

Code Review Checklist

Review Information:

Name of Reviewer: 黄凯文

Name of Coder: 蒋嘉荣

File(s) under review: UserAccountManager.java

Brief description of change being reviewed:

考虑到 UserAccountManager 类是属于 servlet 方法实现，作为一个 controller 方法实现应该清晰明朗，在这里主要修改和建议是考虑到 UserAccountManager 类中的功能实现方法过于集中，不易于维护和扩展，不符合 Clean Code 的 small, do one thing 的原则，需要重构。最后考虑到 session 的操作的严谨性，需要多加些相关注释。

Review Notes (problems or decisions):

1. 源代码的风格规范

1) 遵守约定编码规范，coder 在这方面做的很好。

```
if(action.equals("modifyPassword")) {  
    try {  
        if(UserInfoChange.changePassword(request, response))  
            request.getRequestDispatcher("accountMenu.jsp").
```

2. 注释

注释方面主要需要留意 session 和 request 的存留实现问题，由于两者存在明显的不同，所以有必要对两者的操作和修改时加些注释，方便日后 checker 的检查，这里 coder 就没有很好的区分两者差别的重要性，所以没有添加注释，建议在每次使用 session 时单方面对 session 添加注销，这样就可以很好的区分两种存储方式。

```
if(action.equals("logout")) {  
    session.removeAttribute("userName");  
    session.removeAttribute("curUser");  
    request.getRequestDispatcher("index.jsp").forward(re
```

3. 使用的算法

由于 servlet 仅作为一个 controller 来对数据库操作和对 jsp 进行页面相应，而且系统的实现也较为简单，所以没有很高的算法要求，也可以说基本没必要使用相关算法。

4. 错误处理

由于要对数据库进行查询和修改处理，所以必须要使用相关的数据库的异常处理方法，coder 在这里也做得很好，采用 try, catch, 的异常捕捉形式，确保 JavaBean 里对数据库的操作成功。

```
if(action.equals("modifyQQ")) {  
    try {  
        if(UserInfoChange.changeQQ(request, response))  
            request.getRequestDispatcher("accountMenu.jsp")  
        else  
        {  
            request.setAttribute("msg", "修改失败,");  
            request.getRequestDispatcher("modifyQQ.jsp")  
        }  
    } catch (SQLException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
}
```

5. Clean code

前面已经说到过此类中有一个代码很长方法，不符合Clean Code的 small, do one thing的原则，需要重构。建议这里直接使用java的相关重构工具，方便检查。

需要重构的doPost方法如下：

```

    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        request.setCharacterEncoding("GBK");
        response.setCharacterEncoding("GBK");
        //从请求中获取session对象
        HttpSession session=request.getSession();
        //如果session对象为空则返回
        if(session.getAttribute("curUser") == null) {
            return;
        }

        String action = request.getParameter("action");

        //处理请求
        if(action.equals("modifyEmail")) {
            try {
                if(UserInfoChange.changeEmail(request, response))
                    request.getRequestDispatcher("accountMenu.jsp").forward(request, response);
                else
                {
                    request.setAttribute("msg", "修改失败, 请尝试再次登录");
                    request.getRequestDispatcher("modifyEmail.jsp").forward(request, response);
                }
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }

        if(action.equals("modifyPassword")) {
            try {
                if(UserInfoChange.changePassword(request, response))
                    request.getRequestDispatcher("accountMenu.jsp").forward(request, response);
            }
        }
    }
}

```

建议重构后的一个实例如下：

```

        //处理请求
        if(action.equals("modifyEmail")) {
            modifyEmail(request, response);
        }

        private void modifyEmail(HttpServletRequest request,
            HttpServletResponse response) throws ServletException, IOException {
            try {
                if(UserInfoChange.changeEmail(request, response))
                    request.getRequestDispatcher("accountMenu.jsp").forward(request, response);
                else
                {
                    request.setAttribute("msg", "修改失败, 请尝试再次登录");
                    request.getRequestDispatcher("modifyEmail.jsp").forward(request, response);
                }
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }
}

```

6. 安全性

由于对 servlet 的安全性不是很了解，加上 servlet 仅仅充当一个 controller 角色，所以安全性考虑较少。

SVN Versions (if applicable):

Before review: Version_1

After revisions: Version_2