

成 绩: _____
教师签名: _____



湖北工业大学工程技术学院
HUBEI UNIVERSITY OF TECHNOLOGY ENGINEERING AND TECHNOLOGY COLLEGE

需求分析说明书

项目名称: 美博城——数据库设计、ER 用例图、我的痕迹、猜你喜欢

教 师: 胡峡进

姓 名: 唐可寅

学 号: 201710253110

班 级: 17gb 计算 3 班

日 期: 2020 年 11 月 29 日

目录

课程设计任务书.....	I
一、 需求分析.....	1
1. 任务概述.....	1
1. 任务描述.....	1
2. 运行环境要求.....	1
3. 前言.....	2
2. 需求功能介绍.....	2
1. 客户界面部分.....	3
2. 管理界面部分.....	4
二、 系统设计.....	5
1. 系统体系结构设计.....	5
1. 项目环境和依赖.....	5
2. 分层分析.....	5
2. 数据流图.....	10
3. 数据库设计与实现.....	11
1. 数据库概念结构设计.....	11
2. 数据库逻辑结构设计.....	12
3. 数据库的设计创建.....	12
三、 应用程序设计.....	15
1. 设计页面.....	15

2. 程序代码.....	20
3. 运行结果.....	20
四、 小结.....	21

课程设计任务书

2017 年级本科计算机科学与技术专业 3 班

题目：美博城——数据库设计、ER 用例图、我的痕迹、猜你喜欢

课程设计与要求：

课程设计要求：

课程设计要求：

课程设计开始日期 2020 年 11 月 23 日 指导老师 胡峡进
课程设计完成日期 2020 年 11 月 29 日

一、需求分析

1. 任务概述

1. 任务描述

本次课程设计要求完成电子商务系统——美博城网上购物系统设计。美博城是中国 B2C 市场的最大的 3C 网购专业平台是中国电子商务领域最受消费者欢迎的最具影响力的电子商务网站之一，目前该网站主要实现以品牌专区，特价专场，关于我们，为主打的特色购买平台，该网站主要完成的是后台商品的数据的 CRUD 操作和个人我的订单相关模块。

本次课程设计报告完成如下模块功能：

a. 数据库设计

数据库表设计。

b. 系统 ER 图、用例图设计

c. 猜你喜欢

利用之前用户浏览的商品的痕迹进行相关的产品推荐。

d. 我的痕迹

对我在该网页中浏览的商品进行痕迹的记录。

本项目设定的目标如下：

- a. 系统能够提供友好的用户交互界面，使操作人员的工作量最大限度的减少；
- b. 系统具有良好的运行效率，能够得到提高生产率的目的；
- c. 系统应有良好的可扩充性，可以容易的加入其它系统的应用；
- d. 通过这个项目可以锻炼队伍，提高团队的开发能力和项目管理能力；
- e. 本系统能够提供用户网上购物页面，方便用户操作。

2. 运行环境要求

程序设计语言：Java 8

系统数据库：MySQL

程序设计框架：Spring Boot

3. 前言

网上购物是电子商务的一种应用形式,电子商务建设的最终目的是发展业务和应用。网上购物系统的建设,可以建立起电子商务服务的门户网站,是现实社会到网络社会的真正体现,为广大网上商家以及网络客户提供一个符合中国国情的电子商务网上生存环境和商业运作空间。

电子商务(EC)源于英文 Electronic Commerce,简写 EC(也有人称之为 Electronic Business 两者有一定的区别)。顾名思义,其内容包含两个方面一是电子方式,二是商贸活动。电子商务指的是利用简单、快捷、低成本的电子通信方式,买卖双方不谋面地进行的各种商贸活动。

电子商务从交易对象上划分,可以分为以下三种类型:

企业与消费者之间的电子商务(Business to Customer 即 B To C)

企业与企业之间的电子商务(Business to Business 即 B To B)

企业与政府方面的电子商务(Business to Government 即 B To G)

从网上购物的交易对象上来看,网上购物系统是属于 B To C(企业与消费者之间的电子商务)形式。

随着计算机技术的发展和网络人口的增加,网络世界也越来越广博,越来越丰富,电子商务已经成为网上的一股潮流。很多大型商店已转向在互联网上购物业务,以节约成本,增加利润增长点。相信要不了太长有时间,顾客就可以在网络世界上获得他们在现实世界上可以获得的所有商品和服务。所以,网上购物已是大势所趋。

电商系统致力于提供产品展示及订购为核心的网上购物服务宣传自己商店的产品并将自己的产品展现给客户,让客户通过网站便能对自由的选择地购买产品。

该网站是通过用户登录浏览商品、查看公告、购买、确定购买、实现用户模块功能。其中订单的生成,网站后台系统,通过系统管理员管理商品、订单、用户来实现。

2. 需求功能介绍

本系统是有一系列网页组成的,它通过由商品信息构成,这些商品信息的排列顺序由推荐算法计算产生,能按照用户的需求进行日期、关键词、产品属性、相关性等检索能力获得。商品页面的构成有商品图片、商品介绍、商品属性、商品评价、买家秀、商品购买链接等。同时包含用户账户注册信息维护、商家商品管理、系统管理等功能页面。正是电子商务系统提供的强大而丰富的在线购物能力,能够帮助人们在网络上购买到日常生活所需的各类商

