**一．功能模块：**

程序

管理员

导入出勤表

导入成员列表

导入每个成员的课表

有查看三个表的权利

导入请假表

生成预览

显示制作状态

计算每人每天的出勤情况

可以打印列表

**二．具体的功能：**

**a)程序：**

**>>导入用户列表：从数据库直接抓就可以；（userBase、user）**

**>>导入每个成员的课程表：(subjectTable)**

**核心在于如何去确定课程表的格式：(subjectTable properities)**

**姓名+学号+星期几+第几节课+初始时间（需要做个转换，以日期为单位）**

**+终止时间+课程名称**

**如何处理内存和效率的问题：**

**目前的模式采用的是，一个用户对应一个class（效率上是有问题的）**

1. **先要对所有的课表查询一下有没有创建(只查询状态，不获取数据，subjectTable)**
2. **管理员点击具体某个用户的列表，再加载；**
3. **管理员点击制作出勤表，再全部加载；**

**>>导入每个成员的请假表**

**请假表格式：(RequestTable)**

**学号+姓名+开始时间+结束时间+请假理由(RequestTable properities)**

**模型和控制，没有解耦，控制做了模型该做的事情，需要重构；**

**请假表: (BrowerInfo)**

**也有一个请假表查看界面**

**？是不是所有的查看界面可以同时导向一个地方（查询界面）**

**怎么导入，数据怎么分配（controller、user.requestTable）**

**（？谁去控制数据导入？数据怎么分配）**

1. **user这样一个数组，采用一个For循环，就能找到各自匹配的数据。（数据分配解决了）**
2. **状态更新让controller来做就可以了**

**>>显示制作状态(AdvanceTableCtl、controller)**

**初始化，导入用户后，开始查询用户请假表和用户课表，这块也可能需要重构**

**管理员导入出勤表后，需要更新出勤表情况**

**>>计算每人每天的出勤状态：(user)**

**核心算法：匹配算法(user)**

**同时调用三个数据，进行比较匹配：算法需要设计一下**

**b)管理员  
>>导入出勤表(Worktable)**

**生成预览(controller：不做数据转换)**

**出勤表的数据库格式(workTable的properities)**

**将数据进行分配(调用worktable的方法)**

**原始数据保留下来，保存至数据库中(调用worktable的方法)**

**>>查看三个表**

**查看的内容，传递到查询界面就可以(Brower)**

**(？要不要建一个查询类，查询类定义一个标准的查询接口，这个接口还可以在查询类中使用)**

**>>生成预览（根据出勤表的时间周期:暂时不打算解耦）**

**预览表的内容：**

**低级版：学号、姓名、迟到几次、请假几次、早退几次、未签几次**

**高级版：有具体的未签记录**

**>> 打印列表**

**生成各种格式的就可以。**

**三．操作图**

**导入用户列表**

查询课表状态，并显示

导入用户列表

**用户操作**

统计每一天的出勤

生成预览

更新数据库

生成报表

更新数据库

查看出勤表

生成请假表预览

查看请假表

导入出勤表

查看课表

**统计出勤**

**四．类图**