

实 训 手 册

实习类型： 大数据实训

学 院： 软件学院

专 业： 软件工程

班 级： 20级大数据2班

学 号： 202026701078

姓 名： 高嘉仪

指导教师： 张鹏飞

实习单位： 软通动力技术服务有限公司

### 目录

目录

[目录 2](#_Toc13291)

[一、 项目名称 3](#_Toc3836)

[二、 意义背景 4](#_Toc22958)

[三、 需求分析 4](#_Toc28887)

[四、 功能设计 5](#_Toc25810)

[4.1概要设计 5](#_Toc16110)

[4.1.1功能概要 5](#_Toc10015)

[4.1.2系统整体功能用例图 6](#_Toc239)

[4.1.3系统数据流图 6](#_Toc6794)

[4.2详细设计 7](#_Toc32263)

[4.2.1登录注册 7](#_Toc4687)

[4.2.2购买商品 7](#_Toc27178)

[五、 数据库设计 8](#_Toc29108)

[5.1数据库设计概述 8](#_Toc1587)

[5.2数据库物理结构设计 8](#_Toc32270)

[5.3数据表结构 9](#_Toc4715)

[六、 亮点设计 13](#_Toc19899)

[6.1购买商品 13](#_Toc11779)

[6.2数据可视化 13](#_Toc9268)

[七、 运行截图 14](#_Toc21365)

[7.1注册登录 14](#_Toc19300)

[7.1.1注册 14](#_Toc24820)

[7.1.2登录 15](#_Toc15579)

[7.2购买界面 16](#_Toc21840)

[7.2.1商品界面 16](#_Toc16806)

[7.2.2购物界面 17](#_Toc27892)

[7.3数据分析 17](#_Toc9675)

[7.3.1统计购物总数最多的前十，并求出平均值 17](#_Toc26607)

[7.3.2商品类型分布 19](#_Toc17595)

[7.3.3统计男女用户比例 19](#_Toc7734)

[7.3.4男女每天消费金额比较 20](#_Toc8207)

[7.3.5前两名用户的消费喜好 20](#_Toc14806)

[八、 实习总结 20](#_Toc27302)

# 项目名称

“天猫商城”数据收集与分析

# 意义背景

由于公司的要求，运用主要以hadoop的技术框架为主，配合后端Java的代码的综合使用，学会使用springboot。同时，运用在实习期间学到的知识，基于springboot+hadoop+mysql框架，采用jsp前端页面，以springboot后台为主，完成项目的登录、注册、密码找回、数据展示、上传、下载、修改、删除、签到、预览、回收站、查看个人文件和群组文件等功能。

# 需求分析

|  |  |  |  |
| --- | --- | --- | --- |
| **主要业务** | **Story模块** | **主要功能** | |
| 进入系统 | 注册 | 用户可以通过邮箱或者手机号注册账号，注册成功后，用户可通过输入用户名和密码登录账号。 |
| 登录 |
| 修改个人信息 |
| 购买商品 | 选择商品 | 用户可以点击自己喜欢的商品并加入购物车，然后进行线上支付。 |
| 加入购物车 |
| 购买商品 |
| 数据采集 | 埋点 | 在登录和商品加入购物车进行埋点，数据直接收集到集群上 |
| 需求设计 | 落地到MySQL | 将数据落地到MySQL |
| 可视化 | 用echart对需求设计进行可视化 |

