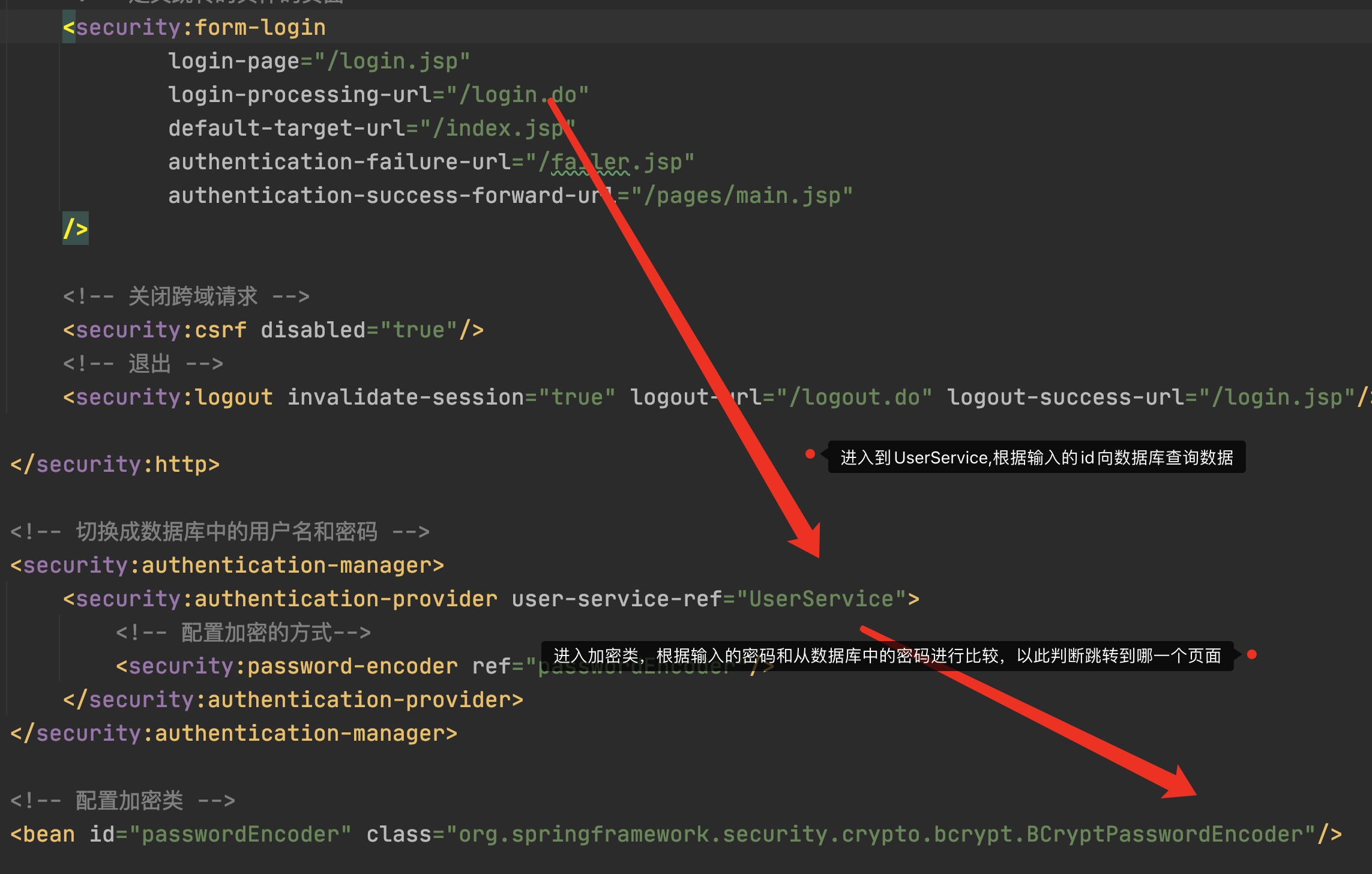
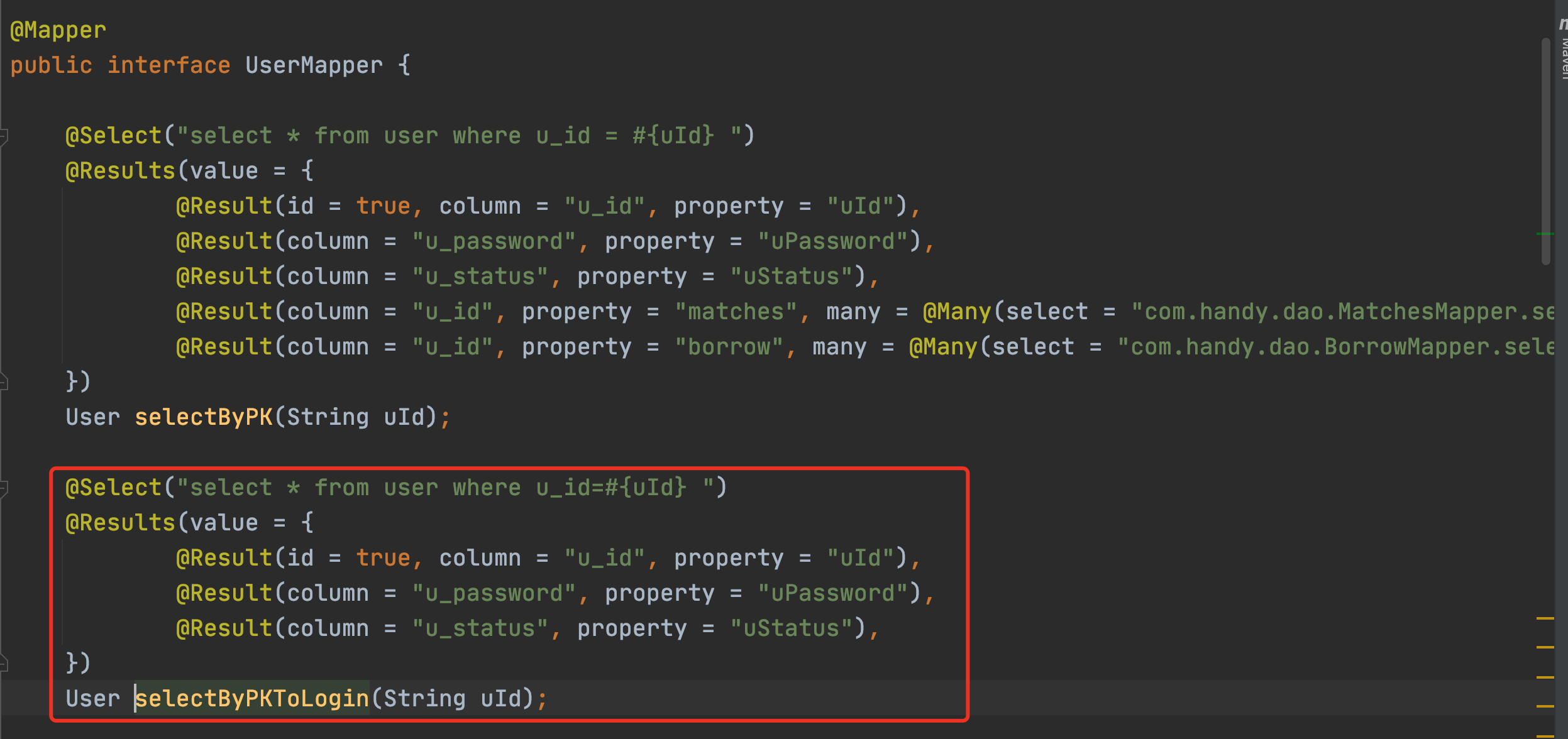
tomcat启动后会读取web.xml.依次加载其中的context,listener,filter,servelt创建ServeltContext对象。

