附件 1：

汕头大学大学生创新创业训练计划项目
进展日志

项目编号：201710560046

项目名称：“货比百家”的美妆导购系统

项目负责人：张敏华

项目组成员：刘彩君、卢程、何铭宜、林鹏珊

指导教师：蔡伟鸿

资助金额：8000 元

项目预计开展时间：2017 年 5 月 至 2018 年 5 月

表 1：　　研究（创新项目）记录表

<table>
<tr><td colspan="2">项目名称</td><td colspan="3">“货比百家”的美妆商品导购系统</td></tr>
<tr><td colspan="2">研究时间</td><td>2018.02.04</td><td>研究地点</td><td>汕头大学</td></tr>
<tr><td rowspan="6">参与人员</td><td>姓名</td><td>学号</td><td>年级专业</td><td>任务分工</td></tr>
<tr><td>张敏华</td><td>2014101021</td><td>14 计算机</td><td>参与讨论、功能设计</td></tr>
<tr><td>刘彩君</td><td>2014101043</td><td>14 计算机</td><td>参与讨论、后台开发</td></tr>
<tr><td>何铭宜</td><td>2014101023</td><td>14 计算机</td><td>参与讨论、后台开发</td></tr>
<tr><td>林鹏珊</td><td>2013010123</td><td>13 计算机</td><td>参与讨论、测试人员</td></tr>
<tr><td>卢程</td><td>2014011047</td><td>14 计算机</td><td>参与讨论、前端开发</td></tr>
<tr><td colspan="2"></td><td colspan="3">√工学 □理学 □农学 □医学 □经济学 □管理学
□法学 □哲学 □教育学 □文学 □历史学 □艺术学</td></tr>
</table>

一、　研究（实验）概述

搜索功能优化研究，用户对于需要的商品在前端进行输入关键词搜索，后台对用户的搜索进行处理，到数据库查询相关的商品，返回推荐列表。对于搜索功能需要进行性能优化，加快搜索的速度，提高用户体验。

二、　研究（实验）初步分析

（研究的工作原理、存在主要问题、限制条件、目前解决方案、已有专利、类似产品的解决方案、仍存在问题和不足等）

搜索功能存在的主要问题是数据库比较庞大，直接根据用户的搜索关键词去数据库使用 like 语句进行模糊查询效率会非常低，因为 like 语句是进行全表查询，当数据库中数据多的时候效率会很低，因此需要进行优化。

三、 研究（实验）拟解决问题

（运用创新方法解决问题，运用创新思维提出解决方案）

目前的解决方案是使用全文索引，数据库的搜索引擎使用 innodb，对于需要搜索的字段进行中文分词，因为全文索引是针对空格进行匹配的，我们的数据库如果存的是一整段描述，将无法匹配到符合的关键字，因此需要分词。所有的最短的索引字符串默认值为 4，这里需要改为 1，因为中文单词通常是两个字的，如果最小值为 4 则很多搜索都不能完成，修改最短的索引字符串后必须重建索引文件。

四、 研究（实验）技术方案和评价

MySQL 索引的建立对于 MySQL 的高效运行是很重要的，索引可以大大提高 MySQL 的检索速度。我们的搜索对需要匹配的列建立 fulltext 索引，当数据库数据很多的时候，对数据库的查询效率会有很大的提高。虽然索引大大提高了查询速度，同时却会降低更新表的速度。索引适合建立在数据更新不频繁的表上，否则维护索引会耗费很大的资源，反而会得不偿失，因为我们的数据查询比更新频繁很多，所以建立索引非常适合。

五、 研究（实验）结论

建立了全文检索之后，进行数据库的搜索，对比 like 模糊搜索，根据数据库返回搜索结果的时间显示，对于同样的关键字搜索，使用全文检索速度明显比使用 like 模糊搜索快。

六、 研究（实验）应用情况/现实指导意义

目前搜索功能的优化已基本实现。我们的项目数据都需要提前使用爬虫获取、处理后再存入数据库中，数据库的使用非常频繁，因此对于数据库使用的优化就尤为重要。做项目的时候不应该满足于实现功能，对于功能的优化也很重要，追求同样的功能用更好的方式实现，才能有进步，这才是我们做项目的意义所在。

七、 项目发表论文情况

论文是否正式发表过	否	**刊　号**	
期刊名称		**期刊期次**	

<table>
<tr><td colspan="2">项目名称</td><td colspan="3">“货比百家”的美妆商品导购系统</td></tr>
<tr><td colspan="2">研究时间</td><td>2018.03.08</td><td>研究地点</td><td>汕头大学</td></tr>
<tr><td rowspan="7">参与人员</td><td>姓名</td><td>学号</td><td>年级专业</td><td>任务分工</td></tr>
<tr><td>张敏华</td><td>2014101021</td><td>14 计算机</td><td>参与讨论、功能设计</td></tr>
<tr><td>刘彩君</td><td>2014101043</td><td>14 计算机</td><td>参与讨论、后台开发</td></tr>
<tr><td>何铭宜</td><td>2014101023</td><td>14 计算机</td><td>参与讨论、后台开发</td></tr>
<tr><td>林鹏珊</td><td>2013010123</td><td>13 计算机</td><td>参与讨论、测试人员</td></tr>
<tr><td>卢程</td><td>2014011047</td><td>14 计算机</td><td>参与讨论、前端开发</td></tr>
<tr><td colspan="2">学科类别：</td><td colspan="3">√工学 □理学 □农学 □医学 □经济学 □管理学
□法学 □哲学 □教育学 □文学 □历史学 □艺术学</td></tr>
</table>

