技术点1、开发步骤

1.1、确定并统一开发平台（JDK,TOMCAT等等）

1.2、需求功能分析

1.3、数据库设计

1.4、确定开发平台及所需要的技术

1.5、确定包结构

1.6、搭建开发环境（导入相应的包，资源以及各种工具类）

1.7、具体代码实现

技术点2、编写一个通用的Servlet

自定义一个servlet类继承`BaseServlet extends HttpServlet`

重写 `public void service(HttpServletRequest r, HttpServletResponse re)` 方法之后内容见代码：父类

```
public void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    //解决请求乱码
    request.setCharacterEncoding("UTF-8");
    //响应乱码
    response.setContentType("text/html;charset=UTF-8");
    try {
        String method = request.getParameter("method");      //获取请求方法名称
        Class clazz = this.getClass();  //获取类对象
        //通过方法名，参数 获取方法
        Method m = clazz.getMethod(method, HttpServletRequest.class,HttpServletResponse.class);
        //执行真实子类的方法，并返回一个转发的路径
        String path = (String) m.invoke(this, request,response);
        if(path!=null){
            //转发
            request.getRequestDispatcher(path).forward(request, response);
        }
    }catch (Exception e) {
        e.printStackTrace();
    }
}
```

继承的子类：

```java
public class UserServlet extends BaseServlet implements IUserServlet {
    public String register(HttpServletRequest request, HttpServletResponse response) {
        try {
            // 获得请求参数并封装到JAVABean类
            Shopuser user = new Shopuser();
            ConvertUtils.register(new DateLocaleConverter(), Date.class);
            BeanUtils.populate(user, request.getParameterMap());// 可以自动转换4类8种
            user.setId(UUIDUtils.getID());// 手动添加id
            user.setActivation(UUIDUtils.getCode()); // 手动添加激活码
            // 调用业务
            UserDao us = new UserDao();
            us.addUser(user);
        } catch (Exception e) {
            e.printStackTrace();
        }
        // 分发转向
        return "/user/login.jsp";
```

集成类　　实现自定义接口方法

多个请求表单最终在同一个类进行处理，只是根据不同的请求参数进入不同方法处理

```jsp
<!--登录的表单 提交到userServlet类处理 -->
<form class="form-horizontal"
    action="${pageContext.request.contextPath }/userServlet"
    method="post">
    <!-- 提交的数设置 为隐藏域 -->
    <input type="hidden" name="method" value="login" />

<!--注册的表单 提交到userServlet类处理 -->
<form action="${pageContext.request.contextPath }/userServlet"
    method="post" class="form-horizontal" style="margin-top: 5px;">
    <!-- 提交的数设置 为隐藏域 -->
    <input type="hidden" name="method" value="register" />
```

通过对请求的参数不通执行不同的方法

技术点3、

发邮件技术：借助邮件工具类和JAR包实现

SendJMail.java 工具类， mail.jar

技术点4、远程改变项目数据库

邮箱中点击【马上激活】链接

提交到Servlet:

String emailMsg = "马上激活";

* 根据激活码修改用户状态：　　给用户发送了一封邮件 邮件里只有一句话（带连接带参数的标签连接标签），用户点击之后会从用户的邮箱跳转到登录界面，并且在后台改变用户的激活状

态（数据库中状态0 改 1）

技术点5、用户退出功能（注销功能）

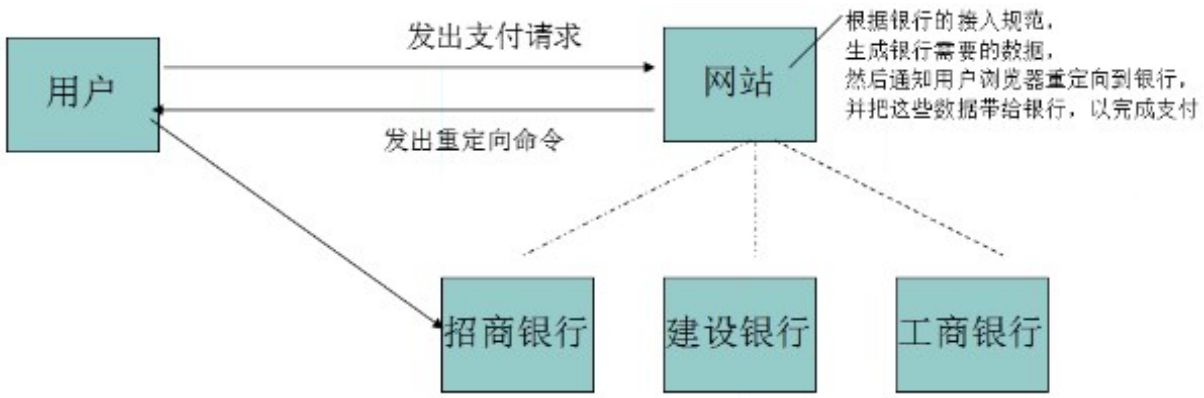
销毁session

```
req.getSession().invalidate();
```