在spring-security中配置了from-login属性，设置了登陆成功或者失败后该跳转向对应页面。

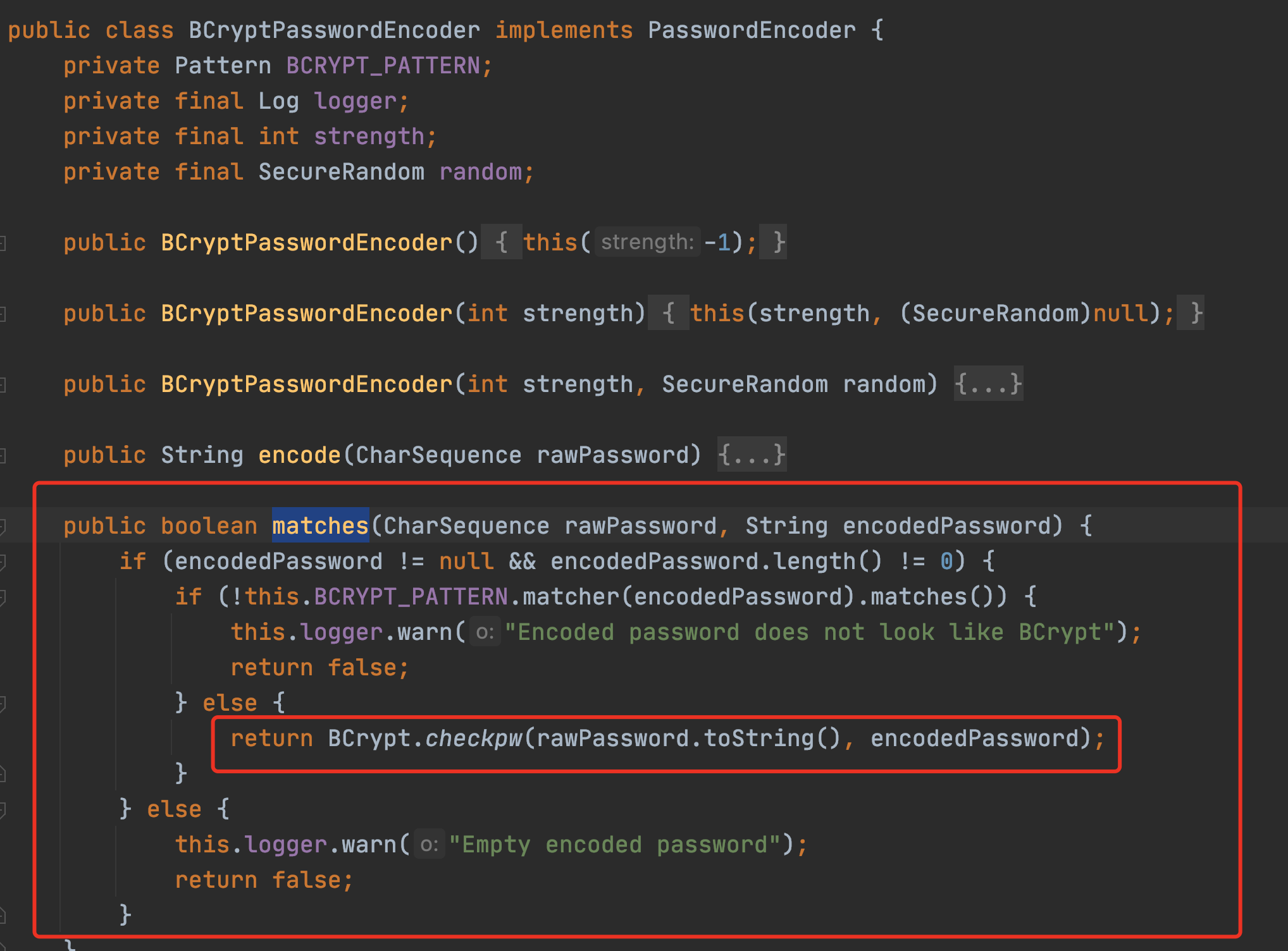
首先会展示登陆界面，当输入完账号密码点击登录时，会以psot方式提交表单.之后会进行如下操作，



在UserServiceImpl中调用loadUserByUsername，而在loadUserByUsername中又会调用UserMaper中的selectByPKUsername查询数据库

