|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **北京科技大学项目实践报告** | | | | | | |
| **学院** |  | | 计算机与通信工程学院 | | | |
| **专业** | 信息安全 | | | **班级** | 信安182 | |
| **项目名称** |  | | 密码学应用与实践 | | | |
| **小组编号** |  | | 第8组 | | | |
| **指导教师** |  | | 朱岩 姚宣霞 孙旭 黄红冈 | | | |
| **实践起止日期** |  | | 2020.9.7-2020.9.11 | | | |
| **小组成员与职责分工** | **姓名** | **主要职责** | | | | **代码行数** |
| 梁逸飞 | 在线支付功能 | | | | 650 |
| 赵佳宾 | 会员管理功能 | | | | 450 |
| 王永壮 | 首页展示功能 | | | | 500 |
| 薛璞 | 旅游线路功能 | | | | 550 |
| 黎子梁 | 线路收藏功能 | | | | 500 |
|  |  | | | |  |
| **报告日期** |  | | 2020年9月12日 | | | |

|  |  |
| --- | --- |
| **项目信息** | |
| **实践目标** | 1. 分析web开发中的安全需求  2. 搭建实习必备的软件开发环境  3. 了解项目开发阶段及各阶段的工作流程  4. 了解项目开发各阶段主要文档及产出物  5. 熟悉代码开发规范  6. 掌握Web项目常用开发技术  7. 学会使用项目管理工具Maven  8. 掌握Web项目部署  9. 了解测试流程及测试用例编写  10.数字证书的生成与使用  11. SQL注入攻击的防护  12. 数据校验和加密  13. 身份认证  14. 掌握支付功能的实现，以及密码学在实际Web项目中的应用。 |
| **项目简介** | 本项目采用JavaWeb技术开发一个类似于去哪儿的旅游网站，通过该网站的开发，掌握一个完整的web项目的开发过程和相关安全问题的解决方案，包括对功能需求和安全问题的分析，数据库的设计，页面的设计，后端服务端的处理和安全方案的设计等，使得所设计的网站能够抵御sql注入攻击、实现数据的加密和校验、数字证书的生成与应用、身份认证和通过支付宝当面付等功能。 |
| **主要功能** | 1、实现会员的安全注册与登录  2、表单异步验证  3、限流控制  4、产品展示  5、数字证书的生成与应用  6、网上下订单与支付。 |

|  |  |
| --- | --- |
| **项目实践环境** | |
| **开发环境与工具** | 操作系统：Windows 10 |
| 开发语言：Java |
| 数据库系统：MySQL 5.7 |
| 开发IDE： Eclipse |
| 源码版本管理：SVN |
| 单元测试方法：JUnit |
|  |
|  |
| **项目部署环境** | MySQL数据库-5.0.22  JDK8  Eclipse  Navicat15  tomcat-8.0.21。 |

**网络用户安全认证系统设计**

1. **问题提出**

网络用户安全认证系统设计中，主要存在的安全问题有：

1. 用户资源需要得到保护，访问时需要登录验证。

网页系统中有部分网页是用户的相关信息，如用户的收藏信息、购买信息等，当没有登录时不能访问这些信息，保护用户的隐私资源不被窃取和泄露。

1. 用户的恶意注册。

在注册时由于没有人工的审核，会有许多人通过脚本代码自动注册生成数不胜数的账号，一是想通过多账号通过不同途径和测试方法获取该系统的各种信息，以便于正式账号在选择时的可以避开某些不必要的操作而选择直接了当的途径，二是可以恶意攻击系统，爆破数据库。因为每注册一个账户，数据库中就会记录一条信息，每次登录时，平均就要从数据库中多查找一条记录。当恶意注册的账户上千上万时，会给数据库造成巨大的压力，从而使数据库崩溃瘫痪，以造成攻击破坏的目的。

1. 数据库中用户密码的保护。

注册时如果在数据库中直接保存账户和密码，则很容易通过对数据库的查询、监听或者攻击直接获取密码，这样会降低账户的安全性。

1. 防SQL注入。

对数据库的CRUD操作（增删查改）会用到SQL语句，如果直接通过字符串拼接来生成SQL语句，很容易遭到SQL注入攻击，比如可以将一个SQL语句变成两个等等，从而达到对数据库的破坏或者系统操作错误等。

1. 用户支付安全。

通过网页端支付输入支付密码，会相应地将输入密码通过jsp页面返回到后端控制器，某些想要恶意攻击的人可以通过监听jsp或者查看请求信息等一些操作来获取用户支付密码。

