

星巴克会员管理系统项目建议书

项目建设单位：	北京航空航天大学
编制日期：	2020 年 12 月 19 日
负责人：	姬轶-ZY2006109

目录

1 . 项目简介	1
1.1. 项目名称	1
1.2. 项目概况	1
1.3. 主要结论和建议	1
2 . 引用文档	2
3 . 需求分析	2
3.1. 系统场景设计	2
3.2. 功能需求分析	2
3.3. 非功能需求分析	4
3.4. 用例分析	4
3.5. 业务流程分析	6
3.6. 数据流程分析	6
4 . 本期项目建设方案	8
4.1. 系统架构设计	8
4.2. 系统功能结构	9
4.3. 功能模块设计	10
4.4. 数据库设计	12
4.3.1. E-R 图	12
4.3.2. 表结构设计	13
5 . 技术比对	16
5.1. 前后端分离	16
5.2. JSP 技术	17
5.3. B/S 模式	17

1. 项目简介

1.1. 项目名称

星巴克会员管理系统(Starbucks Member Management System, SMMS)

1.2. 项目概况

当今社会,随着 Internet 的发展,网上购物已经成为了一种新的购物流念,人们可以足不出户便可以购买所需物品。随着"互联网+"、轻量级框架、分布式、高并发高可用等概念的提出,使得 Web 应用的开发技术得到了进一步的推进及深入发展,对星巴克会员管理系统的设计、实现、管理产生了深远的影响,并在星巴克会员管理系统的开发中得到了广泛的应用。电子商务通过营造一个虚拟市场环境,使贸易双方没有时空障碍,从而增加贸易机会,降低交易成本,改善服务质量,提高商务活动效率。互联网和电子商务的全面整合不仅极大地丰富了各类产品的销售渠道,而且还提升了企业进行改革与转型的效率,推动了企业的发展步伐。然而,由于电子商务在"互联网+"时代并未经历较长的发展阶段,目前还不够成熟,所以在此程中仍存在很多问题,只有尽快发现这些问题,并及时采取针对性的解决措施,才能使电子商务在未来取得更好的发展。

在信息飞速发展的今天,网络已成为人们重要的信息交流平台。星巴克每天都有大量的信息需要通过网络发布,总体目标拟开发一个基于 B/S(浏览器/服务器)模式的星巴克会员管理系统。系统采取组件化的方式对系统进行拆分,并对数据库中各个表的增删查改、表与表之间的约束关系进行分析与设计,最终实现符合用户需求功能的商业级应用。

对应用户的每一步操作,系统都能高响应地做出反馈。系统具体包含对用户的管理、登录注销、商品发布、商品维护、订单结算、订单维护等功能,为星巴克提供一个管理平台,实现对数据地高效、安全地管理。

1.3. 主要结论和建议

我国互联网发展的起步较晚,目前我国内的网络商城的结算依旧采用 B2C 结算。我国网络上出现的首批网络商城服务到现在不过才二十多年。但是近年来我们国家计算机相关行业的迅猛发展,其中网络商城是其中的佼佼者。线上购物逐渐赶上甚至超过了线下购物,成为我国消费者的主要消费方式,同时网络商城

的发展也逐渐改变我们的消费方式、生活习。是我们国家未来消费方式的趋势。

2. 引用文档

本章应列出本文档引用的所有文档的编号、标题、修订版本和发行日期，也应标识不能通过正常的供货渠道获得的所有文档的来源。引用文档应包括：

- (1) 项目任务书
- (2) 其它文档（如设计文档应引用需求文档）

3. 需求分析

3.1. 系统场景设计

系统应用场景如图 3-1 所示，以系统为中心对外提供输入输出服务，响应数据读写请求。系统使用用户分 2 种，使用购买功能的为普通用户，使用管理功能的为管理员用户，普通用户可以浏览到系统展示的商品，然后进行商品购买，接着形成个人购买订单；管理员用户对系统的数据进行维护，包括商品信息的维护，对商品的数量、价格等进行更新和编辑，对客户的订单进行确认和维护等。

