**微 信 考 勤 文 档 说 明 书**

1. **引言**
   1. **编写目的**

对于程序系统的设计考虑，以及软件开发中将要面临的问题及其解决方案进行设计和安排——包括程序系统的基本处理流程，程序系统的组织安排，模块划分，接口设计，功能分配，明确开发风险以及项目规划等，需编写此文档供开发人员分析讨论。

* 1. **项目背景与功能**

随着智能手机的发展，微信作为时下最热门的社交信息平台，也是移动端的一大入口，正在演变成为一大商业交易平台，其对营销行业带来的颠覆性变化开始显现。微信考勤系统是基于微信平台的一款软件，最大的特点就是使用方便，节省用户的时间，功能众多并且便于维护。老师能自助考勤，抽点，在线进行请假和请假认定，并能手动修改考勤记录，实时显示考勤结果，也能显示历史统计和出勤成绩的输出等常用功能。对于学生来说能方便的使用自助考勤，自助请假节省时间，并且能在线查看自己的考勤记录等功能。对于系统的管理者也能提供友好的服务，方便管理者管理,修改和维护，工作将更加轻松有效，整体运作更加协调。

1. **需求分析**

**2.1 系统的功能需求**

1．管理员用户可以在学期初选修课时导入数据包括班级，教师，课程信息表。

2. 系统可以检查外部数据格式，数据是否重复，数据是否逻辑。

3. 教师可以在上课开始前或者开始后打开自助考勤功能。

4. 若老师在上课期间开启自助考勤窗口，那么可以开启抽点功能。

5. 若考勤窗口关闭后，教师可以为某次课程手工增加记录。

6. 教师可以在没有开启考勤窗口的时间统计某门课某次上课的出勤率和该课的平均出勤率以及统计汇总信息。

7. 学生可以在考勤窗口关闭之前申请请假并提供请假证明。

8. 系统根据已经设置的时间窗口（默认是100分钟）自动关闭自助考勤和自助请假功能，结束之后不再接收学生提交的考勤和请假请求。

9. 教师可以在上课中，上课期间，上课之后查看最近一次上课的出勤率，请假人数，以及缺勤人员的信息。

10. 老师可以使用管理功能维护自己课程内的学生姓名和学号等信息。

11. 教师在课下集中认定所授课程中，学生提交的假条，并将相应的学生请假状态从[ Submitted]修改成[approve] 或者[Absence]。后补的假条可以冲抵学生某次的早退，迟到，缺勤记录。

**2.2 系统的非功能需求**

1. 系统应该保持界面简洁，看起来简单不复杂。

2. 系统应尽量减少用户的操作但是保证能满足用户的需求。

3. 系统应满足具有24小时不间断运行的能力。

4. 系统能够让没有使用经验的人在10分钟内完成所有功能的使用。

5. 系统应保证在考勤高峰期的响应时间不超过一秒。

1. **系统的需求分析**

**3.1 系统的用例图**

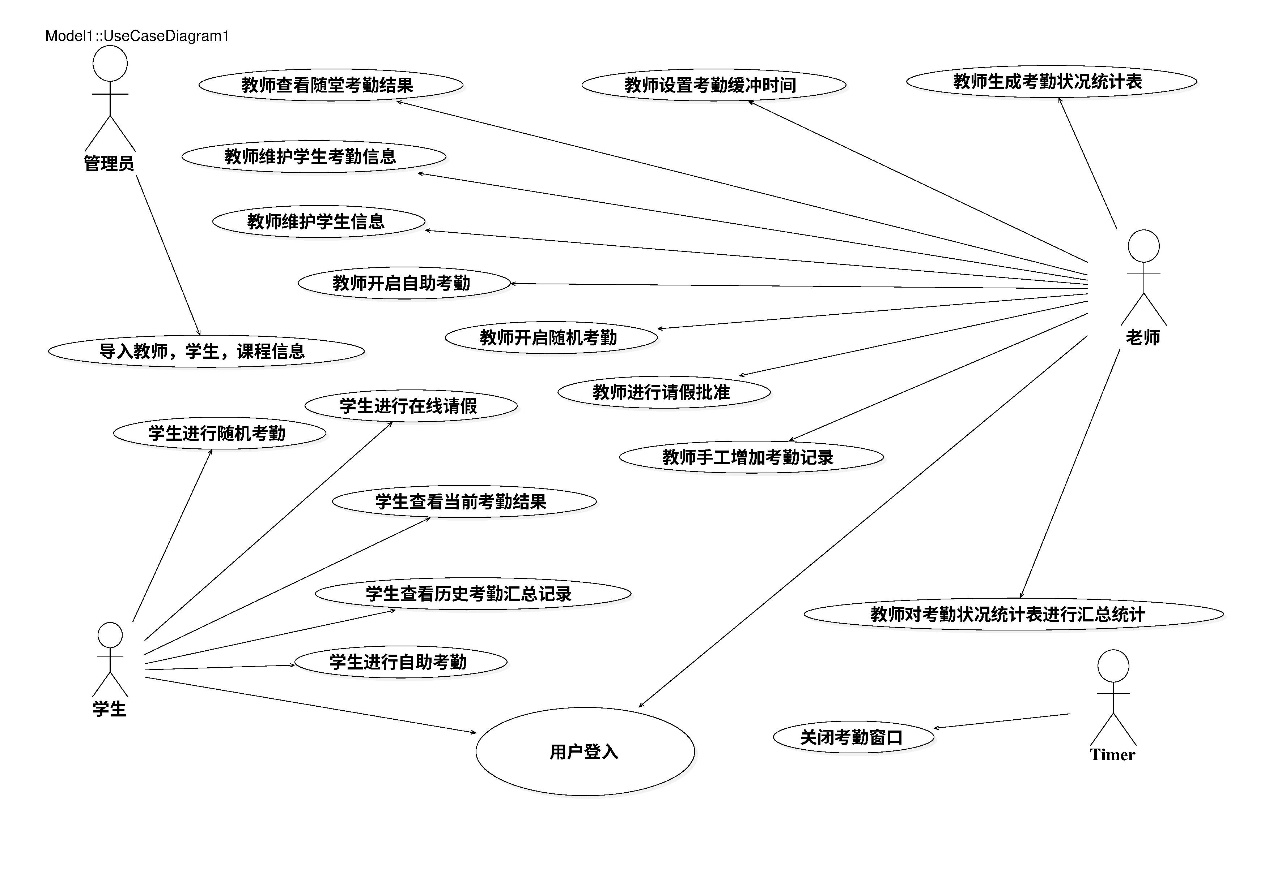


图1 . 用例图

**3.2 用例描述**

**一. 导入教师，学生，课程信息**

1. 目的: 本用例允许管理员将学生信息，课程信息和教师信息导入系统。

**2. 事件流**

**2.1. 基本流程**

当管理员需要导入选课数据时，进入系统后台，打开数据导入功能，本用例开始执行。

1. 管理员选择需要导入的文件。
2. 系统打开文件。
3. 系统检查文件格式，每个单元格的有效性，以及完整性。
4. 系统对外部数据进行格式化操作。
5. 系统检查数据无误后把数据导入内部。