一、 研究（实验）概述

相似商品功能：当用户点击商品的“找相似”按钮时，会跳转到相似商品的推荐页面，向用户推荐与当前选中商品相似度高的商品（同个商品类别、同个品牌等）

二、 研究（实验）初步分析

（研究的工作原理、存在主要问题、限制条件、目前解决方案、已有专利、类似产品的解决方案、仍存在问题和不足等）

相似商品功能：将计算商品的相似性转化为计算商品名称文本的相似度

三、 研究（实验）拟解决问题

（运用创新方法解决问题，运用创新思维提出解决方案）

基于商品名称的文本相似度

Tf-idf 模型：

1>将商品名称分词

2>列出所有的词

3>计算词频

4>形成词频向量

5>计算两个向量的余弦值，余弦值越大，相似度越高

$$\cos\theta = \frac{\sum_{i=1}^{n}(A_i \times B_i)}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} = \frac{A \cdot B}{|A| \times |B|}$$

Simhash 计算文本相似度:

1>将商品名称分词

2>找出词语库中关键词

3>将每个词语进行哈希操作

4>计算海明距离

四、 研究（实验）技术方案和评价

基于本项目中的样本数据，余弦值计算的时间长但准确率较高；simhash 计算的效率高于余弦值但准确率较低；当数据的基数较小时采用第一种方案，当数据海量时采用第二种方案。

五、 研究（实验）结论

通过计算商品名称的文本相似度来推荐商品还是比较合理的，一般来说，商品名称是精又简，且会涵盖该商品的关键词，推荐的商品精准度高

六、 **研究（实验）应用情况/现实指导意义** 相似商品推荐的实现增加了相似商品的曝光率，让用户能在短时间内浏览多种相似商品，可以进行比较后选择自己更为心仪的商品，适合当前快节奏生活

七、 项目发表论文情况			
论文是否正式发表过	否	**刊 号**	
期刊名称		**期刊期次**	

<table>
<tr><td colspan="2">项目名称</td><td colspan="3">“货比百家”的美妆商品导购系统</td></tr>
<tr><td colspan="2">研究时间</td><td>2018.01.17</td><td>研究地点</td><td>图书馆研讨室</td></tr>
<tr><td rowspan="6">参与人员</td><td>姓名</td><td>学号</td><td>年级专业</td><td>任务分工</td></tr>
<tr><td>张敏华</td><td>2014101021</td><td>14 计算机</td><td>参与讨论设计实现方案</td></tr>
<tr><td>刘彩君</td><td>2014101043</td><td>14 计算机</td><td>参与讨论设计实现方案</td></tr>
<tr><td>何铭宜</td><td>2014101023</td><td>14 计算机</td><td>参与讨论设计实现方案</td></tr>
<tr><td>林鹏珊</td><td>2013010123</td><td>13 计算机</td><td>参与讨论设计实现方案</td></tr>
<tr><td>卢程</td><td>2014011047</td><td>14 计算机</td><td>参与讨论设计实现方案</td></tr>
<tr><td colspan="2">学科类别：</td><td colspan="3">√工学 □理学 □农学 □医学 □经济学 □管理学
□法学 □哲学 □教育学 □文学 □历史学 □艺术学</td></tr>
</table>

一、 研究（实验）概述

前端界面的优化：

在中期答辩的基础上，根据系统所要实现的功能，设计、实现一个可用性更好的 Web 前端界面。

二、 研究（实验）初步分析

根据系统所要实现的功能，设计、实现一个可用性更好的 Web 前端界面。

美妆导购系统主要包括：模糊搜索主页、相似商品显示页面、降价通知商品页面、今日热搜商品页面，总体上来说，页面大同小异，所以可以把系统的页面分块，把可以通过传参而控制共用的页面封装起来，以减少代码的冗余。

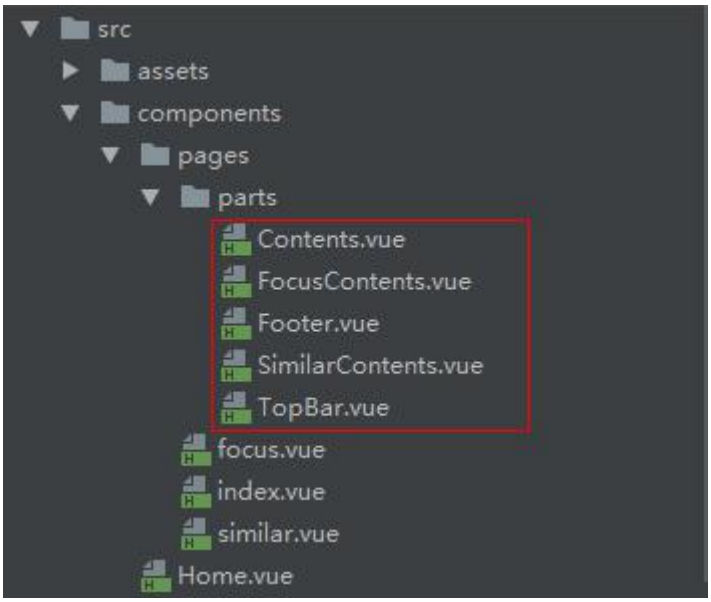
同时，由于降价通知商品页面是需要用户登录了才能使用的商品，所以前端的实现还需要配合服务器的 session 设置。

三、 研究（实验）拟解决问题

1. 如何在 vue.js 的框架应用下，把页面大同小异的部分封装起来，成为一个子组件，通过参数来控制页面显示的内容。
2. 如何配合服务器的 session（或 access_token）记录用户信息，并能取用。