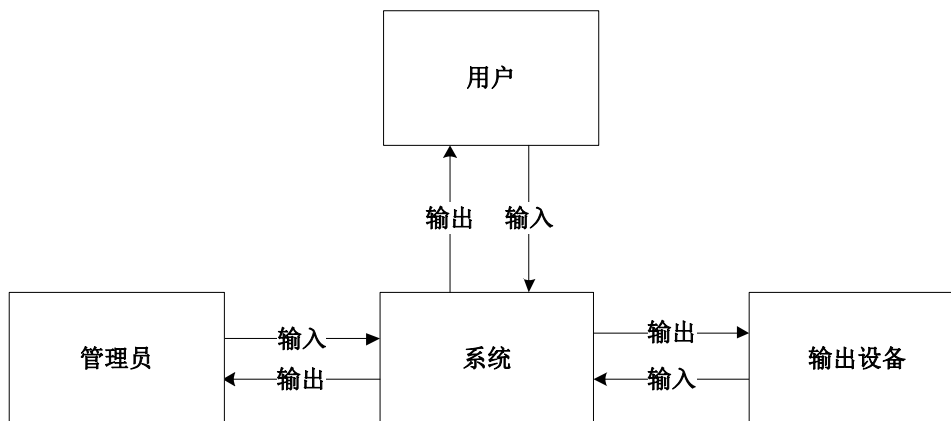


图 3-1 系统场景图

3.2. 功能需求分析

1.注册：首先检查用户名和邮箱是否存在，如果都不存在才能进行注册，并将用户

输入的密码进行 MD5 加密，注册成功后提示注册成功，并将用户信息保存到数据库中。

2.商品展示功能模块主要包括产品搜索、查看商品详情功能。未注册用户（游客）和已注册用户均可以在首页进行产品搜索及查看商品详情。具体功能需求如下：

（1）产品搜索：前台用户可以在前台输入关键词进行搜索，搜索成功后返回该商品的动态排序列表，且能够进行分页展示。

（2）查看商品详情：用户可点击对应商品，从而查看该商品的商品详情。

3.购物车管理功能模块主要包括加入购物车、更新购物车商品数量、删除购物车商品、

加入购物车：购物车是存放用户想要下单的商品，包括商品、下单数量和总金额。

（2）更新商品数量：用户可在查看购物车时更新所需商品的数量。

（3）全选/单选：用户可全选/单选所需商品，系统将对所选中的商品计算总价。

4.收货地址管理：

（1）新增收货地址：

（2）删除收货地址：

（3）更新收货地址：

5.在线支付：

（1）提交订单：用户提交订单后会自动跳转到支付页面，支付页面显示订单金额和二维码。

（2）支付：用户可扫描二维码进行支付。

（3）返回订单状态：付款完成后将订单状态改为已付款状态，返回订单页面。

6.商品管理：

（1）商品列表：展示所有商品信息，形成的列表样式。

（2）商品搜索：通过商品名、商品类目来搜索商品。

（3）图片上传：给商品添加商品图片。

（4）增加商品；

（5）更新商品：更新商品的信息，如商品名、商品图片、价格、数量等。

(6) 商品上下架;

7.商品类目管理:

(1) 添加品类: 管理员可对品类进行管理, 点击添加品类选择父类并输入品类名称即可添加品类。

(2) 修改品类: 管理员可对品类的名称进行修改。

(3) 查看子品类: 管理员可查看所有平级品类的品类 ID, 品类名称等。

8.订单管理:

(1) 按订单号查询订单:

(2) 查看订单详情:

9.资讯管理:

(1) 资讯列表: 展示所有资讯信息, 形成的列表样式。

(2) 资讯搜索: 通过资讯标题、资讯编号来搜索资讯。

(3) 图片上传: 给资讯添加资讯图片。

(4) 增加资讯;

(5) 更新资讯: 更新资讯的信息, 如资讯标题、资讯图片、内容等。

(6) 资讯上下架;

3.3. 非功能需求分析

系统非功能需求有非常多, 比如性能需求、可承载最大用户数、稳定性、易用性需求等。本系统分析时考虑到易用性需求, 因为系统是给人使用的, 所以必须充分从用户的角度出发, 考虑用户体验, 使系统易理解易上手易操作。