品，为人们的日常生活提供便捷，让人们热爱上网上购物，同时页开拓新了新的商业模式，网上商城将会变得越来越普及，越来越为更多的人接受。

1. 客户界面部分

本部分系统包括面向消费者用户和商家用户两类用户的服务功能,其中包含丰富的商品展示、新增、维护页面，同时也提供用户信息维护、订购、订单查询等页面和功能。

本部分系统功能需求如下：

- a. 提供面向消费者的功能需求：
 - a) 提供用户登录和注册的功能；
 - b) 提供面向用户的个人信息信息维护功能；
 - c) 提供用户地址信息的维护管理功能；
 - d) 提供依照关键词、日期、产品属性、相关性在平台内检索信息的能力；
 - e) 提供商品的展示、搜索、详细信息展示等功能；
 - f) 提供商品的购买、退货等功能；
 - g) 提供订单的信息展示、检索、购买、取消等功能；
 - h) 提供商品购物车、收藏夹、优惠券、红包的功能；
 - i) 提供商品向用户主动推荐、浏览历史记录的能力；
 - j) 提供用户对商家进行投诉的功能；
- b. 提供面向商家用户的功能需求：
 - a) 提供商家登录、商家主账户和员工账户注册、员工账户统一分配的功能；
 - b) 提供商家进行对自有域账户进行组织架构管理的能力；
 - c) 提供面向商家用户的个人信息信息维护功能；
 - d) 提供面向商家的商品维护功能；
 - e) 提供订单的信息展示、检索、购买、取消等功能；
 - f) 提供面向商家优惠券、红包的发放、回收、验证等功能；
 - g) 提供商家店铺的数据采集、分析的能力；

2. 管理界面部分

本部分系统包括面向平台管理员用户的服务功能,这是对平台进行统一管理和服务的功能,其中包含商品维护、下架页面,同时也提供用户信息维护、订单查询等页面和功能。

本部分系统功能需求如下:

- a. 提供管理员用户维护本系统内商品维护的功能;
- b. 提供管理员用户维护消费者用户和商家用户账户可用性的能力;
- c. 提供管理员用户维护平台数据的能力;
- d. 提供平台运行数据采集、分析和改进的能力;
- e. 提供用户数据数据采集和分析的能力。

二、系统设计

1. 系统体系结构设计

1. 项目环境和依赖

- 开发环境
 - 开发语言: JDK 1.8 (1.8.0_221)
 - 开发工具: IntelliJ IDEA 2020.3.EAP
 - 数据库: MySQL 8.0.22
 - 开发框架: Spring Boot 2.3.4.RELEASE
 - ORM 框架: MyBatis
 - 前端模版引擎: Art-Template
 - 后端模版引擎: Freemarker
- 环境配置
 - 访问路径: <http://127.0.0.1:9004/ecommerce/>
 - 服务端口号: 9004
- 第三方库依赖
 - org.springframework.boot: spring-boot-starter-freemarker 2.3.4.RELEASE
 - org.springframework.boot: spring-boot-starter-jdbc 2.3.4.RELEASE
 - org.springframework.boot: spring-boot-starter-web 2.3.4.RELEASE
 - org.mybatis.spring.boot: mybatis-spring-boot-starter 2.1.3
 - org.springframework.boot: spring-boot-devtools 2.3.4.RELEASE
 - org.springframework.boot: spring-boot-configuration-processor 2.3.4.RELEASE
 - mysql: mysql-connector-java 8.0.21
 - org.projectlombok: lombok 1.18.12
 - org.springframework.boot: spring-boot-starter-test 2.3.4.RELEASE
 - junit: junit 4.13
 - org.springframework.restdocs: spring-restdocs-mockmvc 2.3.4.RELEASE
 - com.alibaba: druid 1.1.20
 - com.alibaba: fastjson 1.2.74
 - cn.kevinkda: java-utils 0.0.8-RELEASE
 - com.google.code.gson: gson 2.8.6
 - org.webjars: jquery 1.12.4
 - org.webjars.npm: vue 3.0.2

2. 分层分析

1. 开发框架层

a) Spring

Spring 是一个开源框架，Spring 是于 2003 年兴起的一个轻量级的 Java 开发框架，由 Rod Johnson 在其著作 Expert One-On-One J2EE Development and Design 中阐述的部分理念

和原型衍生而来。它是为了解决企业应用开发的复杂性而创建的。框架的主要优势之一就是其分层架构，分层架构允许使用者选择使用哪一个组件，同时为 J2EE 应用程序开发提供集成的框架。Spring 使用基本的 JavaBean 来完成以前只可能由 EJB 完成的事情。然而，Spring 的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言，任何 Java 应用都可以从 Spring 中受益。Spring 的核心是控制反转（IoC）和面向切面（AOP）。简单来说，Spring 是一个分层的 JavaSE/EE full-stack(一站式) 轻量级开源框架。

b) Spring Boot

Spring Boot 是所有基于 Spring 开发的项目的起点。Spring Boot 的设计是为了让你尽可能快的跑起来 Spring 应用程序并且尽可能减少你的配置文件。简单来说就是 Spring Boot 其实不是什么新的框架，它默认配置了很多框架的使用方式，就像 Maven 整合了所有的 jar 包，Spring Boot 整合了所有的框架。

Spring Boot 四个主要特性：

- a. Spring Boot Starter：他将常用的依赖分组进行了整合，将其合并到一个依赖中，这样就可以一次性添加到项目的 Maven 或 Gradle 构建中；
- b. 自动配置：Spring Boot 的自动配置特性利用了 Spring 4 对条件化配置的支持，合理地推测应用所需的 bean 并自动化配置他们；
- c. 命令行接口：（Command-line-interface, CLI）：Spring Boot 的 CLI 发挥了 Groovy 编程语言的优势，并结合自动配置进一步简化 Spring 应用的开发；
- d. Actuator：它为 Spring Boot 应用的所有特性构建一个小型的应用程序。但首先，我们快速了解每项特性，更好的体验他们如何简化 Spring 编程模型。

2. 数据库驱动和中间介层及 ORM 框架层

a) 数据库驱动-JDBC

JDBC (Java Database Connectivity, Java 数据库连接) 是一种用于执行 SQL 语句的 Java API 可以为多种关系数据库提供统一访问，它由一组用 Java 语言编写的类和接口组成。JDBC 提供了一种基准，据此可以构建更高级的工具和接口，使数据库开发人员能够编写数据库应用程序。