# 功能设计

## 4.1概要设计

### 4.1.1功能概要

**（1）注册**：新用户在第一次使用天猫商城，需要进行注册。用户输入自己的用户名、密码以及要绑定的邮箱等信息即可注册成功。

**（2）登录**：用户可以输入用户名和密码来登录自己的账号页面。

**（3）选择商品**：用户可以进入商品界面选择自己喜欢的商品

**（4）加入购物车**：用户可以把自己喜欢的商品加入到购物车中

**（5）购买商品**：用户可以线上支付自己想要购买的商品

### 4.1.2系统整体功能用例图

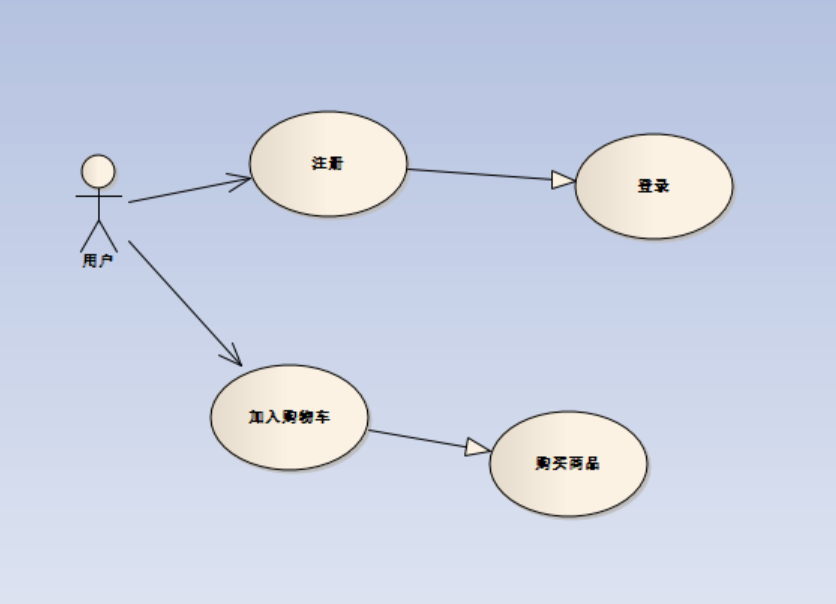


图4.1

### 4.1.3系统数据流图

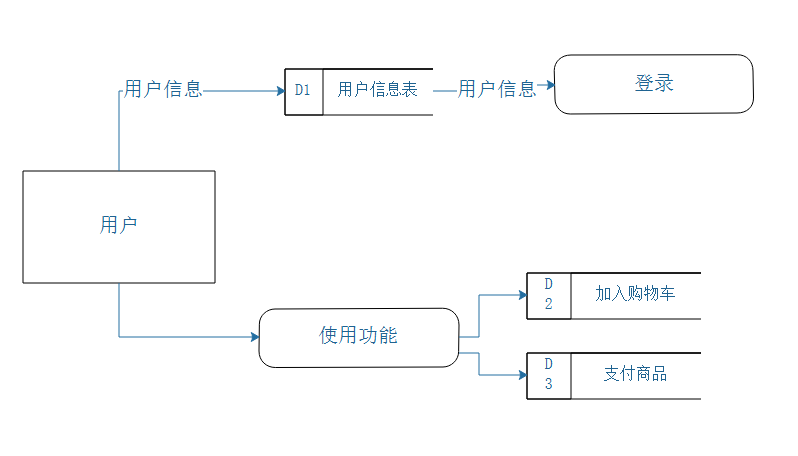


图4.2

## 4.2详细设计

### 4.2.1登录注册

1. 注册

用户输入用户名、密码以及个人邮箱在天猫商城上注册自己的账号。

（2）登录

用户完成注册后，可以通过用户名和密码登录天猫商城。

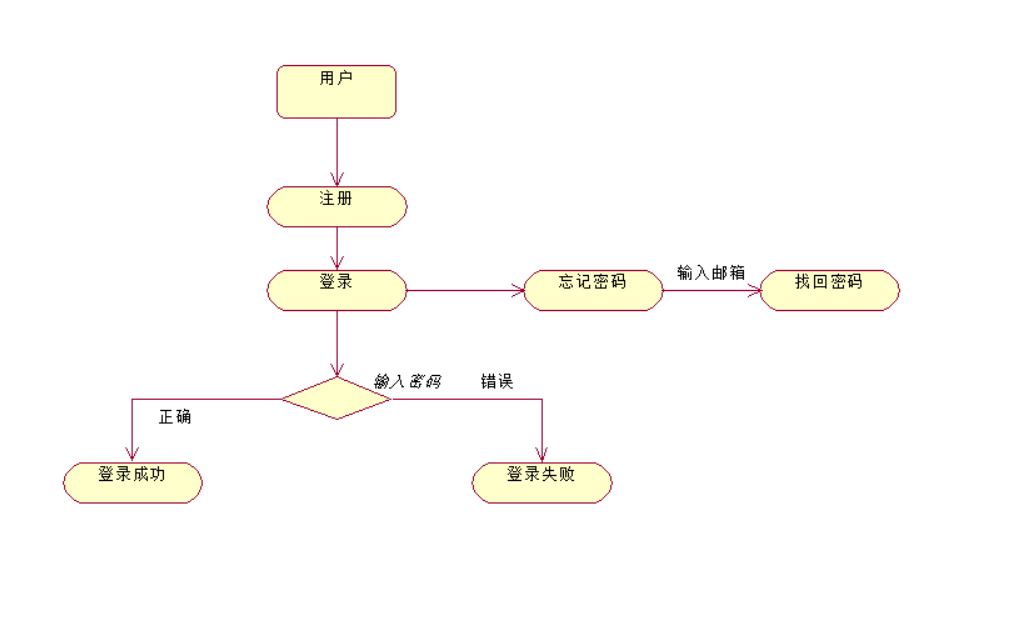


图4.3

### 4.2.2购买商品

1. 加入购物车：用户可以点击自己喜欢的商品并加入购物车
2. 线上支付：用户可以直接线上支付自己喜欢的商品或者支付已经加入购物车的商品

# 数据库设计

## 5.1数据库设计概述

本系统采用MySQL数据库进行开发，本系统共有七张表。

## 5.2数据库物理结构设计

CREATE TABLE `address` (

`address\_areaId` char(6) NOT NULL COMMENT '地区编码',

`address\_name` varchar(50) NOT NULL COMMENT '省市名称',

`address\_regionId` char(6) NOT NULL COMMENT '父级省市id',

PRIMARY KEY (`address\_areaId`) USING BTREE,

KEY `address\_regionId` (`address\_regionId`) USING BTREE,

CONSTRAINT `address\_ibfk\_1` FOREIGN KEY (`address\_regionId`) REFERENCES `address` (`address\_areaId`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 ROW\_FORMAT=DYNAMIC COMMENT='地址表';

DROP TABLE IF EXISTS `admin`;

CREATE TABLE `admin` (

`admin\_id` int(10) NOT NULL AUTO\_INCREMENT COMMENT '编号',

`admin\_name` varchar(25) CHARACTER SET utf8 COLLATE utf8\_bin NOT NULL COMMENT '账户名',

`admin\_nickname` varchar(50) NOT NULL DEFAULT '' COMMENT '昵称',

`admin\_password` varchar(50) CHARACTER SET utf8 COLLATE utf8\_bin NOT NULL COMMENT '密码',

`admin\_profile\_picture\_src` varchar(255) DEFAULT NULL COMMENT '头像地址',

PRIMARY KEY (`admin\_id`) USING BTREE,

UNIQUE KEY `un\_admin\_name` (`admin\_name`) USING BTREE

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8 ROW\_FORMAT=DYNAMIC COMMENT='管理员表';

/\*Table structure for table `category` \*/

DROP TABLE IF EXISTS `category`;