**2.2，备选流程**

1）若系统无法打开文件，则提示错误信息，让用户重新选择。

2）若系统检测到数据有错，则将错误的原因写入到日志中，并继续检查直到所有的数据都被检查完毕。

3） 系统应该检查即将导入的数据不会重复，若重复后来数据为准。

4) 对于内部重复的数据系统应对其覆盖或者删除。

**3. 特殊要求**

使用教务系统导出的数据作为外部数据。

**4. 前提条件**

以管理员身份进入系统。

**5.后置条件**

用例执行成功后，系统中的学生，教师，选课信息将会被更新，否则系统信息不变。

**二. 用户登入**

1. 目的: 本用例允许用户通过微信服务端直接进入本系统。

**2. 事件流**

**2.1 基本流程**

微信服务器向本系统发出登录请求时，本用例开始执行。

1. 系统根据登录的微信号自动从内部文件中匹配用户信息。
2. 如果是老师则显示该用户则显示教师名下的所有的课头列表,系统提示相应操作的菜单，包括：开启考勤，手动考勤，随机考勤，维护考勤记录，学生信息维护，出勤状况随堂统计。
3. 如果是学生则系统提示相应操作的菜单，包括: 自助考勤，自助请假，考勤记录查看，最近出勤状况查看。

**2.2 备选流程**

若微信号不是系统内登记的用户则提示用户登陆失败。

**3. 前提条件**

无

**4. 后置条件**

无

**三. 开启自动考勤**

1. 目的 : 教师上课前或者上课中开启考勤功能，成功开启后等待学生自助考勤或者自助请假。

**2. 事件流**

**2.1 基本流**

1）教师在上课前或上课中，尝试打开开启考勤功能，打开成功后，系统针对该老师维护一个考勤时间窗口，在此时间范围内允许该课头的学生参与自助考勤或自助请假。

2）系统为该课头生成相应的考勤次序号。

**2.2 备选流**

由于同一个老师只能在一个时间内给一个课头的班级上课，同一个班级也只能在一个时间内听一个老师上课，当系统检测到有逻辑冲突发生时需要分下列情况处理。

1. 两个老师尝试开启考勤的时间在相同上课节次区间内，后来的教师尝试打开时间窗口时，系统会提示失败，并提示开启失败的原因。
2. 两个老师尝试开启考勤的时间在不同的上课节次区间内，后来的教师尝试打开时间窗口时，之前教师的时间窗口自动关闭，然后开启后来的教师的时间窗口。

**3. 前置条件**

Settings.int 内的上课节次区间sec已经设置完成。

**4. 后置条件**

系统为成功获取时间窗口的教师维护新的时间窗口，被关闭的时间窗口自动结束，不再接收任何的考勤，请假请求。

**四．教师查看最近随堂考勤结果**

1. 目的： 本用例允许教师在开启自动考勤或者完成考勤后查看最近一次的考勤结果。

**2.事件流**

**2.1 基本流**

考勤窗口正在进行或者下课后时，教师打开查看最近考勤状态功能本用例开始执行。

1. 系统根据seq表查找该老师最近一次的考勤状况得到课程号和考勤次序号。
2. 系统根据教师ID和课程号找到老师生成的考勤汇总表并查找满足考勤次序号的考勤结果，若不存在考勤汇总表则找到该次的考勤细节表并统计在此表学生的考勤状况。
3. 系统根据计算结果显示课头下所有学生当前的考勤状态。

**2.2 备选流程**

若该课头不存在任何一次考勤记录则提示失败信息。

**3. 前置条件**

课头此时存在考勤记录。

**4 后置条件**

无

**五．抽点考勤**

1.目的 ： 本用例允许教师通过系统对学生以随机抽点的方式进行考勤。

**2. 事件流.**

**2.1 基本流**

当教师在上课期间开启自动考勤后想再次了解当前的大致出勤情况打开抽点考勤功能本用例开始执行。

1. 系统根据当前开启的考勤窗口获取当前被考勤的课头名称。
2. 教师选择抽点学生的百分比之后系统随机生成满足该数量的学生学号。
3. 系统把这些学生的考勤记录记为缺勤并追加到当前的考勤细节表中，并且提示被抽点的学生进行随机考勤。

**2.2 备选流程**

1）若教师没有开启自动考勤窗口则提示失败信息。

2）若老师输入的数量大于该课头学生的总数量或者小于0提示错误信息并让用户重新输入。

3）若老师输入的是浮点数则对结果进行取整。

**3. 前置条件**

1)当前存在该老师对该课头自动考勤的时间窗口。

2）课头信息完备。

**4. 后置条件**

老师开启随机考勤功能成功之后向考勤细节表追加被抽点的学生记录，若开启失败系统内部信息不变。

**六． 教师维护学生信息**

1.目的：本用例允许教师对自己课程内的学生姓名，学生学号，微信号，个人特征路径进行维护。

**2.事件流**

**2.1 基本流**

教师需要修改课头下学生信息时打开维护学生信息功能本用例开始执行。

1. 教师输入需要修改学生信息的学号，若学生信息不存在则提示错误。
2. 系统根据输入的学号显示当前学生的信息并提示老师是否修改。
3. 教师根据提示修改学生信息并提交给系统。
4. 系统根据修改数据的类型判断提交的信息是否合法，若不合法提示错误信息，否则写入文件并覆盖学生之前的信息。

**2.2 备选流程**

1）若教师输入的信息与当前存在的信息重复系统则提示错误信息并禁止覆盖。

2）若修改的学生学号不在该老师的课程内系统提示无权修改并舍弃老师提交的内容。

**3. 前提条件**

1) 课头信息完备，学生信息存在。

2）用户以教师的身份登录。

**4. 后置条件**

若老师提交经过系统的检测成功则覆盖之前的数据，否则系统提示错误信息并禁止修改。

**七．教师生成考勤状况统计表**

1. 目的： 本用例允许老师对该自己发起的历史发生考勤记录进行统计并汇总生成考勤汇总表。

**2. 事件流**

**2.1 基本流**

当老师完成某一次考勤想生成该次考勤的汇总记录，打开生成考勤状况统计表功能本用例开始执行。

1. 教师输入需要统计的课头名称和考勤次序号，系统查找内部文件是否有满足输入的考勤细节文件，若没有提示错误信息。
2. 系统根据细节表的考勤记录进行计算若发现有学生存在请假情况反馈给老师。
3. 老师完成请假认定之后追加此次考勤记录到老师该课头名下的考勤汇总表。

**2.2 备选流程**

1）若该老师是第一次进行统计则生成一个该课头的汇总表文件。

2）若该老师重复统计同一次的考勤记录那么后面的计算结果会覆盖掉之前的记录。

3）若教师在上课期间想对考勤结果进行统计系统将提示错误信息并禁止执行。

**3. 前置条件**

1) 该教师至少完成一次自助考勤。

**4． 后置条件**

若系统统计成功（课头所有的学生都有确定的考勤结果）则追加记录到汇总表文件，否则系统信息不变。

**八． 教师进行请假批准**

1.目的 ： 本用例允许老师在课下集中认定所授课程中，学生提交的假条并修改学生的请假状态从提交到缺勤或者批准。

**2.事件流**

**2.1 基本流**

教师在统计课程考勤记录时遇到学生提交的未认定的假条时本用例开始执行。

1. 系统向教师显示假条的提交者，请假内容以及提交时间。
2. 教师根据假条的内容进行判断后对假条进行认定向系统输入认定结果(只能是批准或缺勤)。
3. 系统判断结果是否合法，向教师显示认定结果信息，之后在考勤细节表中修改此学生的请假状态并记入汇总表。

