1. 这是一个web网页应用，作用是社团管理。实现的功能有用户个人信息管理，加入社团，参加社团活动。创建社团、发布社团活动、社团内部成员管理等。
2. 我们采用的框架是MVC，应用的技术是Servlet+JSP
3. 我主要负责的是servlet部分的逻辑编写，由于正好处于整个项目的中间部件，对前端界面及后端数据库都有一定接触，所以能够对整体有更好的把握。
4. 回忆当时主要涉及到的几个技术点：

* 乱码的解决：

乱码问题：数据库、tomcat等支持ISO8859-1，读取字节。java语言使用Unicode编码、中文使用gb2312编码。

https://blog.csdn.net/qq\_31368673/article/details/80949488

统一编码。

统一为gb2312

1. 在英文浏览器中使用中文，直接在page里将charset设置成gb2312
2. 表单填写中文信息，传递给request对象，需要将request的编码格式设置成gb2312；request.setCharacterEncoding(“gb2312”)
3. 从数据库读取和存入数据库出现乱码。要使用中文字符，要转换成GB2312格式

统一为UTF-8：因为UTF-8支持世界上所有的语言，且相比于UTF-16，采用变长编码，更节省空间。

1. 页面设置编码：其中contentType中charset用来设置服务器发送给客户端时的内容编码；pageEncoding 用来设置JSP源文件本身和响应正文中的字符编码。
2. 传参数乱码：原因是tomcat采用ISO8859-1编码。方法，在service.xml中设置URIEncoding为UTF-8，在servlet界面通过request.setCharacterEncoding("UTF-8"); response.setCharacterEncoding("UTF-8");
3. 数据库读取乱码：首先在数据库设置字符集为utf-8；然后在连接数据库时加上?characterEncoding=utf8即可。

* 跳转

界面间的跳转

servlet定向到前端特定界面

RequestDispatcher view = request.getRequestDispatcher("/AddCommunity/applyCommunity.jsp");

view.forward(request, response);

response.sendRedirect("/mainpages/afterlogin.jsp");

两种跳转方式的区别：

<https://www.cnblogs.com/walle2014/p/3659990.html>

getRequestDispatcher().forword：服务器端跳转，界面网址不会改变、携带原来的request对象和response对象跳转、可以通过request和response传值。只能定向到同一个web应用中的url。

sendRedirect：客户端跳转、网址会改变、新建request对象、只能通过url传值。可以重定向到任何url。

* 传值

前端不同界面间的传值

jsp与sevlet之间的传值

1. 用到的三个基本对象：session、request、response
2. servlet传值到jsp：request.getDispatcher().forword()传值时，request.getSession().setAttribute("comID",community.getComID());

<% Student student = (Student)request.getSession().getAttribute("loginStudent");%>

request.setAttribute()

request.getAttribute()

1. 通过sendRedirect（）传值，通过url传值

jsp与jsp传值

https://github.com/welikeyou/community

* 路径
* 项目结构