一、需求分析

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。

4、首页轮播图

在网页首页会有几个商品信息进行轮播显示。

5、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。

6、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。

7、旅游线路详情

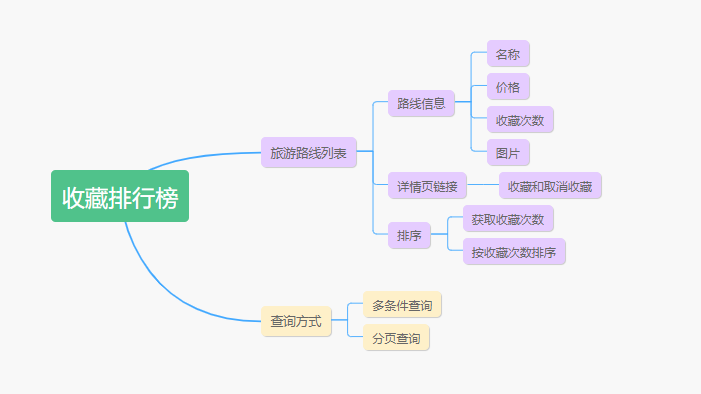
点击路线中的“查询详情“用户可以进入一个新的界面，在该界面中用户可以看到关于该路线的所有图片并且这些图片以轮播的形式播放，用户还可以在此界面进行收藏与购买操作并且观察到该路线已被收藏的次数，值得注意的是用户必须是登录状态下才可进行收藏与购买操作。

8、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。

9、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。



10、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。

11、选择支付方式

在这里我们将提供支付宝的支付方式。

12、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。

13、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。

1. 成员分工

梁逸飞（组长）：在线支付功能。

赵佳宾：会员管理功能。

王永壮：首页展示功能。

薛璞：旅游路线功能。

黎子梁：线路收藏功能。

1. 功能模块复用代码

1、数据库连接工具类：

|  |
| --- |
| package com.web.util;  import java.sql.Connection;  import java.sql.SQLException;  import java.util.Properties;  import javax.sql.DataSource;  import org.apache.commons.dbcp.BasicDataSourceFactory;  /\*\*\*\*  \* 操作数据库连接对象的工具类  \* @author lyf  \*/  public class ConnUtil {  private static DataSource ds;  //这是一个每个用户都有的一个保存Connection对象的容器  private static ThreadLocal<Connection> local = new ThreadLocal<>();  static {//静态代码块，特点：只运行一次。  try {  // 加载资源文件信息的文件对象  Properties prop = new Properties();  // 从mysql5.properties文件中加载信息  prop.load(ConnUtil.class.getResourceAsStream("/mysql5.properties"));  // 使用加载的信息创建数据源对象  ds = BasicDataSourceFactory.createDataSource(prop);  } catch (Exception e) {  e.printStackTrace();  }  }  /\*\*\*  \* 获得一个数据库连接对象的方法  \* @return 连接对象  \* @throws SQLException  \*/  public static Connection getConn() throws SQLException {  Connection conn = local.get(); //从容器中取出连接对象  if(conn == null || conn.isClosed()) {  conn = ds.getConnection();  local.set(conn);  }  return conn;  }    /\*\*\*  \* 关闭数据 库连接对象的方法  \*/  public static void closeConn() {  Connection conn = local.get();  try {  if(conn != null && !conn.isClosed()) {  conn.close(); //释放连接  }  }catch (Exception e) {  e.printStackTrace();  }finally {  conn = null;  local.set(null);  }  }  } |

1. 日期类处理代码

|  |
| --- |
| package com.web.util;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Date;  public class DateUtil {  private static SimpleDateFormat sdf =new SimpleDateFormat("yyyy-MM-dd");  /\*\*  \*  \* @param date日期对象  \* @return 日期对象的字符串形式  \*/  public static String getString(Date date) {  return sdf.format(date);  }  /\*\*  \*  \* @param str 日期字符串形式  \* @return 日期对象  \* @throws ParseException解析出现错误抛出异常  \*/  public static Date getDate(String str) throws ParseException {  return sdf.parse(str);  }  } |

1. SQL语句处理工具类

|  |
| --- |
| package com.web.util;  import com.web.vo.SearchVO;  import com.web.vo.SearchVO1;  public class SQLUtil {    public static String getLimit(int page, int recordOfPage) {  return " limit " + ((page - 1) \* recordOfPage) + ", " + recordOfPage;  }    public static String getWhere(SearchVO searchVO) {  String sql = " WHERE 1=1 ";  if (searchVO.getRouteName() != null && !"".equals(searchVO.getRouteName())) {  sql += " AND rname like '%" + searchVO.getRouteName() + "%' ";  }  if (searchVO.getMinRoutePrice() != null) {  sql += " AND price >= " + searchVO.getMinRoutePrice();  }  if (searchVO.getMaxRoutePrice() != null) {  sql += " AND price <= " + searchVO.getMaxRoutePrice();  }  return sql;  }    public static String getWhered(SearchVO1 searchVO) {  String sql = " WHERE 1=1 ";  System.out.println(searchVO.getRname());  if (searchVO.getRname() != null && !"".equals(searchVO.getRname())) {  sql += " AND rname like '%" + searchVO.getRname() + "%' ";  }    return sql;  }  } |