2.2 可选流程

1 ) 教师对于该学生的假条不予理会，系统默认其未批准此假条学生考勤状态设置为缺勤。

2）教师输入的结果不合法，系统显示错误信息并把该假条的结果置为缺勤。

3）老师在开启考勤窗口时执行本用例时系统将提示失败信息。

3．前置条件

1）学生至少提交一次假条。

2）系统在进行统计考勤信息。

4. 后置条件

若老师批准学生的假条就把该学生的考勤状态设置为考勤已批准，否则设置该学生的考勤状态为缺勤。

**九． 教师对考勤状况统计表进行汇总统计**

1. 目的：本用例允许教师在学期末对本学期所有课头下的汇总表进行统计并对每个学生的出勤状态和出勤率进行显示。

**2. 事件流**

**2.1 基本流**

当老师打开考勤状态汇总统计功能时本用例开始执行。

1. 系统获取老师课程内所有的课头列表的信息。
2. 系统挨个获取每个课头列表的考勤汇总表信息，并对每个课头中学生的考勤状态进行计算。
3. 计算完毕后按课头列表顺序显示此课头下每个学生的考勤信息，包括： 出勤数，缺勤数，迟到数，请假数，早退数，之后显示该课头汇总信息的平均出勤率。

**2.2 备选流程**

1）老师没有完成一次考勤时执行本用例系统将提示错误信息。

2）若之前存在老师没有统计的考勤细节表系统将会显示错误信息并显示考勤次序号和课程号。

**3. 前置条件**

1）用户登录的身份为教师

2）当前存在至少一次完整的考勤细节表。

**4）后置条件**

系统自动对所有的考勤汇总表进行统计，显示，并对没有汇总表的考勤细节表生成汇总表。

**十． 教师手工增加考勤信息**

1. 目的：系统支持教师在考勤窗口关闭后为某次课程手工增加考勤记录。

**2. 事件流**

**2.1 基本流**

当教师需要手工增加考勤记录时选择系统的手工考勤本用例开始执行。

1. 教师向系统输入需要手工考勤的课头列表。
2. 系统读取课头列表信息并初始化每一个学生的考勤记录存入一个列表中，之后向教师显示当前学生的考勤信息并提示老师输入对应的考勤状态。
3. 教师根据考勤实际情况输入该学生的考勤状态并提交给系统。
4. 系统检查输入信息是否合法并反馈给教师提交结果，若不通过显示错误信息把当前学生的考勤状态置为缺勤。
5. 依次执行学生考勤信息的录入直到完成所有学生的考勤记录，最后生成考勤细节表并在考勤记录表中追加此课头的考勤信息。

**2.2 备选流程**

若当前教师存在考勤窗口系统提示发起手动考勤失败。

**3. 前置条件**

1) 当前教师没有开启考勤窗口。

2）当前教师的课头列表信息完备。

**4．后置条件**

本用例执行成功后生成此课头的考勤细节表并增加此课头的考勤次序数。

**十一. 教师设置考勤缓冲时间**

1. 目的 : 本用例允许教师给学生设置考勤信息上传时间，超过此时间上传信息的学生被记为迟到。
2. **事件流**

**2.1 基本流**

教师希望调整学生的信息上传时间时选择系统的设置考勤缓冲时间功能，本用例开始执行。

1. 系统提示教师输入1-10分钟之内的分钟数（必须是1-10之间的整数）。
2. 教师输入数据并提交给系统。
3. 系统对数据的有效性进行检查，若不合法允许用户重新输入，否则记入考勤规则表。

**2.2 备选流程**

无

1. **前置条件**

1）以教师的身份登入。

2）存在完备的课头列表。

4 **后置条件**

系统在考勤规则表中设置此记录，在教师下次发起考勤时启用此规则。

**十二. 学生进行随机考勤**

1.目的 ： 本用例允许学生通过系统对教师发起的随机抽点进行考勤。

**2. 事件流.**

**2.1 基本流**

当学生被告知随机考勤发起后选择系统的进行随机考勤功能本用例开始执行。

1. 系统向被抽点学生显示抽点的课头列表信息以及发起者。
2. 学生向系统提交考勤证据路径。
3. 系统对考勤证据进行计算写入到考勤细节表并反馈考勤结果。

**2.2 备选流程**

1）若教师没有开启随机考勤窗口则提示失败信息。

2）若学生的考勤证据路径被系统确定为失败则减少一次此学生上传信息的次数，并提示失败信息和剩余上传次数。

3）若学生用完了上传次数则无法提交考勤信息。

4）若学生已经完成了考勤再次向系统提交考勤证据路径，系统将忽略他的考勤证据路径并显示对应的错误信息。

**3. 前置条件**

1)当前存在该老师对该课头列表随机考勤的时间窗口

2）学生信息存在于课头列表之中。

**4. 后置条件**

学生开启进行随机考勤功能成功之后向考勤细节表修改此次抽点的学生记录为正常，否则记为缺勤。

**十三. 学生进行在线请假**

1. 目的：本用例允许学生在线向发起自助考勤的课头列表进行请假。
2. **事件流**

**2.1 基本流**

当学生选择系统的在线请假功能时本用例开始执行。

1. 系统根据学生的个人信息判断当前是否存在满足要求的课头列表的考勤窗口。
2. 若存在则提示学生输入请假证据。
3. 学生向系统提交请假证明系统记录上传时间，向考勤细节表追加请假信息并修改此学生的考勤状态为请假已提交。
   1. **备选流程**
4. 当前没有满足要求的考勤窗口或者说没有任何一个老师发起与他有关的考勤窗口则提示错误信息并忽略请假要求。
5. 若当前此学生已经提交过一次假条则提示已存在请假信息忽略后来的请假要求。

**3. 前置条件**

1. 当前存在与此学生有关的考勤窗口。
2. 学生没有在当前考勤窗口进行过在线请假。
3. **后置条件**

若学生请假成功则修改考勤记录为已提交，否则系统信息不变。

**十四. 学生查看当前考勤结果**

1. 目的： 本用例允许学生在响应自动考勤后查看实时的考勤结果。

2.事件流

**2.1 基本流**

学生完成自助考勤后打开查看当前考勤状态功能本用例开始执行。

1. 系统根据学生查看考勤状态的时间从Seq表中获取与该学生有关的最近一次的教师号，课程号和考勤次序号。
2. 系统根据教师号和课程号查询汇总表中考勤次序号对应的考勤信息。
3. 若该次考勤不存在考勤汇总表则查询对应的考勤细节表获取考勤信息。
4. 系统从考勤信息中获取该学生的考勤结果进行计算。
5. 系统根据计算结果显示学生当前的考勤状态。