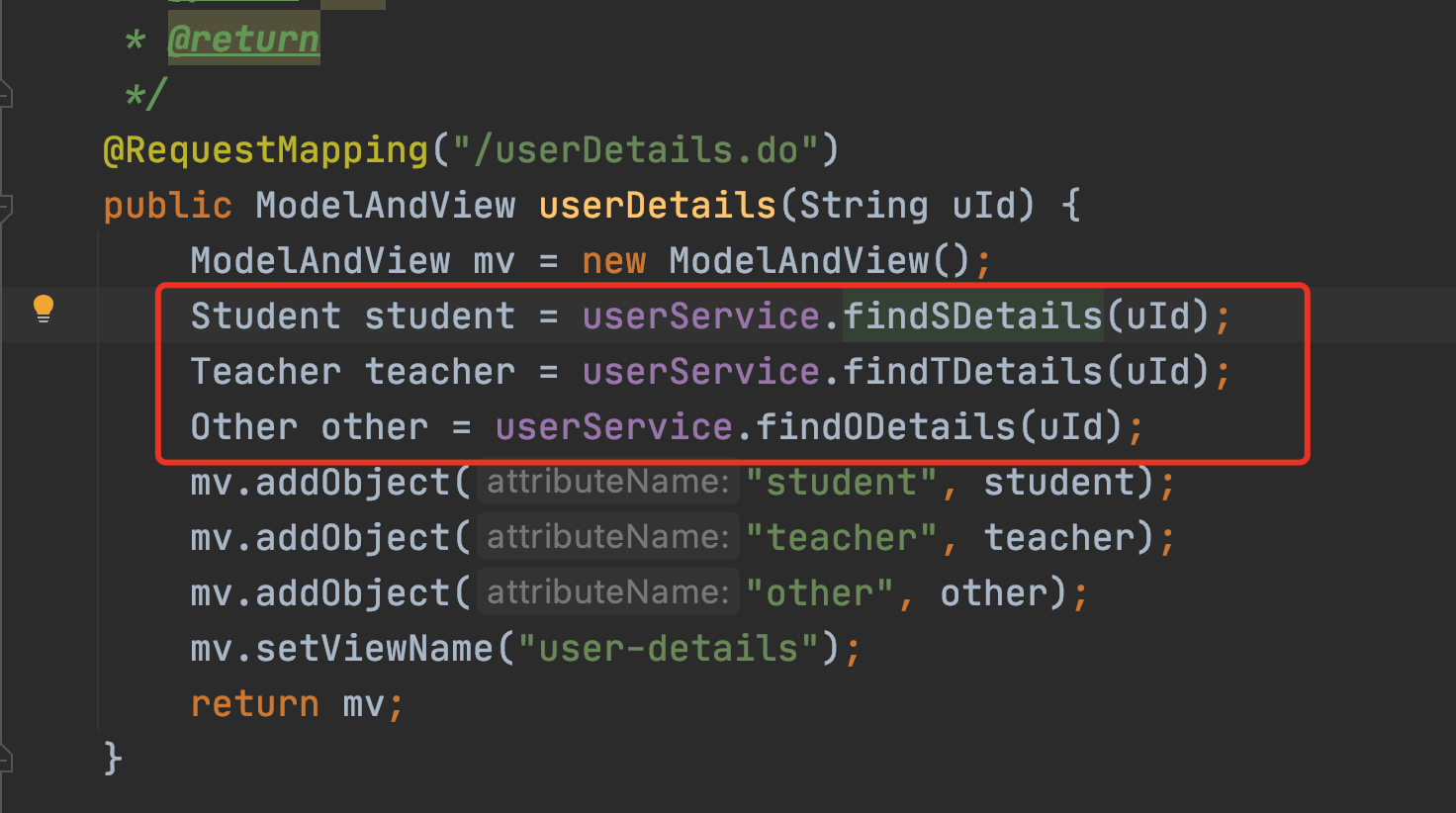


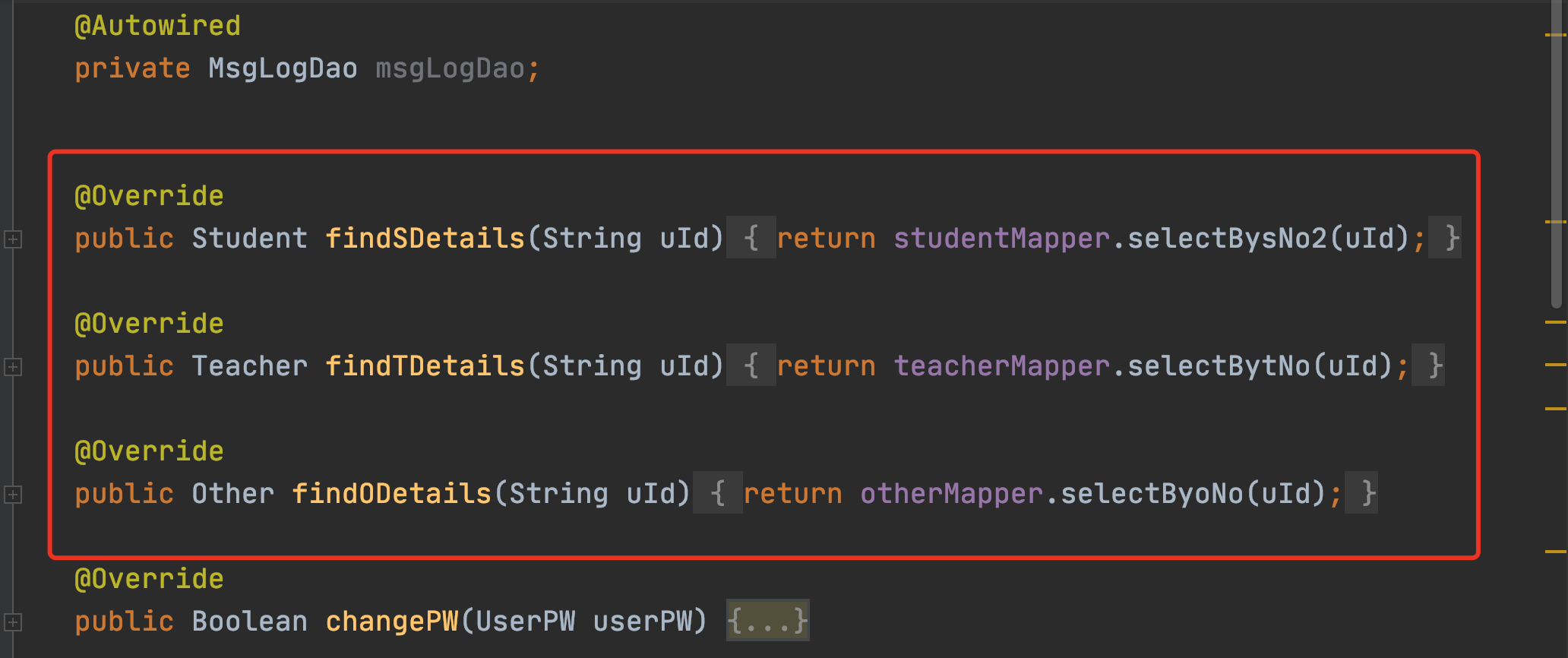
在加密类中调用marches来对输入的密