**名牌手表在线销售系统**

**系统需求分析说明书**

**变更记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 变更说明 | 作者 |
| YYYY-MM-DD | 1.0 | 初始版本 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**签字确认**

|  |  |  |  |
| --- | --- | --- | --- |
| 系统模块 | 对应章节 | 对应部门 | 负责人签字 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目录**

[1 概述 4](#_Toc521482059)

[1.1 系统开发背景 4](#_Toc521482060)

[1.2 系统开发意义 4](#_Toc521482061)

[1.3 参考资料 4](#_Toc521482062)

[1.4 任务概述 5](#_Toc521482063)

[1.4.1 目标 5](#_Toc521482064)

[1.4.2 用户的特点 5](#_Toc521482065)

[2 运行环境 5](#_Toc521482066)

[2.1 软件环境 5](#_Toc521482067)

[2.2 硬件环境 6](#_Toc521482068)

[3 需求规定 6](#_Toc521482069)

[3.1 用户角色以及功能需求描述 6](#_Toc521482070)

[3.2 功能框图 7](#_Toc521482071)

[4 关键技术、代码以及截图 8](#_Toc521482072)

[4.1 关键技术及代码 8](#_Toc521482073)

[4.2 功能截图 12](#_Toc521482074)

# 概述

## 系统开发背景

名牌手表在线销售系统顾名思义就是网店系统，作为电子商务的一种形式，是一种能够让人们在浏览的同时进行实际购买，并且通过各种支付手段进行支付完成交易全过程的系统，是电子零售商业的典型组织形式，是建立在Internet上的商场。一个可以让顾客用计算机购物且有送货上门服务、商人可以提供产品服务，又可以缩减维护实际店面管销成本的地方。它改变了从传统的店面销售到网络上虚拟销售的一种模式，从很大程度上节约了购物时所需的人力，物力。

现在随着Internet的高速发展和个人、企业上网的迅速普及，企业的WWW网站在商业活动中也发挥着越来越大的作用。在信息时代，建立一个功能强大，界面美观的电子商务网站，建立电子商务系统平台，对企业的发展是至关重要的。

## 系统开发意义

在信息化高速发展的今天，时间，效率这样的名词正主导着人们的生活和发展。网店通过廉价的通讯手段，将买家与卖家、厂商和合作伙伴紧密结合在一起，消除了时间和空间带来的局限性，从而大大节约了交易成本，扩大了交易范围。

## 参考资料

1. Java核心技术
2. 《数据库系统概论》
3. 《网页设计与Web数据库发布技术》
4. Spring框架技术
5. Mybatis框架技术
6. AJAX技术

## 任务概述

### 目标

本系统通过强大的计算机技术给卖家和顾客带来便利。目标包括：

1.减少业务人员与管理支出；

2.提高信息准确度；

3.改进管理和服务；

4.建立高效的信息传输和服务平台，提高信息处理速度和利用率；

5.系统设计优良，界面设计精美、友好、快捷，人性化，后台管理功能强大、效率高；

6.更简便、信息化程度更高的网上销售管理流程；

### 用户的特点

本软件的用户是面向管理员、顾客，他们都具有基本的计算机应用知识，可以对计算机进行一些简单的操作。

# 运行环境

## 软件环境

操作系统：Windows 7

数据库平台：Mysql

语言：Java、JavaScript

技术：Spring框架、Mybatis框架、AJAX、Java web。

## 硬件环境

|  |  |  |
| --- | --- | --- |
| **硬件配置** | **最低配置** | **推荐配置** |
| 服务器 | 1CPU：P4 2.0G | 1CPU：P4 2.8G |
| Mem：1G | Mem：2G |
| HD：40G | HD：120G |
|  |  |
|  |  |
| 客户端 | 1CPU：P4 1.0G | 1CPU：P4 2.0G |
| Mem：512M | Mem：1G |
| HD：40G | HD：120G |
|  |  |

# 需求规定

## 用户角色以及功能需求描述

|  |  |
| --- | --- |
| 角色 | 功能描述 |
| 普通用户 | 按类型、品牌、销量、相关产品、产品关键字查找手表，查看手表详情，将手表加入购物车并管理购物车，通过个人中心查看订单、修改用户信息 |
| 管理员 | 进行手表信息、品牌的增删改查以及对用户订单状态进行操作 |

## 功能框图

网上购物

查看手表

查找手表

购物车

订单管理

购物

车管理

注册登录

用户管理

查看手表

管理模式

名牌手表销售系统

手表管理

# 关键技术、代码以及截图

## 关键技术及代码

Spring+Mybatis+Ajax+Javascrip实现手表添加至购物车，并更新左上角购物车的件数。在加入购物车的过程中判断手表是否存在，若存在则修改手表数量，不存在则新增手表信息。

代码:

@PostMapping("/SingleChk")

**public** **void** SingleChk(Model model,HttpSession session,HttpServletRequest request, HttpServletResponse response) **throws** IOException{

PrintWriter out = response.getWriter();

String watch\_id=request.getParameter("watch\_id");

**if**(session.getAttribute("user")!=**null**){

Cart ca = cartMapper.chkCart(Integer.*parseInt*(watch\_id),((User) session.getAttribute("user")).getUserid());

**if**(ca!=**null**){

**int** Tcount=1+ca.getCartcount();

**int** i=cartMapper.updateCart(Tcount,ca.getCartid());

String tip="N";

out.print(tip);

}

**else**{

**int** i = cartMapper.insertCart(Integer.*parseInt*(watch\_id),1,((User) session.getAttribute("user")).getUserid());

String tip="Y";

out.print(tip);

}

}

**else**{

String tip="W";

out.print(tip);

}

}

<script>

**function** chk(watch\_id){

**var** watch\_id =watch\_id;

$.post

(

"${pageContext.request.contextPath}/SingleChk",{"watch\_id":watch\_id},

**function**(data){

**if**(data=="N"){

alert("添加成功，在购物车等待亲");

}

**else**{

**if**(data=="W"){

alert("添加失败，你尚未登陆，请登陆再进行购物");

}

**else** **if**(data=="Y")

{

alert("添加成功，在购物车等待亲");

**var** a=document.getElementById("ph1").innerHTML;

**var** b=(a-0)+1;

**var** c=document.getElementById("ph1").innerHTML=b;

}

}

}

)

}

</script>

Spring+Mybatis实现用户结算时，将订单信息添加至订单表。

@SuppressWarnings("unchecked")

@GetMapping("/editAll")

**public** String editAll(Model model,HttpSession session,Order order,HttpServletRequest request){

**int** j = ((User) session.getAttribute("user")).getUserid();

List<Cart> c=(List<Cart>)session.getAttribute("CartOrder");

**for**(Cart a:c){

**int** w\_count=a.getCartcount();

**int** watch\_id=a.getWatch\_id();

order.setW\_ordertime((**new** Date()).toString());

order.setState("未送达");

order.setUser\_id(((User) session.getAttribute("user")).getUserid());

order.setW\_count(w\_count);

order.setWatch\_id(watch\_id);

**int** k = orderMapper.insertOrder(order);

**int** m = watchMapper.updateWcount(w\_count,w\_count,watch\_id);

}

**int** i = cartMapper.deleteAllCart(j);

model.addAttribute("Cart",cartMapper.findAllCart(((User) session.getAttribute("user")).getUserid()));

model.addAttribute("Ccart", cartMapper.findCCart(((User) session.getAttribute("user")).getUserid()));

model.addAttribute("RandWatch",watchMapper.findRandWatch());

System.***out***.print(i);

**return** "index";

}

实现手表的关键字搜索、按男士、女士、销量来对手表进行筛选

@GetMapping(value="/men/{type}")

**public** String men(@PathVariable("type") String type,@RequestParam(value="search",required=**false**) String watchname,Model model,HttpSession session){

**if**(type.equals("搜索")){

model.addAttribute("Typewatch",watchMapper.findSearchWatch(watchname));

model.addAttribute("watch1", brandsMapper.findAllBrands());

model.addAttribute("Cwatch",watchMapper.findCWatch());

}

**else**{

**if**(type.equals("热销")){

model.addAttribute("Typewatch",watchMapper.findXlWatch(type));

model.addAttribute("watch1", brandsMapper.findAllBrands());

model.addAttribute("Cwatch",watchMapper.findCWatch());

}**else**{

model.addAttribute("watch", watchMapper.findAllWatch());

model.addAttribute("Typewatch",watchMapper.findTypeWatch(type,type));

model.addAttribute("watch1", brandsMapper.findAllBrands());

model.addAttribute("Cwatch",watchMapper.findCWatch());

}}

**if**(session.getAttribute("user")!=**null**){

model.addAttribute("Ccart", cartMapper.findCCart(((User) session.getAttribute("user")).getUserid()));

}

**return** "men";

}

Java+XML实现对用户的监听，若存在相同用户重复登录，则将第一个登录的用户踢掉

**public** **class** LoginListener **implements** HttpSessionAttributeListener {

//只要有人登录，就把登录信息保存map,

//key-》用户名

**private** Map<String,HttpSession> sessionMap = **new** HashMap<String,HttpSession>();

@Override

**public** **void** attributeAdded(HttpSessionBindingEvent sessionEvent) {

// **TODO** Auto-generated method stub

System.***out***.println(sessionEvent.getName()+"：名字");

System.***out***.println(sessionEvent.getValue()+"：值");