1. HTTP数据传输未加密。

一般网页系统采用的是HTTP协议，尽管传输效率高，但是没有对传输的信息进行加密，在传输过程中很不安全。

1. **功能需求分析**

**会员管理功能：**

1、注册

在用户的注册界面，在用户输入用户名时，要对用户输入的用户名进行异步验证，来判断注册的用户名是否存在。同时要进行验证码的登录、注册认证以防止用户恶意注册，来拦截机器人。对用户的密码加密，防止从数据库泄露用户密码信息。

2、登录

登录同样利用异步验证判断输入的用户名是否存在，通过添加过滤器对登录用户的我的收藏和支付功能进行过滤，只有在用户登录后才能收藏路线，及对订单进行支付。

3、收藏

用户登录后可以访问自己的收藏界面，显示查看自己所收藏的商品信息，并对这些收藏的商品信息进行分页查询。

**首页展示功能：**

1、首页轮播图

在网页首页会有几个商品信息进行轮播显示。

2、首页分类数据

在首页会有人气旅游、最新旅游、国内游、境外游的分类，需要在不同的分类下查询数据库里相应的数据并显示出来。

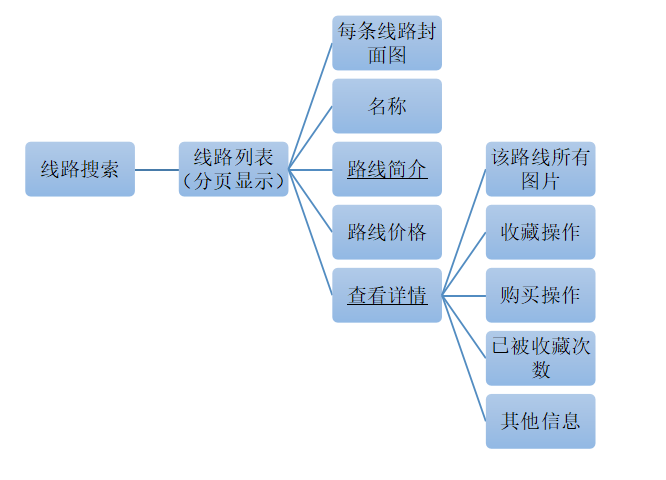
**旅游路线功能：**

1. 旅游线路搜索

在搜索框输入内容时，能通过模糊查询查出路线名称中含有该内容的所有路线，将在列表中展示每条路线的封面图、名称、路线简介、路线价格，并且将所有路线按价格进行升序排列。然后每页最多展示七条路线，根据搜索到的路线数目确定分页数目，用户可以点击首末页、上下页以及导航条中出现的页码。

1. 旅游线路详情

点击路线中的“查询详情“用户可以进入一个新的界面，在该界面中用户可以看到关于该路线的所有图片并且这些图片以轮播的形式播放，用户还可以在此界面进行收藏与购买操作并且观察到该路线已被收藏的次数，值得注意的是用户必须是登录状态下才可进行收藏与购买操作。



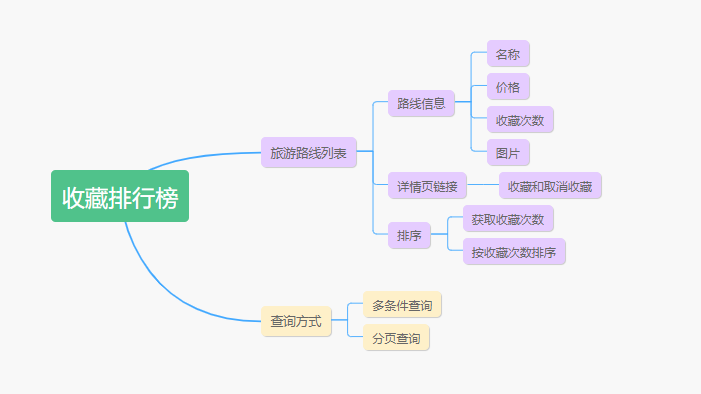
**线路收藏功能**

1、旅游线路收藏

点击名称或图片则会进入路线的详情界面，在详情界面里可以通过点击按钮实现收藏和取消收藏的功能。

2、收藏排行榜

点击分类栏中的收藏排行榜，能将旅游路线列表按照收藏次数从多到少的顺序排列出来，可以输入路线名称和价格区间进行多条件查询，如果都不输入则默认查询全部路线。查询的结果按每8个路线一页，实现分页查询，同时可以点击页码进行跳转页。查询的路线上有路线名称、价格、收藏次数和图片的信息。鼠标悬停在名称或图片上都会显示路线的描述。