码和数据库中查询到的密码进行匹配



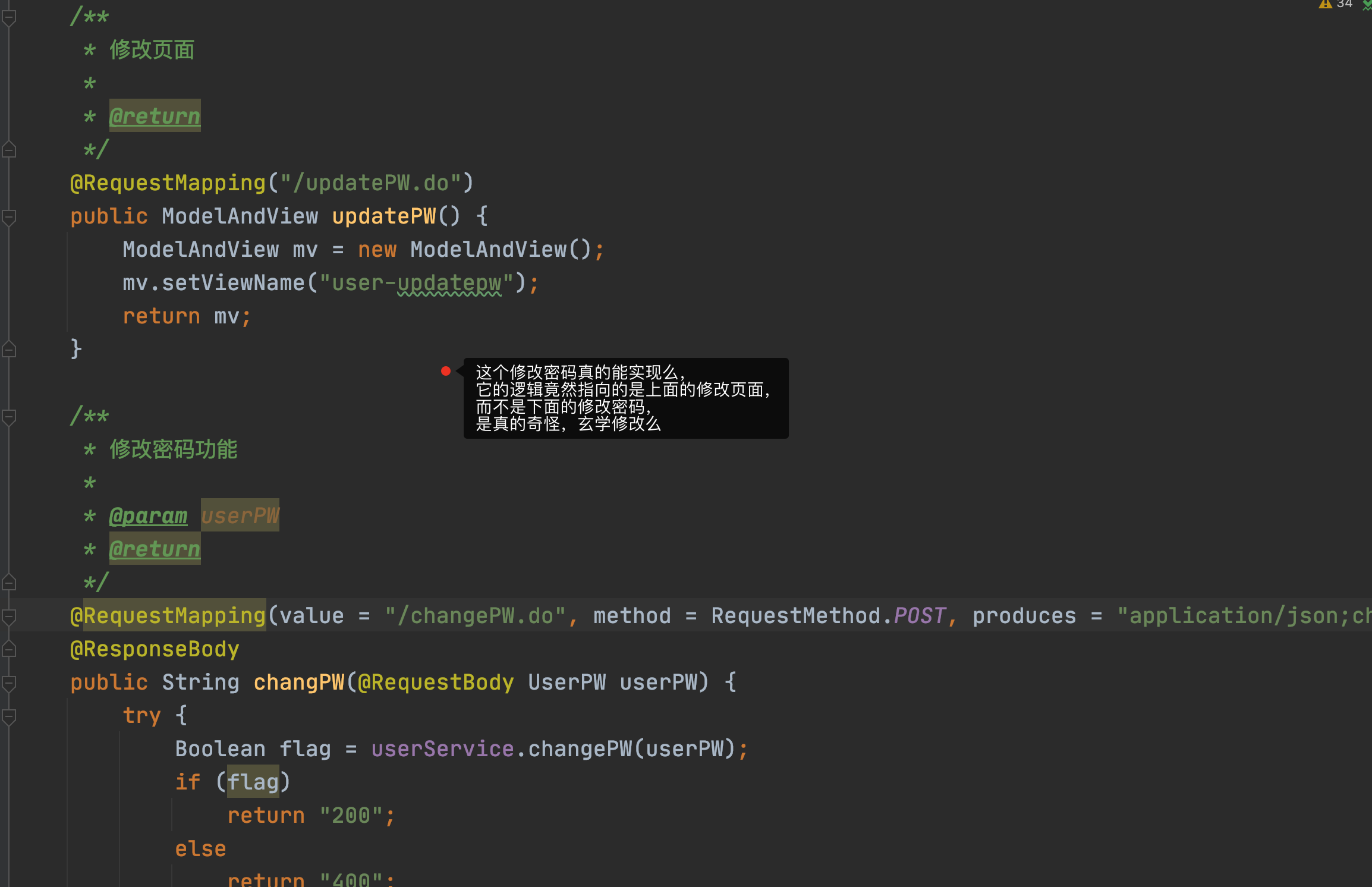
成功后会进入到主界面，主界面主要包括三部分，header,aside,footer