CREATE TABLE `category` (

`category\_id` int(10) NOT NULL AUTO\_INCREMENT,

`category\_name` varchar(20) NOT NULL COMMENT '类别名称',

`category\_image\_src` varchar(255) NOT NULL COMMENT '类别图片',

`del\_flag` tinyint(1) DEFAULT '0' COMMENT '删除标识(1删除 0未删除）',

PRIMARY KEY (`category\_id`) USING BTREE

) ENGINE=InnoDB AUTO\_INCREMENT=17 DEFAULT CHARSET=utf8 ROW\_FORMAT=DYNAMIC COMMENT='类别表';

DROP TABLE IF EXISTS `product`;

CREATE TABLE `product` (

`product\_id` int(10) NOT NULL AUTO\_INCREMENT,

`product\_name` varchar(100) NOT NULL COMMENT '产品名称',

`product\_title` varchar(100) DEFAULT NULL COMMENT '产品标题',

`product\_price` decimal(10,2) DEFAULT NULL COMMENT '原价',

`product\_sale\_price` decimal(10,2) NOT NULL COMMENT '促销价',

`product\_create\_date` datetime NOT NULL COMMENT '创建日期',

`product\_category\_id` int(10) NOT NULL COMMENT '类别id',

`product\_isEnabled` tinyint(1) NOT NULL DEFAULT '0' COMMENT '是否可用',

PRIMARY KEY (`product\_id`) USING BTREE,

KEY `product\_ibfk\_1` (`product\_category\_id`) USING BTREE,

CONSTRAINT `product\_ibfk\_1` FOREIGN KEY (`product\_category\_id`) REFERENCES `category` (`category\_id`) ON DELETE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=90 DEFAULT CHARSET=utf8 ROW\_FORMAT=DYNAMIC COMMENT='产品表';

DROP TABLE IF EXISTS `productImage`;

CREATE TABLE `productImage` (

`productimage\_id` int(10) NOT NULL AUTO\_INCREMENT,

`productimage\_type` tinyint(1) unsigned NOT NULL COMMENT '类型(0:概述图片 1:详情图片)',

`productimage\_src` varchar(255) NOT NULL COMMENT '图片地址',

`productimage\_product\_id` int(10) NOT NULL COMMENT '产品id',

PRIMARY KEY (`productimage\_id`) USING BTREE,

KEY `productimage\_product\_id` (`productimage\_product\_id`) USING BTREE,

CONSTRAINT `productImage\_ibfk\_1` FOREIGN KEY (`productimage\_product\_id`) REFERENCES `product` (`product\_id`) ON DELETE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=1037 DEFAULT CHARSET=utf8 ROW\_FORMAT=DYNAMIC COMMENT='产品图片表';

DROP TABLE IF EXISTS `productOrder`;

CREATE TABLE `productOrder` (

`productorder\_id` int(10) NOT NULL AUTO\_INCREMENT,

`productorder\_code` varchar(30) NOT NULL COMMENT '订单号',

`productorder\_address` char(6) NOT NULL COMMENT '产品地址',

`productorder\_detail\_address` varchar(255) NOT NULL COMMENT '产品详细地址',

`productorder\_post` char(6) DEFAULT NULL COMMENT '邮政编码',

`productorder\_receiver` varchar(20) NOT NULL COMMENT '收货人',

`productorder\_mobile` char(11) NOT NULL COMMENT '联系方式',

`productorder\_pay\_date` datetime NOT NULL COMMENT '支付日期',

`productorder\_delivery\_date` datetime DEFAULT NULL COMMENT '发货日期',

`productorder\_confirm\_date` datetime DEFAULT NULL COMMENT '确认日期',

`productorder\_status` tinyint(1) NOT NULL COMMENT '订单状态(0:待付款 1:待发货 2:待确认 3:交易成功 4:交易关闭)',

`productorder\_user\_id` int(10) NOT NULL COMMENT '用户id',

PRIMARY KEY (`productorder\_id`) USING BTREE,

UNIQUE KEY `un\_productorder\_code` (`productorder\_code`) USING BTREE,

KEY `productorder\_address` (`productorder\_address`) USING BTREE,

KEY `productorder\_ibfk\_2` (`productorder\_user\_id`) USING BTREE,

CONSTRAINT `productOrder\_ibfk\_1` FOREIGN KEY (`productorder\_address`) REFERENCES `address` (`address\_areaId`),

CONSTRAINT `productOrder\_ibfk\_2` FOREIGN KEY (`productorder\_user\_id`) REFERENCES `user` (`user\_id`) ON DELETE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=222 DEFAULT CHARSET=utf8 ROW\_FORMAT=DYNAMIC COMMENT='产品订单表';

DROP TABLE IF EXISTS `productOrderItem`;

CREATE TABLE `productOrderItem` (

`productorderitem\_id` int(10) NOT NULL AUTO\_INCREMENT,

`productorderitem\_number` smallint(5) unsigned NOT NULL COMMENT '数量',

`productorderitem\_price` decimal(10,2) NOT NULL COMMENT '单价',

`productorderitem\_product\_id` int(10) NOT NULL COMMENT '关联产品id',

`productorderitem\_order\_id` int(10) DEFAULT NULL COMMENT '关联订单id',

`productorderitem\_user\_id` int(10) NOT NULL COMMENT '关联用户id',

`productorderitem\_userMessage` varchar(255) DEFAULT NULL COMMENT '用户备注',

PRIMARY KEY (`productorderitem\_id`) USING BTREE,

KEY `productorderitem\_product\_id` (`productorderitem\_product\_id`) USING BTREE,

KEY `productorderitem\_order\_id` (`productorderitem\_order\_id`) USING BTREE,

KEY `productorderitem\_user\_id` (`productorderitem\_user\_id`) USING BTREE,

CONSTRAINT `productOrderItem\_ibfk\_1` FOREIGN KEY (`productorderitem\_product\_id`) REFERENCES `product` (`product\_id`),