**在线支付功能：**

1、实现生成订单

当点击购买的时候，会让用户二次确认购买，在这里会显示购买的相关信息，如产包名、数量、价格等，确认购买后需要根据用户选择产品包的信息，自动生产产品包的订单，包括产品包名、数量、价格等相关信息，并且还要将订单信息保存到会话中（如果保存到请求则需要一直传下去）以便于后续生成订单的二维码。

2、选择支付方式

在这里我们将提供支付宝的支付方式。

3、生成二维码

当用户选择支付宝的支付方式后，利用支付宝提供的库和接口，根据订单信息生成一个订单二维码。

4、实现订单当面付

用户通过扫码在手机APP上面进行支付，在网页界面可以同步查看订单详情，支付成功后页面将会跳转到支付成功界面。

**三、总体设计**

1、利用过滤器实现未登录点击购买和收藏时会跳转到登录界面以保护用户资源。

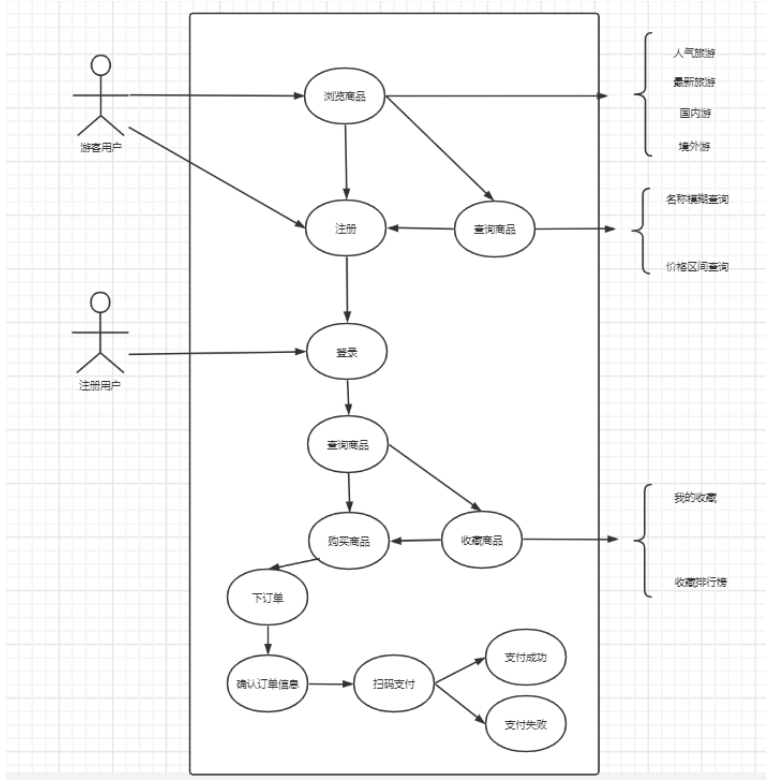
2、注册和登录通过验证码验证以防止恶意注册。

3、注册时利用SHA256对密码进行加密后，将加密后的密码和随机数保存到数据库中。

4、利用SQL预编译防止恶意的SQL注入。

5、通过支付宝二维码在手机端APP上支付，防止网页端支付时被窃取密码。

6、启用https协议，生成tomcat的数字证书，修改端口并启用加密过得端口，对传输数据进行加密，使得浏览器获得信任。



**四、详细设计**

1、用户注册用户名的异步验证

用户在输入用户名时，通过与数据库已存在的用户名进行搜索、判断，当数据库中存在用户输入的用户名时，在UserExistsController控制器向register.jsp注册界面返回boolean 型数据1，通过函数将错误提示显示给用户，如果用户不更改已有的用户名，当用户点击注册时，将在RegisterController控制器将register.jsp传递的数据拦截，并返回到注册界面。

1. 登录权限验证

用户未登录可以进行旅游路线、收藏排行榜的浏览，当用户想要对旅游路线进行收藏或者对旅游路线订单支付时，要先进行登录。用户登录是输入的密码要在UserLoginService中进行二次加密，和数据库中的加密后密码进行比对。在我的收藏界面，用户可以进入收藏路线的详情，还可以在我的收藏界面分页查询已收藏路线。