**2.2 备选流程**

若该课头不存在任何一次考勤记录则提示失败信息。

若学生没有完成考勤认证则显示考勤状态为缺勤。

**3. 前置条件**

课头此时存在考勤记录。

**4 后置条件**

无

**十五. 学生查看历史考勤汇总信息**

1. 目的：本用例允许学生查看所有课程下该学生的出勤状态和出勤率。

2. 事件流

2.1 基本流

当学生打开历史考勤状态汇总统计功能时本用例开始执行。

1) 系统获取学生课程内所有的课头列表的信息。

2) 系统挨个获取每个课头列表的考勤汇总表信息，并对每个课头中考勤状态进行计算。

3) 计算完毕后按课头列表顺序显示课程下每一次考勤该学生的考勤结果，最后显示课程下该学生的出勤率。

2.2 备选流程

1）学生没有完成一次考勤时执行本用例系统将提示错误信息。

2）若之前存在老师没有统计的考勤细节表系统，则向学生显示此教师没有对考勤记录进行统计，无法显示。

3. 前置条件

1）用户登录的身份为学生

2）当前存在至少一次完整的考勤细节表。

4）后置条件

无

**十六. 学生进行自助考勤**

1.目的 ： 本用例允许学生通过系统对教师发起的自助考勤进行考勤。

**2. 事件流.**

**2.1 基本流**

当学生被告知自助考勤发起后选择系统的进行自助考勤功能本用例开始执行。

1） 系统向课头列表内所有被考勤的学生显示考勤的课头列表信息以及发起者。

2） 学生向系统提交考勤证据路径。

3） 系统记录信息上传时间并对证据的有效性进行验证，若验证成功写入到考勤细节表并反馈考勤结果，否则显示错误信息并提示剩余考勤次数。

**2.2 备选流程**

1）若教师没有开启自助考勤窗口则提示失败信息。

2）若学生的考勤证据路径被系统确定为失败则减少一次此学生上传信息的次数，并提示失败信息和剩余上传次数。

3）若学生用完了上传次数则无法提交考勤信息。

4）若学生已经完成了考勤再次向系统提交考勤证据路径，系统将忽略他的考勤证据路径并显示对应的错误信息。

**3. 前置条件**

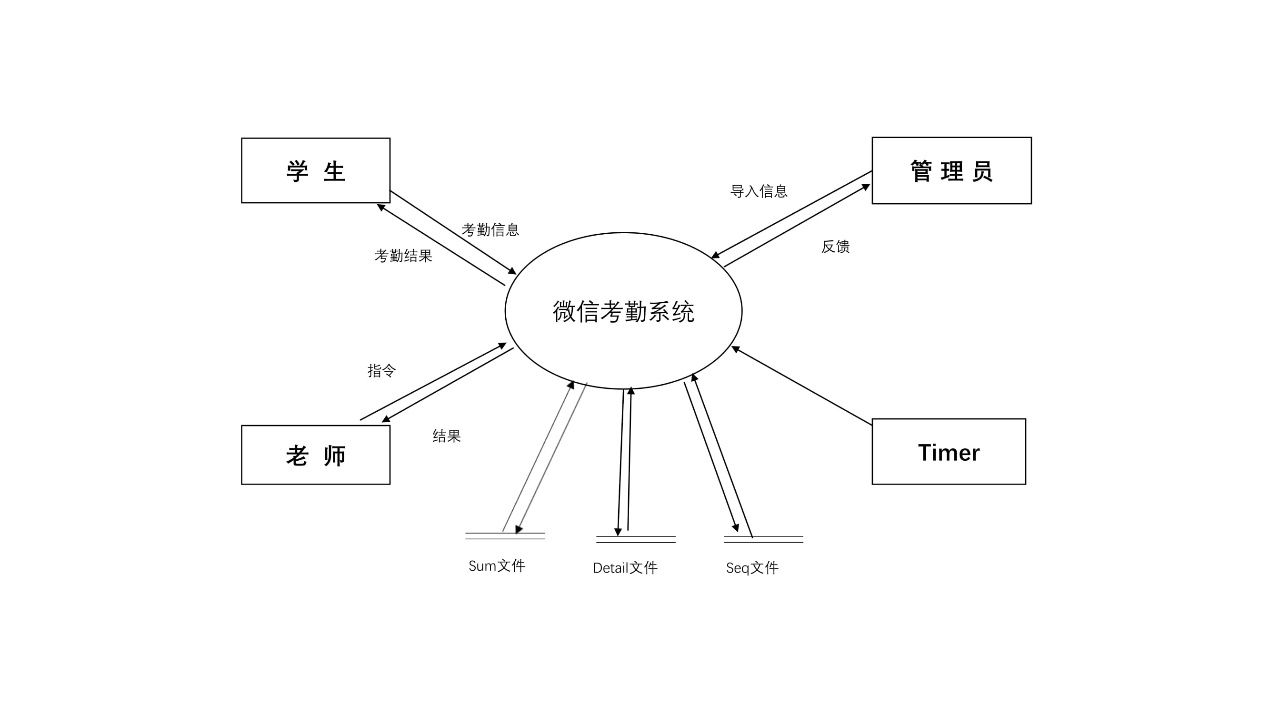
1)当前存在该老师对该课头列表自助考勤窗口

2）学生信息存在于课头列表之中。

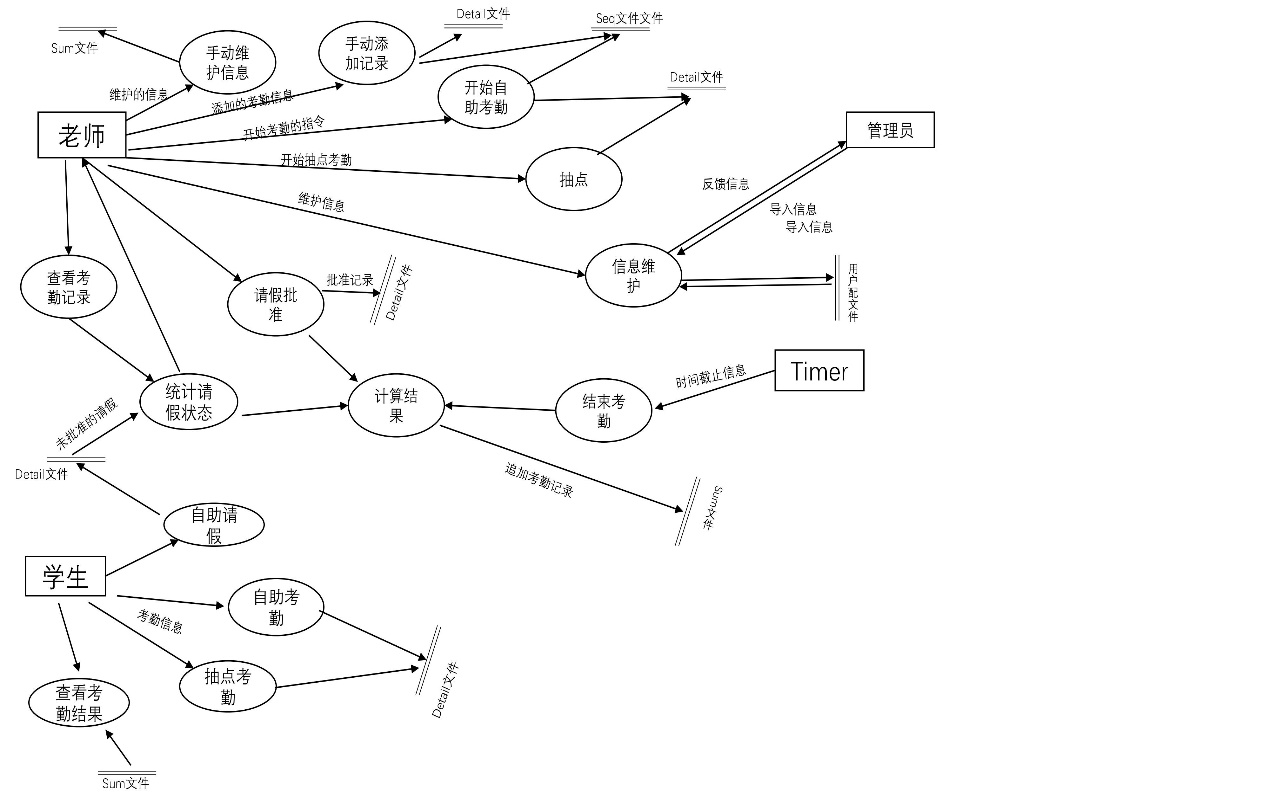
**4. 后置条件**

学生开启自助考勤功能成功之后向考勤细节表修改此次自助考勤的学生记录为正常，否则记为缺勤。

**3.3 系统的数据流图**



图二. 顶层数据流图



图三. 一层数据流图

**四．概要设计**

**4.1 系统架构设计**

系统的架构可分为四个层次。

1. 数据底层

提供对原始文件的读写和追加服务，并对外提供一个接口函数，负责对数据的基础操作，不负责数据的逻辑操作。该层有三个子模块分别为读，写和追加，分别对应文件读写的三种形式。每个模块只负责完整的把数据读取或者写入到文本而不管文本信息是否合法，外部通过该接口调用内部服务。

