

4．商场信息表(marketsql)

商场信息表主要存在，该导航系统已经进入的商场。主要包括商场编号，商场名称，商场介绍，楼层信息，主营业务。其中商场编号，作为主键。该数据库设计表如4-8示。

表4-8 商场信息表(marketsql)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **数据类型** | **长度** | **是否为空** | **是否为主键** | **字段说明** |
| 1 | gid | 自动编号 | 4 | 否 | 是 | 商场编号 |
| 2 | gname | 字符 | 32 | 否 | 否 | 商场名称 |
| 3 | gintroduce | 文本 |  | 否 | 否 | 商场介绍 |
| 4 | ginfo | 文本 |  | 否 | 否 | 楼层信息 |
| 5 | gimportant | 字符 | 255 | 否 | 否 | 主营业务 |
| 6 | glocation | 字符 | 255 | 否 | 否 | 商场位置 |

5．打折促销信息表(discountsql)

打折促销信息表主要存放平台管理员发布的关于打折促销商品的各项信息。主要包括促销编号，促销名称，发布人员，发布日期，促销价格，发布商家，商品位置。其中促销编号，作为主键。并且，促销的商品，是已经发布的商品。该数据库设计表如4-9所示。

表4-9 打折促销信息表(discountsql)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **数据类型** | **长度** | **是否为空** | **是否为主键** | **字段说明** |
| 1 | did | 自动编号 | 4 | 否 | 是 | 促销编号 |
| 2 | dname | 字符 | 32 | 否 | 否 | 促销名称 |
| 3 | duser | 字符 | 32 | 否 | 否 | 发布人员 |
| 4 | ddate | 日期 |  | 否 | 否 | 发布日期 |
| 5 | dprice | 整型 | 8 | 否 | 否 | 促销价格 |
| 6 | dcomsign | 字符 | 255 | 否 | 否 | 发布商家 |
| 7 | dlocation | 字符 | 255 | 否 | 否 | 商品位置 |

6．商家信息表(merchantsql)

商家信息表主要存放平台管理员发布的关于商品的各项信息。主要包括促销编号，促销名称，发布人员，发布日期，促销价格，发布商家，商品位置。其中促销编号，作为主键。并且，促销的商品，是已经发布的商品。该数据库设计表如4-10所示。

表4-10 商家信息表(merchantsql)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **数据类型** | **长度** | **是否为空** | **是否为主键** | **字段说明** |
| 1 | mid | 自动编号 | 4 | 否 | 是 | 商家编号 |
| 2 | mname | 字符 | 32 | 否 | 否 | 商家名字 |
| 3 | mphone | 字符 | 11 | 否 | 否 | 联系方式 |
| 4 | mcomsign | 字符 | 255 | 否 | 否 | 已发布商品 |
| 5 | mimportant | 字符 | 255 | 否 | 否 | 主营业务 |
| 6 | mlocation | 字符 | 255 | 否 | 否 | 商家位置 |

## 4.3 数据库连接

Node中连接数据库，我们只需要使用公共的Db类进行操作，在连接数据库的时候。我们需要安装MySql的npm包，这个是官方提供的并且及其稳定的包。连接如下所示：

module.exports = {

mysql: {

host: 'localhost',

user: 'root',

password: '',

database: 'market',

port: '3306'

} }

var conn = mysql.createConnection(models.mysql);

conn.connect();

## 4.4 详细设计

### 4.4.1 用户登录、注册

登录系统分为用户、管理员和超级管理员（admin）管理员只能由admin分配，通过注册可以成为用户。系统在登录的的时候通过输入的用户名进行判断用户、管理员和admin。如图4-11所示。



图4-11 系统登录界面图

代码如下

router.post('/login', function (req, res) {

let params = req.body;

conn.query('select username from usersql where username="' + params.username + '" and password="' + params.password+'"', function (err, rows) {

if (err || rows.length == 0) {

console.log(err)

res.send({code: 0})

}else {

res.send({code: 1})

}

})

})