四、 研究（实验）技术方案和评价

封装前端代码：

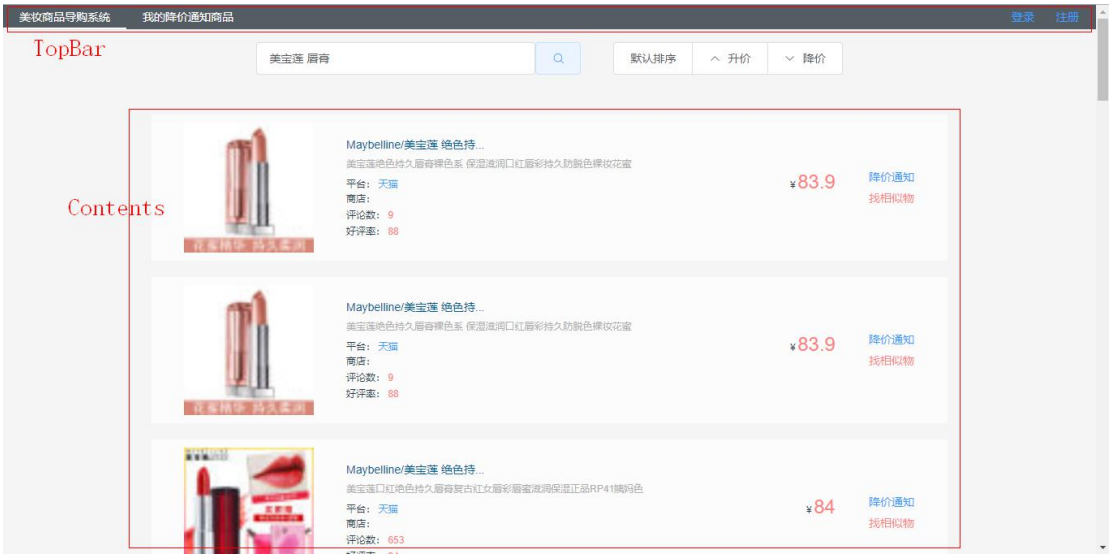


图上，我们把一个页面划分成三个基本的块——TopBar， Contents， Footer。

Footer 基本不会有什么变化，所以只需要保证在每个页面，其 Footer 都是在底部的，不仅需要保证其，在需要滚动条滚下的长页面中，是处于底部，还需要保证，Contents 部分所占的高度太小（没有撑满页面），Footer 能在可是范围的最底部。

Contents 部分是变化比较大的部分，其中显示的商品信息的界面，是根据不同的页面而不同。所以 Contents 还会分成 FocusContents 和 SimilarContents 来嵌套。

而 TopBar 看起来是最不起眼的，反而是最需要花心思的。“登录”按钮在用户登录之后，其对应文本将显示用户的昵称，且在 Contents 中是需要获取用户信息的，这时候需要把用户信息从 TopBar 中传到 Contents 使用。总而言之，TopBar 和 Contents 部分的数据传递次数比较频繁。



五、 研究（实验）结论

1. 把页面大同小异的部分封装起来，成为一个子组件，通过参数来控制页面显示的内容能够大大减少前端冗余代码。
2. 使用 session+cookie 记录用户信息能够方便某些需要用户信息的接口的调用。

六、 研究（实验）应用情况/现实指导意义

代码封装有效地减少了前端代码冗余，减少了工作量。使用 session+cookie 保证了缓存用户信息，方便接口调用。

七、 项目发表论文情况

论文是否正式发表过	否	**刊　号**	
期刊名称		**期刊期次**	

<table>
<tr><td colspan="2">项目名称</td><td colspan="3">“货比百家”的美妆商品导购系统</td></tr>
<tr><td colspan="2">研究时间</td><td>2018.02.22</td><td>研究地点</td><td>图书馆研讨室</td></tr>
<tr><td rowspan="6">参与人员</td><td>姓名</td><td>学号</td><td>年级专业</td><td>任务分工</td></tr>
<tr><td>张敏华</td><td>2014101021</td><td>14 计算机</td><td>参与讨论、实现接口</td></tr>
<tr><td>刘彩君</td><td>2014101043</td><td>14 计算机</td><td>参与讨论、查找资料</td></tr>
<tr><td>何铭宜</td><td>2014101023</td><td>14 计算机</td><td>参与讨论、完成文档</td></tr>
<tr><td>林鹏珊</td><td>2013010123</td><td>13 计算机</td><td>参与讨论、实现爬虫</td></tr>
<tr><td>卢程</td><td>2014011047</td><td>14 计算机</td><td>参与讨论、实现前端</td></tr>
<tr><td colspan="2">学科类别：</td><td colspan="3">√工学 □理学 □农学 □医学 □经济学 □管理学
□法学 □哲学 □教育学 □文学 □历史学 □艺术学</td></tr>
</table>

一、研究（实验）概述

商品降价通知功能的设计与实现。

二、研究（实验）初步分析

由于系统的总体目标是实现多个电商平台的商品比价，让用户可以搜索到物美价廉的商品，所以如果用户特别关注某个或某些商品的价格变动，且商品价格降低时，系统需要能够及时发送降价通知短信给用户，通知用户商品的价格降低了。