1. 支付宝接口工具类

|  |
| --- |
| package com.web.util;  import java.math.BigDecimal;  import java.util.ArrayList;  import java.util.List;  import java.util.Objects;  import java.util.UUID;  import org.apache.commons.beanutils.BeanUtils;  import org.apache.commons.beanutils.PropertyUtils;  import com.alipay.api.AlipayClient;  import com.alipay.api.DefaultAlipayClient;  import com.alipay.api.domain.AlipayTradePrecreateModel;  import com.alipay.api.domain.GoodsDetail;  import com.alipay.api.request.AlipayTradePrecreateRequest;  import com.alipay.api.response.AlipayTradePrecreateResponse;  public class PayUtil {  private static final String URL = "https://openapi.alipaydev.com/gateway.do";  private static final String APP\_ID = "2021000116668453";  private static final String APP\_PRIVATE\_KEY = "";  private static final String CHARSET = "utf-8";  private static final String ALIPAY\_PUBLIC\_KEY = "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAlJQlZCpkMrcelVRUlcaDC3JMOgE3D8iOSJL5H0XNgE7wioHvfxRrMyrGEk6ZFe13fJtjrOgVOV6NECtTbz/BZHMjirnre+FpXcdOo2/ioarAcmzTaiKdSBsadypK/bl/B3ahyyCmI8Pyx5kbfS6GegPdyH2fqeP7VjIak5TFITuonOXEUgXzHXFvpfXSPtyob+tOlpICEdC3DfatPDJaCULupqD7ygq4RJ3aFnKGWaB0fgSvWa5VvkJwlDNFeQhjBs0YgAVthu/pCHC4AaahFY740dZxdZzZrVnxAkJ5QlTDZ8nDvK3iHCveu7AULWAuUaLKre5yK2HqdCzKLNTgKQIDAQAB";  private static final String returnURL = "http://localhost:9527/training-web/result.do";  private static final String notifyURL = "http://localhost:9527/training-web/result.do";  // 获得初始化的AlipayClient  public static AlipayClient alipayClient = new DefaultAlipayClient(URL, APP\_ID, APP\_PRIVATE\_KEY, "json", CHARSET,  ALIPAY\_PUBLIC\_KEY, "RSA2");  /\*\*  \* 创建收款单  \*  \* @param orderNo  \* 商户订单号,64个字符以内、只能包含字母、数字、下划线；需保证在商户端不重复  \* @param orderSubject  \* 订单标题 256个字符以内  \* @param totalAmount  \* 订单总金额，单位为元，精确到小数点后两位，取值范围[0.01,100000000]  \* @param orderDesp  \* 对交易或商品的描述 128个字符以内  \* @param TradeGoodsDetail  \* 商品明细说明  \* @param notifyURL  \* 支付结果异步通知地址URL  \*/  public static TradePreCreateResp tradePrecreateRequest(String orderNo, String orderSubject, Double totalAmount,  String orderDesp, List<TradeGoodsDetail> goodsList, String notifyURL) {  AlipayTradePrecreateRequest request = new AlipayTradePrecreateRequest();  AlipayTradePrecreateModel model = new AlipayTradePrecreateModel();  model.setOutTradeNo(orderNo);  model.setTotalAmount(new BigDecimal(totalAmount).setScale(2, BigDecimal.ROUND\_HALF\_UP).toString());  model.setSubject(orderSubject);  model.setBody(orderDesp);  model.setQrCodeTimeoutExpress("1m");  model.setTimeoutExpress("1m");  try {  if (!Objects.isNull(goodsList)) {  List<GoodsDetail> goodsDetailList = new ArrayList<GoodsDetail>();  for (TradeGoodsDetail detail : goodsList) {  GoodsDetail goods = new GoodsDetail();  PropertyUtils.copyProperties(goods, detail);  goodsDetailList.add(goods);  }  model.setGoodsDetail(goodsDetailList);  }  request.setBizModel(model);  request.setNotifyUrl(notifyURL);  AlipayTradePrecreateResponse response = alipayClient.execute(request);  if (response.isSuccess()) {  TradePreCreateResp resp = new TradePreCreateResp();  BeanUtils.copyProperties(resp,response);  return resp;  }  throw new RuntimeException("error");  } catch (Exception ex) {  throw new RuntimeException(ex.getMessage());  }  }  public static String genUUID(){  String uuid = UUID.randomUUID().toString();  return uuid.replace("-","");  }  public static String getUrl() {  return URL;  }  public static String getAppId() {  return APP\_ID;  }  public static String getReturnURL() {  return returnURL;  }  public static String getAppPrivateKey() {  return APP\_PRIVATE\_KEY;  }  public static String getCharset() {  return CHARSET;  }  public static String getAlipayPublicKey() {  return ALIPAY\_PUBLIC\_KEY;  }  public static String getNotifyurl() {  return notifyURL;  }  } |