3.4. 用例分析

1.普通用户主要使用系统的在线购买业务, 下图所示为普通用户的用例图。

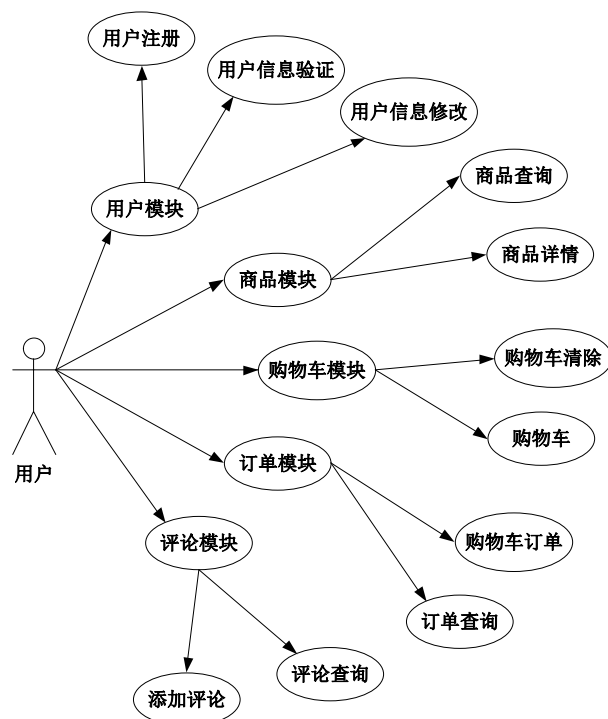


图 3-2 普通用户用例图

管理员主要负责系统管理、商品信息管理、商品类别管理、用户信息管理、订单管理，下图所示描述管理员的用例图。

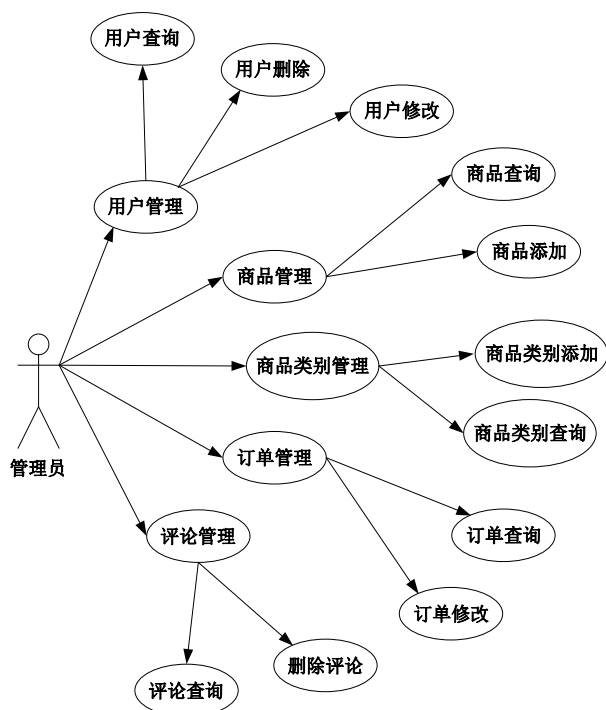


图 3-3 管理员用例图

3.5. 业务流程分析

系统业务流程图如下所示。

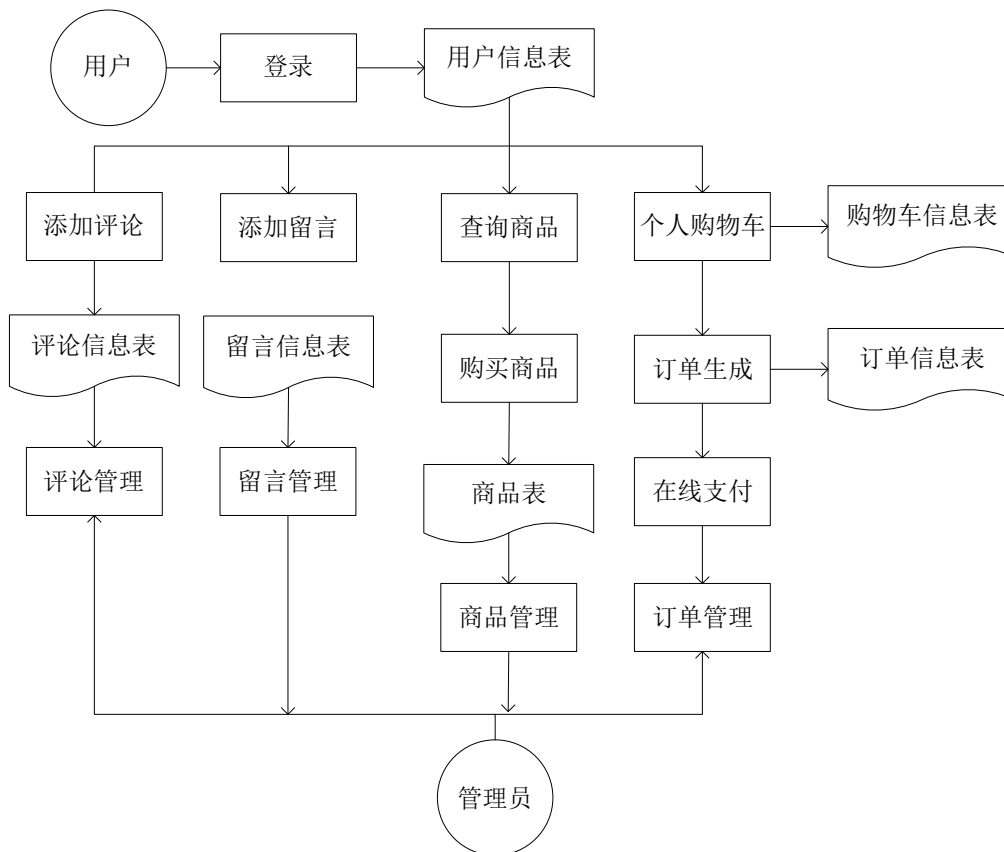


图 3-4 系统业务流程图

首先系统使用用户分 2 种，使用购买功能的为普通用户，使用管理功能的为管理员用户，普通用户注册成功后得到账号和密码，在登录页输入账号密码后登录到系统，普通用户便可以浏览到系统展示的商品，然后进行商品购买，接着形成个人购买订单；管理员用户对系统的数据进行维护，包括商品信息的维护，对商品的数量、价格等进行更新和编辑，对客户的订单进行确认和维护等。