三、**研究（实验）拟解决问题**

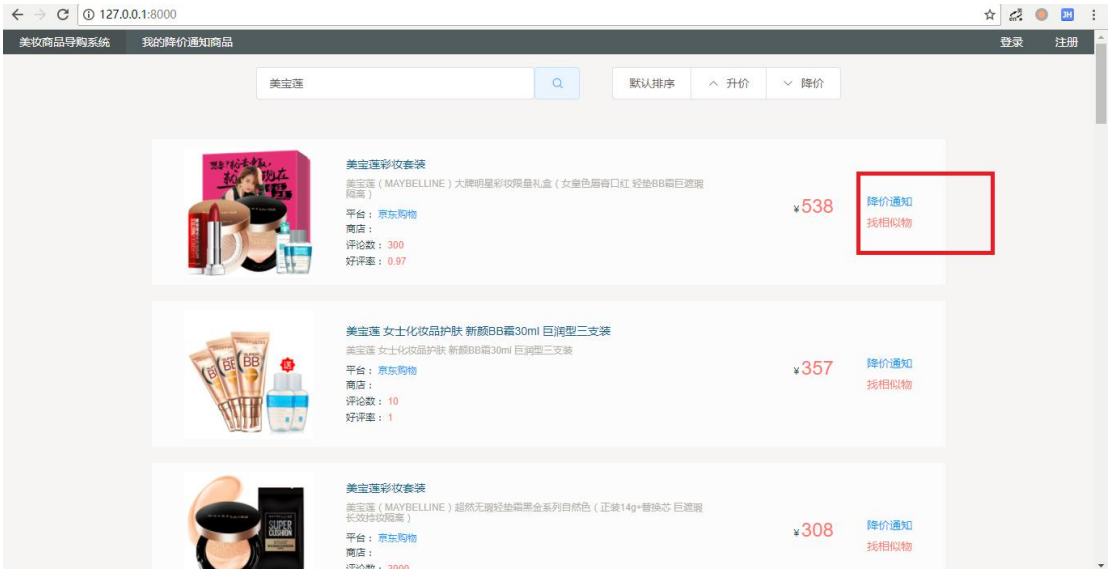
1)用户在系统的前端界面搜索商品后，可以点击商品信息展示区域的“降价通知”按钮，然后系统将用户和关注的商品信息录入数据库中。

2)系统通过爬虫实时更新商品价格时，如果发现某个商品新的价格比旧的价格低，则从数据库中获取关注该商品的用户信息

3)系统推送降价通知短信给用户

四、研究（实验）技术方案和评价

1） 用户在系统的前端界面搜索商品后，可以点击商品信息展示区域的“降价通知”按钮，对应的前端页面如下图：



对应接口设计如下：

请求 URL：

http://xx.com/beauty/cut_price/add_product

请求方式：

POST

参数：

参数名	必选	类型	说明
user_phone	是	string	当前登录用户的手机号码
item_url	是	string	item 原地址
name	是	string	商品名字
img_url	是	string	图片 url
price	是	int/string	价格
platform	是	string	平台
comment_count	是	string	评论总数

返回示例

成功：

```
{
  "error_code": 0,
  "msg": "success"
}
```

失败：

```
{
  "error_code": 1,
  "msg": "增加降价商品异常报错 XXXXX! "
}
```

返回参数说明

参数名	类型	说明
error_code	int	状态码，0：成功；1：失败
msg	string	状态信息，成功："success"；失败："xxxxxxx 报错信息 xxx

2) 系统将用户和关注的商品信息录入数据库中，数据库设计如下：

字段	类型	空	默认	注释
id	int(10)	否		
use_phone	varchar(20)	否		用户手机号码
product_address	varchar(50)	否		商品的 url
product_name	varchar(50)	否		商品的名称
product_img_url	varchar(50)	否		商品图片
product_current_price	varchar(50)	否		商品当前价格
product_current_platform	varchar(50)	否		商品的平台

3) 系统通过爬虫实时更新商品价格时，如果发现某个商品新的价格比旧的价格低，则从数据库中获取关注该商品的用户信息，然后推送降价通知短信给用户，这部分的主要的代码如下：

```
# 每小时更新一次商品价格
def update_price_comment(self,product = ProductLipstick):
    while True:
        try:
            url = product.address
            new_price = ""
            new_comment_count = ""
            old_comment_count = ""
            new_good_comment_rate = ""

            if (self.checkPlatfrom(url) == 1):
                skuId = url.split('/')[-1].strip(".html")
                print(skuId)
                new_price = self.get_jd_price(skuId)
                # print(new_price)
                result = self.get_jd_comment(skuId)
                new_comment_count = result['comment_count']
                print(new_comment_count)
                new_good_comment_rate = result['good_comment_rate']
                old_comment_count = product.new_comment_count
                print(old_comment_count)
            new_price = float(new_price)
```

```
                    old_price = float(product.price)

                    print("更新前的价格" + str(old_price))
                    print("更新后的价格" + str(new_price))
                    print("更新前的评论总数" + str(product.comment_count)+"第 2 次"+str(product.new_comment_count))

                    # 更新价格和降价通知
                    if new_price != old_price:
                        if new_price < old_price:
                            # 短信通知
                            self.inform_price_reduce(url)
                            pass
                        # 更新价格
                        product.price = new_price
                        print(product.price)
                    # 更新评论总数
                    product.comment_count = product.new_comment_count
                    product.new_comment_count = new_comment_count
                    print("更新后的评论总数" + str(product.comment_count) + str(product.new_comment_count))
                    # 更新好评率
                    product.good_comment_percentage = new_good_comment_rate
                    product.get_time = Now()
                    print("更新后的好评率" + str(new_good_comment_rate))
                    product.save()
                except Exception as e:
                    print(e)
                    pass
                time.sleep(60)#自动休眠，每一小时爬一次数据
```

4) 运行结果：每当用户关注的商品价格降低时，用户可以收到短信通知，通知内容如下：

五、研究（实验）结论

该方案能够完成降价通知的功能，但存在一个问题：用户有时候可能会重复关注同一个商品，多次点击前端页面的“降价通知”按钮；如果同一个用户重复点击同一个商品时，都将用户关注的商品录入数据库，数据库将多次很多冗余数据。解决问题的方案是：当用户第二次点击同一个商品的“降价通知”按钮时，提示用户“该商品已经关注，可以到【我的降价通知商品】页面查看”，然后前端页面增加【我的降价通知商品】页面，用户可以在该页面浏览自己关注的商品。另外，数据库表中将用户的手机号码和商品的唯一标识设为主键，确保唯一性

