## 第二周总结：

**大致总结**

本周在上周的基础上进行改进加强。在熟悉SSM框架的同时，通过文件上传操作熟悉Java的IO操作、通过模糊查询加深对非空与字符串的控制判断，在将代码规范的同时，利用视图简化对数据库的操作。大致总结如下：

（1）一个合理化的设计流程是多么的重要

在简单的逻辑与分类不太复杂的情况下，预先设计设计流程似乎显得有些多余，可是在进入到复杂的项目与团队协作时又是那么的不可缺少，而这正是自己所缺少的。

（2）考虑问题是不要怕复杂，这样容易考虑不全

本次又9中情况，可进行3大类区分，由于没有仔细分析，导致在控制层的逻辑处理与mybatis的xml语句配置都出现大的问题。

（3）JS的重要性

在文件上传时后台过于复杂并且不好实现的情况下，深刻意识到JS的重要性，而JS在项目中的比重也日渐加重，需要对这部分知识进行强化学习。

（4）代码规范的重要性

上午开会时，深刻意识到代码规范的重要性。代码规范不仅仅是团队协作的基础，也可以看出一个人对自己的要求。

**具体内容与问题**

以下是在项目练习中的具体内容与问题。

1. 视图创建，增加用户表图片字段

在设计视图时一定要全面考虑将来需要用到的数据，以免重复劳动。（确定所需列数）

2.将代码先规范

将业务代码移到service中

3.session问题

将user处理为全局

4.登录问题

1)越过登录直接进入到主页面的问题

2）密码验证问题

3）前端登录页面 各种格式问题 前端处理

5.边界问题

空指针问题，在代码设计时一定要注意对象的非空、是否异常等问题

6.模糊查询

1）其中最大的问题莫过于在用户搜索时如何处理用户误操作输入，此次处理的是空格字符问题。既要考虑输入是否为空，也要考虑非空情况下是否有值，加上是两个对象，情况变得相对复杂。

2）其次问题也在mybatis的xml配置书写，一是对于like后的格式也查了很多资料，最终落实；二是如何根据不同的情况书写查询语句。

7.优化查询

利用视图查询----减少多表查询和连接数据库次数（T）

8.删除操作优化

由物理删除改为逻辑删除

9.图片上传操作

1）不能简单有效地获取普通表单项。

在文件上传并不太难，难的是表单中除了提交文件外还要提交其他信息，可能导致获取前端数据的困难，以及如何保持一次请求(调用commonsMultipartResolver.resolveMultipart(httpRequest); 方法时，原request会消亡！)；

当表单<form>标签加入enctype="multipart/form-data"后，不能使用request.getParametre()传值

2）编写代码的混乱

发现编码十分困难。

后端：既要单独书写文件上传代码，也要书写controller获取其他前端数据，

前端：考虑前端如何写才能让后端

3）解决方法

直接使用ajax处理解决这些复杂的问题

$.ajax({

id: '',

url : '${\_ctx}/index/addUserInformation.do',//1.do的调用

type: 'post',

data : $('#addFrom').serialize(),//2.将数据直接系列化封装（)

dataType : 'json',

success : function () {

//3.处理返回数据

}

})