CONSTRAINT `productOrderItem\_ibfk\_2` FOREIGN KEY (`productorderitem\_order\_id`) REFERENCES `productOrder` (`productorder\_id`) ON DELETE CASCADE,

CONSTRAINT `productOrderItem\_ibfk\_3` FOREIGN KEY (`productorderitem\_user\_id`) REFERENCES `user` (`user\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=286 DEFAULT CHARSET=utf8 ROW\_FORMAT=DYNAMIC COMMENT='产品订单详细表';

DROP TABLE IF EXISTS `property`;

CREATE TABLE `property` (

`property\_id` int(10) NOT NULL AUTO\_INCREMENT,

`property\_name` varchar(25) NOT NULL COMMENT '属性名称',

`property\_category\_id` int(10) NOT NULL COMMENT '关联类别id',

PRIMARY KEY (`property\_id`) USING BTREE,

KEY `property\_category\_id` (`property\_category\_id`) USING BTREE,

CONSTRAINT `property\_ibfk\_1` FOREIGN KEY (`property\_category\_id`) REFERENCES `category` (`category\_id`) ON DELETE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=245 DEFAULT CHARSET=utf8 ROW\_FORMAT=DYNAMIC COMMENT='类别属性表';

DROP TABLE IF EXISTS `propertyValue`;

CREATE TABLE `propertyValue` (

`propertyvalue\_id` int(10) NOT NULL AUTO\_INCREMENT,

`propertyvalue\_value` varchar(100) NOT NULL COMMENT '属性值',

`propertyvalue\_property\_id` int(10) NOT NULL COMMENT '关联属性id',

`propertyvalue\_product\_id` int(10) NOT NULL COMMENT '关联产品id',

PRIMARY KEY (`propertyvalue\_id`) USING BTREE,

KEY `propertyvalue\_property\_id` (`propertyvalue\_property\_id`) USING BTREE,

KEY `propertyvalue\_product\_id` (`propertyvalue\_product\_id`) USING BTREE,

CONSTRAINT `propertyValue\_ibfk\_1` FOREIGN KEY (`propertyvalue\_property\_id`) REFERENCES `property` (`property\_id`) ON DELETE CASCADE,

CONSTRAINT `propertyValue\_ibfk\_2` FOREIGN KEY (`propertyvalue\_product\_id`) REFERENCES `product` (`product\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=716 DEFAULT CHARSET=utf8 ROW\_FORMAT=DYNAMIC COMMENT='产品属性管理表';

DROP TABLE IF EXISTS `review`;

CREATE TABLE `review` (

`review\_id` int(10) NOT NULL AUTO\_INCREMENT,

`review\_content` mediumtext NOT NULL COMMENT '内容',

`review\_createdate` datetime NOT NULL COMMENT '创建日期',

`review\_user\_id` int(10) NOT NULL COMMENT '关联用户id',

`review\_product\_id` int(10) NOT NULL COMMENT '关联产品id',

`review\_orderItem\_id` int(10) NOT NULL COMMENT '关联订单详细id',

PRIMARY KEY (`review\_id`) USING BTREE,

KEY `review\_user\_id` (`review\_user\_id`) USING BTREE,

KEY `review\_product\_id` (`review\_product\_id`) USING BTREE,

KEY `review\_orderItem\_id` (`review\_orderItem\_id`) USING BTREE,

CONSTRAINT `review\_ibfk\_1` FOREIGN KEY (`review\_user\_id`) REFERENCES `user` (`user\_id`),

CONSTRAINT `review\_ibfk\_2` FOREIGN KEY (`review\_product\_id`) REFERENCES `product` (`product\_id`),

CONSTRAINT `review\_ibfk\_3` FOREIGN KEY (`review\_orderItem\_id`) REFERENCES `productOrderItem` (`productorderitem\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=76 DEFAULT CHARSET=utf8 ROW\_FORMAT=DYNAMIC COMMENT='评论表';

DROP TABLE IF EXISTS `user`;

CREATE TABLE `user` (

`user\_id` int(10) NOT NULL AUTO\_INCREMENT,

`user\_name` varchar(25) CHARACTER SET utf8 COLLATE utf8\_bin NOT NULL COMMENT '用户名',

`user\_nickname` varchar(50) NOT NULL COMMENT '昵称',

`user\_password` varchar(50) CHARACTER SET utf8 COLLATE utf8\_bin NOT NULL COMMENT '密码',

`user\_realname` varchar(20) DEFAULT NULL COMMENT '姓名',

`user\_gender` tinyint(1) NOT NULL COMMENT '性别',

`user\_birthday` date NOT NULL COMMENT '出生日期',

`user\_address` char(6) NOT NULL COMMENT '所在地地址',

`user\_homeplace` char(6) NOT NULL COMMENT '家乡',

`user\_profile\_picture\_src` varchar(100) DEFAULT NULL COMMENT '用户头像',

`del\_flag` tinyint(1) DEFAULT '0' COMMENT '删除标识(1删除 0未删除）',

PRIMARY KEY (`user\_id`) USING BTREE,

UNIQUE KEY `un\_user\_name` (`user\_name`) USING BTREE,

KEY `user\_address` (`user\_address`) USING BTREE,

KEY `user\_homeplace` (`user\_homeplace`) USING BTREE,

CONSTRAINT `user\_ibfk\_1` FOREIGN KEY (`user\_address`) REFERENCES `address` (`address\_areaId`),

CONSTRAINT `user\_ibfk\_2` FOREIGN KEY (`user\_homeplace`) REFERENCES `address` (`address\_areaId`)

) ENGINE=InnoDB AUTO\_INCREMENT=23 DEFAULT CHARSET=utf8 ROW\_FORMAT=DYNAMIC COMMENT='用户表';