JDBC 是个“低级”接口，也就是说，它用于直接调用 SQL 命令。在这方面它的功能极佳，并比其它的数据库连接 API 易于使用，但它同时也被设计为一种基础接口，在它之上可以建立高级接口和工具。高级接口是“对用户友好的”接口，它使用的是一种更易理解和更为方便的 API，这种 API 在幕后被转换为诸如 JDBC 这样的低级接口。

在关系数据库的“对象/关系”映射中，表中的每行对应于类的一个实例，而每列的值对应于该实例的一个属性。于是，程序员可直接对 Java 对象进行操作；存取数据所需的 SQL 调用将在“掩盖下”自动生成。此外还可提供更复杂的映射，例如将多个表中的行结合进一个 Java 类中。

随着人们对 JDBC 的兴趣日益增涨，越来越多的开发人员一直在使用基于 JDBC 的工具，以使程序的编写更加容易。程序员也一直在编写力图使最终用户对数据库的访问变得更为简单的应用程序。例如，应用程序可提供一个选择数据库任务的菜单。任务被选定后，应用程序将给出提示及空白供填写执行选定任务所需的信息。所需信息输入应用程序将自动调用所需的 SQL 命令。在这样一种程序的协助下，即使用户根本不懂 SQL 的语法，也可以执行数据库任务。

JavaSoft 提供三种 JDBC 产品组件，它们是 Java 开发工具包(JDK)的组成部份：JDBC 驱动程序管理器、JDBC 驱动程序测试工具包和 JDBC-ODBC 桥。

JDBC 驱动程序管理器是 JDBC 体系结构的支柱。它实际上很小，也很简单；其主要作用是把 Java 应用程序连接到正确的 JDBC 驱动程序上，然后即退出。

JDBC 驱动程序测试工具包为使 JDBC 驱动程序运行您的程序提供一定的可信度。只有通过 JDBC 驱动程序测试的驱动程序才被认为是符合 JDBC 标准 TM 的。

JDBC-ODBC 桥使 ODBC 驱动程序可被用作 JDBC 驱动程序。它的实现为 JDBC 的快速发展供了一条途径，其长远目标提供一种访问某些不常见的 DBMS（如果对这些不常见的 DBMS 未实现 JDBC）的方法。

使用的时候，需要首先下载相应的驱动，然后配置 Java 工程里面的环境，再书写代码。因为 JDBC 是一个开放的标准，所以在使用的时候。需要先用反射加载驱动。可以使用 ClassLoader，也可以直接用 Class.forName 进行反射，反射的时候会自动运行驱动程序类的静态代码块，在静态代码块里面会进行数据库连接的初始化。同时因为这个关系，如果静态代码块初始出现异常，数据库功能将不能使用，除非重启程序。

b) MyBatis

MyBatis 是一个半 ORM（对象关系映射）框架，它内部封装了 JDBC，开发时只需要关注 SQL 语句本身，不需要花费精力去处理加载驱动、创建连接、创建 statement 等繁杂的过程。程序员直接编写原生态 SQL，可以严格控制 SQL 执行性能，灵活度高。

MyBatis 可以使用 XML 或注解来配置和映射原生信息，将 POJO 映射成数据库中的记录，避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集。

通过 XML 文件或注解的方式将要执行的各种 statement 配置起来，并通过 java 对象和 statement 中 SQL 的动态参数进行映射生成最终执行的 SQL 语句，最后由 MyBatis 框架执

行 SQL 并将结果映射为 Java 对象并返回。（从执行 SQL 到返回 result 的过程）。

3. Model 和 View 层

Model 层等同于 Entity 层。存放我们的实体类，与数据库中的属性值基本保持一致。

Model 和 View 层的关系如下：



图 1Model 和 View 层的关系 E-R 图

4. Constant 层

该层根据阿里巴巴 Java 代码设计在应用模块包下建立 constant 包，建立 ConstConfig 程序类存放应用级共享常量，这将可以消除编码过程中可能出现的绝大多数魔法值，使应用返回信息统一，实现一处修改多处同步，使程序复杂度降低。

5. Mapper 层

Mapper 层等同于 DAO 层，现在用 Mybatis 逆向工程生成的 Mapper 层，其实就是 DAO 层。对数据库进行数据持久化操作，它的方法语句是直接针对数据库操作的，而 Service 层是针对我们 Controller，也就是针对我们使用者。Service 的 impl 是把 Mapper 和 Service 进行整合的文件。

这里提供了针对 Mapper 层各表的操作方法。根据以上实体之间的关系，设计出本层的 UML 类图。从图中我们可以清楚得看出类间依赖关系。

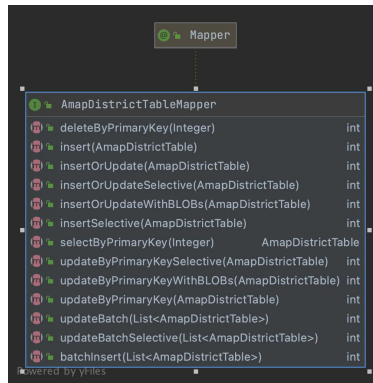


图 2Mapper 层 UML 样例

6. Service 层

该层包含存放业务逻辑处理，也是一些关于数据库处理的操作，但不是直接和数据库打交道，他有接口还有接口的实现方法，在接口的实现方法中需要导入 Mapper 层，Mapper 层是直接跟数据库打交道的，他也是个接口，只有方法名字，具体实现在 mapper.xml 文件里，Service 是供我们使用的方法。