六、研究（实验）应用情况/现实指导意义

这次研究的系统功能“降价通知功能”是中期检查答辩时，老师建议增加的功能，我们小组通过讨论分析，发现这个功能是较为重要的，实现了这个功能，系统便能更加快速地给用户推送商品价格变动的信息。

七、项目发表论文情况

论文是否正式发表过	否	**刊　号**	
期刊名称		**期刊期次**	

<table>
<tr><td colspan="2">项目名称</td><td colspan="3">“货比百家”的美妆商品导购系统</td></tr>
<tr><td colspan="2">研究时间</td><td>2018.03.20</td><td>研究地点</td><td>图书馆研讨室</td></tr>
<tr><td rowspan="6">参与人员</td><td>姓名</td><td>学号</td><td>年级专业</td><td>任务分工</td></tr>
<tr><td>张敏华</td><td>2014101021</td><td>14 计算机</td><td>参与讨论设计实现方案</td></tr>
<tr><td>刘彩君</td><td>2014101043</td><td>14 计算机</td><td>参与讨论设计实现方案</td></tr>
<tr><td>何铭宜</td><td>2014101023</td><td>14 计算机</td><td>参与讨论设计实现方案</td></tr>
<tr><td>林鹏珊</td><td>2013010123</td><td>13 计算机</td><td>参与讨论设计实现方案</td></tr>
<tr><td>卢程</td><td>2014011047</td><td>14 计算机</td><td>参与讨论设计实现方案</td></tr>
<tr><td colspan="2">学科类别：</td><td colspan="3">√工学 □理学 □农学 □医学 □经济学 □管理学
□法学 □哲学 □教育学 □文学 □历史学 □艺术学</td></tr>
</table>

一、 研究（实验）概述

天猫平台商品数据抓取

二、 研究（实验）初步分析

（研究的工作原理、存在主要问题、限制条件、目前解决方案、已有专利、类似产品的解决方案、仍存在问题和不足等）

爬取天猫平台的美妆商品的信息，包括商品的基本信息、价格、商品评论数和好评率等等，采集到大量的数据便于后续的数据分析和对比。

三、 研究（实验）拟解决问题

（运用创新方法解决问题，运用创新思维提出解决方案）

上网查找相关资料、查阅相关书籍、小组讨论、编写程序。

四、 研究（实验）技术方案和评价

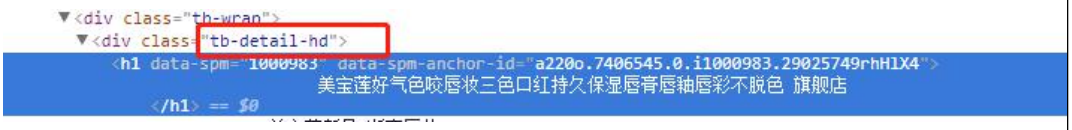
1）以天猫平台的唇彩化妆品做为研究对象；
2）通过分析天猫平台商品的详情的网页标签，以获取商品详情数据
3）分析天猫反爬虫的机制，并对其进行处理

五、 研究（实验）结论

1. 天猫平台的标签分析

① 商品标题

标签：



程序实现：

```
#获取标题
def __getDescription(self):
    if self.pageSoup is not None:
        title=self.pageSoup.find("div",class_="tb-detail-hd").find("h1").string
        strip()
        return title
```

② 商品详细信息

标签：

程序实现：

```
#获取产品名称、颜色分类、保质期、功效、适合肤质、产地
def __getProductDetail(self):
    if self.pageSoup is not None:
        params=["产品名称","颜色分类","保质期","功效","适合肤质","产地"]
        results={}
        for param in params :
            results[param]="null"
        try:
            attributes = self.pageSoup.find('ul',attrs={"id":"J_AttrUL"}).find_all("li")
            for attr in attributes:
                if '\xa0' in attr:
                    attr=attr.replace(u'\xa0',u'')
                attr = attr.string.strip()
                for i in range(len(params)):
                    strIndex = re.search(params[i],attr)
                    if strIndex is not None:
                        end = strIndex.span()[1]
                        #因可能出现“化妆品保质期：36个月”这种情况，所以需要知道“保质期”的最后匹配
                        置，然后截取，才能得到干净的字段
                        #results[ params[i] ] = attr[len(params[i])+1:]
                        results[ params[i] ] = attr[end+1:]
                        del params[i]
                        break
```

③ 商品价格信息

价格信息是通过异步请求得到，请求地址例如：

https://mdskip.taobao.com/core/initItemDetail.htm?&itemId=5290850

64706

程序实现：

```
#获取每一种型号的价格
def __getPrices(self):
    url="https://mdskip.taobao.com/core/initItemDetail.htm?&itemId={}".format(self.id)
    header={
        'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, li
        Gecko) Chrome/64.0.3282.186 Safari/537.36',
        'Referer':"https://detail.tmall.com/item.htm?id={}".format(self.id)
        #这里应对阿里云服务器的反爬虫策略
    }
    html=requests.get(url,headers=header).text
    #print(html)
    try:
        datas=json.loads(html.strip())
    except(JSONDecodeError,Exception):
        print("JSONDecodeError")
        return None
    try :
        prices = datas['defaultModel']['itemPriceResultDO']['priceInfo']
        keys = list(prices.keys())
        result = None
        if keys is None:
            raise Exception('NullException')
        key = keys[0]
```

④ 商品评论信息

价格信息是通过异步请求得到，请求地址例如：

https://dsr-rate.tmall.com/list_dsr_info.htm?itemId=529085064706

程序实现：

```
#获取商品评价总数和好评率
def __getComments(self):
    comments={}
    url="https://dsr-rate.tmall.com/list_dsr_info.htm?itemId={}".format(self.id)
    html=requests.get(url).text
    tmp = re.search("\(.*\)",html).span()
    html = html[tmp[0]:tmp[1]][1:-1]
    datas=json.loads(html)
    comments['评论总数']=datas['dsr']['rateTotal']
    comments['好评率'] = round(float(datas['dsr']['gradeAvg'])/5*100,2)
    print(comments)
    return comments
```