3.6. 数据流程分析

如下为系统的顶层数据流图，外部实体有用户和管理员，分别对系统输入数据，从而得到要输出的数据。

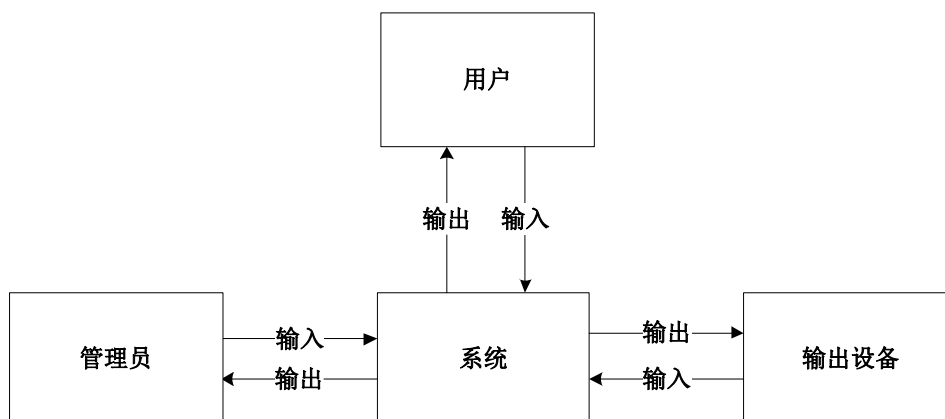


图 3-5 系统顶层数据流图

如下图所示为系统底层数据流图。

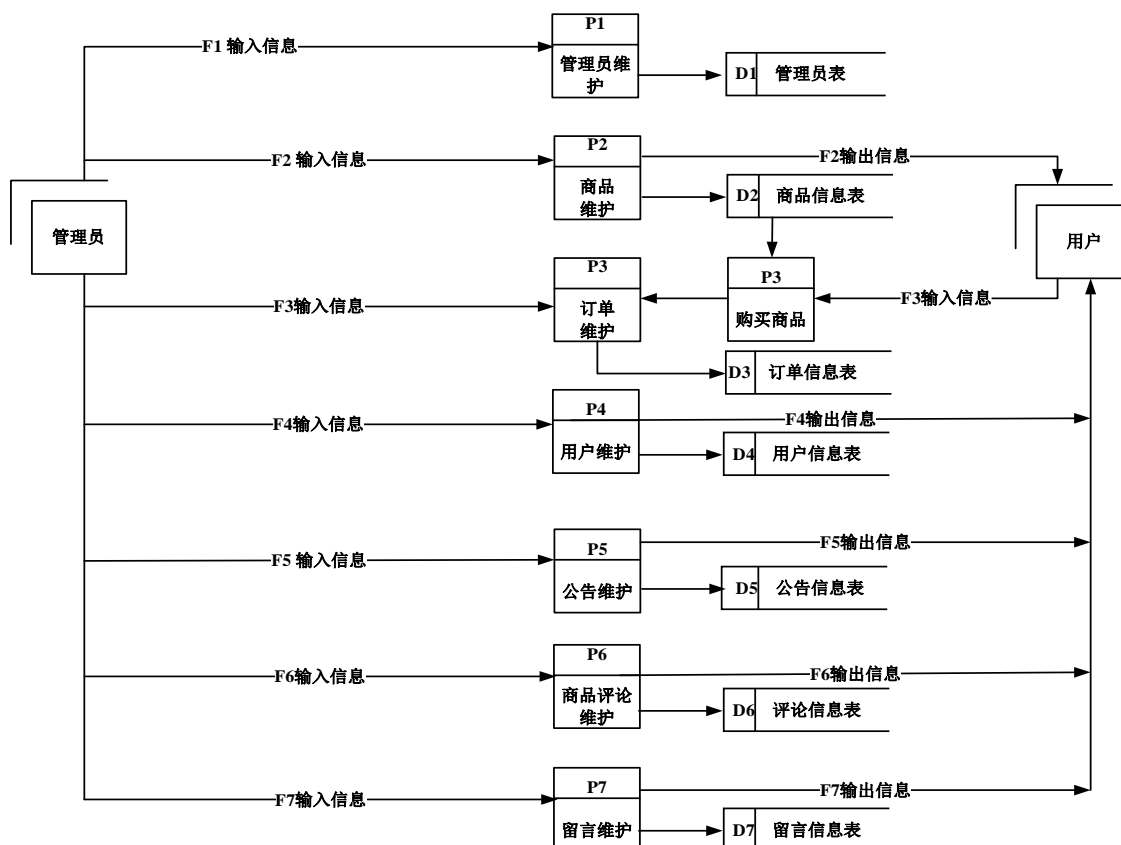


图 3-6 系统底层数据流图

星巴克会员管理系统底层数据流图主要描述系统底层模块的数据流、数据载体、数据处理、外部实体的具体交互。本系统就是管理员和普通用户在商品维护、订单维护、用户维护等模块上的数据交互。