//保存在session里面的是否跟用户相关，约定user

**if**("user".equals(sessionEvent.getName())){

//取出值

//整个用户对象保存在session

User user = (User) sessionEvent.getValue();

//判断前面是否登录

String uName = user.getUsername();

**if**(sessionMap.containsKey(uName)){

//得到原来session，踢掉

HttpSession orgSession =sessionMap.get(uName);

orgSession.invalidate();

}

//新的session保存起来

sessionMap.put(uName, sessionEvent.getSession());

}

}

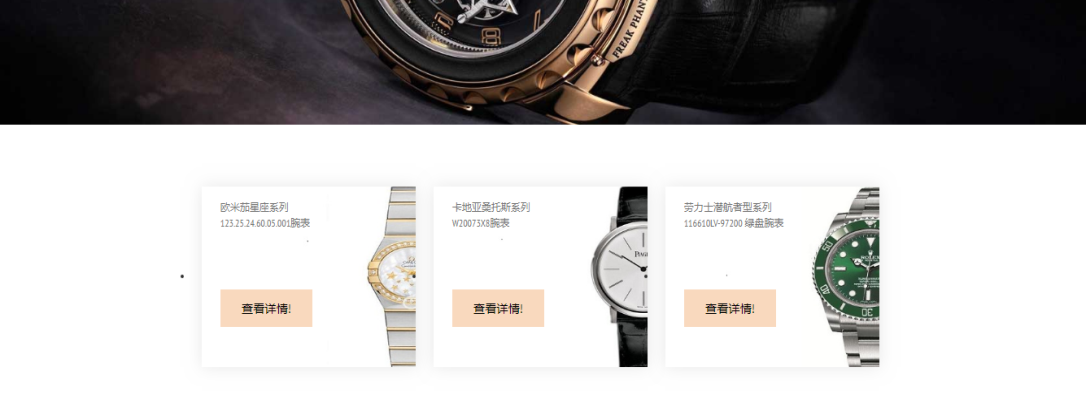
## 功能截图

首页：用户未登录状态

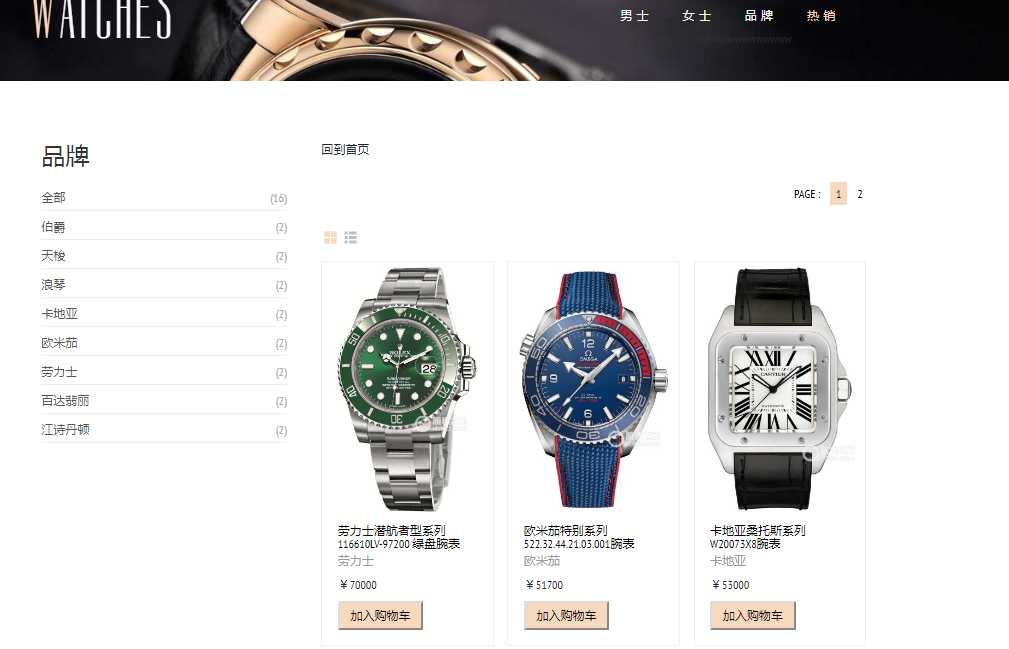
首页:用户已登录状态，显示用户名，点击进入用户个人中心



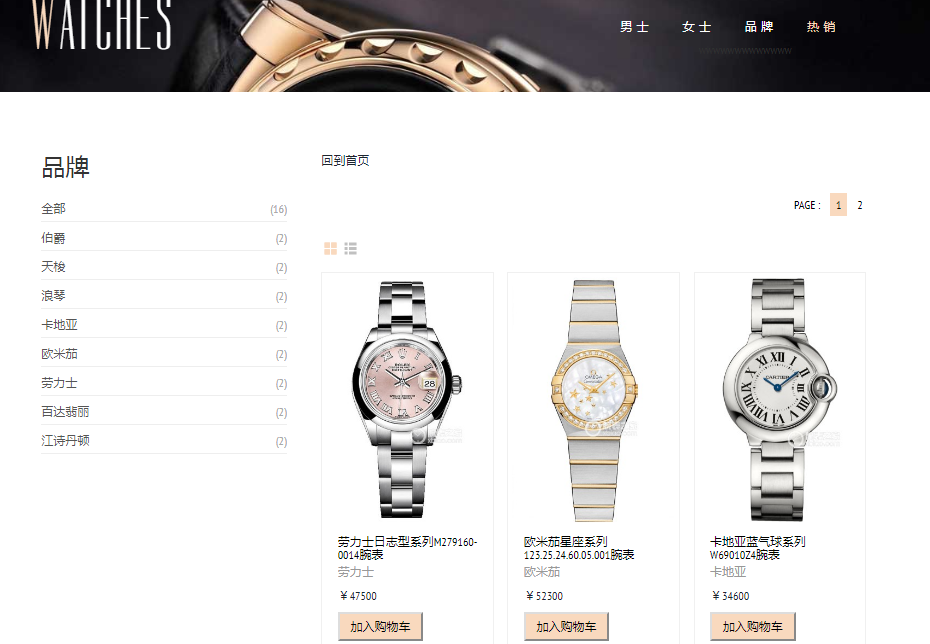
首页:在首页实现数据库中3条手表的随机生成 ,点击查看详情进入单品信息



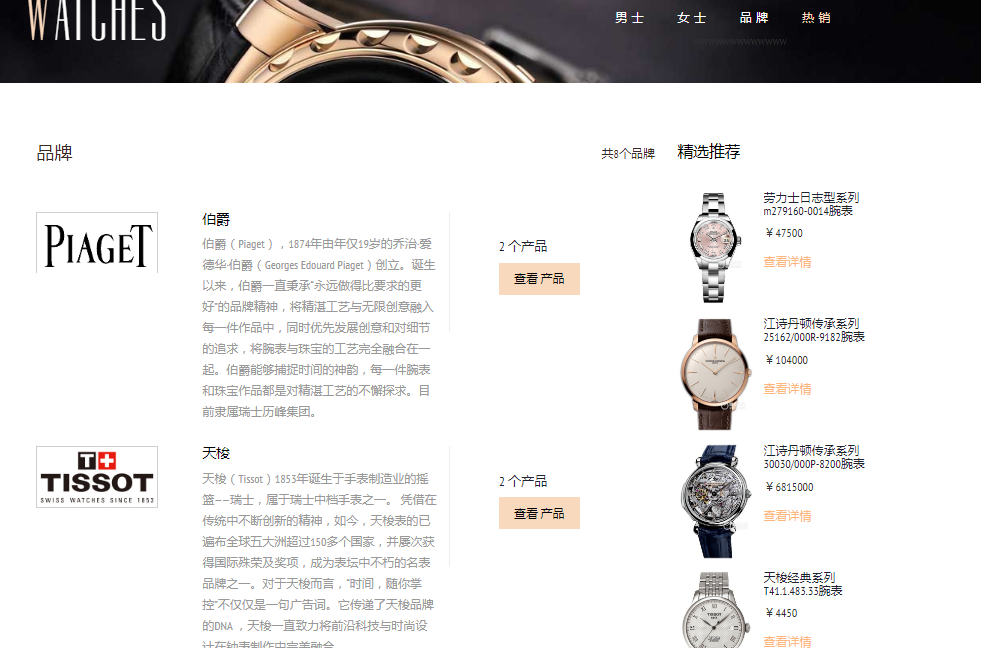
手表页面:点击男士进入相应的手表展示页面，点击手表可用查看详情，点击全部可查看全部的手表，也可以选择对应的品牌进行筛选