其中header中主要是记录和操作用户个人的一些信息，当查看用户信息时，会进行如下操作，首先UserController会调用userDetails,而userDetails又会调用UserServiceImpl中的相关方法，同样的会进入到对应的Maper中查询数据库。

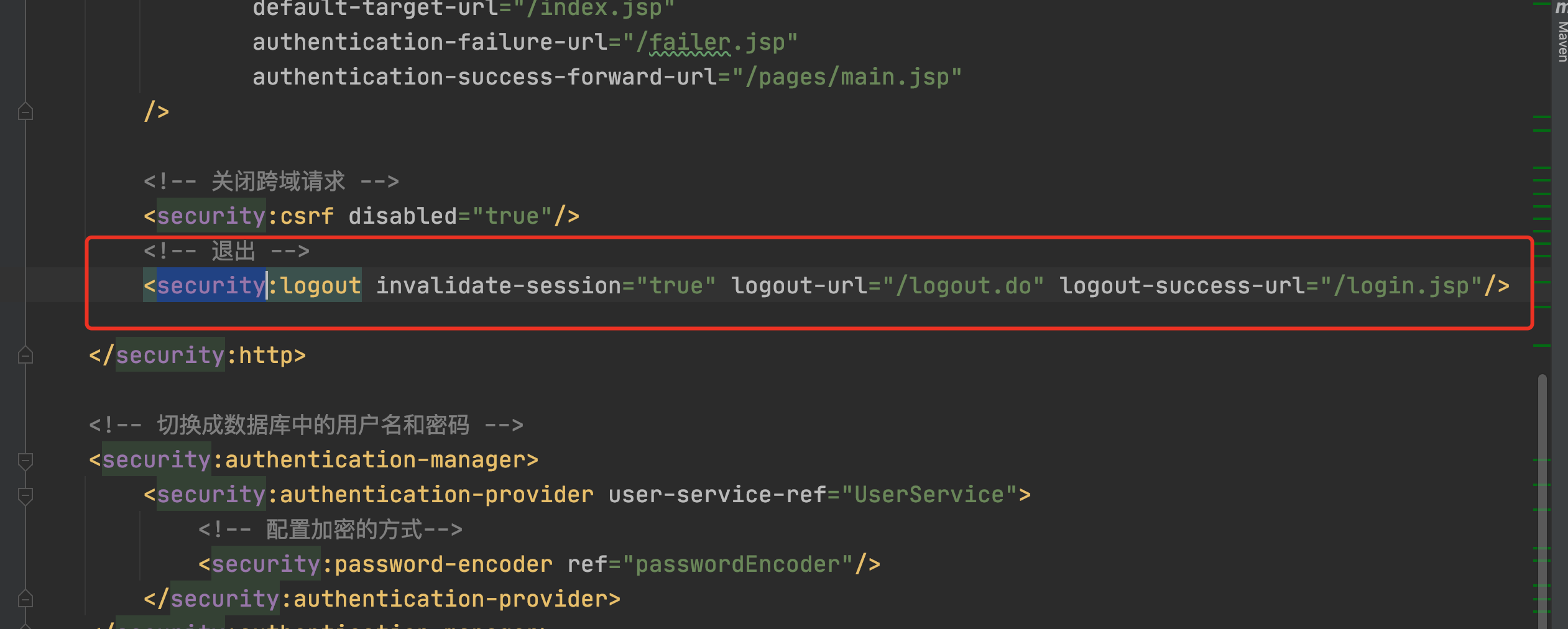




当用户修改密码时，

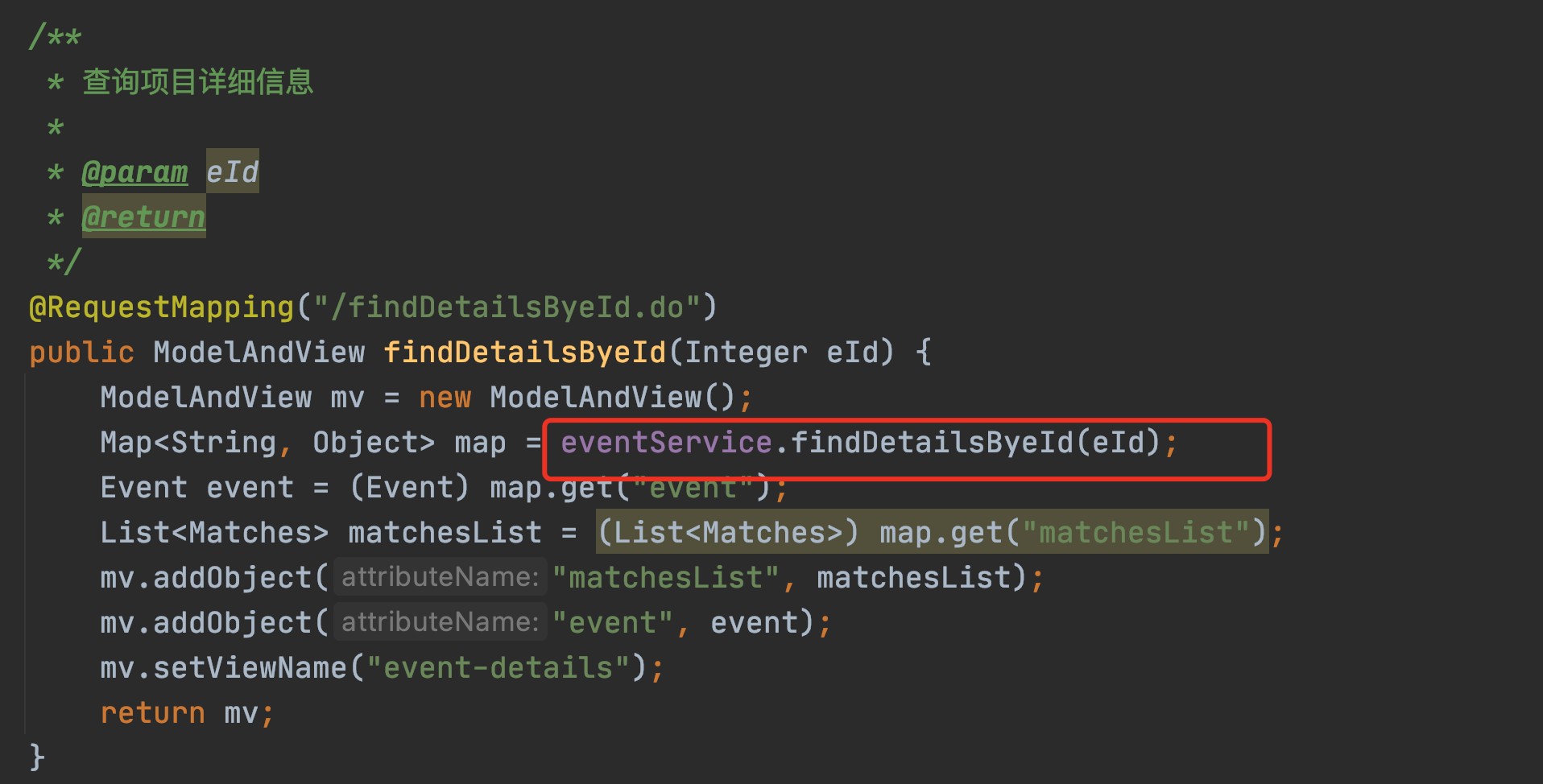