4. 本期项目建设方案

4.1. 系统架构设计

本系统采用 MVC 结构，结构包括了 View、Controller、Service、ServiceImpl、Dao 等模块。系统架构图如下图所示。

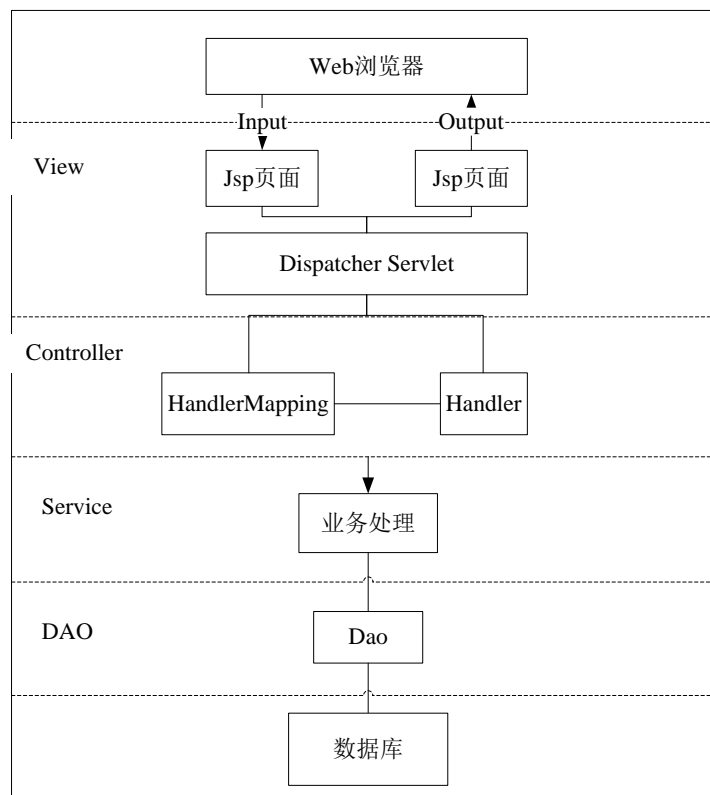


图 4-1 系统架构图

根据本系统需求分析，应为系统搭建多个包，分别为 Controller、Service、ServiceImpl、Dao、Entity、Util。

现对上述的包进行简要的功能阐述，Controller 层负责接收前台传来的请求并对取到的数据进行封装。Service 层主要负责处理业务逻辑。Dao 层负责数据库方面的交互。Entity 包放置实体类，对应数据库里面一张张的表。Util 包放置工具类，比如以 json 数据的格式返回前台的 JSONUtil 类，自动生成 id 的 GenerateID 类等。通过 applicationContext.xml，applicationContext-dataSource.xml，web.xml 这三个配置文件来搭建本系统。

首先配置 `applicationContext.xml`，主要是自动扫描控制器，视图模式，注解的启动。

`applicationContext.xml` 是 `springMVC` 的一些相关。

4.2. 系统功能结构

为了将系统从“做什么”落实到“怎么做”，基于系统设计原则，对系统功能进行更适合编码实现的功能划分。根据上图的系统架构设计，整理出规范的系统功能结构图，为系统的实现编码做好准备。