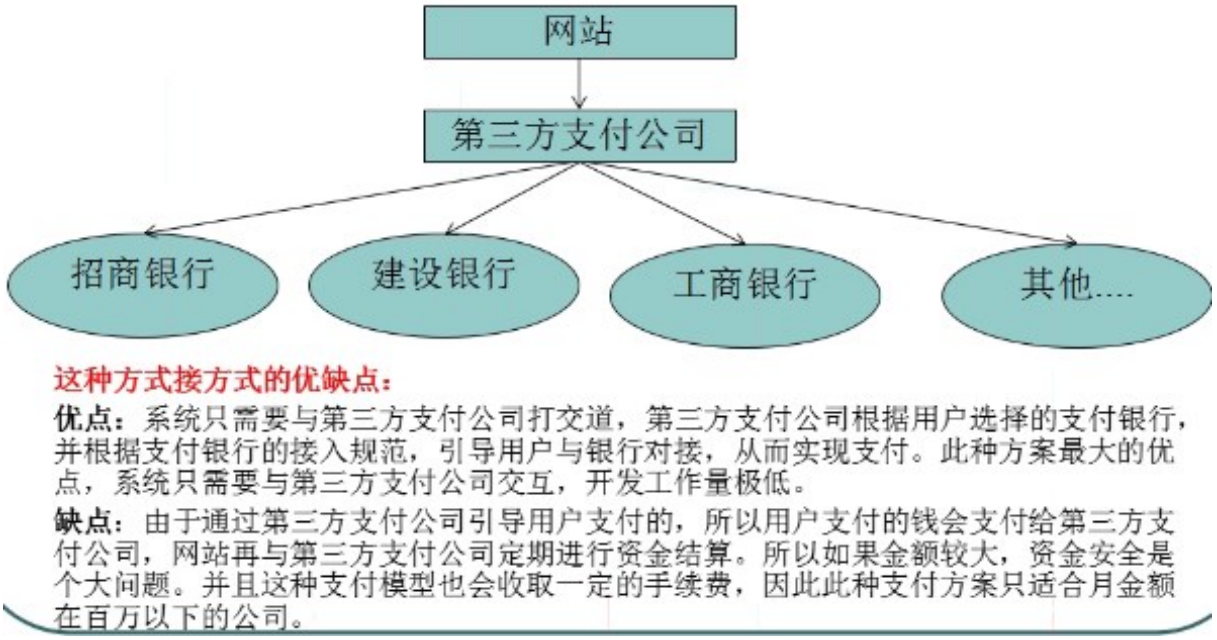
重新设定一个会话给浏览器并跳转到首页

```
Cookie cookie1 = new Cookie("autologin", "");
cookie1.setPath("/");
cookie1.setMaxAge(0);
resp.addCookie(cookie1);
return "/user/index.jsp";
```