## 5.3数据表结构

表1 admin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 字段长度 | 描述 | 是否可为空 | 备注 |
| id | int | 10 | id | not null | 自增，主键 |
| uname | varchar | 20 | 用户名 | not null |  |
| uphone | varchar | 50 | 手机号 | not null |  |
| password | varchar | 20 | 密码 | not null |  |
| email | varchar | 20 | 邮箱 | not null |  |
| create\_time | timestamp |  | create\_time | not null | DEFAULT  CURRENT\_TIMESTAMP |
| update\_time | timestamp |  | update\_time | null |  |

表2 user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 字段长度 | 描述 | 是否可为空 | 备注 |
| id | int | 10 | id | not null | 自增，主键 |
| uname | varchar | 20 | 用户名 | not null |  |
| uphone | varchar | 50 | 手机号 | not null |  |
| password | varchar | 20 | 密码 | not null |  |
| email | varchar | 20 | 邮箱 | not null |  |
| create\_time | timestamp |  | create\_time | not null | DEFAULT  CURRENT\_TIMESTAMP |
| update\_time | timestamp |  | update\_time | null |  |

表3 user\_info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 字段长度 | 描述 | 是否可为空 | 备注 |
| id | int | 10 | id | not null | 自增，主键 |
| uname | varchar | 20 | 用户名 | not null |  |
| uphone | varchar | 50 | 手机号 | not null |  |
| password | varchar | 20 | 密码 | not null |  |
| email | varchar | 20 | 邮箱 | not null |  |
| img | varchar | 30 | 头像 | not null |  |
| update\_time | timestamp |  | update\_time | not null |  |
| create\_time | timestamp |  | create\_time | not null | DEFAULT  CURRENT\_TIMESTAMP |

表4 file\_info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 字段长度 | 描述 | 是否可为空 | 备注 |
| id | int | 10 | id | not null | 自增，主键 |
| filename | varchar | 20 | 文件名 | not null |  |
| uphone | varchar | 50 | 手机号 | not null |  |
| create\_time | timestamp |  | create\_time | not null | DEFAULT  CURRENT\_TIMESTAMP |
| update\_time | timestamp |  | update\_time | null |  |

表5 teacher

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 字段长度 | 描述 | 是否可为空 | 备注 |
| id | int | 10 | id | not null | 自增，主键 |
| uname | varchar | 20 | 用户名 | not null |  |
| uphone | varchar | 50 | 手机号 | not null |  |
| password | varchar | 20 | 密码 | not null |  |
| email | varchar | 20 | 邮箱 | not null |  |
| create\_time | timestamp |  | create\_time | not null | DEFAULT  CURRENT\_TIMESTAMP |
| update\_time | timestamp |  | update\_time | null |  |

表6 recycle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 字段长度 | 描述 | 是否可为空 | 备注 |
| id | int | 10 | id | not null | 自增，主键 |
| time | varchar | 20 | 文件删除时间 | not null |  |
| way | varchar | 50 | 文件原路径 | not null |  |
| filename | varchar | 20 | 文件名 | not null |  |
| create\_time | timestamp |  | create\_time | not null | DEFAULT  CURRENT\_TIMESTAMP |
| update\_time | timestamp |  | update\_time | null |  |

表7 sign

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 字段长度 | 描述 | 是否可为空 | 备注 |
| id | int | 10 | id | not null | 自增，主键 |
| time | varchar | 20 | 签到时间 | not null |  |
| date | varchar | 50 | 签到日期 | not null |  |
| create\_time | timestamp |  | create\_time | not null | DEFAULT  CURRENT\_TIMESTAMP |
| update\_time | timestamp |  | update\_time | null |  |