这里提供了针对 Service 层的操作方法进行了举例。根据以上实体之间的关系，设计出本层的 UML 类图。从图中我们可以清楚得看出类间依赖关系。

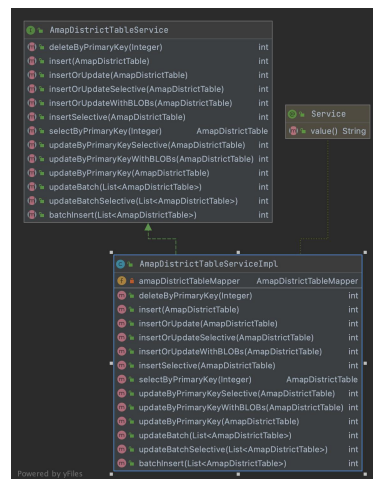


图 3Service 层 UML 样例

7. Controller 层

Controller，从字面上理解是控制器，所以它是负责业务调度的，所以在这一层应写一些业务的调度代码，而具体的业务处理应放在 Service 中去写，而且 Service 不单纯是对于 Mapper 的增删改查的调用，Service 是业务层，所以应该更切近于具体业务功能要求，所以

在这一层,一个方法所体现的是一个可以对外提供的功能,比如购物商城中的生成订单方法,这里面就不简单是增加个订单记录那么简单,我们需要查询库存,核对商品等一系列实际业务逻辑的处理。

2. 数据流图

数据流图（Data Flow Diagram）：简称 DFD，它从数据传递和加工角度，以图形方式来表达系统的逻辑功能、数据在系统内部的逻辑流向和逻辑变换过程，是结构化系统分析方法的主要表达工具及用于表示软件模型的一种图示方法。