在商场中消费者可以通过注册成为该商场中的用户。如图4-12。



图4-12 用户注册界面

代码如下

注意的问题

1．用户名必须只是唯一的，不可以重复。

2．并且两次输入的密码要一致

router.post('/register', function(req, res) {

let params = req.body;

conn.query('select username from usersql where username="' + params.username + '"', function(err, rows) {

if(err || rows.length == 0) {

console.log(err);

conn.query('insert into usersql set ?', {username: params.username,password:params.password}, function(err, rows) {

if(err) {

console.log(err);

res.send({code: 0, ms: '服务器出错'})

} else {

res.send({code: 1})

}

})

} else {

res.send({code: 0, ms: '账户已存在'})

}

})

})

### 4.4.2 主界面设计

主界面主要是该系统的主页，主页包括各个模块的列表以及最新的打折促销消息。用户可以通过该主页提供的消息，寻找到自己所需要的商品。管理员则可以进入后台管理页面对对各个商品进行修改。如图4-13所示

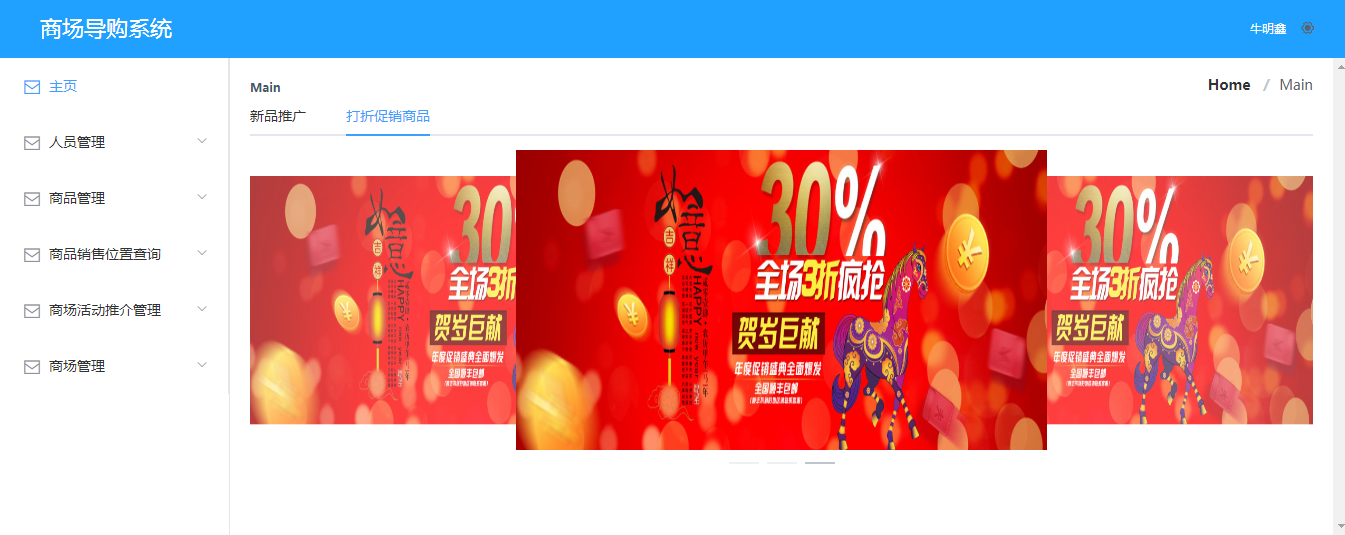


图4-13 系统主界面效果

代码如下

<template>

<el-row class="container">

<el-header :span="24" class="header">

<el-col style="width:230px" class="logo" >

商场导购系统

</el-col>

<el-dropdown class="userinfo">

<i class="el-icon-setting" ></i>

<el-dropdown-menu slot="dropdown">

<el-dropdown-item>个人中心</el-dropdown-item>

<el-dropdown-item divided @click.native="logout">退出登录</el-dropdown-item>

</el-dropdown-menu>

</el-dropdown>

<span class="userinfo">{{userName}}</span>

</el-header>

<el-col :span="24" class="main">

<aside >

<!--导航菜单-->

<el-menu class="el-menu-vertical-demo" :default-active="$route.path" unique-opened router>