1. 数据逻辑操作层

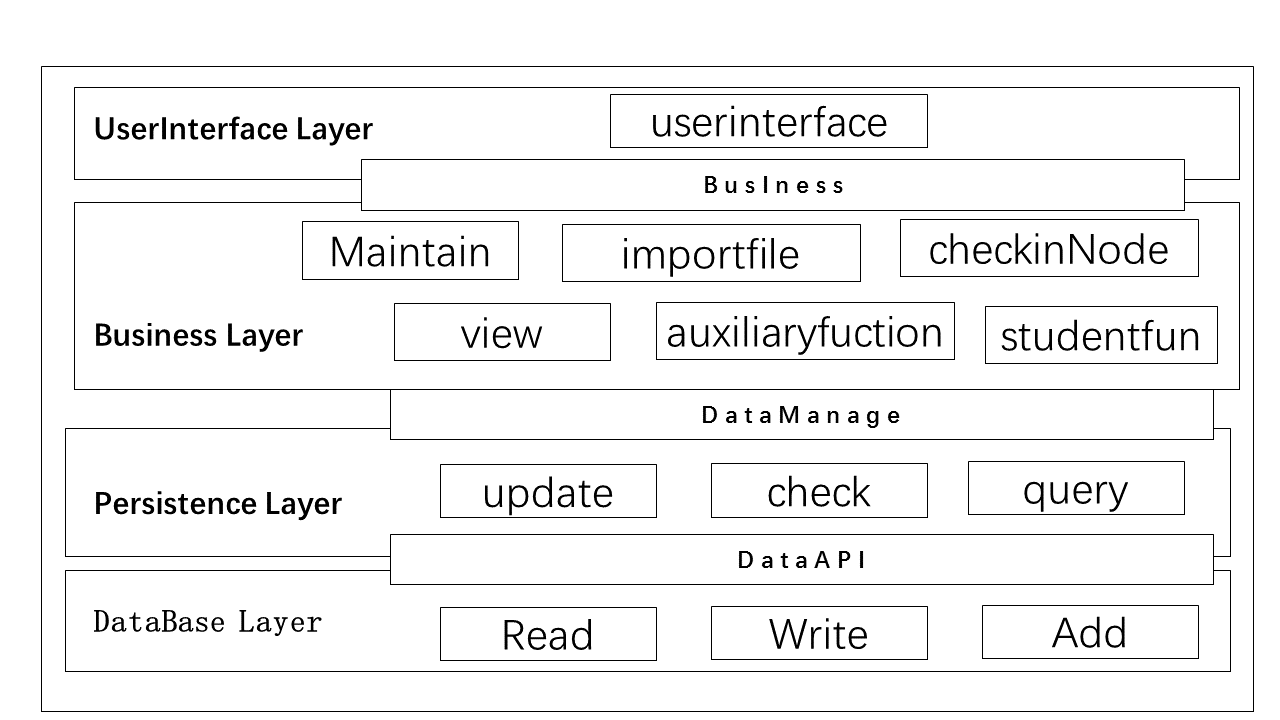
该层负责提供对数据的搜索，核查，更新并向外提供一个接口。该层也具有三个子模块。搜索数据模块接收搜索的地址和搜索的关键词返回满足关键词的信息。执行搜索信息时不依赖于具体的参数，而是返回满足要求的键值，由调用者自己筛选，调用者提供的可选参数越多返回的信息越精确。更新模块是以一次更新为操作对象，而不是以写和追加。核查模块提供对数据的核对，去重。三种服务被封装向上层提供对数据的逻辑操作接口。

1. 业务层

业务层向上层提供业务服务和业务逻辑服务。按功能分为以下模块，学生服务模块，教师服务模块，考勤结点模块，辅助以及计算模块。上层的用户界面层提供输入参数调用本层接口，本层调用下层的服务接口对信息进行查询或者核对满足要求后提供对应的服务，并更新内部文件。

1. 用户界面层

该层向系统提供输入数据，调用下层的业务层提供功能支持，本层不负责对数据的验证和查询而是传递参数调用下层的接口，并显示返回的信息。



图四. 系统的架构图

**4.2 总体实现方案**

**读取文件模块设计描述：**通过对读入数据方式的抽象发现任何一次对文件信息的有效获取都需要至少两个参数，一是文件信息的有效地址，二是对信息的过滤原则。通过文件地址读取文件所有的信息以过滤原则对信息进行筛选得到满足要求的数据集合。备选流程：若关键词为空时代表此时对文件信息的筛选没有限制即返回文件的所有信息，若两者都为空即返回一个空的集合代表没有符合要求的读取。

**写和追加文件模块设计描述：**本系统的基本文件类型是csv文件，csv文件的读取规则比较依赖键和值，所以以字典的方式进行读写是非常合适的。追加文件的用处就是加快文件的IO访问时间，有些文件是不需要判断是否重复的，比如老师抽点时批量增加学生记录，单纯的写入需要先读取文件再写入效率很低而加入追加模式可以快速的完成IO操作。但是由于写入文件后下一次追加时追加的信息可能就不会满足文件的表头规则，即值与键映射错误。所以写入文件时要对写入的顺序进行排序后写入，而对于追加文件可能会出现对空文件的追加，所以就要考虑文件为空时自动添加表头的功能不然写入文件时就会缺少表头。

**查询信息模块设计描述:** 该模块负责对数据的查询逻辑操作。对底层的读取服务进行了抽象扩大了使用范围并对读取服务封装提供了查询服务。该模块提供三种功能，查询目标信息，查询目标键值，查询符合要求的文件名集合，对上层提供数据查询服务。

1. **查询目标信息**

给定文件的地址，以及查询的关键字就能得到所有满足规则的数据集合。若关键字为空则返回文件的所有信息，若文件名为空但是含有参数，则返回项目根目录之内所有文件中满足关键字的信息，若两者都为空则返回一个空集。参数可用统配符进行缺省查询，‘\*’则返回所有满足的信息，若为‘？’则直接返回满足的第一个信息。可利用管道进行连续查询，因为返回的是字典，而查询的关键词也是字典，所以可以连续查询。以前一个的结果作为输入传给后一个作为输出，比如查询某老师的所有课程先查询教师信息作为搜索原则更换目标文件为courseInfo.csv查询课程信息。也可分开查询，区别是后者可以指定搜索规则结果具体而前者搜索结果太笼统。