根据系统功能的分析可得出如下关于会员购物过程、会员退货过程、商品管理过程和会员管理过程的数据流图。

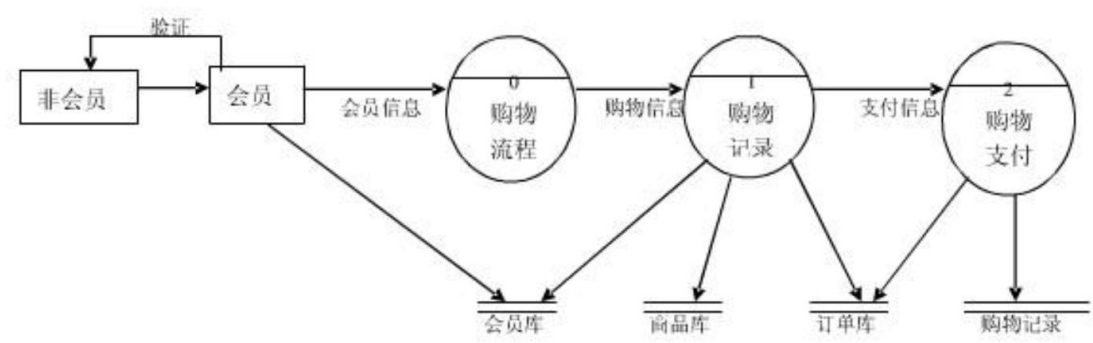


图 4 会员购物过程

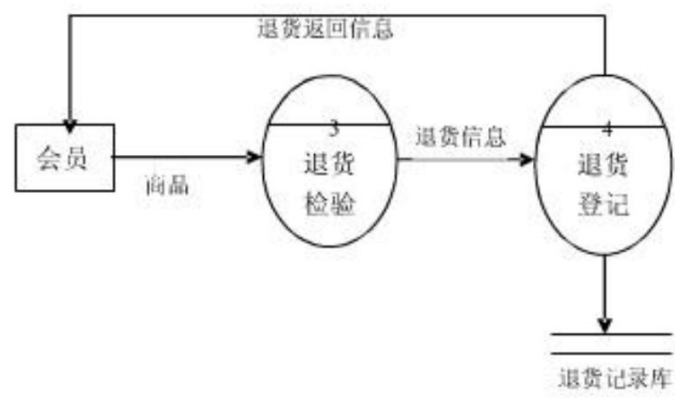


图 5 会员退货过程

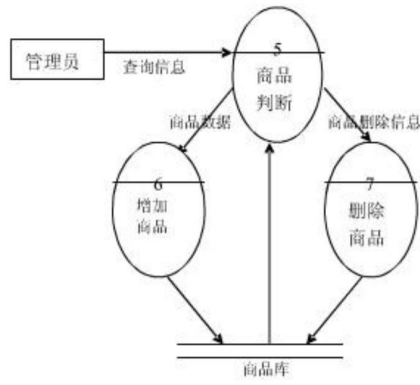


图 6 商品管理过程

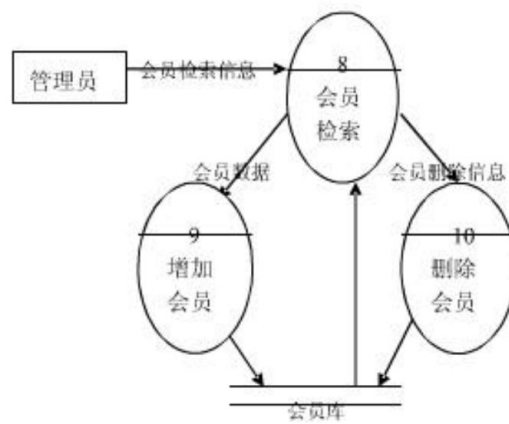


图 7 会员管理过程

3. 数据库设计与实现

1. 数据库概念结构设计

在建立数据库之前，要对系统数据进行概念模型设计，根据概念设计，得到如下系统总体 ER 图。

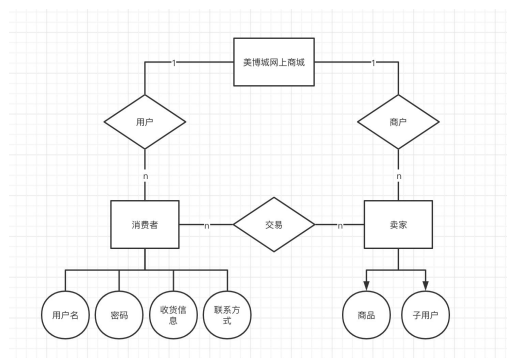


图 8 系统总体 ER 图

2. 数据库逻辑结构设计

依照数据库概念结构设计得到如下图所示的结构设计。

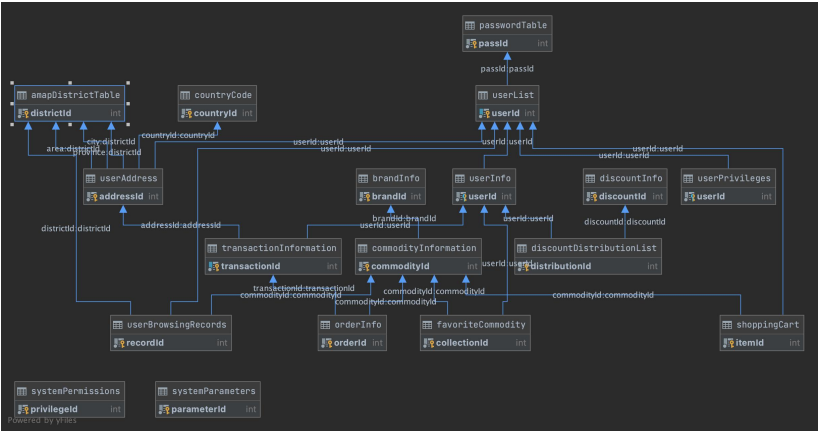
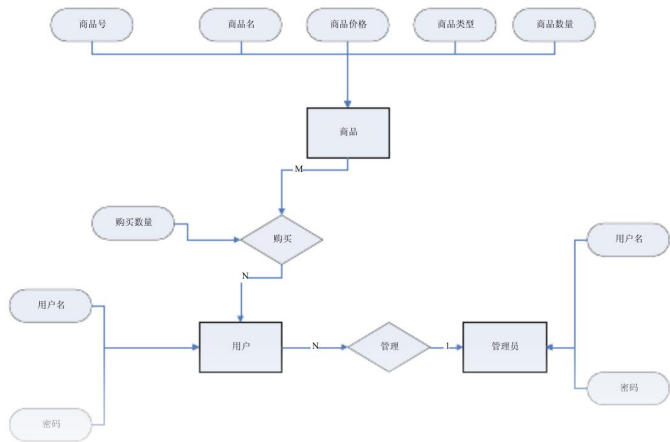


图 9 数据库结构设计



3. 数据库的设计创建

1) 表 amapDistrictTable

表 amapDistrictTable，为行政区划对照表，该表从高德地图第三方行政区划 API 接口上爬取得到，该表完成本系统中地址信息三级联动服务，同时为其他需求地理、地址信息的模块提供服务。

字段名	字段说明	字段类型	主外键、属性
districtId	区域 ID	int	PK、自动递增

citycode	城市编码	varchar(10)	
adcode		int	
name	行政区名	varchar(50)	
polyline		text	
center	边界坐标	varchar(200)	
level	行政区级别	varchar(50)	
parentId	上级行政区 ID	int	

2) 表 brandInfo

表 brandInfo，为品牌信息表，记录商品信息列表中商品对应的品牌信息。表设计中预留 parameter 字段存储 JSON 信息，以容纳在未来一段时间对数据库的小规模扩充需求，也可存储非固定字段，使数据表具有非关系型数据库特性。

字段名	字段说明	字段类型	主外键、属性
brandId	品牌 ID	int	PK、自动递增
brandName	品牌名	varchar(255)	
brandPhotoUrl	品牌图片路径	text	
parameter	商品参数 JSON	json	
updateTime	更新时间	datetime	当前时间自动更新

3) 表 commodityInformation

表 commodityInformation，为商品信息列表，记录所有商品信息。表设计中预留 parameter 字段存储 JSON 信息，以容纳在未来一段时间对数据库的小规模扩充需求，也可存储非固定字段，使数据表具有非关系型数据库特性。

字段名	字段说明	字段类型	主外键、属性
commodityId	商品 ID	int	PK、自动递增
commodityName	商品名	varchar(255)	
brandId	品牌 ID	int	FK(brandInfo.brandId)
commodityPrice	商品价格	decimal(10,2)	
commodityPhotoUrl	商品图片路径	text	
commodityStatus	商品状态	int	无符号、零填充
parameter	商品参数 JSON	json	
updateTime	更新时间	datetime	当前时间自动更新

4) 表 favoriteCommodity

表 favoriteCommodity，为收藏夹列表，记录所有用户的收藏夹项目。表设计中预留 parameter 字段存储 JSON 信息，以容纳在未来一段时间对数据库的小规模扩充需求，也可存储非固定字段，使数据表具有非关系型数据库特性。

字段名	字段说明	字段类型	主外键、属性
collectionId	收藏夹 ID	int	PK、自动递增
commodityId	商品 ID	int	FK(commodityInformation.commodityId)
userId	用户 ID	int	FK(passwordTable.userId)
collectionStatus	收藏状态	int	无符号、零填充
parameter	商品参数 JSON	json	
updateTime	更新时间	datetime	当前时间自动更新

5) 表 userList

表 userList，为用户列表，记录所有用户的账户基础信息，用户详细信息由其他表提供。表设计中预留 userConfiguration 字段存储 JSON 信息，以容纳在未来一段时间对数据库的小规模扩充需求，也可存储非固定字段，使数据表具有非关系型数据库特性。