3、首页数据查询

最新旅游、国内游、境外游的查询都比较简单，最新旅游查询所有结果按时间顺序排显示前4个即可，国内游、境外游则按数据库中种类查询然后按价格从高到低显示前6个即可。稍微复杂麻烦一点的便是人气旅游。“人气”我们定义为收藏总次数，一个旅游线路可以被多个用户收藏，一个用户可以收藏多个旅游线路，因此他涉及到两张表的连接查询，因此设计的SQL查询语句如下。

|  |
| --- |
| String sql = "select tab\_route.rid,rname,rimage,price "  + " from tab\_route left join tab\_favorite on tab\_route.rid = tab\_favorite.rid"  + " group by tab\_route.rid "  + " order by count(\*) desc , price asc "  + SQLUtil.*getLimit*(1, 4); |

4、旅游线路搜索

在搜索框内第一次输入时，RouteListController只需接收rname一个参数（page参数设置为1），然后调用业务逻辑实现相应的功能（分页查询，按价格升序排列...），当点击下一页、末页、以及具体的页码时，将接收rname和page两个参数，实现与之前一样的功能。

点击查看详情时，RouteDetailController需要接收该路线对应的rid，通过rid查出关于该路线的所有内容并且在页面中显示出来。

|  |
| --- |
| **public** **class** RouteListController **extends** HttpServlet{  //接收rname和page两个参数  ......  //调用业务逻辑  RouteService routeService = **new** RouteService();//业务层  PageVO1<Route> pageVO = routeService.*search*(searchVO1, page, recordOfPage);  //实现页面导航  request.getRequestDispatcher("/route\_list.jsp").forward(request, response);  }  **public** List<Route> selected(SearchVO1 searchVO,**int** page,**int** recordOfPage) **throws** SQLException{}// 获取当前页的路线的所有信息  **public** **class** RouteDetailController **extends** HttpServlet{  //接收路线编号rid和用户编号uid两个参数  ......  //调用业务逻辑  FavoriteService favoriteService = **new** FavoriteService();  RouteService routeService = **new** RouteService();  Route route = routeService.findById(rid);//通过路线编号获取该路线的所有信息  //实现页面导航  request.getRequestDispatcher("/route\_detail.jsp").forward(request, response);  }  **public** **boolean** isFavorite(**int** rid,**int** uid) **throws** SQLException{}//判断用户是否收藏该路线 |

5、收藏按钮的实现

在商品详情界面有一个收藏按钮，未收藏时会显示“点击收藏”，已收藏时会显示“取消收藏”。需要先编写一个判断商品是否被当前用户收藏的函数，再在控制器中接收请求获取请求参数。

|  |
| --- |
| **public** **boolean** isFavorite(**int** rid,**int** uid) **throws** SQLException{  String sql ="SELECT 1 FROM tab\_favorite WHERE rid = ? AND uid = ?";  Connection conn = ConnUtil.*getConn*();  PreparedStatement pstat = conn.prepareStatement(sql);  pstat.setInt(1, rid);  pstat.setInt(2, uid);  ResultSet rs = pstat.executeQuery();  **if**(rs.next()){  **return** **true**;  }**else**{  **return** **false**;  }  } |

之后便是调用业务逻辑实现页面导航。

6、收藏排行榜

首先需要获取各个路线的收藏次数。在数据库中有各用户和路线的收藏关系表，于是写一个方法，通过sql语句得到各路线的总收藏次数并按收藏次数排序。

|  |
| --- |
| String sql = "select tab\_route.rid,rname,rimage,price "  + " from tab\_route left join tab\_favorite on tab\_route.rid = tab\_favorite.rid"  + " group by tab\_route.rid "  + " order by count(\*) desc , price asc "  + SQLUtil.*getLimit*(1, 4); |

同时也得到了路线名称、价格等信息，用来显示在界面上。

然后实现多条件查询和分页查询。将多条件查询需要的参数封装成一个SearchVO类。

|  |
| --- |
| **public** **class** SearchVO {  // 1 路线名称的模糊查询 routeName  **private** String routeName;  // 2 路线的最小值 minRoutePrice  **private** Float minRoutePrice;  // 3 路线的最大值 maxRoutePrice  **private** Float maxRoutePrice;  .  .  . |

将分页查询需要的参数封装成一个PageVO类。

|  |
| --- |
| **public** **class** PageVO <T>{  // 1 当前页码 page int 用户指定，如果用户没有指定默认第1页0  **private** **int** page;  // 2 每页记录数量 recordOfPage int 用户指定和程序员指定。  **private** **int** recordOfPage;  // 3总记录数 recordCount int 查询，select count(\*) from goods where 多条件  **private** **int** recordCount;  // 4 总页数 pageCount int 计算，pageCount =（（recordCount-1）/ recordOfPage）+ 1  **private** **int** pageCount;  // 5 当前页数据集合 list List<T> 查询，limit （page-1）\* recordOfPage,recordOfPage  **private** List<T> list;  .  .  . |