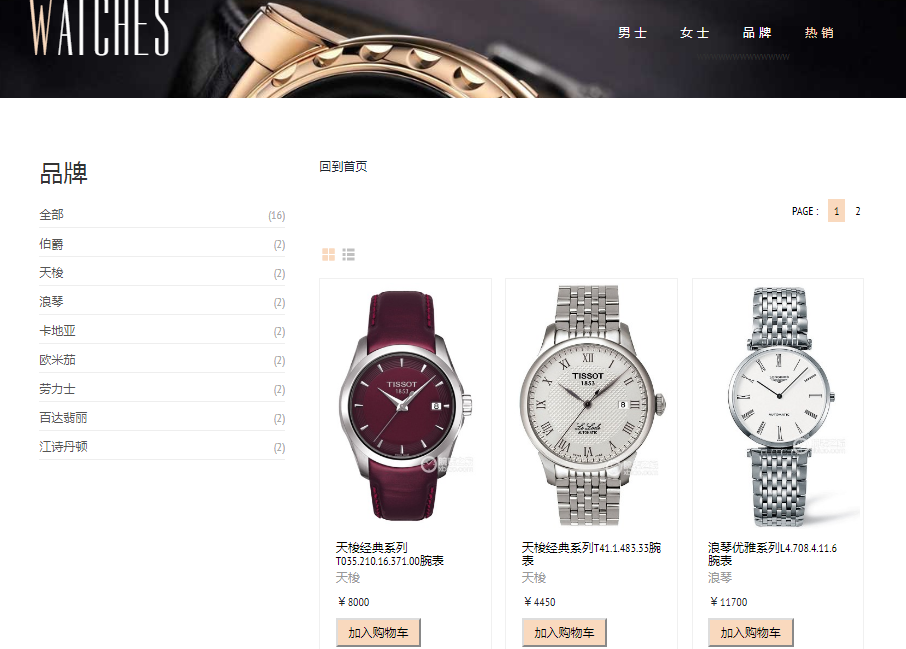
手表页面:点击女士进入相应的手表展示页面，点击手表可用查看详情



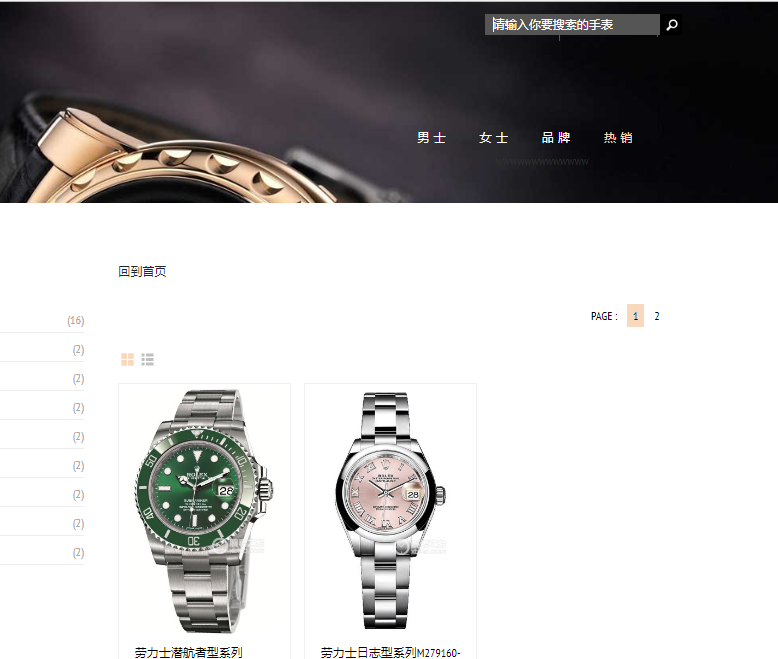
品牌页面:点击品牌，查看所有品牌,点击查看产品可以看到该产品的手表

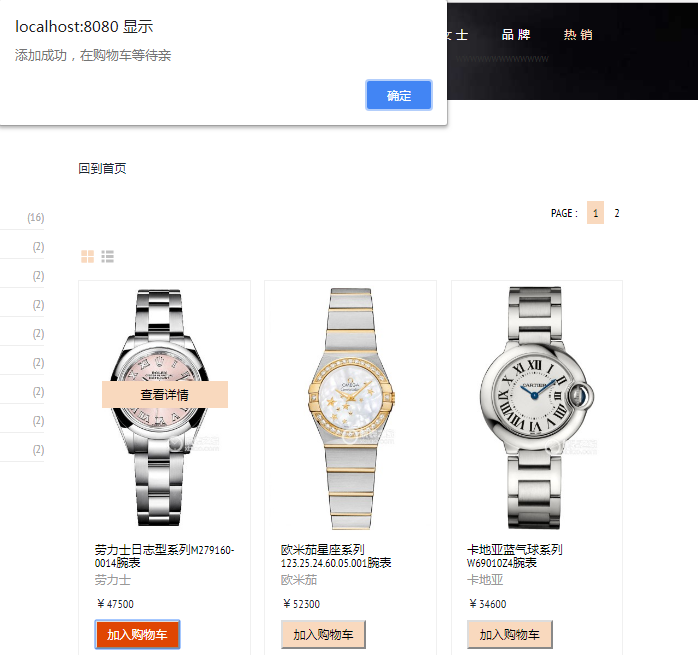