# 亮点设计

## 6.1购买商品



图6.1

## 6.2数据可视化



图6.2

# 运行截图

## 7.1注册登录

### 7.1.1注册

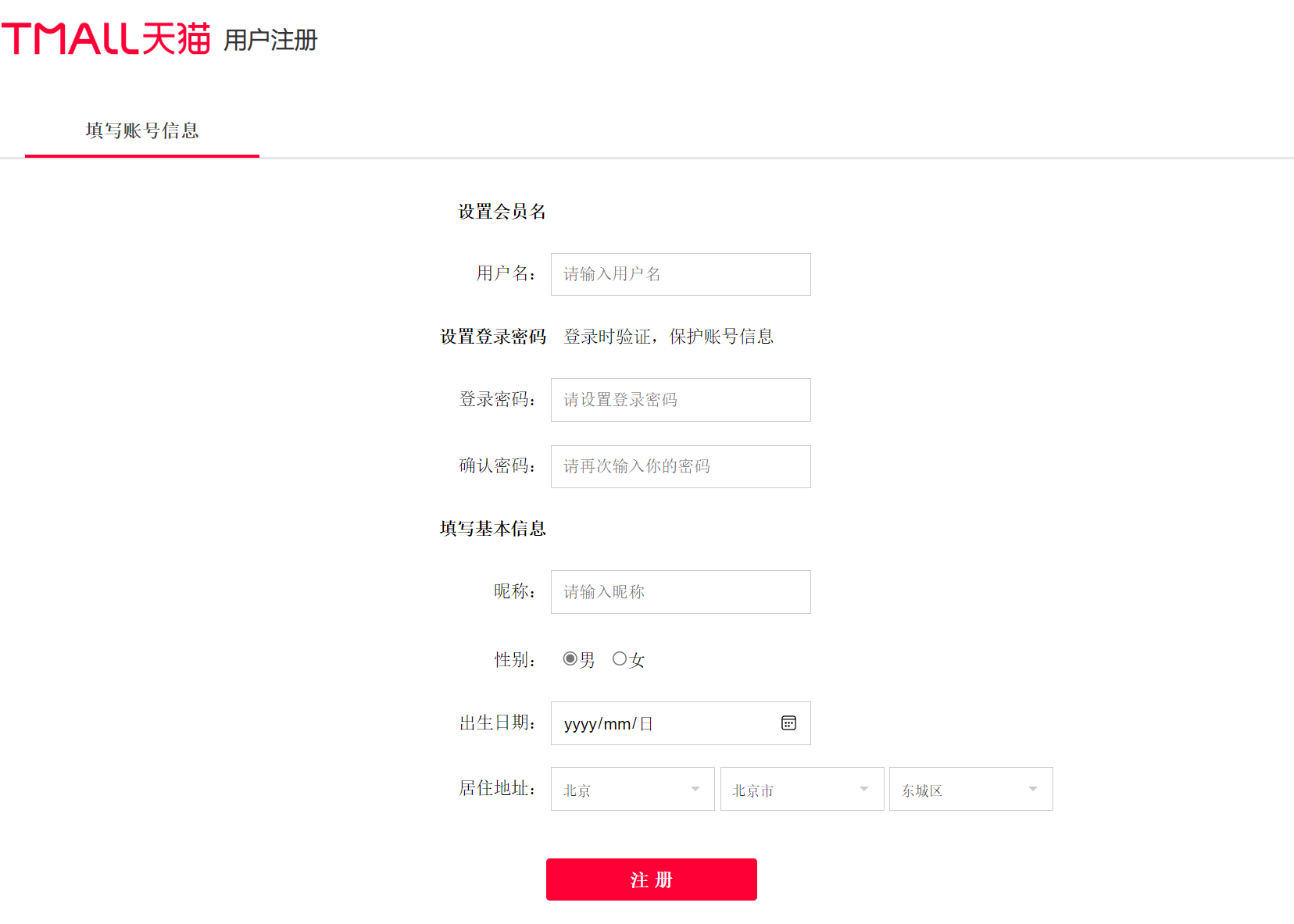


图7.1

### 7.1.2登录



图7.2

## 7.2购买界面

### 7.2.1商品界面

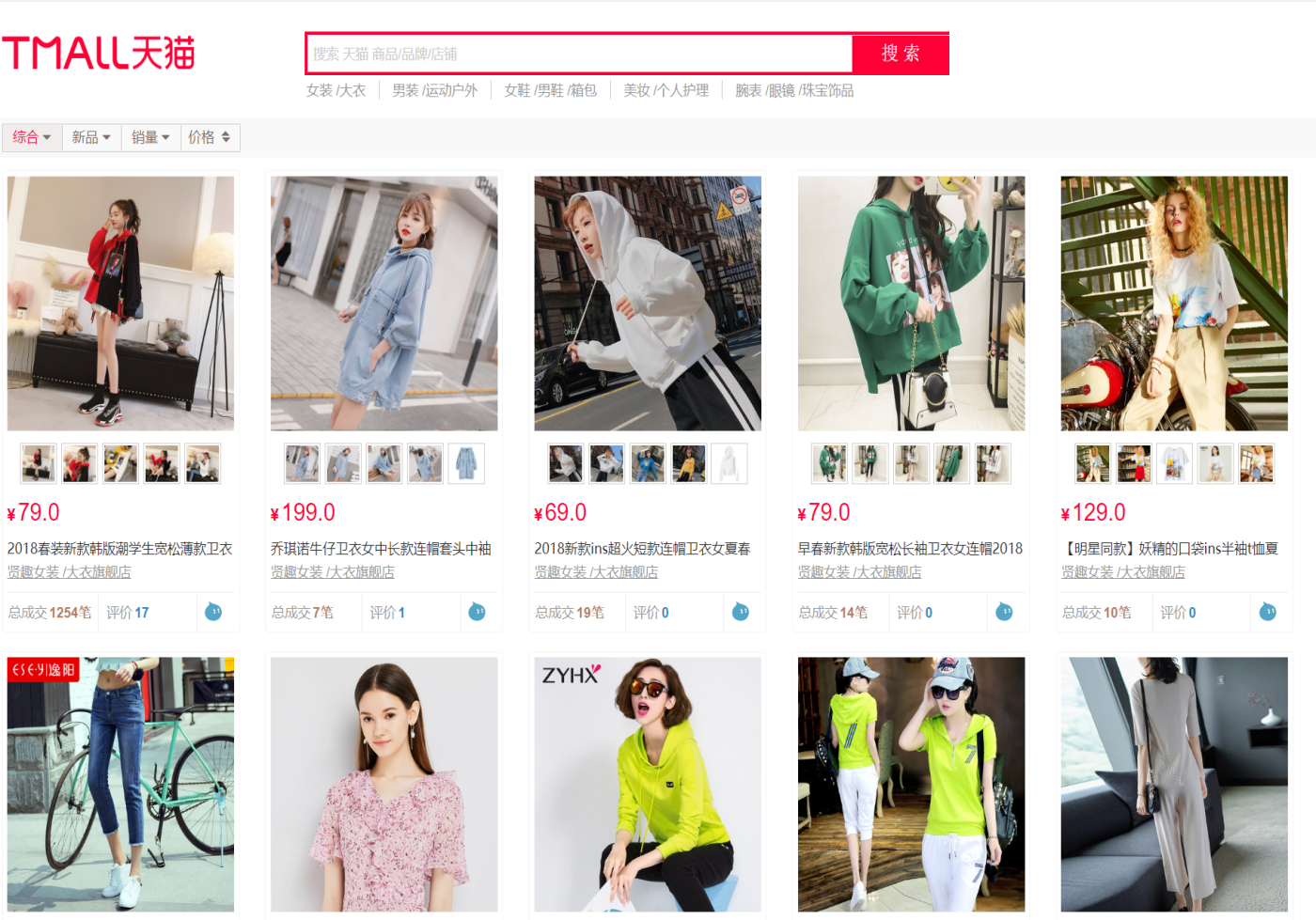


图7.3

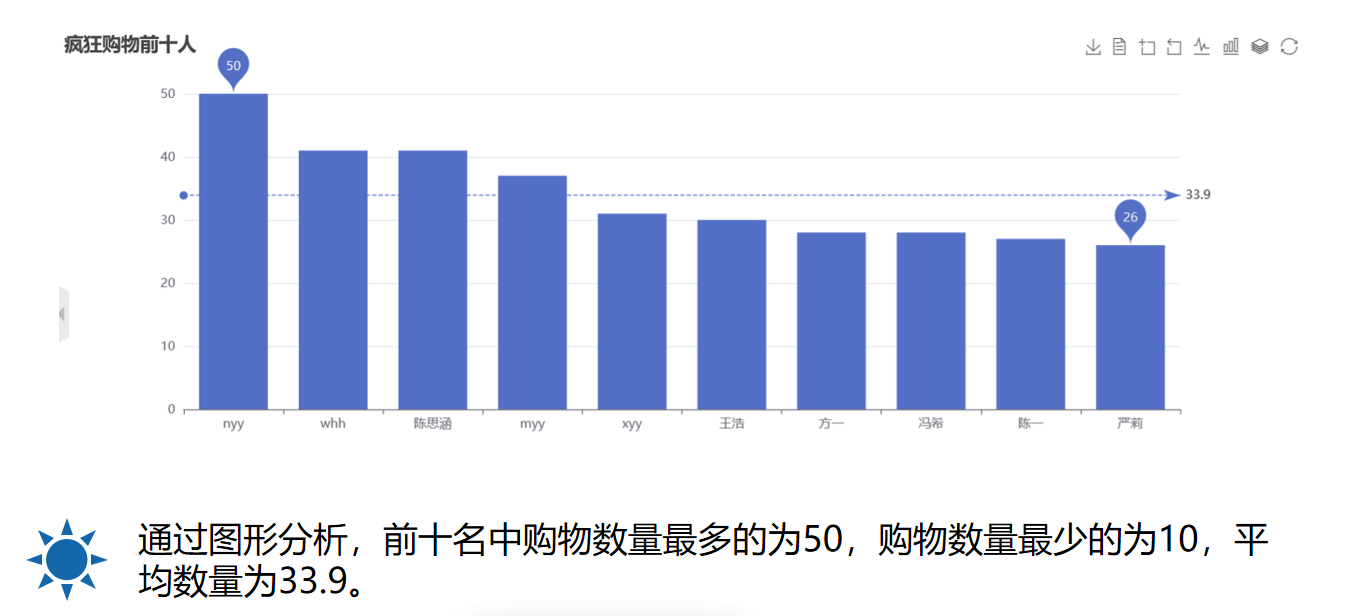
### 7.2.2购物界面



图7.5

## 7.3数据分析

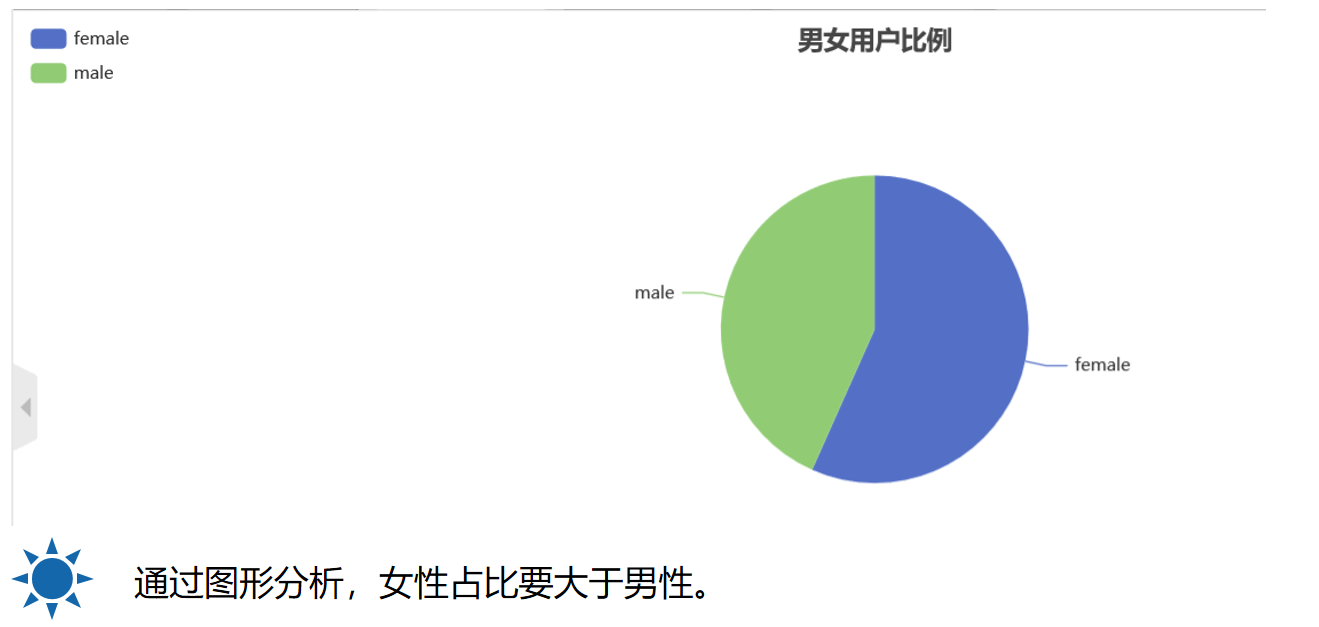
### 7.3.1统计购物总数最多的前十，并求出平均值



### 7.3.2商品类型分布

### 

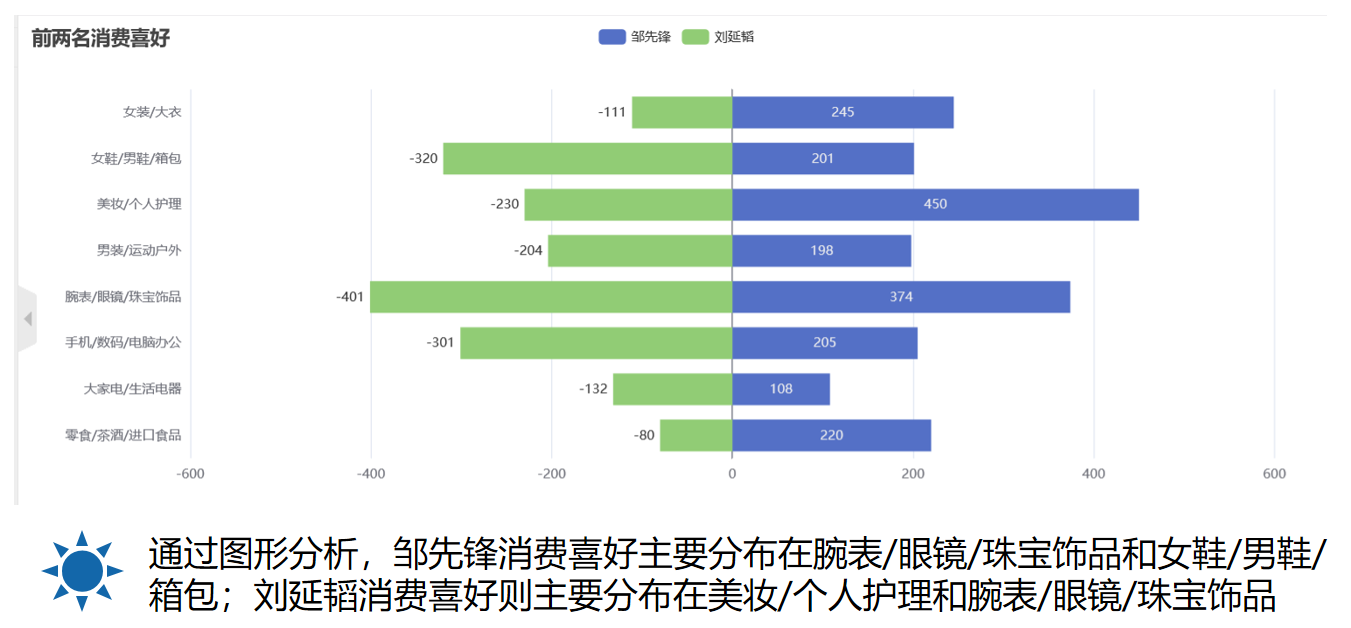
### 7.3.3统计男女用户比例



### 7.3.4男女每天消费金额比较

### 

### 7.3.5前两名用户的消费喜好



# 实习总结

在这次的实训中，我经历了更为接近企业真实软件项目的开发流程，了解了企业工作的真实要求，也更明白了自己的不足。