通过sql语句根据查询条件进行筛选。

|  |
| --- |
| String sql = "select tab\_route.rid,rname,rimage,price,count(\*) favorite "  + " from tab\_favorite left join tab\_route on tab\_route.rid = tab\_favorite.rid"  + SQLUtil.*getWhere*(searchVO)  + " group by tab\_route.rid "  + " order by count(\*) desc , price asc ,sourceId asc"  + SQLUtil.*getLimit*(page, recordOfPage); |

7、订单生成二维码

在点击购买后，根据商品信息生成订单，当确认购买方式为支付宝后，根据页面获取订单信息，并将这些信息通过支付宝提供的接口生成一个二维码。核心代码如下。

|  |
| --- |
| package com.web.controller;  import ......  @WebServlet("/qrCode.do")  public class QRCodeController extends HttpServlet {  protected void doGet(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  String qrtext = request.getParameter("qrCode");  ByteArrayOutputStream out = QRCode.from(qrtext).to(ImageType.PNG).stream();  response.setContentType("image/png");  response.setContentLength(out.size());  OutputStream outStream = response.getOutputStream();  outStream.write(out.toByteArray());  outStream.flush();  outStream.close();  }  } |

用户通过在支付宝沙箱APP上扫描这个二维码获取订单信息以及需要支付的金额，然后可以在手机APP上完成支付。

**五、程序运行相关界面及简介**

注册界面，当输入已注册用户aaa时显示如图1。



图1 注册界面（1）

成功注册新用户以及数据库显示信息如图2。



图2 注册界面（2）

数据库显示的密码为进行SHA256加密后的密码，最后一列为随机生成的“盐”，出生日期处理正确。如图3。

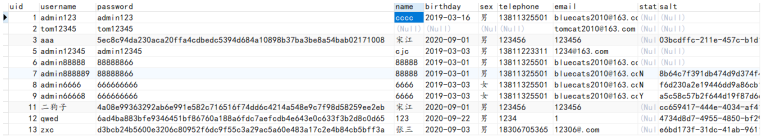


图3 数据库用户表

登录已注册的账户，能够显示登陆账户名，如图4。



图4 登陆成功显示用户名

首页的人气旅游、最新旅游、国内游、境外游如图5-8。



图5 人气旅游



图6 最新旅游

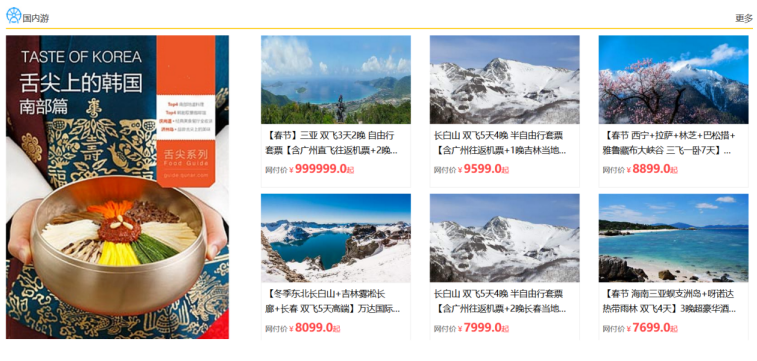


图7 国内游

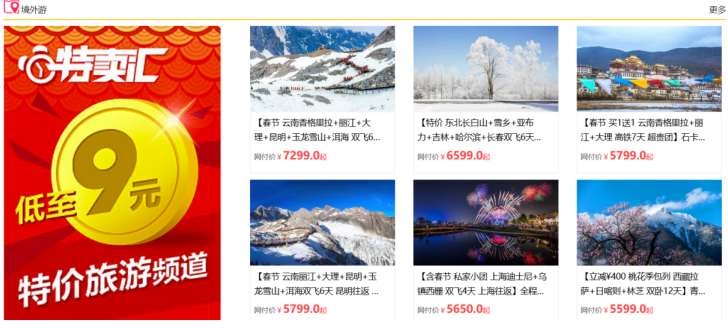
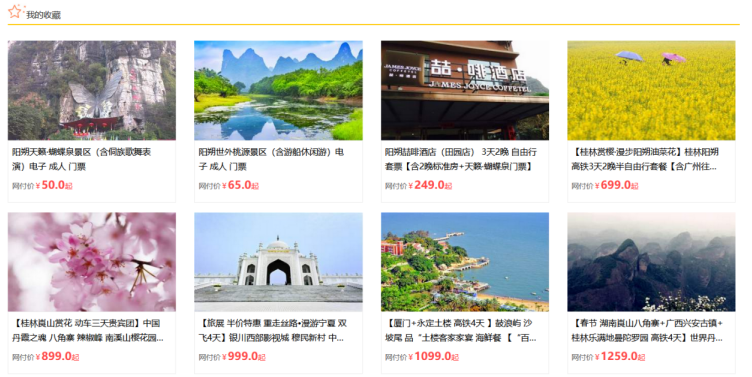