<div v-for="(item, index) in dataNav.data" :key="index" >

<el-menu >

<div v-if="index == 0">

<router-link :to="`${item.route}`">

<el-menu-item index="1-1">

<i class="el-icon-message"></i>{{item.name}}

</el-menu-item>

</router-link>

</div>

<el-submenu index="1" v-else>

<template slot="title"><i class="el-icon-message"></i>{{item.name}}</template>

<el-menu-item-group>

<router-link :to="`${item.routeFirst}`">

<el-menu-item index="1-1">

{{item.first}}

</el-menu-item>

</router-link>

<router-link :to="`${item.routeSecond}`">

<el-menu-item index="1-2">

{{item.second}}

</el-menu-item>

</router-link>

</el-menu-item-group>

</el-submenu>

</el-menu>

</div>

</el-menu>

</aside>

<section class="content-container">

<div class="grid-content bg-purple-light">

<el-col :span="24" class="breadcrumb-container">

<strong class="title">{{$route.name}}</strong>

<el-breadcrumb separator="/" class="breadcrumb-inner">

<el-breadcrumb-item v-for="item in $route.matched" :key="item.path">

{{ item.name }}

</el-breadcrumb-item>

</el-breadcrumb>

</el-col>

<el-col :span="24" class="content-wrapper">

<transition name="fade" mode="out-in">

<router-view></router-view>

</transition>

</el-col>

</div>

</section>

</el-col>

</el-row>

</template>

### 4.4.3 用户管理管理模块

该模块只有管理员可以对用户信息进行修改，通过编辑可以修改用户的各项信息，这项只能管理员进行操作。当然用户也可以在个人页面进行修改个人信息。如图4-14所示。

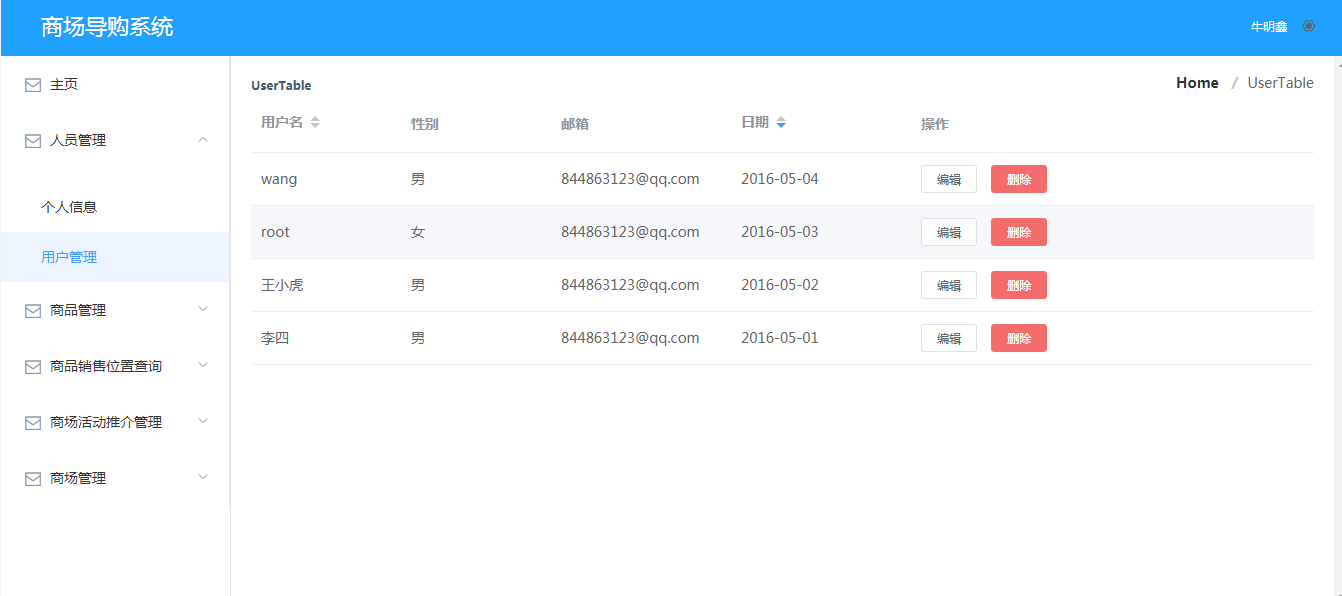


图4-14 用户管理信息界面

代码如下

router.post('/delete\_user', function(req, res) {

let params = req.body;

conn.query('select username from usersql where username="' + params.username + '"', function(err, rows) {

if(err || rows.length == 0) {

res.send({code: 0, ms: 'not exist'})

console.log(err);