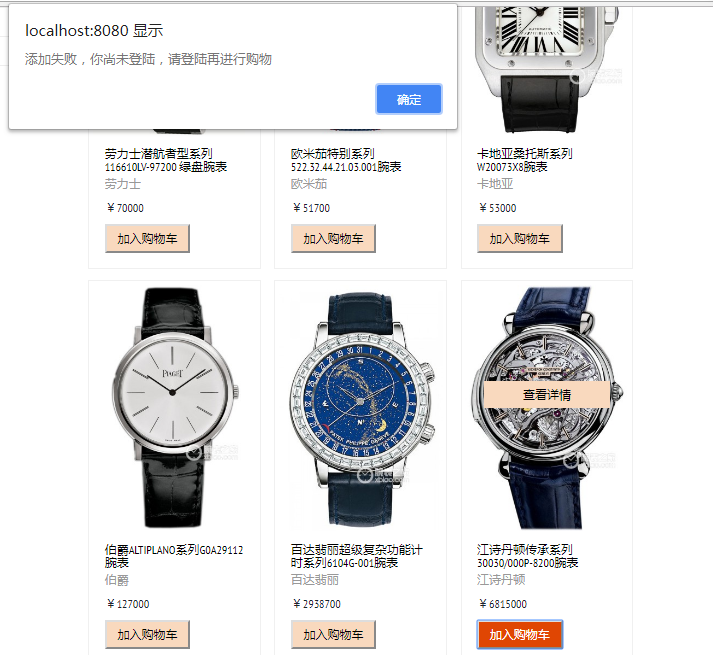
手表页面:点击热销，跳转到手表页面，出现6条销量最高的手表，并按顺序排列



手表页面:点击右上角的搜索框，搜索手表名字关键字，筛选出相应的手表



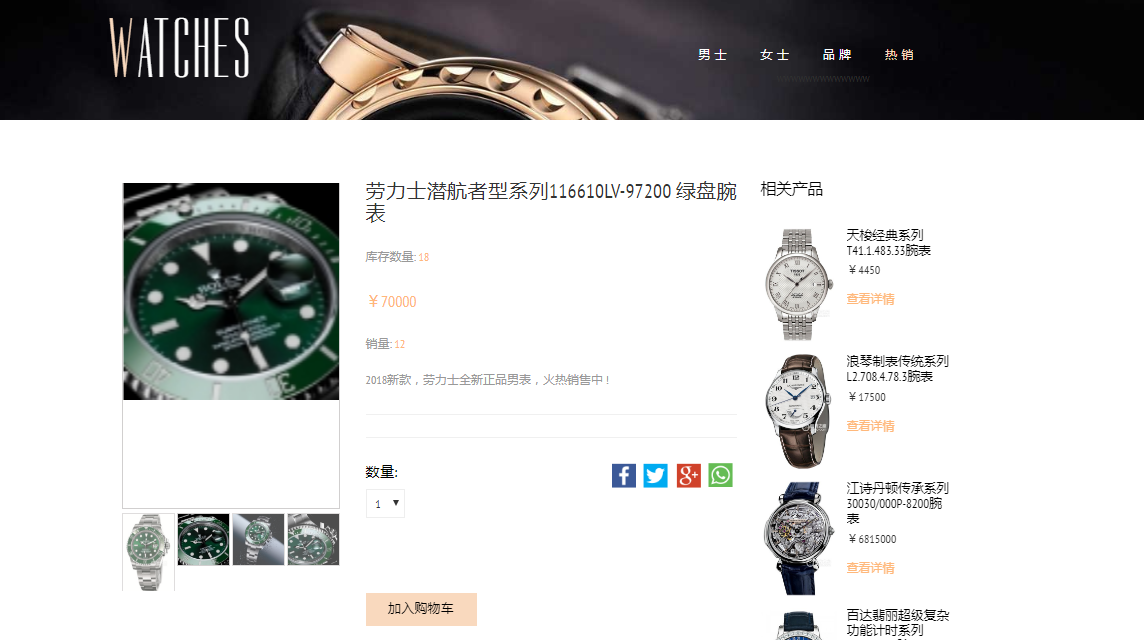
手表页面:点击手表下方的加入购物车，若未登录，弹窗提示，已登录，则加入购物车，并判断是否已存在相同的手表，若存在，则数量加1