图8 境外游

进入我的收藏功能，能够查看收藏商品详情，能够分页查询：

第一页内容如图9：

图9 我的收藏第一页

第二页内容如图10：



图10 我的收藏第二页

在搜索框中输入（这里我们输入三亚），点击搜索，将罗列出与三亚有关的路线如图11所示。

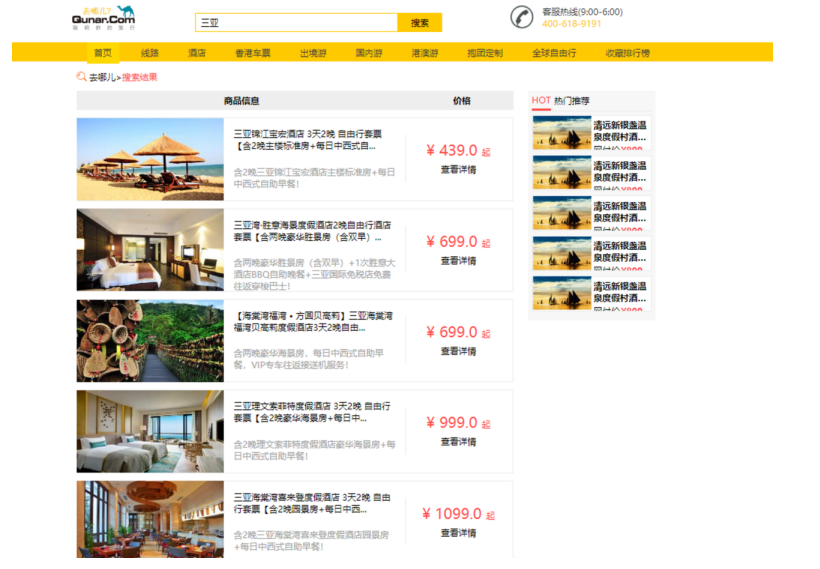


图11 “三亚”搜索结果

如图12所示，将该列表进行分页处理，每页七条记录。



图12 搜索界面一页记录

如图13-14所示，页面可以正确进行分页查询功能。

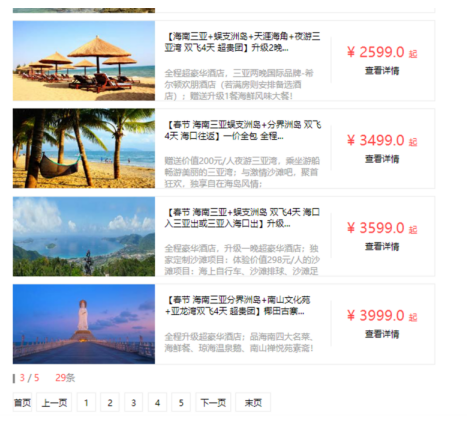


图13 搜索界面分页查询（1）



图14 搜索界面分页查询（2）

点击查看详情，将会显示该路线的详细信息，如图15所示。

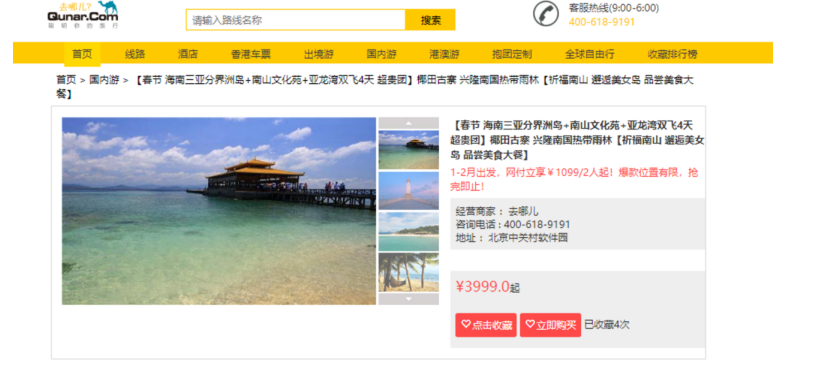


图15 详情界面

点击收藏排行榜后按收藏次数排序分页显示所有路线和路线的基本信息，如图16。



图16 收藏排行榜

切换页码，显示各页的结果。如图17-18。



图17 分页显示（1）



图18 分页显示（2）

输入查询条件如“厦门”，价格为1000到1500，显示出符合条件的所有路线，如图19。



图19 多条件查询

点击路线图片或名称即可进入如图15的详情页面。

未收藏时显示如图20。



图20 未收藏