2. 天猫平台的反爬虫机制和解决对策

① 问题：在天猫首页时爬取数据时，需要登录

解决方案：有两种，一种是在请求的头部添加 cookie 字段，另一种是通过天猫搜索功能中添加搜索“唇彩”关键字搜索可得，可以顺利绕开天猫登录机制。程序实现中采用第二种解决方法，因为第一种的 cookie 值是有有效期的。

② 问题：爬取价格信息时，遇到 403

解决方案：在程序中，在发起请求的头部添加 referer 字段，即可解决。

3. 最终抓取的数据结果

```
mysql> select count(*) from product_lipstick;
+----------+
| count(*) |
+----------+
|     1138 |
+----------+
```

六、 **研究（实验）应用情况/现实指导意义**

该实验作为整个系统的基础，对系统核心比价功能的实现和运用提供数据支撑。

七、 项目发表论文情况

论文是否正式发表过	无	**刊　号**	
期刊名称		**期刊期次**	

附件 2：

项目小组研讨与交流会议纪要

时间	2017-12-15	地点	线上讨论	记录人	刘彩君
参加人员	张敏华、林鹏珊、何铭宜、刘彩君、卢程				
会议主题	模糊搜索功能实现思路和接口设计				

会议记录：

会议主要内容：
1）制订接下来一个月的任务计划；
2）进行任务分工并确定完成时间。

会议过程以及会议结果：

编号	任务内容	参与人员	完成时间
1	前端设计和实现	卢程	2018.01.10
2	统一整理数据库表的信息	张敏华	2018.01.10
3	模糊搜索功能后端代码实现	何铭宜	2018.01.10
4	模糊搜索功能后端代码实现	刘彩君	2018.01.10
6	收集淘宝平台的唇彩数据	林鹏珊	2018.01.10
7	对数据库相应内容更新	各自更新自己爬取的内容	2018.01.10

备注：数据库有些商品已下架，还有新商品上架，需要更新数据库内容，每人负责更新自己负责的彩妆类别。

时间	2018-01-03	地点	微信群讨论	记录人	林鹏珊
参加人员	张敏华、何铭宜、刘彩君、卢程、林鹏珊				
会议主题	讨论在爬虫时遇到的反爬虫机制与解决方案				

会议记录

<1>分析遇到的反爬虫机制

问题：对淘宝平台美妆商品主题页抓取商品 id 时，出现输入验证码的问题。

<2>讨论之后的解决方案

针对验证码处理的方法有三种，分别是：

第一种：把验证码 down 到本地之后，手动输入验证码验证，此种成本相对较高，此时不能做到完全自动抓取，需要认为干预；

第二种：图像识别验证码，自动填写验证，但是验证码噪声较多复杂度较大，而且组内没有熟悉图像识别技术的组员，故较难实现；

第三种，接入自动打码平台。

综合讨论以上的处理方法和组员的技术架构，以及成立项目的目标之后，最终选择更换平台，改为天猫平台。

时间	2018-01-23	地点	微信	记录人	何铭宜
参加人员	张敏华、林鹏珊、何铭宜、刘彩君、卢程				
会议主题	相似商品功能实现方案讨论				

会议记录：

会议主要内容：

（1）汇报前两周的任务进展、制订接下来一周的任务计划；

（2）讨论相似商品功能

会议结果：

相似商品的功能方案：

相似性的计算是基于商品名称的文本相似度，因为商品名称既短又精，包含关键词

Tf-idf 模型：

6>将商品名称分词

7>列出所有的词

8>计算词频

9>形成词频向量

10>计算两个向量的余弦值，余弦值越大，相似度越高

$$\cos\theta = \frac{\sum_{i=1}^{n}(A_i \times B_i)}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} = \frac{A \cdot B}{|A| \times |B|}$$

Simhash 计算文本相似度:

5>将商品名称分词

6>找出词语库中关键词

7>将每个词语进行哈希操作

8>计算海明距离

返回相似度高的前 n 个商品

备注：需要测试两种方案的时间复杂度和准确率来择优选择

编号	任务内容	参与人员	完成时间
1	相似商品功能接口实现	何铭宜	2018.03.25
2	相似商品功能接口前后端对接	何铭宜、卢程	2018.03.28

时间	2018-02-10	地点	G 座教学楼	记录人	何铭宜
参加人员	张敏华、林鹏珊、何铭宜、刘彩君、卢程				
会议主题	相似商品功能实现思路和接口设计				

会议记录：

会议主要内容：

（3）汇报前两周的任务进展；

（4）讨论相似商品功能实现思路和接口设计

会议结果：
接口设计如下：

请求 URL：

- `http://xx.com/beauty/productsList/getAllSimilarProducts?category=*&pname=*`

请求方式：

- GET

参数：

参数名	必选	类型	说明
category	是	string	商品类别，决定在哪张表查找数据（数据库表的字段名 third_category）
pname	是	string	点击的商品对应的商品名称