如下图所示为系统功能结构图。

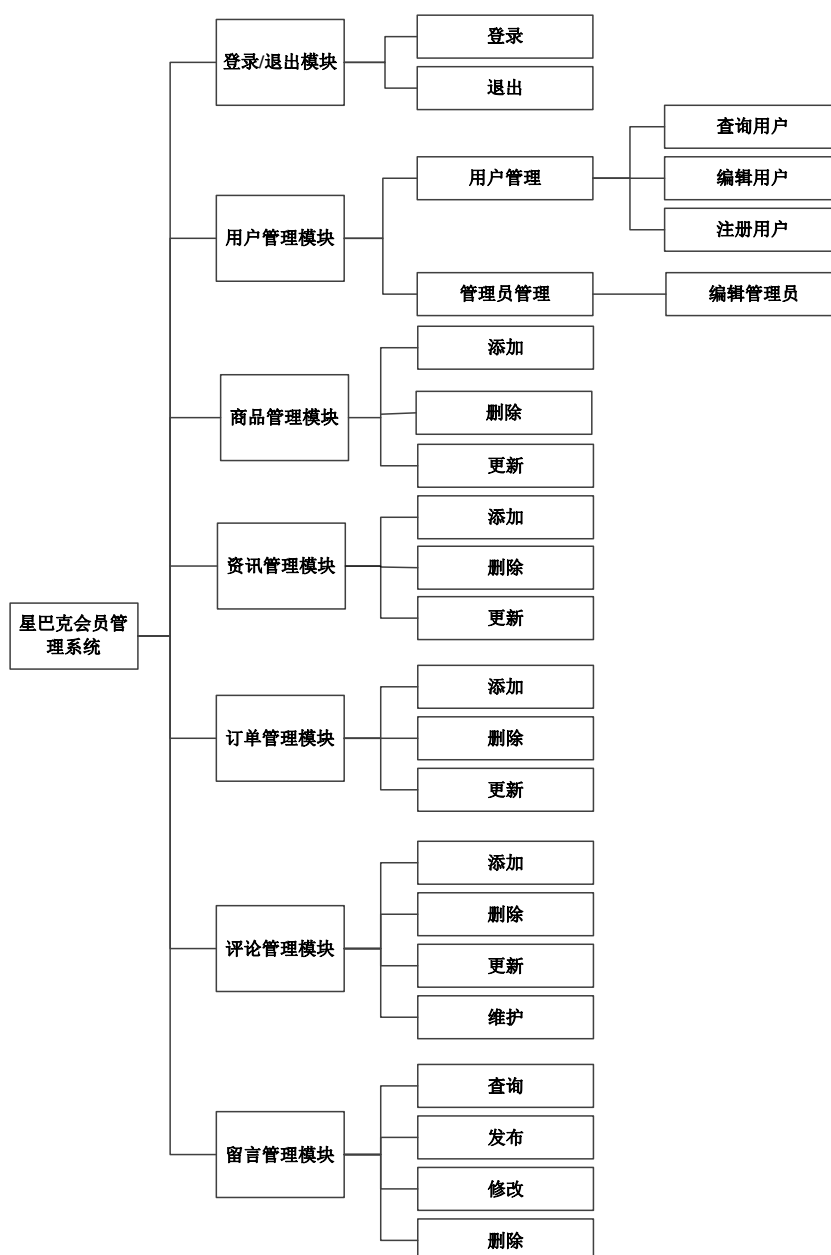


图 4-2 系统功能结构图

(1) 用户信息的管理：系统管理员可以对已经注册过系统的用户数据信息进行 CRUD 操作。

(2) 商品数据信息资源管理：管理员可以对系统的商品信息进行增删改查的基本操作。

(3) 显示商品分类信息：所有商品的类别都会显示在首页中，用户可根据这些类别快速的找到自己喜欢的商品。

(4) 下单和购物车功能：用户对自己喜欢的商品资源，在登录的前提下可以自己购买或者先添加到购物车之后一块购买。

(5) 搜索商品：物品很多时，用户不容易查找，因此提供了商品搜索功能，可以通过物品标号对商品进行查找。

4.3. 功能模块设计

用户管理针对所有用户和管理员。未注册用户点击注册，进入注册页面填写新用户信息，得到自定义生成账号后，注册完成。未注册用户注册的活动图如下图所示。

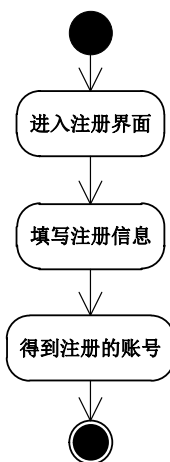


图 4-3 未注册用户注册活动图

用户可进行登录使用更多功能，首先进入用户登录界面，输入用户账号和密码，后台对账号和密码信息进行核对验证，验证成功则页面直接显示登录用户昵称代表登录成功，否则返回用户登录界面。用户登录活动图如下图。

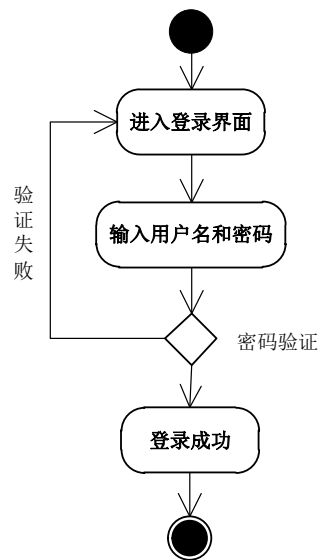


图 4-4 用户登录活动图

用户登录后，进入功能页面，可对自己的信息进行管理，包括个人信息，个人信息，购物车管理、个人订单管理、在线支付。用户使用功能活动图如下图所示。

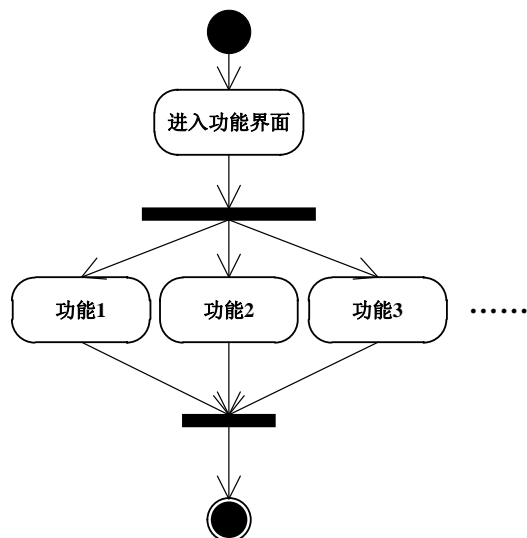


图 4-5 用户使用功能活动图

用户功能界面可对自己的密码进行修改。进入修改密码界面后核对当前密码，核对成功后填写新密码。如下图所示。

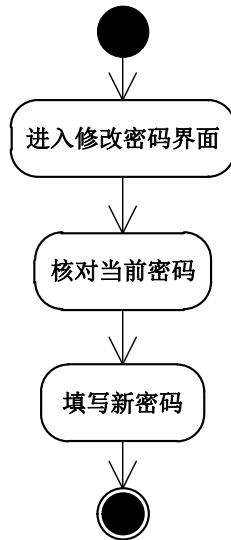


图 4-6 修改密码活动图

4.4. 数据库设计

星巴克会员管理系统采用的数据库是 Mysql2008，该部分将根据系统需求 and 设计，设计合理的数据库。本系统考虑到之后系统的扩展集群，数据库将不使用外键。此外，不用外键时数据管理也较为简单，操作更加方便，性能更高。