1. **查询目标键值**

给定一个文件名即可得到该文件的表头。该函数有衍生功能即判断文件是否存在，若存在是否是标准的csv文件。若符合要求返回目标的表头集合，否则返回空集。

1. **查询文件名**

给定一个调用者定义的函数作为参数即可返回根目录下所有满足函数规则的文件名集合，用于对文件的查询功能。

**核对模块设计描述：**该模块对数据的检查和去重以及删除功能。

1. **格式检查**

给定外部数据集合和标准格式对外部数据进行对比验证并返回数据的错误信息。若标准格式为缺省参数则以数据单元格的完整性，数据元素的键值是否相同，以及数据是否为有效参数类型为标准格式进行检查。

1. **数据去重**

给定内部数据和外部数据，主键的键值，和操作参数返回满足要求的数据格式，内部数据为缺省参数时只对外部数据进行操作。外部数据中的原子字典以参数主键为键的值相同则认为是重复内容进行删除，然后判断内部数据是否也存在重复内容进行删除之后，判断外部是否与内部数据重复，若重复则覆盖内部数据，否则追加到内部数据。

当操作参数为删除时删除内部数据中所有以参数主键为键对应值与外部数据以参数主键为键对应值系统的项。

1. **更新模块设计描述：**该模块提供对数据的更新,需要导入地址，外部数据，主键列表以及操作参数。当操作参数为写入的时自动完成对数据的去重和覆盖，若主键为空则覆盖掉完全相同的项否则覆盖指定键值相同的项。当操作参数为删除时删除内部数据与外部数据主键对应系统的项，当操作参数为追加时自动追加数据到文件末尾。

**业务逻辑模块设计描述：**系统对业务的处理直接表现在界面层的显示信息上，间接体现在对文本的记录和信息的统计，所以对数据处理方式的好坏间接影响着系统的效率。

**业务操作的依赖：对文件的读写权限**

业务按使用者可分为学生，老师，管理员，按对数据的操作方式可分为主观写入，直接写入，间接写入，主观读取，受限读取。对于学生的业务就是间接写入，受限读取。学生的业务函数不能对系统文件直接读写，学生提交请假或者考勤信息时必须要在考勤窗口存在时间内，学生查看记录时也只能从考勤汇总表中查看本人信息。所以业务接口层对于学生用户提交的请求要经过层层验证，学生只是提供证据路径，完整信息的创建和写入是系统代替学生完成的。

老师业务是主观写入，主观读取。老师能选择读取文件，老师能对已经存在的数据进行读写并生成汇总文件，手动生成自己课头列表的考勤细节表并修改自己课头列表下学生的信息但不能对其他老师创建的文件进行读写，也不能对记录考勤信息的细节表进行读写，原因有二：

1. 该文件是系统自动创建的老师无修改权。
2. 该文件相对于汇总表文件具有更大的客观性，可当做上课考勤证明出现，任意修改会污染数据。

管理员业务是直接写入。管理员负责向内部数据进行直接导入而无法主观的修改每个数据的值。

**业务的核心结构：考勤队列**

队列中的每一个个正在上课的课头列表都是一个动态的考勤对象。该对象含有本节课的所有信息包括课头列表的键值，是否开始考勤，开始时间，结束时间，当前随机窗口的学生信息，老师自定义的缓冲时间，也含有对该课头列表的操作方法，包括向课头列表申请考勤，申请再次随机抽点。一个考勤对象若没进入考勤队列就是静态的，此时可进行手动考勤，统计信息等操作。动态对象和静态对象最大的区别就是处于动态对象中的用户对文件的读写权限。

**考勤队列的逻辑规则：空闲让进，忙则等待，有限等待，让权等待。**

前提假设 : 对象信息都是存在且有效的

**大前提**

若对象存在且时间有效的那么该对象就是不可考勤的，否则就是可考勤的。(有效指仍然在上课的时间内，存在指在考勤队列中）。

**小前提**

若一节课不能开启考勤那么该课下的老师或者学生至少有一个不能考勤。

得到以下四点结论：

空闲让进: 若一个考勤队列是空的就允许老师对其课头列表下的班级发起考勤。

忙则等待：若教师希望对其课头列表下的班级发起考勤，但是该班级或者该老师本身都处于考勤队列中一个未完成的考勤对象中那么就忽略该老师发的考勤请求。

有限等待:任何一个动态考勤对象都是有存在时间限制的，超过限制的考勤对象将被清除。

让权等待：若一个考勤对象已经到了下课时间但是仍然存在于考勤队列中，后面的考勤对象就有权使此考勤对象让出考勤权并退出考勤队列。

**考勤结果认定的原则:高优先级者优先**

一个学生在完成一门课程的考勤时可能出现多种考勤结果，但是我的设计是只记录最严重的情况来形容一个学生的考勤状态。如果一个学生迟到，早退，那么就记录早退。这样设计的好处是简化了操作流程，同时利用优先级可快速计算考勤结果。

一个学生的状态存在五种--正常，迟到，早退，缺勤，请假已提交，请假已批准

明确正常和迟到是必须‘在这’，早退和缺勤就是‘不在这’是两者本质的区别，所以后者的优先级高于前者。

首先正常的优先级最低　谁都可以覆盖它。

迟到即虽然过了自动考勤的有效时间但是完成了考勤认证或者自动考勤时是缺勤但抽点是正常那么就是迟到，后续考勤状态中只有优先级比他高的才能覆盖。

早退就是之前是正常或者迟到但是某次出现缺勤就是早退。

缺勤就是一直是缺勤从来没有在这的证据。

特例：请假已提交和请假已批准　。

这两者优先级比前四种要高，一个学生只能向同一历史细节表插入一次假条若该细节表该学生已经存在请假记录则无法进行请假。

通过下文的算法流程迭代该学生的状态就被定下来了

考勤细节表只是详细的记录每一次的考勤结果，内部没有逻辑性

但是考勤汇总表有逻辑性，它只记录运算后的状态。

首先假设每个学生的状态都是空 0，权限最低

学生的考勤状态　正常　１　迟到　２　早退　３　缺勤　４　请假已提交　５　请假已批准　６

考勤细节表出现的考勤状态的优先级只能为１　２　４　５　６

也就是说早退考勤细节表没法直接判断

初始化时若一个学生的任意一次考勤置为null

设考勤细节表中的数据为外部数据。

算法流程如下:

若当前状态大于外部权限。

若当前为空　状态直接修改为外部状态。

若当前为正常或者迟到　若外部是缺勤　则修改为早退　否则覆盖。

若当前为早退　除了　５，６　其他的都不能修改。

若当前为缺勤　遇见５，６就修改为请假

其他若大于直接覆盖

否则

若当前为缺勤　存在正常或者迟到就修改为迟到

不会存在一个学生的请假既是已提交又含有已批准的状况　因为一个学生只能向同一文件提交一次请假。

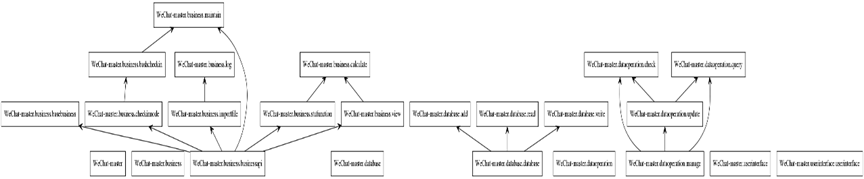
考勤信息的汇总及其统计的批处理计算：

临时信息和汇总信息共用一套处理方法，通过对汇总表和细节表结构进行观察发现任何一次的考勤统计计算结果都依赖于需要计算的列名以及该列所对行标，行标即学号，列名即考勤状态对应的列名。不同的是汇总表有多列数据表示考勤状态，而细节表只有一列表示考勤状态。所以对其抽象指明需要统计的列名即可自动统计该列所对的各学生的考勤信息。