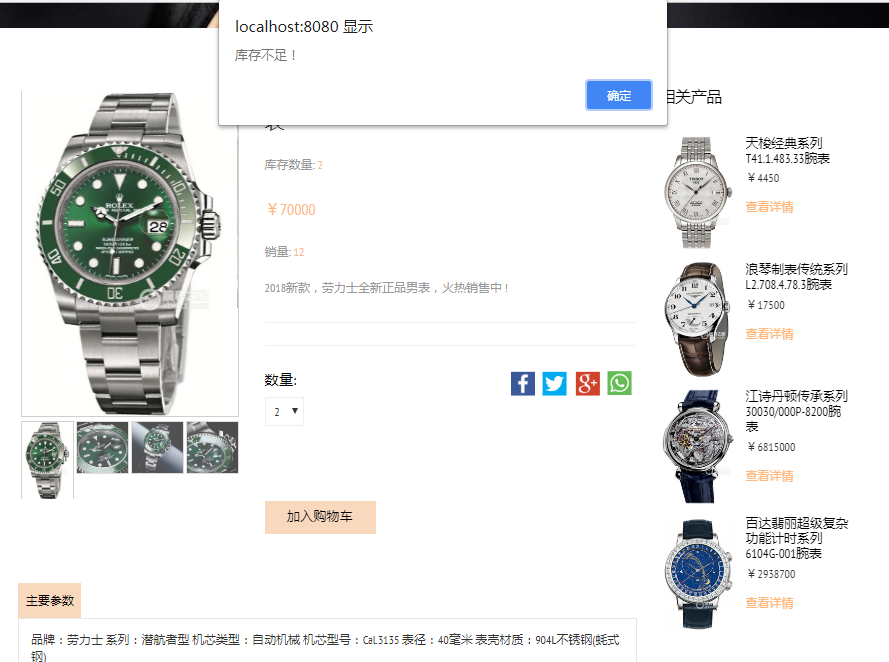
任意页面:加入购物车显示购物车的手表数量,点击清空购物车则把数量置0并删除该用户的购物车表.



单品页面:点击手表进入单品页面,可以查看该手表的详细信息,选择购买数量，并加入购物车,相关产品显示与该手表类型或品牌相同的手表进行显示,点击则进入单品页面.



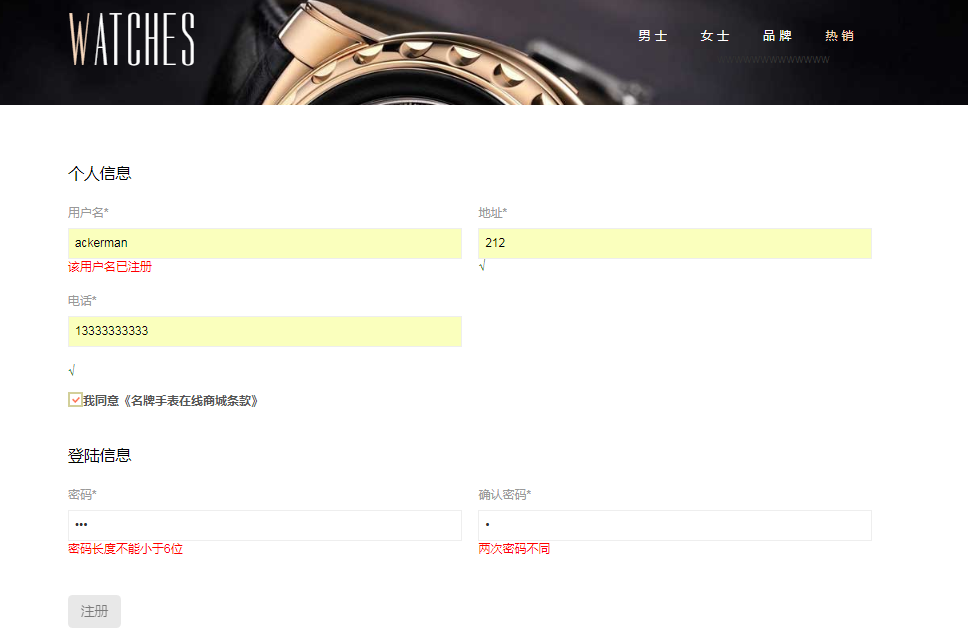
单品页面:点击加入购物车,若选取数量大于库存数量,则提示，并将数量置为1



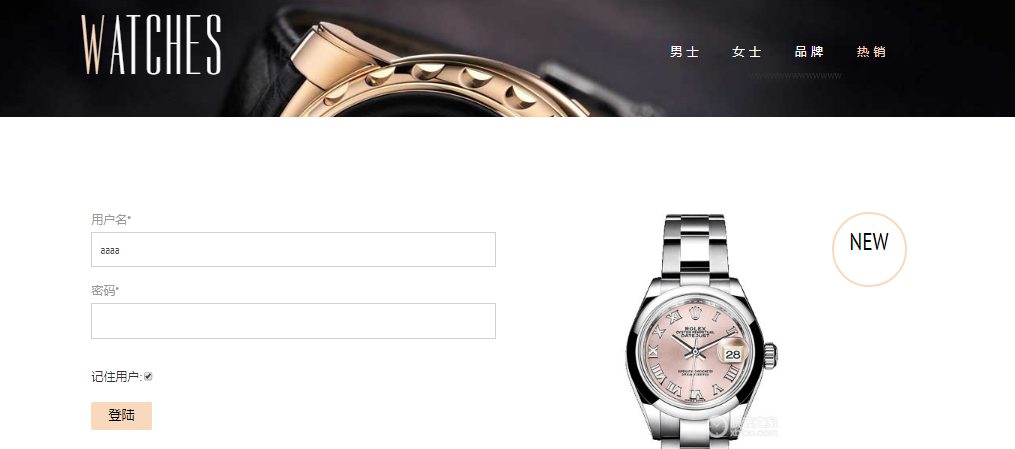
购物车页面:点击购物车进入购物车页面,在该页面可进行相应的清空，结算操作,若结算则生成订单



用户注册页面:用户注册页面,在该页面对全部的注册项进行判断,若验证失败则注册按钮失效.