这里和以往在学校工作室里和老师做项目的感觉很不一样，在这里，虽然这次实训只有短短的二十天，但是在实训老师的教导下，我收获很多，这是一次非常宝贵的经验。

第一周主要学习了一下Linux和Mobax的环境配置，在学习过程中认识到了自己Linux的基础过于薄弱，好在老师带领重新复习了一遍常用指令；我们还初始了Hadoop，学会了搭建HDFS集群和上传下载。第二周老师开始带领我们学习SpringBoot，从头开始带我们搭建随存通项目，登录注册、文件上传、下载、修改名字等等，讲解的非常细致，我感觉我代码能力有逐渐增强。第三周学习了大数据的计算MapReduce，因为没有学习过大数据，学习的有一点吃力，但是感受到了数据计算的有趣，同时我们开始了自己项目的编写，我负责部分的前端代码和后端代码的编写，通过老师同学们的指点和自己摸索，基本完成任务。第四周学习了Hive，比MapReduce代码简洁很多，作业完成较好，在这周我们也基本完善了自己的项目。

在过去的工作室项目开发中，老师教导的更多是项目的理论知识，从项目需求收集到项目设计到开发，过程更为全面，时间也很充裕。。但在实训中，实训老师的教导更为精简，直接的讲述所需实现功能代码，虽然有的代码我还是不太理解，但是已经可以基本参照着完成分配的功能。这次实训，我愈发的明白了学习和实践的重要性，在以后的学习和工作中，我会更加努力的。

实习工作评价与成绩评定

|  |
| --- |
| 学生自我鉴定：  对于本次实训，我做到了按时听讲认真听课，课下能够用按时自行完成作业和任务。遇到的问题基本可以自己查阅资料解决或者询问同学。对于小组项目的制定能够积极参与，能够完成分配给自己的任务。虽然后端代码写的不太好，但有积极参与前端和文档编写，在以后的学习和工作中会努力学习。 |
| 实习单位或指导教师意见（由实习企业填写）：  单位(盖章):  2021年 9月 1日 |
| 实习工作成绩（占综合成绩 50%）  实习报告成绩（占综合成绩 50%）  实习综合成绩(百分制)  指导教师签字:张鹏飞  2021年 9月 1日 |
| 评阅教师评定实习成绩:  评阅教师签名:  年 月 日 |

A：90~100分 A-：85~89分 B+：82~84分 B：78~81分 B-：75~77分

C+：72~74分 C：68~71分 C-：64~67分 D：60~63分 F：<60分