**五，软件详细设计说明**

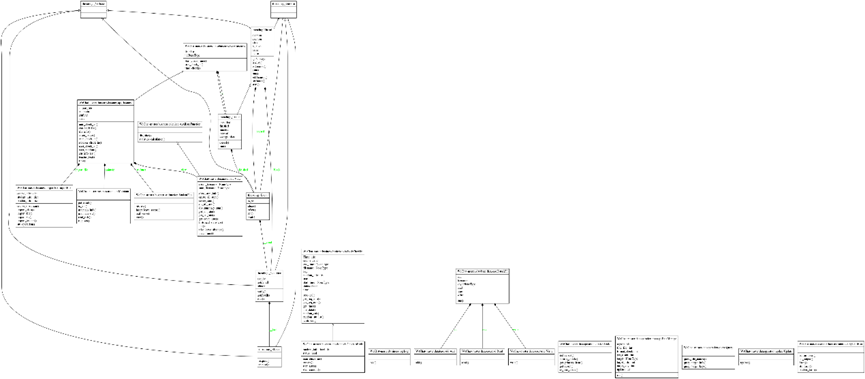
**5.1 系统的结构示意图**

**5.11 系统的包图**

****

图五. 微信考勤系统的包图

**5.12 系统的类图**



图六.微信考勤系统的类图

**5.2 系统功能模块详细设计**

**database 模块**

**模块描述：** 完成微信考勤系统数据的读写,它含有四个类Read,Write,Add,Data,DataAPI。

**功能：**完成对数据的读，写和追加功能。

**交互的模块:** dateoperation

**模块设计:**

该模块的主要文件，文件中包含的主要类及其功能和其他包的交互如下：

**read.py** : Read类是只包含一个静态方法的工具类，它负责对数据的读取。

实现的功能有:

1. 读取指定文件的所有内容。
2. 读取指定文件中符合搜索规则的内容。

接口方法: read(filename, key\_list=None,\_open=’False’)

参数1 filename 即指定的文件名,就是需要读取的文件地址。

参数2 key\_list 是指定搜索的键和值，数据结构类型的为字典。

参数3 \_open 是指定是否开启通配匹配

该方法是原子方法，要么成功，要么失败。

从文中判断是否有数据项包含key\_list，存于一个空列表中，返回创建的列表，若key\_list为空则以列表的形式返回全文内容，方法执行过程中出现运行错误则抛出异常转交控制权给下文。通配方法实现如下： 利用双层循环若内层的值出现\*或者?就跳出循环，并判读具体的符号。若是\*则把匹配的数据加入到列表中继续外层循环，若是?则直接返回匹配的结果。

**write.py:** Write类是只包含一个静态方法的工具类，负责对数据的写入。

实现的功能：

1. 写入给定的外部数据。

接口方法：write(filename, key\_list=None)

参数 1 filename 指定的文件名

参数2 key\_list 外部数据 数据结构类型为列表，元素类型为字典。

该方法是原子方法，要么成功，要么失败，对写入的顺序进行排序。方法执行过程中出现运行错误则抛出异常转交控制权给下文，操作成功返回True。

**add.py:** Add类是只包含一个静态方法的工具类，负责对数据的追加。

接口方法 : add(filename, key\_list=None)

参数 1 filename 指定的文件名

参数2 key\_list 外部数据 数据结构类型为列表，元素类型为字典。

该方法是原子方法，要么成功，要么失败，若文件为空，自动创建表头，对写入顺序进行排序。方法执行过程中出现运行错误则抛出异常转交控制权给下文，操作成功返回True。

**database.py:** DataAPI类是一个模块接口类，负责对整个模块的封装。

接口方法: DataAPI ( filename, \_type, key=None，\_open=’False’).run( )

参数1 filename 给定的文件名

参数2 \_type 给定的操作参数 (必须是 a,w,r)

参数3 key 操作数据

参数4 \_open 是否开启通配匹配

利用该方法调用模块内部服务,函数返回值与内部调用的方法保持一致。

**dateoperation 模块**

**模块描述：**完成对数据操作的封装，含有四个类，Check,Query,Update,DataManage。

**模块功能**: 完成对数据的逻辑操作，实现更新，查询，核对功能。

**交互模块**:business,database

**模块设计:**

该模块的主要文件，文件中包含的主要类及其功能和其他包的交互如下：

**check.py: 类Check是一个工具类，负责对数据的有效性进行校验。**

**函数 format\_check(data, \_format=None)**

给定需要验证的数据集合data数据结构类型为列表，列表元素类型为字典，给定每一个元素的标准格式\_format数据结构类型为字典，返回数据的错误信息。

**函数 get\_primary\_item(keys, line)**

给定一行数据和若干个主键返回所有主键对应值做连接后的字符串。

**函数 del\_repeat(out\_data, in\_data=None, keys=None, \_type='w')**

对out\_data和in\_data进行去重。判断out\_data 数据 是否在in\_data中存在，若不存在直接加入，否则覆盖in\_data中第一次匹配的数据，并删除in\_data之后匹配到的数据。若参数\_type为’dl’则删除in\_data所有与out\_data匹配的值，keys代表判断需要的数据项的键，若keys为缺省值则默认是全部键值。操作成功返回去重后的数据,否则返回None。

**函数 re\_line\_check(line, row, error, \_format=None)**

对一行数据line以给定的正则表达式标准格式\_format进行判断,若错误把行号row和出现的错误追加到error中。操作成功返回True,否则返回False。

**函数 get\_result(dictionary)**

判断错误信息集合dictonary是否存在错误信息，若存在返回True，否则返回False。

**query.py: Query类是一个工具类，调用底层的DataAPI对给定的关键词进行查询和搜索。**

**函数 query\_target\_keys(obj)**

给定目标文件名返回文件名的表头，若文件不存在表头或者文件本身就不存在返回None。

**函数 query\_target\_info(filename="", parameter=None,\_open=’False’)**

给定文件名和搜索关键词返回满足的结果，若关键词为空返回全文内容，若文件名为空但是关键词不为空返回从根目录下所有满足关键词的结果，若两者都为空返回None。

**函数 query\_file\_names(operation)**

给定文件名判断函数operation返回根目录下所有使operation返回值为真的文件名集合，若不存在返回None。

**update.py: Update类是一个工具类，调用底层的DataAPI,Check,Query类对系统数据进行更新。**

**函数 update(filename, \_type, out\_data, keys=None)**

给定文件名，操作参数,外部数据，主键对文件进行更新。若操作参数为w对于给定文件所有主键相同的项进行覆盖对于不同的项进行追加。若操作参数为dl，删除给定文件中所有以主键为匹配规则与out\_data匹配的数据。若操作参数为a，直接追加到文件末尾，不进行去重和覆盖。