点击收藏按钮，改变收藏次数，跳回本页并改变按钮名称，如图21。

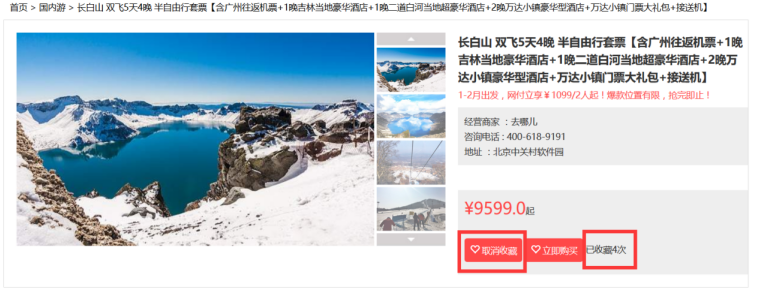


图21 已收藏

点击我的收藏里可以看到收藏的商品信息，如图22。

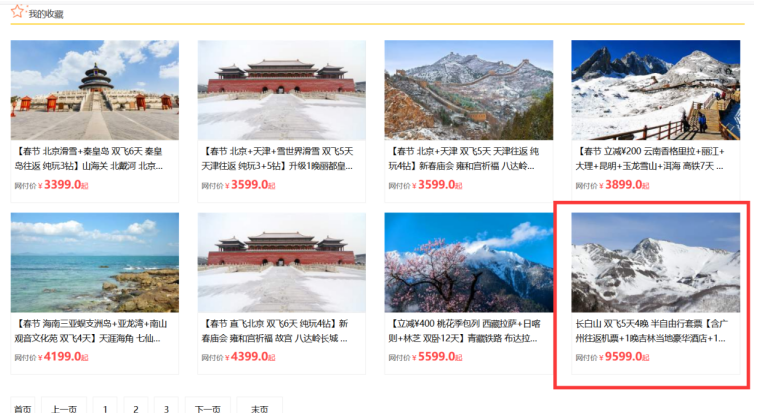


图22 我的收藏里新增加收藏的信息

在商品详情页面选择立即购买后，跳转到提交订单界面，如图23。



图23 提交订单

点击下方确认购买，即可选择支付方式，如图24。



图24 选择支付方式

在这里只有支付宝支付，点击选择支付宝支付，生成订单二维码。如图25。



图25 生成订单二维码

点击查看详情获得订单信息。如图26。

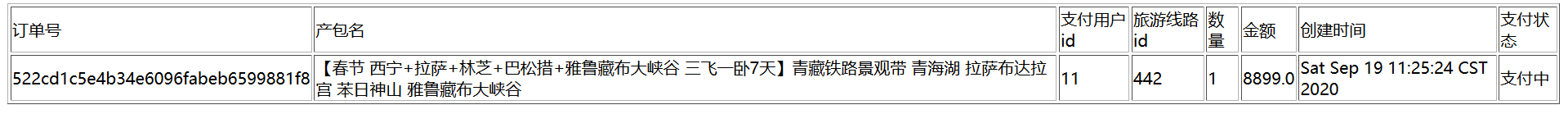


图26 订单信息

支付成功后，跳转到支付成功界面。如图27。



图27 支付成功

在这点击查看详情，会发现订单信息的支付状态变为支付成功。如图28。

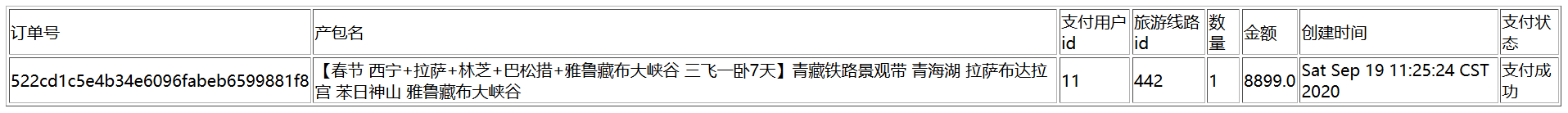


图28 支付成功后的订单信息

1. **程序编制、运行及调试过程中遇到问题的解决方案**

1、在设计我的收藏时，自己设计的方法是使用会话中存放的用户名进行查询路线信息，但是这样的SQL语句跨越了三张数据表，设计起来有难度，经过询问老师，改为使用会话中的用户ID进行两张表查询，这样SQL语句就简单了。

