# 第二周周报

## 邮箱验证登录的实现

使用 commons-email的jar包

|  |
| --- |
| public class SendEmailSmsUtils {  public static void sendEmail(String address, String code) {  // 不要使用SimpleEmail,会出现乱码问题  HtmlEmail email = new HtmlEmail();  try {  // 这里是SMTP发送服务器的名字：，普通qq号只能是smtp.qq.com ；  email.setHostName("smtp.qq.com");  //设置需要鉴权端口  email.setSmtpPort(465);  //开启 SSL 加密  email.setSSLOnConnect(true);  // 字符编码集的设置  email.setCharset("utf-8");  // 收件人的邮箱  email.addTo(address);  // 发送人的邮箱  email.setFrom("2363405213@qq.com", "YUEYU");  // 如果需要认证信息的话，设置认证：用户名-密码。分别为发件人在邮件服务器上的注册名称和得到的授权码  email.setAuthentication("596015571@qq.com", "dsgsdgsgsdhjjytr");  email.setSubject("【邮箱验证】");  // 要发送的信息，由于使用了HtmlEmail，可以在邮件内容中使用HTML标签  email.setMsg("您的验证码为："+code+"\n验证码120s后失效，请尽快使用！");  // 发送  email.send();  log.info("邮箱 {} 的验证码发送成功",address);  } catch (EmailException e) {  log.error("邮箱 {} 的验证码发送失败",address);  }  }  } |

## 左侧导航栏的制作与实现



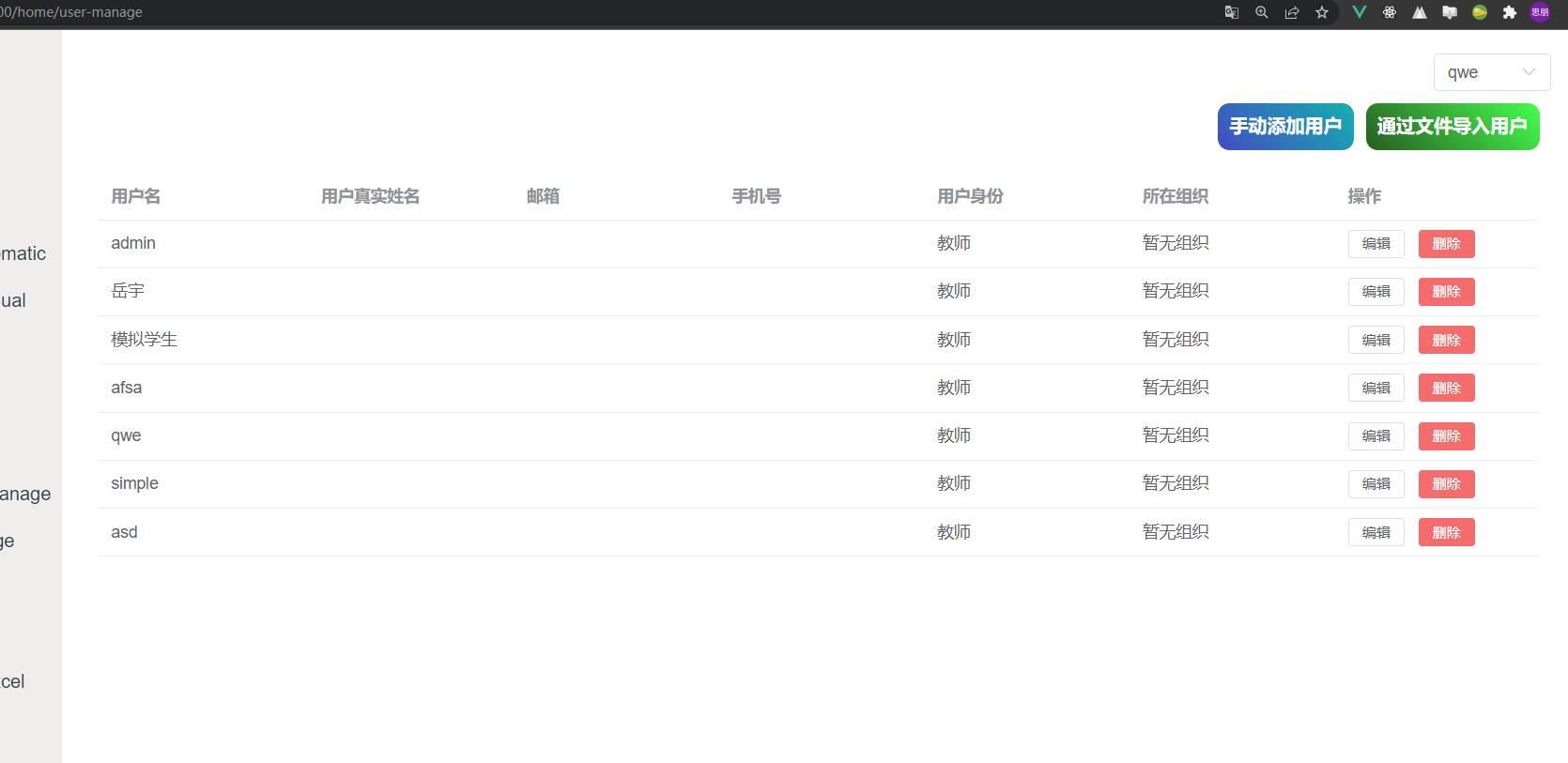
为了尽可能地减少代码的冗余以及优化页面的性能，左侧导航栏的设计显得尤为重要。

考虑到导航功能的层级以及实现的难度。

这里将【导航组件】做成【递归组件】的形式，不管有多深的层级，都可以正确展示以及正确访问相应的路径。

为了减小页面的性能消耗以及优化代码质量，在处理层级事件的时候采用了【事件委托】，即将【子元素的事件】绑定到父元素身上，由父元素统一处理。**这样就不会因为每一个子元素绑定相应的事件而导致浏览器内存开销过大而造成卡顿问题**。

## 用户管理功能的实现



做到对用户的增、删、改、查。

为了高效复用代码，用户的【增、改、查】都使用的同一个页面，也就是注册页面。

### 实现了手动添加用户和通过文件导入用户的功能

手动添加用户就是用户的注册。

通过文件导入用户，就是需要通过excel文件，按照规定的格式输入用户的信息然后上传到后台，对上传的用户信息进行解析，即可完成自动导入