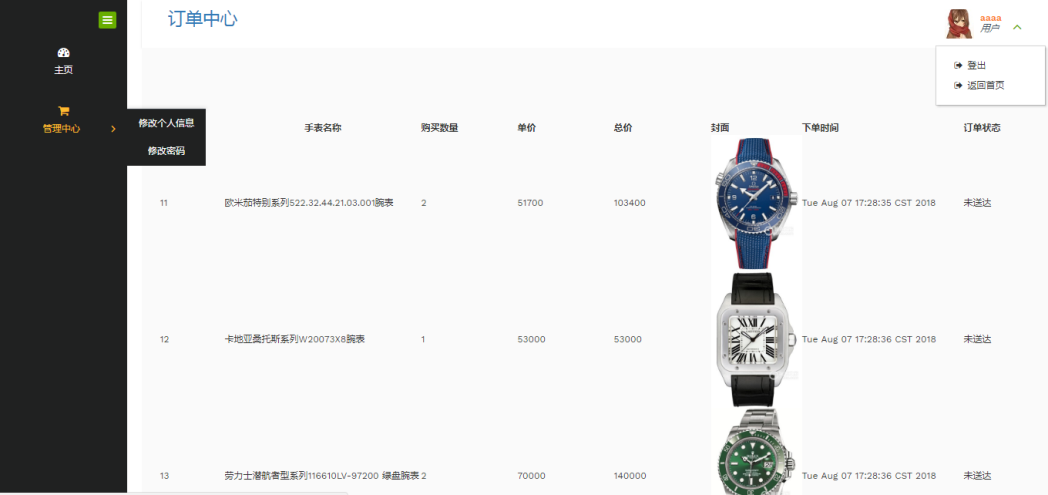


用户登录页面:点击记住用户则将用户名保存在cookie中，并记住checkbox的勾选状态，若密码错误，则返回登录页面,并给出提示.





用户中心页面:用户点击主页的用户名进入用户中心,可以查看订单,还可以进行登出和修改个人信息的操作。



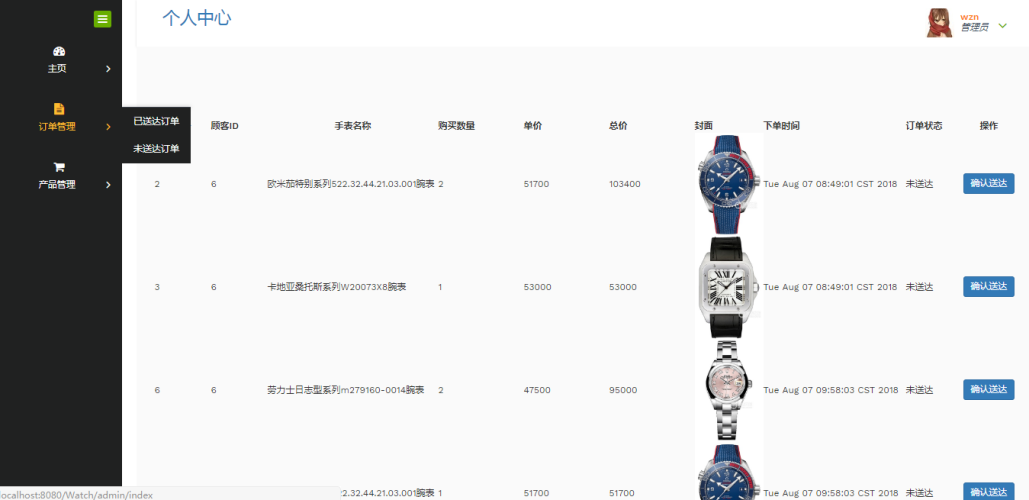
用户中心页面:点击管理中心，可以进行用户信息和密码的修改，若原密码错误，则修改按钮失效,不能进行相应的操作。



管理员页面:管理员进入主页可对品牌和手表进行增删改查。



管理员页面:管理员进入主页对用户的订单进行相应的状态调整。



管理员页面: 管理员进入主页对品牌进行增删改查。



管理员页面:管理员进入主页对手表、品牌进行新增，并选择该手表的封面图，设置手表的具体参数等。