返回示例

```
{
    "error_code": 0,
    "msg": "success",
```

```
    "data": [
    {'address': 'https://item.jd.com/1973278112.html',
    'name': '阿玛尼（ARMANI） 阿玛尼 ARMANI 口红 唇膏 唇釉 红管#501 玫瑰豆沙色 热卖',
    'description': '阿玛尼（ARMANI） 阿玛尼 ARMANI 口红 唇膏 唇釉 红管#501 玫瑰豆沙色 热卖',
    'price': 358.0,
    'platform': '京东全球购',
    'comment_count': 800,
    'img1_address': 'https://img12.360buyimg.com/n5/s75x75_jfs/t5602/357/361649357/215874/8d02a82c/591ecd0aNc4ead577.jpg',
    'good_comment_percentage': '94%'
    },
    {'address': 'https://item.jd.com/10752537660.html',
    'name': '美宝莲新品上市绝色持久丝绒雾感唇釉 唇彩 唇膏 口红 滋润持久 01',
    'description': '美宝莲新品上市绝色持久丝绒雾感唇釉 唇彩 唇膏 口红 滋润持久 01',
    'price': 109.0,
    'platform': '京东商场',
    'comment_count': 8,
    'img1_address': 'https://img10.360buyimg.com/n5/jfs/t8719/364/885795549/339192/3c1652a1/59b0ac67N5c7c0368.jpg',
    'good_comment_percentage': '87%'
    }
  ]
  }
失败：
  {
    "error_code": 1,
    "msg": "搜索异常报错 XXXXX！比如搜索不出对应的产品"
  }
```

返回参数说明

参数名	类型	说明
error_code	int	状态码，0：成功；1：失败
msg	string	状态信息，成功：” success” ；失败：” xxxxxxx 报错信息 xxx”

name	string	商品名字
price	float	价格
img1_address	string	图片 url
address	string	商品链接 url
good_comment_percentage	string	商品好评率
comment_count	int	商品评论总数
platform	string	平台
description	string	描述

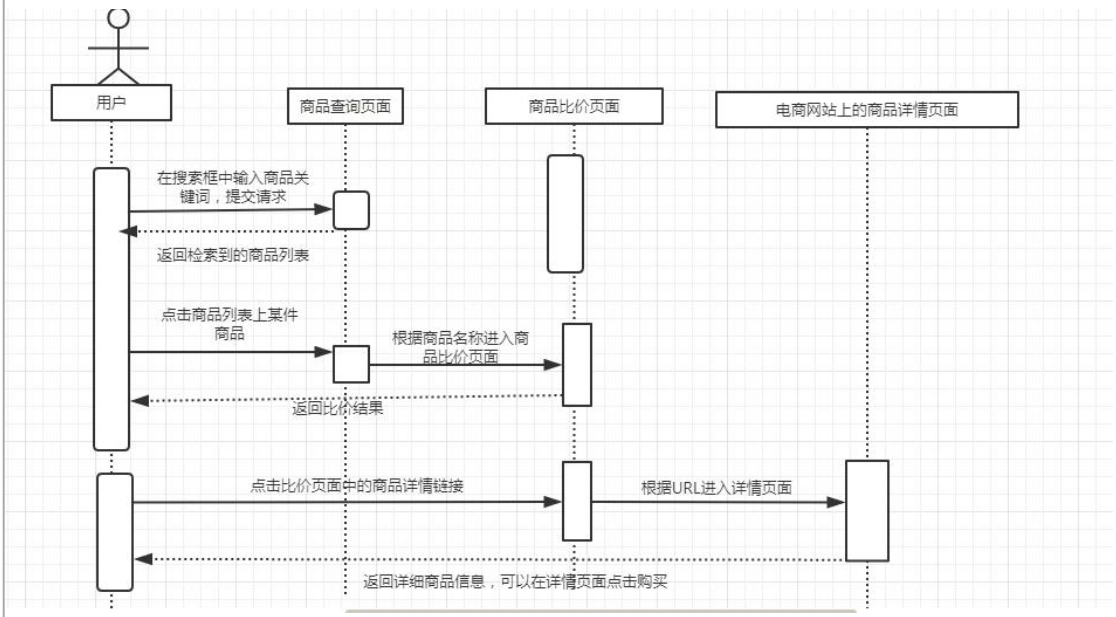
时间	2018-03-01	地点	微信群讨论	记录人	林鹏珊
参加人员	张敏华、何铭宜、刘彩君、卢程、林鹏珊				
会议主题	设计系统的比价功能				

会议记录

<1>明确比价功能的需求

需求明确：系统可以为用户给出某种商品在各种平台的比价结果。

操作流程：



<2>比价功能设计

(1) 平台的选择：京东、天猫和淘宝

(2) 比价排序规则：

暂定为：根据价格和好评率各占 50%的计算结果从大到小排序

(3) 实现步骤：先实现各大平台上的唇彩类型实现比价功能，之后再实现其他美妆类型的比价功能。

<3>讨论分工

成员	分工
张敏华	收集天猫平台的唇彩数据

<table>
<tr>
<td>何铭宜</td>
<td>实现接口：
根据用户点击查询商品列表上的某件商品，根据商品名称返回价格比价信息：包括价格来源网站，该网站该商品的商品详情链接、商品价格
<table>
<tr><td>url</td><td>http://127.0.0.1:8000/product/getAllProductPrice</td></tr>
<tr><td>请求类型</td><td>Post</td></tr>
<tr><td>请求参数</td><td>{"productName": "美宝莲 20ml 清淡型唇彩" }</td></tr>
<tr><td>请求参数格式</td><td>json</td></tr>
<tr><td>返回结果</td><td>[{"productId": "1",
"productName":
"美宝莲 20ml 清淡型唇彩" ,
"productImage":"xxx.jpg",
"priceList":{“priceSource”
“京东商品”，
“price”：“34.00”，
” detailLink“:"xxx.jd.com",
},...]</td></tr>
<tr><td>返回结果格式</td><td>json</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><td>刘彩君</td><td colspan="2">实现接口：
根据用户输入的关键词查询商品，返回指定页数的商品列表;一页默认展示 20 条</td></tr>
<tr><td></td><td>url</td><td>http://127.0.0.1:8000/productsList/getProductsPage</td></tr>
<tr><td></td><td>请求类型</td><td>Post</td></tr>
<tr><td></td><td>请求参数</td><td>{"searchKeyWords":"美宝莲唇彩",
"PageNo":1 ,
"PageNumbers":20 }</td></tr>
<tr><td></td><td>请求参数格式</td><td>json</td></tr>
<tr><td></td><td>返回结果</td><td>[{"productId": "1",
"productName": "美宝莲 20ml 清淡型唇彩" ,
"productImage":"xxx.jpg",
"productXXX":??},
{"productId": "2",
"productName": "美宝莲 20ml 清淡型唇彩 2" ,
"productImage":"xxx.jpg",
"productXXX":??},...]（20 条）</td></tr>
<tr><td></td><td>返回结果格式</td><td>json</td></tr>
<tr><td></td><td colspan="2"></td></tr>
<tr><td>林鹏珊</td><td colspan="2">收集淘宝平台的唇彩数据</td></tr>
<tr><td>卢程</td><td colspan="2">优化网站前端</td></tr>
</table>