} else {

conn.query('delete from usersql where username="' + params.username + '"', function(err, rows) {

if(err) {

console.log(err)

res.send({code: 0})

} else {

res.send({code: 1});

}

})

}

})

})})

### 4.4.4 商品信息管理/查询模块

管理员可以对数据库中以及存在的商品进行查看其详细信息，并且可以对商品进行管理，用户仅仅可以查看商品信息不可以对商品有其他的操作。如图4-15所示。



图4-15 商品信息管理/查询界面

代码如下

var models = require('../db/db');

var express = require('express');

var router = express.Router();

var mysql = require('mysql');

var $sql = require('../db/sqlMap');

var conn = mysql.createConnection(models.mysql);

conn.connect();

router.post('/goodsinfo', function (req, res) {

let params = req.body;

conn.query('select \* from goodssql', function (err, rows) {

if (err || rows.length == 0) {

console.log(err)

res.send({data: 0})

}else {

res.send({data: rows})

}

})

})

### 4.4.5 商场管理模块

商场管理模块，主要用户记录，以及进入驻该系统的商场，并且可以查询该商场所在的位置，了解该商场的主营业务。是否有需要去该商场进行购物。如图4-16所示。



图4-16 商场管理页面

代码如下

getInfo() {

var \_this = this;

this.$refs.ruleForm2.validate((valid) => {

if (valid) {

var url = '/api/marketinfo';

this.$http.post(url, {

},{}).then(function(data) {

// console.log(data)

let info = data.body.data;

if(data.body.data == 0 ) {

this.$message.error('信息为空');

} else {

\_this.tableData.name = info.name;

\_this.tableData.introduce = info.introduce;

\_this.tableData.phone = info.phone;

\_this.tableData.important = info.important;

\_this.tableData.address = info.address;

}

})

} else {

this.$message.error('faile);

return false;

}

});

}

### 4.4.6 商场活动推介模块

商场活动推介模块主要通过一个简短的视频，用来介绍这个某个商家的详细信息，然后相应的商品也会有所介绍，并且在视频旁边，会有相应的商品展示出来，用户可以选择自己自己中意的商品进行购买。如图4-17所示。