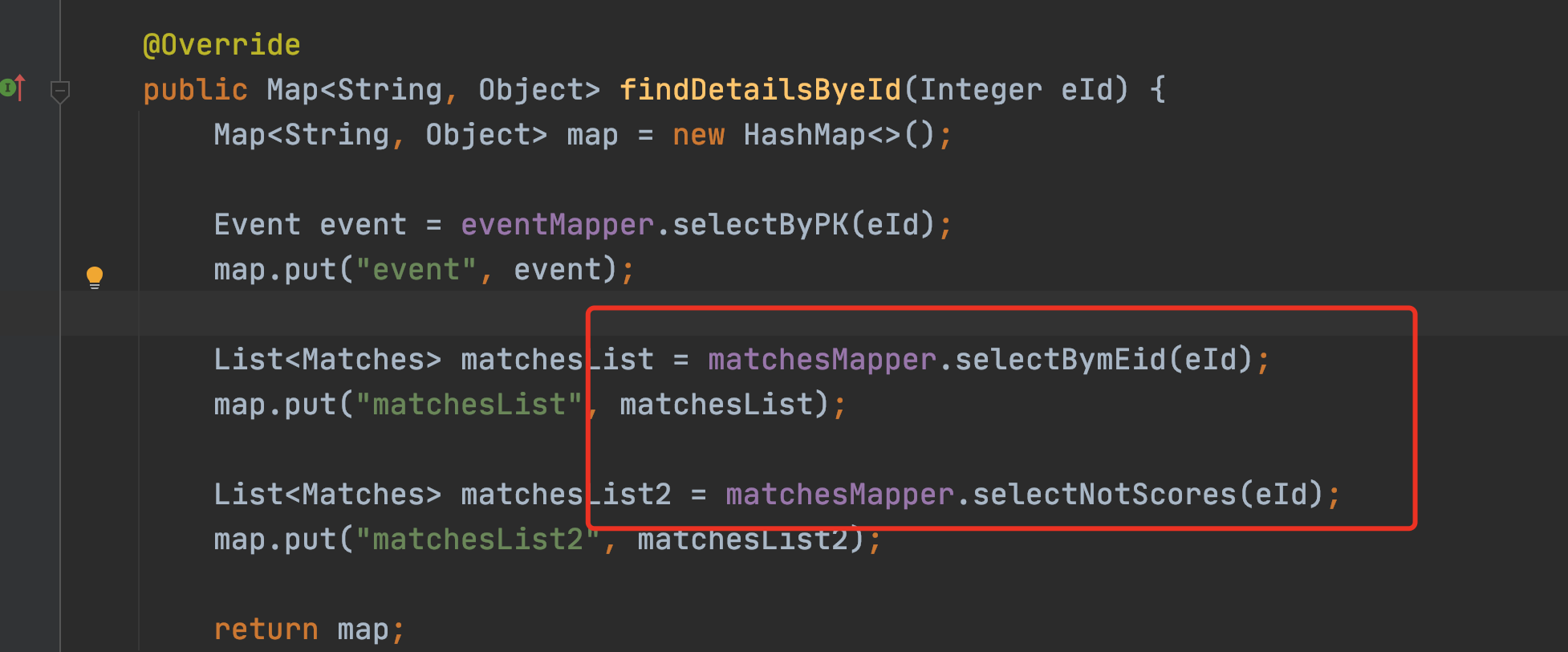
当退出时，会清除会话，然后跳转到登陆界面



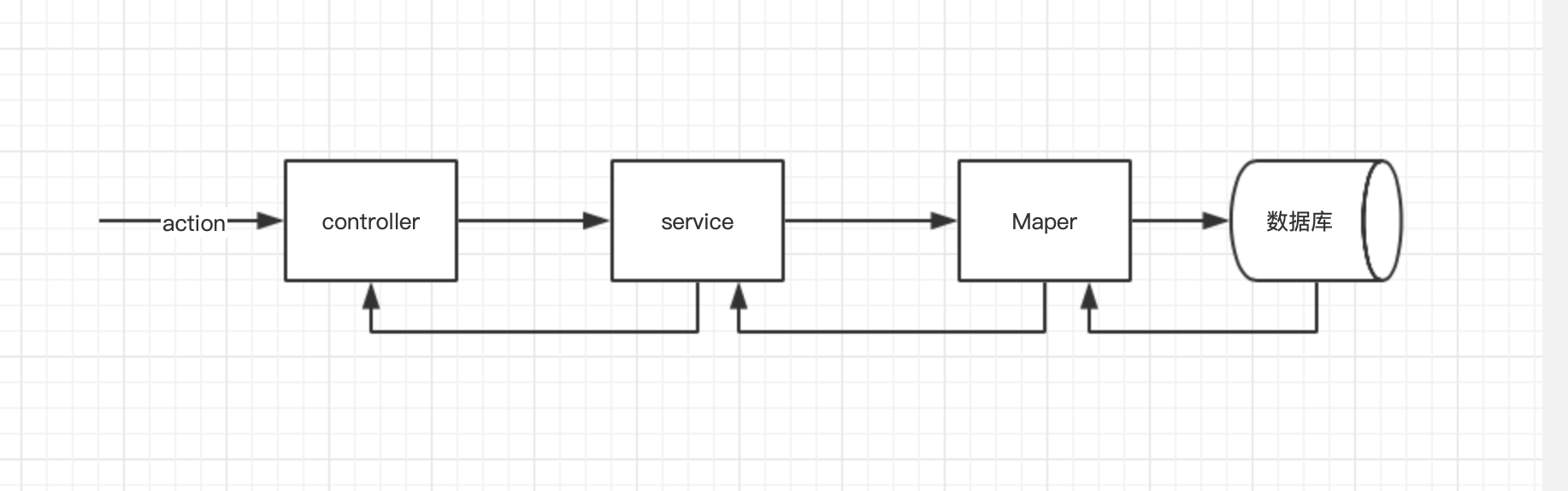
在aside中，对于不同的角色会显示不同的内容，对于user显示的就是自己的参赛相关信息比如项目列表，对于admin显示的就是管理界面，比如学生信息，也有些所有角色都会显示的内容，比如运动会相关信息，而角色的确认，是在登陆校验时一并从数据库中查询得到的。

其中的逻辑也大致相同，就拿user用户下的项目列表来举例子，首先会在EventController中调用findDetailsById,在findDetailsById中又调用EventServiceImpl下的findDetailsByeId，findDetailsByeid又调用响应Maper中的方法查询数据库然后一级一级返回数据





总结起来就是



footer则显示了一些开发版本相关信息