字段名	字段说明	字段类型	主外键、属性
userId	用户 ID	int	PK、自动递增
userName	用户名	varchar(50)	
passId	用户密码 ID	int	FK(passwordTable.userId)
accStatus	账户状态	int	无符号、零填充
regTime	注册时间	datetime	当前时间自动更新
userConfiguration	用户参数	json	
flag	用户标记	json	

6) 表 userBrowsingRecords

表 userBrowsingRecords，记录用户浏览商城商品的所有记录。表设计中预留 parameter 字段存储 JSON 信息，以容纳在未来一段时间对数据库的小规模扩充需求，也可存储非固定字段，使数据表具有非关系型数据库特性。

字段名	字段说明	字段类型	主外键、属性
recordId	记录 ID	int	PK、自动递增
commodityId	商品 ID	int	FK(commodityInformation.commodityId)
userId	用户 ID	int	FK(passwordTable.userId)
districtId	访问地区 ID	int	FK(emapDistrictTable.districtId)
parameter	商品参数 JSON	json	
updateTime	更新时间	datetime	当前时间自动更新

三、应用程序设计

1. 设计页面

a. 商品浏览记录页面

用户在电子商务系统中的浏览记录是推荐系统的重要组织部分。

在推荐系统中用户的数据输入是网上商城进行推荐的主要数据依据,而推荐系统则是根据以往用户的输入数据对用户进行相应的个性化推荐。如果没有用户输入,那么一个推荐系统而产生的推荐结果是无法达到用户个性化的。网上商城系统的输入数据通常包括用户个人信息数据、交易数据以及用户浏览数据三个部分,但根据输入方式可以分为以下几类:

- a) 隐式浏览输入 (Implicit navigation): 指的是不需要用户进行手工输入的情况下而获取到的相关信息,可以由系统自动分析用户与项目之间的交互关系(比如客户的浏览记录和购买行为),进而推出客户的兴趣。即用户与系统的交互为的输入。
- b) 显式浏览输入 (Explicit navigation): 是指需要用户明确地向系统指出个人感兴趣的产品或者有目的地向其他人提供自己的喜好,如对产品的购买、评价等多种方式。
- c) 客户购买历史 (Purchase history): 是指将客户在过去的购买记录作为输入数据输入系统。

用户和商品信息的数据采集模块是本系统设计的关键部分,直接影响到之后数据分析处理的结果,影响到系统推荐算法和功能的实现,甚至该模块工作的正常与否直接关系整个系统工作效果和状态的正常与否。因此信息采集一定要细致、全面、准确,才能训练出更好的模型,更好的为用户服务。

此页面数据由用户访问本系统内商品信息时产生的痕迹系统。用户每次访问商品页面前台页面会向本系统进行记录反馈,由本系统将用户的访问记录存入数据库中,并永久保留。这些数据的记录为本系统的商品推荐提供底层数据支持。

b. 商品推荐页面

a) 背景

基于互联网的电子商务发展日新月异,为用户提供的商品日益多样化,给用户带来丰富选择的同时,也出现了信息过载现象。如何在众多的商品中给用户展现其感兴趣的商品,提升用户体验,提高网站商品的销售转化,是值得我们研究的课题。

推荐系统的研究持续了很多年,从上个世纪九十年代开始就不断有研究者对推荐系统和

算法进行研究和设计，并且不断有实践者在实际的商业系统中应用并且取得很丰富的成果。

本论文立足于一个真实的电子商务环境，结合过往推荐算法设计和推荐系统实践的经验，融入企业自身的一些特点，设计一套适合具体情况并能满足企业商业目标的商品推荐系统。

b) 意义

推荐系统在电子商务系统中到底有何意义，我们希望其能起到什么样的作用，系统总结如下：

第一，增加商品的销量。对于电子商务来说，这是最根本的商业目标，脱离了这个目标对于电子商务来说都是无意义的。推荐的商品越能够满足用户的需求，商品的销量才可能更高。在同样的用户流量情况下，增加商品销量，也就是提升了整个网站商品销售的有效转化率。

第二，售卖更多多样化的商品。推荐系统可以对已经浏览或者收藏或者购买了一件商品的用户推荐与这件商品相关的一些商品，帮助用户快速找到其需要的配套商品，比如对于买一个相机的用户，可以为其推荐相机相关的镜头、相机包、三脚架等配件，买了一个笔记本电脑的用户，可以为其推荐相关的散热器、鼠标清洁用品等。对于电子商务的运营者来说，也是希望通过售卖多样化的商品来提升整体的规模和效率，提升冷门产品的曝光几率和销量，而不仅仅只是售卖热门的商品。

第三，提升用户的满意度。一个良好设计的推荐系统可以改进用户在网站中的体验，用户会发现页面中推荐的商品是其感兴趣的，和当前上下文是相关的，进而会更乐于使用这个网站。有效的、相关的和精准的推荐能够使用户对网站产生积极的评价和好感，而这将增加用户的使用时间和行为次数，进而推荐系统能做出更好的推荐。

第四，提升用户的忠诚度。推荐系统能够记录用户的历史访问行为，并从历史行为中总结用户的偏好，用户的访问次数越多，其偏好数据越完整，推荐系统可以为用户推荐的商品也更符合用户的胃口，用户对于网站的忠诚度也会越高。

第五，更好的理解用户需求。推荐系统会收集用户的显式偏好和隐式行为，并能预测用户的偏好，这些数据同样能够用在电子商务的其他应用中，比如做库存和生产的预测、做个性化邮件营销等等，使得企业能够更好的理解和满足用户的需求。

c) 推荐系统应用场景

在电子商务网站中，未部署推荐系统应用之前，网站本身一般都具有分类展示商品功能和站内商品搜索功能，这两种信息获取的方式都需要用户明确知道自己的意图，然后使用分类浏览或者搜索功能寻找需要的商品，而推荐系统是主动给用户推荐其感兴趣的物品，是以上两种方式的重要补充手段，尤其是在商品数量非常大的情况下，用户寻找商品的成本较高，