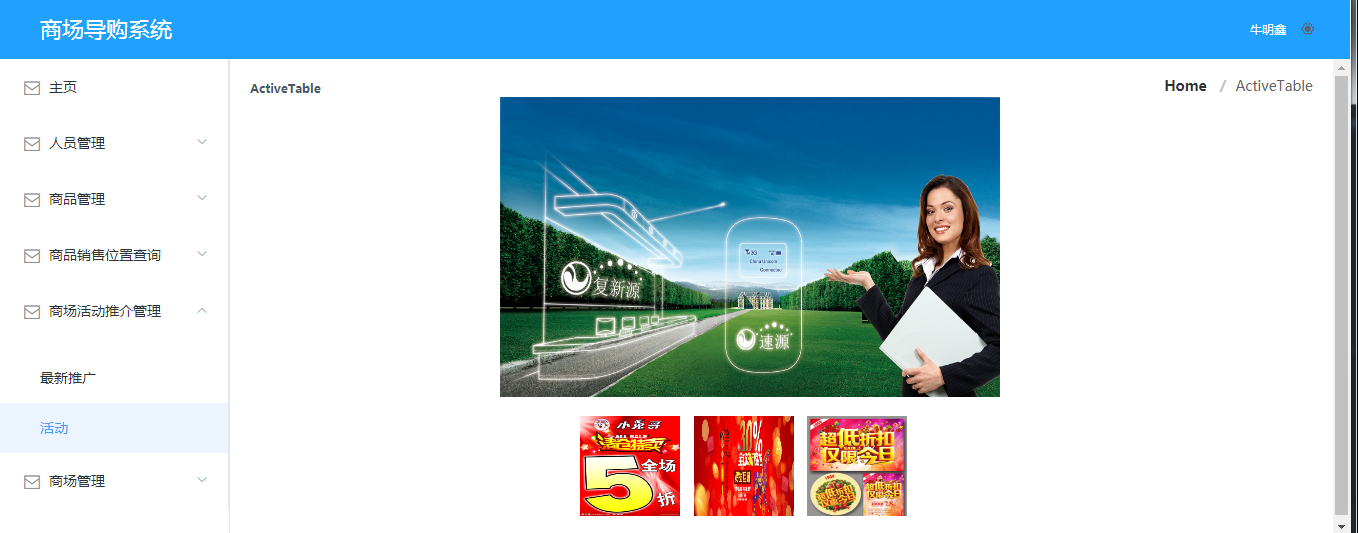


图4-17 商场活动推介页面

代码如下

<template>

<div>

<div class="">

<video src="`${video}`" alt="" class="video">

</div>

<div class="img" v-for="item in active" :key="item">

<a href="item.push">

<img :src="item.img" alt="">

</a>

</div>

</div>

</template>

# 第5章 软件测试与结果分析

软件测试是软件发布后不可或缺的一个环节，一个好的系统是必须可以经受的住的测试的。只有经过测试的系统才可以放心的让用户进行使用。软件测试必须要进行全面的考虑，把系统中所有的路径流向都要进行测试到。在现有的条件下，发现程序中出现的错误，是可以衡量软件质量的重要途径，并且判断该软件是否满足该需求分析要求的重要过程[14]。

## 5.1 测试的目的和原则

软件测试的目的是为了让软件在为投入使用之前及时的发现其中的错误，（当然发现错误不是唯一目的）。我们可以从用户和软件开发着角度来看软件测试。从用户角度看，看待该软件是否出现太多错误，来确定是否要使用该软件。从开发人员来看，软件测试是希望自己开发的产品出现较少的错误，可以极大的满足用户的需求，并且可以被用户所接受使用。通过测试还可以发现软件设计的不足，并且把这些不足进行改正，就可以让自己的软件更加完善。

本系统主要使用白盒测试，白盒测试又称结构测试和透明盒测试，逻辑驱动测试或基于代码的测试。白盒测试法的覆盖标准有逻辑覆盖、循环覆盖和基本路径测试。其中逻辑覆盖包括语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、条件组合覆盖和路径覆盖[15]。六种覆盖标准发现错误的能力呈由弱到强的变化：

1．语句覆盖每条语句至少执行一次。

2．判定覆盖每个判定的每个分支至少执行一次。

3．条件覆盖每个判定的每个条件应取到各种可能的值。

4．判定/条件覆盖同时满足判定覆盖条件覆盖。

5．条件组合覆盖每个判定中各条件的每一种组合至少出现一次。

6．路径覆盖使程序中每一条可能的路径至少执行一次。

## 5.2 测试计划

软件测试是开发软件必不可少的一个步骤，通过软件测试的软件，更能满足用户的使用。制定软件测试计划可以保证软件测试顺利的完成，软件测试计划表如5-1所示。

表5-1 系统测试计划表

|  |  |
| --- | --- |
| **目标** | 系统可以按照需求完成各个功能 |
| **内容** | 系统的登录与注册、商品管理模块、人员管理模块、商场管理模块、销售位置查询模块、打折促销管理模块 |
| **技术** | 白盒测试与黑盒测试，本系统主要应用白盒测试 |
| **前提** | 系统安装运行 |
| **完成标准** | 功能基本实现 |
| **特殊事项** | 权限分配 |

## 5.3 测试过程

因为本系统主要是使用的白盒测试，并且单元测试，集成测试，系统测试是白盒测试所包含的内容[15]。本次测试，主要是对该系统的基本需求进行测试。以下是测试的详细过程。

### 5.3.1 单元测试

单元测试主要针对系统中的较小的模块，来判断该模块的功能是否明确。通常是在某个特定条件下，进行测试该功能是否可行。通过单元测试，可以明确知道该段代码是否可以满足我们的期望。通过单元测试，每一个路径都进行执行一次。如图5-2所示。

