**修改过的用例用蓝色标注**

**Step0 -> Step1 教师、班级、选课数据的导入用例成功执行后**

courseInfo.csv settings.ini

studentInfo.csv teacherInfo.csv

自助考勤详细设计

Public void StartCheckin(String teacherWechatID,String CourseID)

{

1.系统维护全局教师课程队列，将教师、课程信息、班级信息、开始考勤时间入队列，并开启/维护Timer，将timer的定时周期设置为下一个将要出队的时间点。

2.在Seq.csv中增加一行TeacherID,CourseID,SeqID,StartTime

3.系统将时间窗口有交叠的前一门课程需要做时间范围检查，然后将超出上课时间区间的课程直接清除出队列，该课程考勤时间窗口设置为关闭

1) 记录考勤开始时间，更新Timer；

2) 设置一个列表记录当前教师课程队列；

3) 打开teacherInfo.csv文件；

4) 找到匹配WechatID和teacherWechatID的教师信息，获取到该教师的TeacherID；

5) 打开courseInfo.csv文件；

6) 根据课程号，找到该课程的名字

7) 打开seq.csv文件；

8) 初始化一个seqID = 1；

9) 找到该CourseID对应的最新的SeqID,更新seqID；

10) 将获取到的TeacherID，CourseID，更新过的seqID，考勤开始时间StartTime依次追加到文件seq.csv中；

11) 将当前教师课程队列存到全局教师课程队列中

12) 检查Timer，超出上课时间区间的课程清除出队列

13) 生成<工号>\_<课程号>\_<考勤序号>\_checkinDetail.csv

}

Public void StopCheckin(String teacherWechatID,String CourseID)

{

从全局队列中删除该教师的课程信息

重新设置下一个timer的时间间隔

1. 打开teacherInfo.csv文件；

2. 根据教师微信号，获取该教师的TeacherID；

3. 打开courseInfo.csv文件；

4. 根据课程号，获取参加该课程的班级名字；

5. 打开studentInfo.csv文件；

6. 根据班名，获取参加该班的学生ID；

7. 打开该课程的checkinDetail.csv

8. 对考勤的学生进行考勤认定，未参与考勤的学生，系统将在checkDetail的数据行增加一条缺勤记录

9. 从全局队列中删除该教师的课程信息

}

public boolean StuCheckIn(String \_stuWechatID , BufferedInputStream inputStream)

{

系统登记考勤认定成功的信息IsSucc部分为 True

未认定成功的学生系统将其保持在一个临时的变量中，直至考勤时间窗口关闭时再写入checkDetail 的数据行

checkinResult 一直不写入任何信息

checkinType登记为Auto

一直未响应考勤系统的学生在考勤窗口关闭之后，将在checkDetail 的数据行增加一条考勤记录

checkinResult 一直不写入任何信息

checkinType登记为Auto

1. 记录当前时间

2. checkinType为Auto

3. 根据学生微信ID，找到该学生的班级ID，学号，以及样本地址

4. 根据课程号，找到该课程的名字

5. 判断当前队列中有无该课程号和教工号当前队列中有该课程号和教工号

6. 考勤认定

7. 增加考勤记录

8. checkinResult一直不写入任何信息

}

public boolean StuLeave(String \_stuWechatID , BufferedInputStream inputStream)

{

系统登记请假学生的信息IsSucc部分为 为空

checkinResult 写为 “请假”

checkinType登记为Auto

证据路径填写假条的保存路径

时间为当前时间

1. 记录当前时间

2. 根据学生微信ID，找到该学生的班级ID，学号，以及样本地址

3. 根据班级ID，找到该课程的教工号,课程号

4. 根据教师ID，获取seqID

5. 获取文件名

6. 增加考勤记

7. checkinType登记为Auto

8. checkinResult写为 “请假

9. 证据路径填写假条的保存路径

}