如果有推荐系统能够在众多的商品中推荐给用户感兴趣的商品,将大大缩短用户寻找商品的成本,提高用户的使用体验,从而增加商品购买转化。下图是一个电子商务网站中常见的三种获取商品的方式,其中推荐系统商品展示是本论文中所关注的内容。

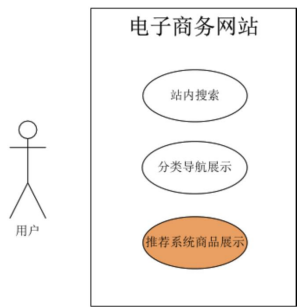


图 10 电子商务网站用户浏览商品的方式

d) 应用场景设计

本推荐系统在电子商务网站中的应用场景如下：

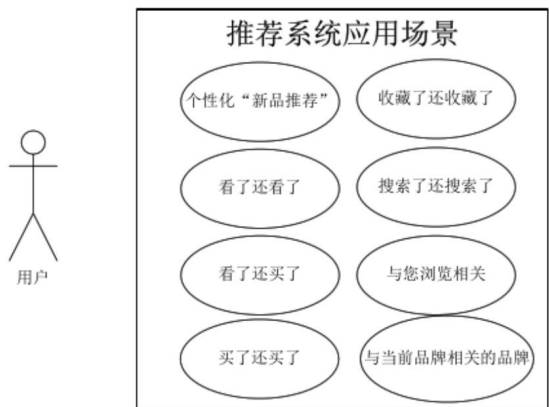


图 11 本课程设计推荐系统应用场景

e) 应用场景目标

上节所述的使用场景都是在现有的各类功能页面上开辟一块位置进行,不同的位置也决定了需要使用不同的数据依据和推荐思路。比如“看过了还看过了”是依据群体用户的商品浏览行为进行的相关推荐,而且放在了商品详情页,因为用户这时候仍在浏览商品,并没有最终决定购买,所以给出“看过了还看过了”的推荐理由也比较符合用户当前的心理,给用户推荐更多相关商品供选择,让用户更快找到自己想要的商品,降低寻找成本,提升用户的体验。

“买过了还买过了”放在购物车页,用户将商品加入了购物车页,说明用户这时候对商品已经有了购买意向,这时候给用户推荐“买过了还买过了”的商品,推荐的理由更具有说服力,而且有利于引导用户加入更多的商品到购物车中,提升客单价和复购率。“看过了还看过了”

和“买过了还买过了”由于应用的位置不同，其要达到的效果也是不同的，前者是为了提高用户的粘性，降低寻找商品成本，提升用户体验，而后者主要是为了提高用户的客单价。

下表详细描述了每个推荐位置的推荐的理由和商业目标。

网站位置	推荐理由	主要目标
首页	个性化新品	依据用户偏好推荐新品, 给用户提供更比较新颖的商品
商品详情页	看过该商品的用户还看过; 看过该商品的用户还买过	相关商品推荐, 降低寻找商品成本, 提高用户体验, 促进购买
购物车	买过该商品的用户还买过	提升客单价
会员中心页	与用户最近的浏览历史相关	推荐与最近浏览历史记录相关的商品, 提升复购率
收藏页	收藏了该商品的用户还收藏了	引导用户多收藏商品, 获取更多用户反馈
搜索结果页	搜索了该词的用户还搜索了	搜索词推荐, 帮助用户发现其需求, 引导用户浏览更多页面
品牌页	与当前品牌相关的品牌	完善内链结构, 使得相关品牌之浏览跳转更流畅
品类页	与当前品类相关的品类	完善内链结构, 使得相关品类之间浏览跳转更流畅

f) 现有推荐系统的核心问题

目前被广泛引用的推荐系统的非形式化概念是 Resnick 和 Varian 在 1997 年给出的：“它是利用电子商务网站向客户提供商品信息和建议，帮助用户决定应该购买什么产品，模拟销售人员帮助客户完成购买”。推荐系统有 3 个核心组成要素：用户、商品、推荐方法。其核心问题如下：

i. 用户特征分析

用户的人口统计学特征：包括用户的年龄、性别等用户在注册时提供的信息。

用户的行为特征：包括浏览商品、将商品加入购物车、购买商品、收藏商品、评论商品、给商品打分、搜索商品、分享商品等。从时间序列上，可以分为用户近期的行为和长期的行为；也可以分为显式的行为和隐式的行为，显式的行为是指用户主动提交的自身能够明显感知到的行为，比如购买、收藏和评论等，隐式的行为包括指浏览网页的行为，用户没有给出主动反馈的行为。

用户的兴趣标签特征：在带标签的系统中，比如豆瓣、新浪微博等，用户可以给自己打出明显的兴趣爱好标签。

如上用户的特征数据存储以不同的形式存储在不同的介质中,注册信息一般存储于关系数据库中,用户行为特征一般存在于 web 日志文件中、或者分布式文件系统如 HDFS 中、或者缓存中。在开发推荐系统时,需要将这些数据进行清理、转换生成到统一的用户偏好数据库中。