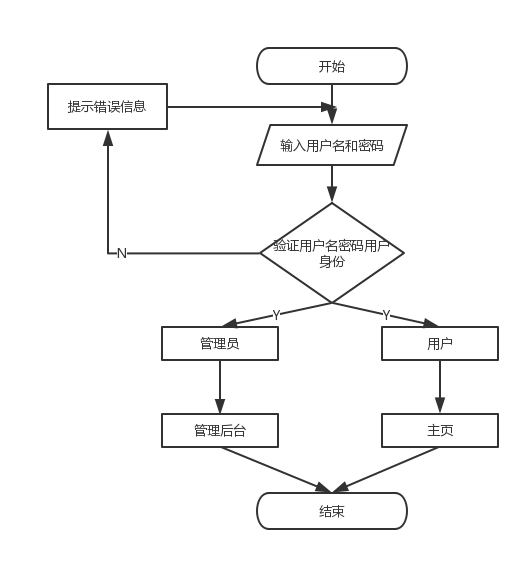


图5-2 登录模块流程图

对应的流图如图5-3示

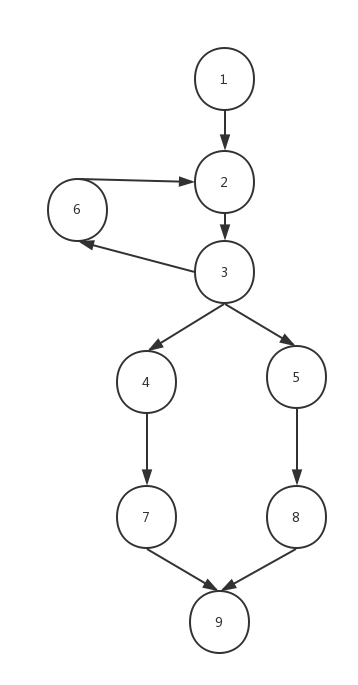


图5-3 登录模块流图

确定基本路径，可以确定如下几条基本的路径

路径1：1-2-3-4-7-9

路径2：1-2-3-5-8-9

路径3：1-2-3-6-2-3...

为以上的每一条路径设计测试用例，让程序至少可以执行一次。

1．路径1测试用例

输入正确账号和密码。判断为管理员

期望结果：登陆成功，进入系统管理首页。

2．路径2测试用例

输入正确账号和密码。判断为用户

期望结果：登陆成功，进入系统首页。

3．路径3测试用例

输入错误账号和密码。不进行判断

期望结果：登录失败，返回错误信息，重新输入。

### 5.3.2 集成测试

集成测试也称为组装测试或者联合测试。是在单元测试的基础上，将所有模块按照设计要求组装成子系统或者系统，进行测试。在把每个要测试的模块组合到一起时，随之可能出现的错误也许都是与每个单元模块之间的接口连接有着一定的关系，集成测试保证了信息在各个模块之间的畅通。在此基础上，即可进行接下来的系统测试。

### 5.3.3 系统测试

启动电脑windows7+操作系统，开启xampp启动mysql来保证可以连接到数据库，进入项目运行vue（npm run dev），运行server（node app.js），在浏览器中输入地址[localhost:8081/#/login（localhost](http://localhost/TourSystem/index.php/Index/index（localhost)为网站根目录，login为网站的登录页，8081为端口），访问本系统进行各项功能的测试。测试结果如图5-3所示

#### 5.3.3.1 注册功能测试

注册功能只可以针对用户进行注册管理员是admin进行分配的。该表如5-3所示。

表5-3 注册功能测试表

|  |  |  |  |
| --- | --- | --- | --- |
| **功能描述** | 用户输用户名和密码，通过访问数据库来进行验证 | | |
| **用例目的** | 验证用户注册功能是否符合系统的设计需求 | | |
| **用例ID** | **操作** | **预期结果** | **实际结果** |
| res\_01 | 直接点击注册 | 提示输入不能为空 | 提示各必填项不能为空 |
| res\_02 | 密码和验证不密码不相同 | 提示俩次输入不一致 | 提示密码输入不一致 |
| res\_03 | 不填写确认密码 | 提示密码输入不一致 | 输入不能为空 |
| res\_04 | 注册已经存在的用户名 | 提示用户名已经存在 | 提示用户名已经存在 |
| res\_05 | 填写完整的正确信息 | 提示注册成功 | 提示注册成功 |