<4>讨论完成时间

暂定在 3 月 12 日前完成。

时间	2018-03-12	地点	线上讨论	记录人	张敏华
参加人员	张敏华、林鹏珊、何铭宜、刘彩君、卢程				
会议主题	1.讨论通过爬虫实时更新商品的数据 2.任务分配				

会议记录：

会议内容

1. 讨论并确定需要更新的商品属性：

- 商品价格
- 评论总数
- 好评率

原因：这三个属性值随着时间而变化，而其他的商品属性值一般不变

2. 讨论如何利用 python 的多线程机制来爬虫实现“实时”更新商品的价格、评论总数、好评率

结论：可以利用 python 的 threading 模块实现多线程爬虫，利用 time 模块实现间隔固定时间进行数据更新

3. 初步确定实时更新方案：

第一步：利用爬虫脚本首次爬取数据

第二步：将爬取的数据进行处理，然后录入数据库

第三步：每间隔 1 小时根据商品的 url 重新爬取商品的价格、评论总数、好评率这三个值

第四步：将新的数据和原本的数据进行对比，如果不同，则更新数据库中的数据

4. 分配任务：

成员	分工
刘彩君	负责完善爬虫脚本
何铭宜	
张敏华	负责实现实时更新数据的程序
林鹏珊	
卢程	

任务完成时间均为 2018-03-26 号。

时间	2018-03-26	地点	线上讨论	记录人	张敏华
参加人员	张敏华、林鹏珊、何铭宜、刘彩君、卢程				
会议主题	1. 讨论降价通知功能的实现 2. 任务分配				

会议记录：

会议内容

1. 讨论并确定降价通知功能的需求：如果用户特别关注某个或某些商品的价格变动，且商品价格降低时，系统需要能够及时发送降价通知短信给用户，通知用户商品的价格降低了

2. 讨论如何实现该功能，并确定初步实现方案如下：

① 用户在系统的前端界面搜索商品后，可以点击商品信息展示区域的“降价通知”按钮，然后系统将用户和关注的商品信息录入数据库中。

② 系统通过爬虫实时更新商品价格时，如果发现某个商品新的价格比旧的价格低，则从数据库中获取关注该商品的用户信息

③ 系统推送降价通知短信给用户

3. 分配任务：

成员	分工
刘彩君	修改并完善实时更新商品价格的爬虫脚本
何铭宜	实现子功能：短信通知的代码
张敏华	设计接口文档，完成降价通知接口的代码
林鹏珊	完成将用户和关注的商品信息录入数据库的代码
卢程	完成降价通知功能的前端代码

任务完成时间均为 2018-04-09 号。

时间	2018-04-10	地点	G 座教学楼	记录人	卢程
参加人员	张敏华、林鹏珊、何铭宜、刘彩君、卢程				
会议主题	前端和后台代码的整合和模糊搜索功能优化讨论				

会议记录：

会议主要内容：

（5）汇报前两周的任务进展；

（6）对前端、后台代码进行整合；

（7）讨论模糊搜索功能的的优化方案；

会议过程以及会议结果：

编号	任务内容	参与人员	备注
1	汇报前两周的任务进展	全体成员	各成员汇报工作的进展和完成情况；
2	对前端、后台代码进行整合	全体成员	部署环境，利用 github 对前端后台代码进行整合
3	讨论模糊搜索功能的的优化方案	全体成员	各成员就搜索精确度和所需时间对模糊搜索功能进行优化方案探讨。

1. 汇报前两周的任务进展

张敏华：爬取唯品会的商品信息。已成功爬取某类型的商品信息。

林鹏珊：尝试爬取淘宝、天猫的商品信息，反爬虫技术很严，有一定难度，还在尝试。

何铭宜：查找资料，设计找相似商品的实现方案，并编写接口文档。已了解相关技术，正在尝试写 demo。

刘彩君：查找资料，设计找模糊搜索的实现方案，并编写接口文档。已了解相关技术，正在尝试写 demo。

卢程：根据成员的意见修改界面，进一步完善界面代码，做好对接接口的工作。

2. 对前端、后台的代码进行整合

全员利用 Github 和 pycharm，对各自代码进行整合并存放在 Github，建立合作者的共同开发模式，各成员笔记本配置好 numpy、 scipy、 gensim、 jieba、 sklearn 等 python 模块，保证各成员能够在各自电脑成功拉取 Github 服务器上最新的代码、能够推送代码到 Github 服务器。

3. 讨论模糊搜索功能的的优化方案

就终期答辩之前的搜索功能进行探讨，探讨的结果是，由于系统的数据库数据量比较大，既要保证模糊搜搜功能的实时性，又要保证其准确性，我们必须采用更为有效的方案。

经过各成员的讨论，我们计划使用 jieba 分词模块，将商品的名称和描述做分词处理，再根据用户的搜索输入进行分词搜索。