4.3.1. E-R 图

E-R 图，更加直观的告诉开发人员系统的各个数据的属性，各个数据之间的联系，各个数据的类型。能通过该模型更直观地了解数据库的设计，并根据对其逐渐改善。本系统图设计，如下图所示。

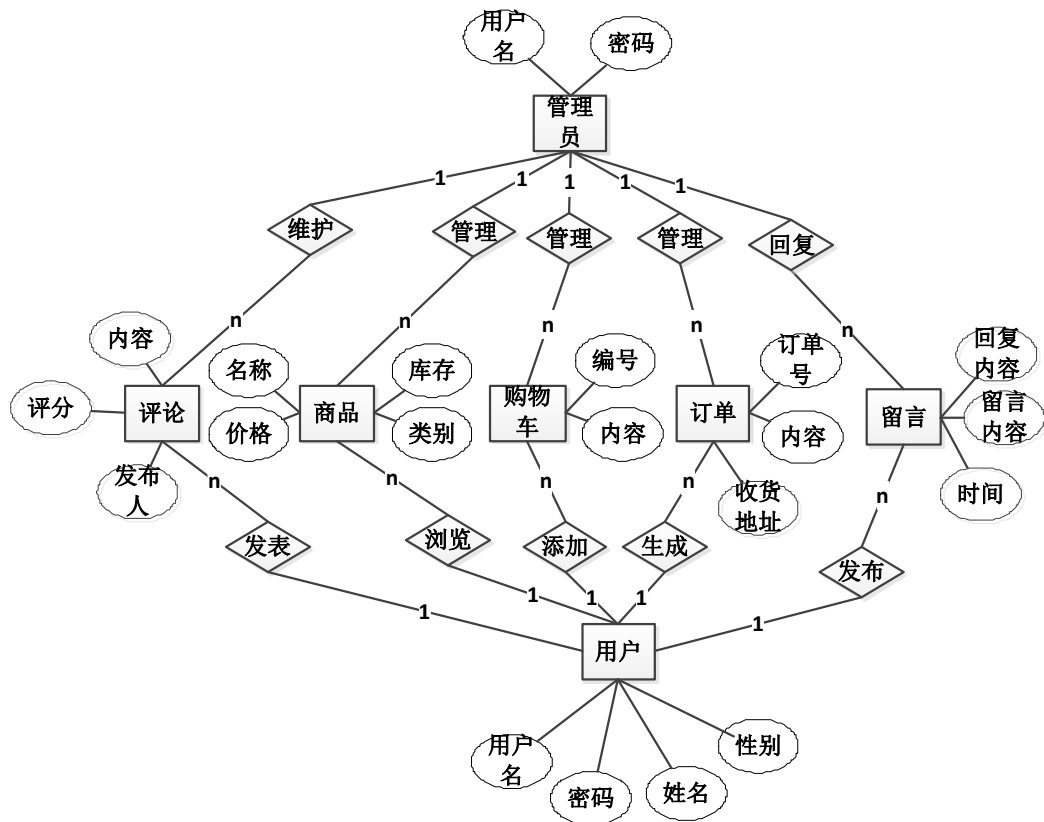


图 4-7 系统 E-R 图

4.3.2. 表结构设计

本系统根据第三范式的思想设计数据库，减少最大程度的冗余，总共建立有多个表，用户表、管理员表、商品表、订单表、购物车表、评价表。

表结构如下所示。

allusers 表:

序号	字段名称	字段类型	大小	允许为空	最大长度	备注
1	ID	Int	4		10	
2	username		150		255	
3	pwd		150		255	
4	cx		150		255	
5	addtime	DateTime	8		19	

dingdanxinxi 表:

序号	字段名称	字段类型	大小	允许为空	最大长度	备注
1	ID	Int	4		10	

2	dingdanhao		150		255	
3	dingdanneirong		1073741823		255	
4	zongjine		150		255	
5	goumairen		150		255	
6	xingming		150		255	
7	shouji		150		255	
8	shouhuodizhi		1073741823		255	
9	beizhu		1073741823		255	
10	issh		6		255	
11	iszf		6		255	
12	addtime	DateTime	8		19	

gouwuche 表:

序号	字段名称	字段类型	大小	允许为空	最大长度	备注
1	ID	Int	4		10	
2	shangpinbianhao		150		255	
3	shangpinmingcheng		150		255	
4	leibie		150		255	
5	jiage		150		255	
6	goumaishuliang		150		255	
7	zongjine		150		255	
8	goumairen		150		255	
9	beizhu		1073741823		255	
10	issh		6		255	
11	addtime	DateTime	8		19	

liuyanban 表:

序号	字段名称	字段类型	大小	允许为空	最大长度	备注
1	ID	Int	4		10	
2	cheng		150		255	
3	xingbie		6		255	
4	QQ		150		255	
5	youxiang		150		255	
6	dianhua		150		255	

7	neirong		1073741823		255	
8	addtime	DateTime	8		19	
9	huifuneirong		1073741823		255	

pinglun 表:

序号	字段名称	字段类型	大小	允许为空	最大长度	备注
1	ID	Int	4		10	
2	xinwenID		150		255	
3	pinglunneirong		1073741823		255	
4	pinglunren		150		255	
5	pingfen		150		255	
6	biao		150		255	
7	addtime	DateTime	8		19	

shangpinleibie 表:

序号	字段名称	字段类型	大小	允许为空	最大长度	备注
1	ID	Int	4		10	
2	leibie		150		255	
3	addtime	DateTime	8		19	

shangpinxinxi 表:

序号	字段名称	字段类型	大小	允许为空	最大长度	备注
1	ID	Int	4		10	
2	shangpinbianhao		150		255	
3	shangpinmingcheng		150		255	
4	leibie		150		255	
5	jiage		150		255	
6	tupian		150		255	
7	xiaoliang		150		255	
8	shangpinjianjie		1073741823		255	
9	issh		6		255	
10	addtime	DateTime	8		19	

yonghuzhuce 表:

序号	字段名称	字段类型	大小	允许为空	最大长度	备注
----	------	------	----	------	------	----

1	ID	Int	4		10	
2	yonghuming		150		255	
3	mima		150		255	
4	xingming		150		255	
5	xingbie		6		255	
6	chushengnianyue		150		255	
7	QQ		150		255	
8	youxiang		150		255	
9	dianhua		150		255	
10	shenfenzheng		150		255	
11	touxiang		150		255	
12	dizhi		1073741823		255	
13	beizhu		1073741823		255	
14	addtime	DateTime	8		19	
15	issh		6		255	

5. 技术比对

5.1. 前后端分离

Ajax 的出现，也带来了另外一个问题，那就是有了 Ajax 以后，之前用模板语言实现起来的功能变得简单，之前模板语言实现不了的功能现在也能实现了。这样就造成越来越多的逻辑转移到了 javascript 上，使其变得越来越复杂。

随着 js 复杂度的增长，原来的开发模式出现了问题，一个程序员搞定全站变得越来越不靠谱，因此在这个时候就把网站开发这个职位划分成了前端和后端两个职位。但是只划分了前后端的职责范围还是远远不够的，在原来的开发模式下，前后端的代码也在一起的。现在既然已经分为前后端两波人在开发了，维护同一套代码就变得不那么方便。项目越复杂，出现你等我，我等你的情况就会越来越多，这样就拖慢了整体团队的节奏。所以为了团队的效率，前后端的代码也要做分离。

前后端的分离方式分为部分分离和全部分离两种，部分分离是只把脚本和样式分离出去，而 html 模板还留在后端通过 jsp，velocity 或者 freemarker 来渲染；另一种就是完全分离，脚本样式以及模板全都放在前端来维护。

部分分离已经很大程度上解决了前后端开发时的协调问题,开发效率也得到了很大程度的提升。但也得承认,这种方式也还是有问题的。当要开发 html 模板的时候,就需要搭起一整套后端的开发环境,或者是找后端程序员来协助。

而完全分离一般有两种方案,第一种就是使用 velocity 这种在 nodejs 和 java 下都可以编译的页面模板,在开发时放到前端项目里,但在发布时,会把模板发布到后端的模板目录下,这样,开发时就做到了完全分离。这种方式最大的好处就是线上模板的渲染还是由 java 来做的,形成的是带有动态数据的 html,比较有利于 SEO。但这种方式下,前端的开发环境和发布系统的复杂度都相对较高,单纯的前端改动也还是要带着后端一起发布。

第二种完全分离的方式,就是把纯静态的 html 模板完全放在前端,数据全部通过 RESTful 接口来进行交互。这样前后端就完全分开了,脱离了后端的模板,而这种方式系统复杂度也会比第一种完全分离的方式低。但这种方案下,所有的页面数据都是用 js 渲染的,没有动态模板,不太利于 SEO。这个不足我们可以通过做 server render 或者给蜘蛛做一套定制页面来解决。

5.2. JSP 技术

JSP 技术对动态 Web 页面的开发作用简直是举足轻重。可以很快的响应到客户端的发送请求。是甲骨文公司旗下的 IT 及互联网技术服务公司 Sun Microsystems 公司主导并创立的动态网页技术的标准。而且能依据请求内容动态地生成 XML、HTML,为用户的网络请求提供技术服务,而且可以与服务器上的其它 Java 程序共同处理先对复杂的业务需求。

JSP 主要优势如下:

- (1)一旦程序有一次成功的编写,就能在多处运行起来。
- (2)支持面特别广,许多平台已经引入该技术。

5.3. B/S 模式

计算机的系统软件主要分为两种:系统软件和应用软件。应用软件则是为了某个特定的应用而开发的一类软件,比如文字处理软件 word、QQ、开发本软件用的 myeclipse 工具等。而应用型软件大致的分为 B/S 和 C/S 两种模式,常用的 QQ、Office 办公软件就是 C/S 结构的,另外常见基于的 B/S 结构开发的有各种网站,比如淘宝、企业的门户网站。

B/S 模式的优势有三个:第一是开发和业务扩展简单、只需要改变网页代码,就可以达到想要的网页效果;第二学生的操作显浅易懂;第三具有分布性特定,可以随时随地地进行查询、浏览等业务的处理。这是 C/S 所无法实现的。综上因素,选择 B/S 模式来设计和实现本系统。