#### 5.3.3.2 后端测试

进入服务器文件，通过node app.js 运行该后端，然后通过postman访问接口。通过对查询和修改，测试结果如表5-4所示。

表5-4 后端测试表

|  |  |  |  |
| --- | --- | --- | --- |
| **测试用例** | **操作** | **预期结果** | **实际结果** |
| 访问getinfo接口 | 直接提交 | 可以得到数据数据库中数据 | 得到数据库中数据 |
| 不填写数据访问updinfo | 直接提交 | 提示不能为空 | 添加要修改的数据 |

#### 5.3.3.3 整体测试

在后端代码测试完整之后，然后进入前端项目，通过npm run dev启动。然后，前后端进行结合，对整体的一系列功能进行测试。测试结果如表5-5所示。

表5-5 整体测试表

|  |  |  |  |
| --- | --- | --- | --- |
| **测试用例** | **操作** | **预期结果** | **实际结果** |
| 登录/注册 | 填写完整/正确信息 | 可以进行提交并且实现 | 实现登录注册 |
| 人员管理 | 访问人员/删除人员 | 管理员可以访问和删除/用户不能 | 和预期一致 |
| 查询商品 | 输入数据库中包含的 | 显示商品信息 | 准确显示 |
| 访问最新消息 | 直接访问主页 | 最新活动以及打折活动 | 可以完整显示 |

## 5.4 测试结论

通过单元测试对每一个模块中逻辑，进行修改完善，直到逻辑是可以正确运行的。通过集成测试修改系统中遇见的每一个较大的错误，直到系统可以使用。本系统从选题到测试结束已经完成了毕业任务书的要求。从我个人来开，基本上达到了我预期的目标，从用户角度来看，也可以满足用户的基本要求，相对来说该系统在之后的维护也比较方便。因此此次开发也是比较成功的。但是作为一个优秀的程序员，在以后的开发过程中，要不断学习，不断开发，不断测试。

# 结 论

Web商场导购系统，一方面是为了满足消费者在大型商场中购物的方便，可以及时的让消费者找到自己心仪的商品，另一方面，可以让商场管理人员，可以很方便的对商品进行管理，对打折促销消息可以及时的传递给用户，让用户可以第一时间，寻找商品。该系统在商场中应用场景十分广泛并且可以很大发挥其实用和实际的价值，

在商品推广方面节约了，人力物力，相对于传统的方式，该系统不仅保护了环境，还打破了传统的习惯，有利于推动科技的进步。此外，该系统使用起来比较简单，并且版本很稳定，系统界面友好。该系统在VSCode下进行开发，吧数据存储在mysql中，具有安全性，符合设计的基本要求。

总体来说，通过本次毕业设计，对我个人软件设计上来说有了很大的提高。本次毕设从选题到设计都经过了多次的思考。在设计之初我在网上和图书馆查阅了很多资料，并且询问一些消费者和大型商场中的人员，最终确定该系统所需要存在的几大模块，然后在把该系统的各个模块进行细化，并且完善到需求分析和概要设计上，把这些工作做好之后，然后再进行前端页面的设计，页面完成后，就要进行代码开发。所有的代码，都是通过之前的积累，以及平常的学习，进行敲上去的。虽然在开发过程中遇见了很多问题，但这些问题并不能阻挡我前进的步伐，每次遇见问题，我都会询问同学，咨询老师，网上查阅资料。每一次的解决问题，都让我更清楚的认识自己的优点和缺点，并且取长补短，让自己有更大的提高。此次毕设，让我更清楚的认识到，团队精神和沟通学习能力是必不可少的一项技能。