技术点6、支付



- 此种模式下，网站需要针对不同的银行开发不同的支付程序，编程工作量很大，并且银行接入规范一旦发生变动，网站程序也要跟着改，维护工作量极大。



操作思路：

1、使用第三方支付公司的接口支付

（申请商家，获得支付秘钥，商户编号 加密验证工具）

2、向第三方支付公司提交相应的数据（重要：订单编号，金额等）

3、用户支付完成返回订单支付结果给商城网站

4、获取返回信息改变订单状态

部分代码：

```
public String pay(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
```

```
//获取请求参数
String oid = req.getParameter("oid");
String money = req.getParameter("money");
String bank = req.getParameter("pd_FrpId");
//调用业务
//修改订单表的收货人信息
String address = req.getParameter("address");
String name = req.getParameter("name");
String telephone = req.getParameter("telephone");
OrderService os = new OrderServiceImpl();
os.updateOrder(address,name,telephone,oid);
//完成支付
String p0_Cmd = "Buy";
String p1_MerId = "10001126856"; //商户编号
String p2_Order = oid; //商户订单号
String p3_Amt = money;//支付金额
String p4_Cur = "CNY";
String p5_Pid = "unknow";
String p6_Pcat = "";
String p7_Pdesc = "";
String p8_Url = "http://localhost:8080/day22_store/payServlet?method=callback";//本网站接收支付成功数据的地址
String p9_SAF = "";
String pa_MP = "";
String pd_FrpId = bank; //用户选择的银行
String pr_NeedResponse = "1";
String keyValue = "69cl522AV6q613Ii4W6u8K6XuW8vM1N6bFgyv769220IuYe9u37N4y7rI4Pl";
String hmac = PaymentUtil.buildHmac(p0_Cmd, p1_MerId, p2_Order, p3_Amt, p4_Cur, p5_Pid, p6_Pcat, p7_Pdesc, p8_Url, p9_SAF, pa_MP, pd_FrpId, pr_NeedResponse, keyValue);
```

```
  //分发转向
  req.setAttribute("p0_Cmd", p0_Cmd);
  req.setAttribute("p1_MerId", p1_MerId);
  req.setAttribute("p2_Order", p2_Order);
  req.setAttribute("p3_Amt", p3_Amt);
  req.setAttribute("p4_Cur", p4_Cur);
  req.setAttribute("p5_Pid", p5_Pid);
  req.setAttribute("p6_Pcat", p6_Pcat);
  req.setAttribute("p7_Pdesc", p7_Pdesc);
  req.setAttribute("p8_Url", p8_Url);
  req.setAttribute("p9_SAF", p9_SAF);
  req.setAttribute("pa_MP", pa_MP);
  req.setAttribute("pd_FrpId", pd_FrpId);
  req.setAttribute("pr_NeedResponse", pr_NeedResponse);
  req.setAttribute("keyValue", keyValue);
  req.setAttribute("hmac", hmac);
  return "/user/confirm.jsp";
 }
 public String callback(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
  String hmac = req.getParameter("hmac");
  String p1_MerId = req.getParameter("p1_MerId");
  String r0_Cmd = req.getParameter("r0_Cmd");
  String r1_Code = req.getParameter("r1_Code"); //支付结果 1代表交易成功
  String r2_TrxId = req.getParameter("r2_TrxId");
  String r3_Amt = req.getParameter("r3_Amt");
  String r4_Cur = req.getParameter("r4_Cur");
  String r5_Pid = req.getParameter("r5_Pid");
  String r6_Order = req.getParameter("r6_Order"); //订单编号
  String r7_Uid = req.getParameter("r7_Uid");
```

```
  String r8_MP = req.getParameter("r8_MP");
  String r9_BType = req.getParameter("r9_BType");
  boolean verifyCallback = PaymentUtil.verifyCallback(hmac, p1_MerId, r0_Cmd, r1_Code,
r2_TrxId, r3_Amt, r4_Cur, r5_Pid, r6_Order, r7_Uid, r8_MP, r9_BType,
"69cl522AV6q613Ii4W6u8K6XuW8vM1N6bFgyv769220IuYe9u37N4y7rI4Pl");
  if(verifyCallback){ //判断结果是否安全
   if("1".equals(r1_Code)){ //表示支付成功
    OrderService os = new OrderServiceImpl();
    os.updateState(r6_Order);
   }
  }
  //跳转到用户订单页面
  return "/orderServlet?method=findOrdersByPage";
 }
```

技术点7、将项目部署到Liunx系统上

1、在linux系统中安装：jdk mysql tomcat redis

2、把数据导入到linux系统的mysql中

3、把应用打成war包 （修改c3p0-config.xml中的密码）

4、部署应用到tomcat