2、在进行设计分页查询时，在网页上显示出多余的空白页码，经过反复检查代码，最后锁定在SQL语句。在Navicat中查询此SQL语句发现出错，经改正后正确显示分页查询。

3、在选择支付方式设计中，静态页面模板给了支付宝和微信两种支付方式，不管点击哪个支付方式都会跳转到支付界面，但我们是利用支付宝来实现的，所以后来将微信支付方式给删除了。

4、在生成订单信息的时候，我们把商品信息存到了请求里面，因为从商品详情界面到点击购买后跳转到的提交订单界面，商品信息都是通过请求进行传递的，因为这两步都还没有正式生成订单。但后来发现支付的2—4步都需要商品信息，商品信息放请求里每跳转到下一个界面我们就需要在控制器里获取前一页面的商品信息然后再存到请求里，下一页面编写的时候又要从请求里提出来商品信息，整个过程就很繁琐，而且中间某一步传漏了，导致后面几步页面显示不正确。所以后来改进了下，在提交订单界面，在点击确认购买前，商品信息都存到请求中，点击确认购买后，再将商品信息放到会话中，这样既能减少许多中间传递过程，也能提高准确性。

**七、密码学应用与实践的总结与体会**

梁逸飞：

通过本次实习，或多或少都会有所收获。这个实习项目是我们第一次以团队形式完成的较大的一个项目。一个项目五个功能分给五个人做，既考验每个人的能力，又考验整个团队的能力。每个功能每个成员需要考虑怎么设计实现这个功能，需要编写哪些代码，并且整个过程一旦出现bug需要自己的调试修改。当每个人完成各自实现的功能后还要将其进行合并汇总形成一个完整的、功能成熟的大项目，在合并过程中又会出现很多问题，比如命名冲突等。

另外通过本次实习初步了解了web应用程序开发过程中涉及的许多安全性问题，以及如何去解决这些安全性问题，我们并不能只是为了实现这个功能就完了，还得考虑这个功能的安全性，这都对以后的学习有很大的帮助。

赵佳宾：

在此次密码学应用于实践的课程中，认识学习了web开发设计项目的整体流程，让我们对前、后端工程师的工作内容有简单的了解。在这周亲身设计web设计，相比于前两周的实习有了更深刻的认识。同时也发现了很多的不足，在设计SQL语句时忘记在分断后空格，这会导致程序报错，在以后的工作中有了参考的经验教训，上一学期学习密码学后，还未亲身使用过密码学在实际程序开发中的用处，在这次实习中我们使用了密码学的知识来完善程序，对密码学有了更进一步的掌握。

王永壮：

代码编写确实比较考验人，有些时候不动手写真不知道自己几斤几两、自己学的如何。在整个实习过程中团队力量非常强大，我很多不会的地方通过同组成员的帮助、调试、改正才得以实现功能，这让我体会到了团队的重要性。

我负责的部分与系统的安全性设计关联的不多，也就只有查询数据库数据时利用SQL预编译防止SQL注入。另外我们之前都是各自写各自的，有很多能够复用的代码写了几份，还有些不同的代码取了相同的名造成命名冲突，这些都是我们第一次团队协作过程中出现的问题，也是以后需要注意改正的地方。

薛璞：

经过这几天的学习，我不仅学到了很多专业知识，而且我意识到了团队协作的重要性，最终成果的实现离不开组长的协调以及每位组员的努力。此外对于经验不丰富的我们，开始做项目时需要一步一步分析整个系统，把项目流程完整的分析出来而不能急于求成。

黎子梁

对于小组任务，每个成员在做的时候不仅要保证自己的部分可以正确实现功能，还要让自己的代码能够很好地与其他成员地代码合并。这就需要开始工作前进行交流，比如规定好变量的统一命名，相互协作的成员交流需要传入的参数，商量好共同需要的实体、方法类等。