虽然本次毕业时间比较短暂，但我却学到了很多东西，但是我还能感觉到个人还有许许多多有待于提高和进步的地方，商场导购系统，还需要后期不断的完善。从软件的开发到发布，测试是不可缺少的一个环节，但是系统不可能做到十分完美，所以需要我们不断的学习，并且利用新的知识来完善该系统。但是完善和维护该系统的同时，要考虑到维护该系统的成本以及商业价值。由于该系统是本人自己设计和开发的，出于个人学习的考虑，会对该系统不断的完善。

商场导购系统是一个很好的开发项目，具有很高的社会实用价值，虽然这个系统是一个毕业设计，但也存在很多不足之处。因此不断的学习，才能让自己提高。

# 致 谢

通过web商场导购系统的开发，让我学习到了很多知识，并且让我更加清楚的

认识了软件开发过程的思想。在此我诚挚的感觉大学无私奉献的老师，以及给予我帮助的同学。因为有了他们，我才能走的更高，看的更远。

商场导购系统是在贺薪宇老师指导下完成的。从该毕设的选题到完成本次毕业论文，贺薪宇老师给予了我无私的帮助以及悉心的指导，从一个细小的问题，再到比较严重的问题，贺薪宇老师都非常耐心的一遍一遍的指导我毕设和论文，才最终有了近乎完美的毕设和论文。贺薪宇老师从大一就是我的老师。并且她是一位非常年轻优秀的老师，她知识渊博，严谨负责，认真对待每一件事。正是因为这种优秀的品质深深的影响了我。让我认真学习每一门学科，充分的掌握理论知识。于此同时，也要扩宽自己的视野，提高自己分析问题和解决问题的能力，同时也提高自己的创新精神。除此之外，我同学，也在毕业设计和论文上给予了我很大的帮助，由于实习的方向不一样。在测试的时候，要请教实习过测试的同学，该如何进行完美的测试。在调试代码的时候，同学也会帮助你。正因为有了他们的帮助，才让我的程序，能够完善。在此，要深深的感谢我的同学。通过本次论文，让我学习了很多知识，也增加了很多经验。

最后，再次深深的感谢我的同学和老师。

# 参考文献

[1] 姚禕炜. 《购物广场互动导购》解决方案[J].科技风，2013，16期.

[2] 田阳，王成东，崔嵩. 项目可行性研究[M].哈尔滨工业大学出版社，2015-06，4-80.

[3] 李杰，柳靖，刘淼. HTML5高级程序设计[M].人民邮电出版社，2011-1-1，3-50.

[4] 林小志.CSS那些事儿[M].电子工业出版社[M]，2009-10，6-59.

[5] 杨涛，王建桥，杨晓云.JavaScript DOM编程艺术[M]，2006-12，6-65.

[6] 朴灵. 深入浅出Node.js[M]. 北京：人民邮电出版社，2013-12-01，1-150.

[7] 未知. Express 4.x Node.js的Web框架[Z]. 论文库.

[8] 明日科技. Java从入门到精通[M]. 北京：清华大学出版社，2016-10-1，2-50.

[9] 李晓明. C#与. NET 4高级程序设计[M]. 北京：人民邮电出版社,2011，5-200.

[10] 宁海元，周振兴，彭立勋，翟卫祥，刘辉. 高性能MySQL[M]. 北京：电子工 业出版社，2013-5-1，1-150

[11] 程显峰. MongoDB权威指南[M]. 北京：人民邮电出版社，2011-5-1，1-50.

[12] 戴志诚. SQL Server 2005 数据库系统开发与实例[M]. 北京：电子工业出版社，2011，3-60.

[13] 冀振燕. UML系统分析与设计教程[M]. 北京：人民邮电出版社，2014-8，20-150.

[14] 王玉玺. 软件测试──提高软件可靠性的关键手段航空计算技术[M]. 北京：电子工业出版社，2011，1-70.

[15] 佟伟光，郭菲菲. 软件测试[M]. 北京：人民邮电出版社，2015-1，37-100.

[16] 方木云，刘辉. 高级软件工程[M]. 北京：清华大学出版社，2011-6.

[17] Shelley Powers. Learning Node[M]. O'Reilly Media,2012-7.

[18] Addy Osmani. Learning JavaScript Design Patterns[M]. O'Reilly Media,2012-8-22.