1. SHA加密工具类

|  |
| --- |
| package com.web.util;  import java.io.UnsupportedEncodingException;  import java.security.MessageDigest;  import java.security.NoSuchAlgorithmException;  /\*\*  \* SHA256验证码  \*  \* @author 41619  \*  \*/  public class SHAUtil {  /\*\*  \* 利用java原生的类实现SHA256加密  \*  \* @param str  \* 参数拼接的字符串  \* @return  \*/  public static String getSHA256(String str) {  MessageDigest messageDigest;  String encodeStr = "";  try {  messageDigest = MessageDigest.getInstance("SHA-256");  messageDigest.update(str.getBytes("UTF-8"));  encodeStr = bytesToHexString(messageDigest.digest());  } catch (NoSuchAlgorithmException | UnsupportedEncodingException e) {  e.printStackTrace();  }  return encodeStr;  }  /\*\*  \* 将字节数组转换成16进制字符串  \*  \* @param bytes  \* 即将转换的数据  \* @return 16进制字符串  \*/  private static String bytesToHexString(byte[] bytes) {  StringBuffer sb = new StringBuffer(bytes.length);  String temp = null;  for (int i = 0; i < bytes.length; i++) {  temp = Integer.toHexString(0xFF & bytes[i]);  if (temp.length() < 2) {  sb.append(0);  }  sb.append(temp);  }  return sb.toString();  }  } |

1. 接口说明
2. 用户资源保护过滤器

处理点击收藏、立即购买的控制器其部署的URL路径为”/users/...”，其过滤器的部署路径为”/users/\*”。

1. 中文处理过滤器

所有控制器部署的URL为”/xxx.do”，中文处理的过滤器部署路径为”\*.do”。

1. JSP页面处理

将所有JSP的头提取出来形成一个新的页面header.jsp，然后再在各JSP种引入header.jsp，以便修改页面头部分时只需要在header.jsp中修改即可。

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>  <%  String path = request.getContextPath();  String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";  %>  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  <html>  <head>  <meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>  <title>Insert title here</title>  </head>  <body>  <!--引入头部-->  <div id=*"header"*><!-- 头部 start -->  <header id=*"header"*>  <div class=*"top\_banner"*>  <img src=*"images/top\_banner.jpg"* alt=*""*>  </div>  <div class=*"shortcut"*>  <c:choose>  <c:when test=*"*${**empty** sessionScope.users}*"*>  <!-- 未登录状态 -->  <div class=*"login\_out"*>  <a href=*"login.jsp"*>登录</a>  <a href=*"register.jsp"*>注册</a>  </div>  </c:when>  <c:otherwise >  <!-- 登录状态 -->  <div class=*"login"*>  <span>欢迎回来，${sessionScope.users.username }</span>  <a href=*"users/myfavorite.do"* class=*"collection"*>我的收藏</a>  <a href=*"logout.do"*>退出</a>  </div>  </c:otherwise>  </c:choose>  </div>  <div class=*"header\_wrap"*>  <div class=*"topbar"*>  <div class=*"logo"*>  <a href=*"/"*><img src=*"images/logo.jpg"* alt=*""*></a>  </div>  <div class=*"search"*>  <script type=*"text/javascript"*>  **function** toSearch(){  document.searchForm.submit();  }  **function** toPage(p){  document.searchForm.page.value=p;  document.searchForm.submit();  }  </script>  <form action=*"routelist.do"* methon=*"post"* name=*"searchForm"*>  <input type=*"hidden"* name=*"page"* value=*""*>  <input name=*"rname"* type=*"text"* placeholder=*"请输入路线名称"* value=*"*${param.rname}*"* class=*"search\_input"* id=*"search\_by"* autocomplete=*"off"*>    <a href=*"javascript:toSearch();"* class=*"search-button"* id=*"qqq"*>搜索</a>  </form>  </div>  <div class=*"hottel"*>  <div class=*"hot\_pic"*>  <img src=*"images/hot\_tel.jpg"* alt=*""*>  </div>  <div class=*"hot\_tel"*>  <p class=*"hot\_time"*>客服热线(9:00-6:00)</p>  <p class=*"hot\_num"*>400-618-9191</p>  </div>  </div>  </div>  </div>  </header>  <!-- 头部 end -->  <!-- 首页导航 -->  <div class=*"navitem"*>  <ul class=*"nav"*>  <li class=*"nav-active"*><a href=*"index.do"*>首页</a></li>  <li><a href=*"routelist.do"*>线路</a></li>  <li><a href=*"route\_list.jsp"*>酒店</a></li>  <li><a href=*"route\_list.jsp"*>香港车票</a></li>  <li><a href=*"route\_list.jsp"*>出境游</a></li>  <li><a href=*"route\_list.jsp"*>国内游</a></li>  <li><a href=*"route\_list.jsp"*>港澳游</a></li>  <li><a href=*"route\_list.jsp"*>抱团定制</a></li>  <li><a href=*"route\_list.jsp"*>全球自由行</a></li>  <li><a href=*"routesearch.do"*>收藏排行榜</a></li>  </ul>  </div></div>  </body>  </html> |

将各JSP中上述代码替换为：

<jsp:include page=*"/header.jsp"*/>

即可引入头页面。