ii. 商品特征分析

商品特征包括基本特征和动态属性特征,不同问题领域涉及的商品或者产品的属性有所不同,如下描述的特征是电子商务零售产品的常见特征。

商品的基本特征:包括品牌、品类、货号、性别、尺码、颜色、上架时间、适应季节等属性,这些属性在商品上架的时候,一般都是固定不变的,又可称为静态属性。

商品的动态属性:包括销售数量、库存、市场价格、销售价格、毛利、曝光数、浏览次数、加购物车次数等,这些属性都是随着用户的行为、公司的价格策略调整等行为发生变化的。

以上是电子商务零售产品的常见特征,这些特征都是比较容易获取的,基于这些特征我们可以做商品相似度计算、商品的搭配度计算等,为推荐算法提供基础数据。有些类型商品属性特征比较容易获取和准确描述,并且这些特征也能够有效表达用户的兴趣特征,比如用户喜欢什么样的品牌,用户喜欢什么颜色的商品等,针对于这类商品比较适合做基于内容的推荐算法。但是有些商品可描述的属性特征较少,或者商品的属性特征无法直观表达用户的兴趣特征,比如电影,图书等可描述属性特征不多,并且用户喜欢一部电影或者图书的原因可能与其现有的属性没有直接关系,这时可以采用用户标签系统来弥补商品的不可描述性,当然此类商品也可以采用协同过滤推荐算法来给用户做推荐,因为协同过滤算法不需要依赖于商品自身的属性特征。

iii. 研究用户和商品、用户和用户以及商品和商品的关联关系

推荐算法的目标就是要将合适的商品推荐给用户,所以研究用户和商品之间的关联关系是选择和设计推荐算法的重要步骤。下图是一些常见的用户和商品之间的关联关系,每条路径代表一类推荐算法的基本思路。

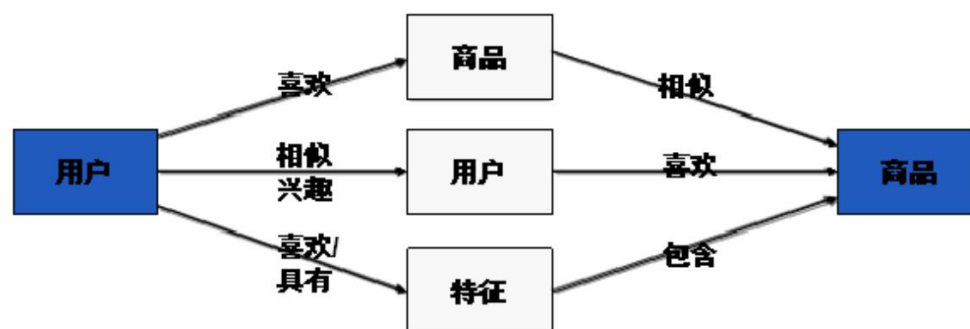


图 12 用户与商品之间的关联关系

用户->商品->商品：这条路径代表了基于内容的推荐算法的主要思想，推荐给用户的商品是与用户喜欢的商品相似的商品，用户喜欢的商品可以通过用户在网站中对商品的浏览、加购物车、浏览、收藏、评论等行为进行获取，用户对商品发生了如上行为，可以认为用户对该商品是感兴趣的，依据不同的行为相关的商品，可以给用户推荐与该商品相关的商品，并且给出推荐的理由。协同过滤算法中的基于商品的推荐算法也是类似的思路，只不过计算商品相似度的时候，不采用内容，而是使用用户行为进行相似度计算。

用户->用户->商品：这条路径代表了基于用户的协同过滤推荐算法的主要思想，通过寻找与当前用户具有相似兴趣的用户，推荐这些用户感兴趣的商品。

用户->特征->商品：这条路径是基于用户标签的推荐系统和基于知识的推荐采用的思路。在用户标签系统中，比如豆瓣、新浪微博，用户都会主动对商品或者给自身贴具有明显特征的标签，依据这些标签搜索对应的商品进行推荐，也是一种有效的推荐手段；基于知识的推荐是依据特定领域的知识进行推荐，如果知道用户具有哪些特征，并且具有如何满足这些特征的专家知识，则可以直接进行推荐，比如医疗专家系统。

2. 程序代码

3. 运行结果

四、小结

这次框架企业级应用综合设计课程设计是一门非常不错的课程,在此之前我写服务器基本都是自己造轮子自己写,但是这次课设使得我有机会使用现在还在使用 Spring Boot + MyBatis + MySQL + Freemarker 进行架构开发,同时,这次实验对我来说是一个巨大的挑战。因为之前只是对 Spring Boot 的快捷开发有所耳闻,原先用 Servlet 的时候也是自己从 GET 和 POST 处理函数为入口自己写处理,所以一开始对于 Spring Boot 的写法还不是很熟练。但是,经过持续约一个星期坚持不懈的努力,我终于完成了本次的实验任务。

当然在开发的过程中也遇到了很多问题。老师曾说,对于我们现在接触到的困难,之前肯定有人接触过,所以要好好利用互联网。课程设计,从安装 MySQL,到配置数据库,到使用 JDBC 进行数据的增删改查,每步都曾遇到很棘手的问题,多亏了老师的帮助,也感谢在社区和论坛上帮我解决问题的朋友们。

采用 B/S 结构与 JDBC 框架。动态 WEB 中,程序依然使用客户端和服务端,客户端依然使用浏览器(Chrome、Safari、Firefox 等),通过网络(Network)连接到服务器上,使用 HTTP 协议发起请求(Request),现在的所有请求都先经过一个 WEB Server Plugin(服务器插件)来处理,此插件用于区分是请求的是静态资源(*.htm 或者是 *.html)还是动态资源。

当然,这次实验的另外一个体会就是,时间安排非常重要。老师在很早的时候就把任务书发了下来,但是当时我没在意,所以在之后做得非常紧。由此了解到,做好一个项目,每一阶段的时间都必须规划好。因为我们学生现阶段采用开发模式基本都是瀑布开发模式,就必须保证每一步的正确性和及时性。总之,本次设计锻炼了实践能力,增长了我发现问题解决问题的能力,大大丰富了自己的专业知识和技能。由于各种原因,系统的功能还没有得到进一步的完善,以后会继续努力,在今后的学习过程中,不断地提高自己。