**manage.py : DataManage类是一个模块接口类，负责对整个模块进行封装。**

**接口方法：DataManage** **(target=None, args=()).run( )**

参数1 为调用的模块方法，已经集成到了类中成为静态常量，直接调用模块的类变量就能引用模块服务，类的内部自动完成对常量的解析。

参数2为给定的对于调用方法的参数。

**business 模块**

**模块描述: 提供对业务的逻辑处理功能，是整个系统最复杂的模块，含有10个类，basebusiness,bashcheckin,businessapi,calculate,checkinnode,importfile,log,maintain,**

**stufunction,view。**

**模块功能：**

1. **老师发起自动考勤，抽点考勤，手动考勤，维护信息，查看随堂考勤记录，查看历史汇总记录，设置考勤缓冲时间，生成出勤状况统计表。**
2. **管理员信息导入。**
3. **学生在线请假，考勤，查看历史记录，查看最近一次考勤状态。**

**交互模块：dataoperation，userinterface**

**模块设计：**

**该模块的主要文件，文件中包含的主要类及其功能和其他包的交互如下：**

**importfile.py :ImportFile类是一个导入信息类完成各种信息的导入功能，调用了底层的DataManage类和业务层的Log类。**

类变量: teacher\_info , course\_info, student\_info 分别存储各个数据类型的标准正则表达式当做格式检查的依据。

**函数 stu\_operation(out\_file, path) 是一个静态工具方法**

给定外部数据的文件路径和外部学生信息路径完成对外部学生信息的格式化。若格式化成功返回格式化后的数据，否则返回None。

**函数 course\_operation(\*out\_file) 是一个静态工具方法**

给定外部课程数据的路径对数据进行格式化，可以拆分班级区间并去掉无关项生成格式化数据，若操作成功返回格式化的数据，否则返回False。

**函数 def import\_file(source\_file, out\_file, \_format=None, primary\_key=None, operation=None, path='') 是一个静态工具方法**

给定需要信息的内部文件路径，外部文件路径，文件数据的标准格式，文件的去重规则完成对文件的导入，当外部数据出现格式户错误时调用Log类的add方法把错误信息和记录时间写入日志文件。若外部文件需要格式化则需要选择格式化函数，若外部文件存在关联的文件路径需要添加文件路径。导入信息对导入功能进行抽象提高了导入文件处理操作的自由度，调用了底层的DataManage.format\_check方法对数据进行校验，利用DataManage.update方法进行更新。文件的去重规则可由调用者根据需要修改，数据格式化方式可由调用者根据需要修改，提高了复用性。

文件的去重规则如下：

一个文件可能有多个主键，若这些主键重复则认为是需要删除的，例如设置主键StuID相同的数据认为是重复的那么去重规则可用[[‘StuID’]]表示。同样的，也可利用逻辑性自行设置多个规则，例如[['TeacherID'], ['WeChatID']] 表示文件有两个主键分别是TeacherID和WeChatID，一个数据项这两个键对应的值不能与其他的数据项系统。原理如下:该功能利用了DataManage的去重功能，该类的去重方法一次只能指定一个主键，但是可以多次去重所以给定多个主键重复调用即可达到目的。

**函数 import\_stu(self, out)**

对import\_file函数进行封装使其能单独对学生信息单独导入，内部信息去重规则设置为[[‘StuID’],[‘WeChatID’]]。因为一个学生信息能被学号或者微信号唯一确定，所以先去除学号系统的数据，再去除微信号相同的数据剩下的数据就是学号和微信号都不同的学生数据，返回值与调用的import\_file方法返回值相同。

**函数 import\_teacher(self, out)**

对import\_file函数进行封装使其能单独对老师信息进行导入，内部信息去重规则设置为[[‘TeacherID’],[’WeChatID’]]原理与学生去重规则相同，返回值与调用的import\_file方法返回值相同。

**函数 import\_class(self, out)**

对import\_file函数进行封装使其能单独对班级信息进行导入，内部信息设置为[['CourseID', 'ClassName']]，我们先来分析一堂课需要什么主键才能确定。去除和CourseID具有相同的代表性的CourseName，剩下CourseID,ClassID,TeacherID三个键，利用排列组合原理共有7种方法。但是任何单独的一个键都不能确定唯一的一节课去掉三种单独的情况剩下四种。三个键同时存在时存在不同教师对同一班级上同一课程的情况无法覆盖所以去掉，还剩下三种情况。首先CourseID和TeacherID相同时ClassID不同是允许的不能覆盖掉所以舍弃，TeacherID和ClassID不能确定唯一一门课舍弃。即得到课程的主键即[['CourseID', 'ClassName']]。

该方法返回值与import\_file方法返回值相同。

**log.py: Log类是一个记录导入数据出错信息的工具类。**

**方法 add(info)**

在ImportFile类导入信息时调用了DataManage类的format\_check方法，该方法会返回出错信息，把出错信息作为参数直接传给add方法即可完成错误日志的记录。

**maintain.py: Maintain类具有维护学生信息和修改考勤状态以及缓冲时间的读写功能。**

**方法 maintain\_info(key)**

该方法用来维护学生的考勤状态信息，参数key传递了修改者的个人信息以及需要修改的考勤次序号，若该次不存在考勤汇总表直接返回False。用户输入需要修改的学生学号，若学号不存在或者该学生不是老师课程范围之内，不满足直接返回False。然后根据需要选择修改的考勤状态后系统自动完成对汇总表的更新，若用户选择的状态不合法则默认修改结果为缺勤。返回值与调用更新内部文件的DataManage.update方法相同。

**方法 is\_can(key)**

该方法用来验证参数key中的信息是否在学生数据库中具有唯一性，若是返回True，否则返回Fakse。

**方法 get\_result(self, class\_list, data)**

该方法用来接收用户输入的修改信息并把输入作为返回结果传递给maintain\_stu方法。

**方法 maintain\_stu(self, key)**

该方法用来修改学生的个人信息，参数key传递了方法执行的必要信息。若要修改信息的学生不存在或者不是该老师课程下的学生则无法修改返回Fasle。验证完成后依次输入需要修改的信息并保证学号和微信号具有唯一性，修改的班级仍在教师的课程范围之下。完成信息的录入后对信息的录入格式进行检查若合法更新系统文件返回值与调用DataManage.update方法相同，否则返回False。

**方法 rule\_set(key)**

该方法用来设置老师指定学生的有效考勤时间，在此时间内有有效考勤时间，否则即为缺勤或者迟到，有效考勤时间必须在1~10分钟之间。设置完成后调用DataManage.update方法更新set.csv文件，返回值与调用方法相同。

**方法 read\_rule(key)**

该方法用来读取老师指定的有效考勤时间，若教师从来没有设置过则默认是三分钟。

**calculate.py : AuxiliaryFunction类是一个辅助计算类只包含需要计算的数据集合返回计算结果或者显示结果，而不关心具体的计算对象。**

**方法 \_\_calculation(absence)**

该方法是一个私有方法，完成对考勤计算结果的格式化输出。对所有考勤类型的人数进行统计，包括缺勤人数，请假提交人数，请假批准人数，迟到人数，早退人数，正常人数以及该批数据的出勤率。并包含了非正常学生的考勤状态信息，包括姓名和考勤结果类型。

**方法 statistics\_calculation(self, stu\_info\_list,\_type='')**

该方法是AuxiliaryFunction类的核心方法，完成对数据集合的计算。stu\_info\_list

代表需要计算的数据集合，\_type代表需要计算结果的键名，表示以该键作为状态参量进行迭代。该方法的执行过程可分为三步:

1. 初始化状态参量。

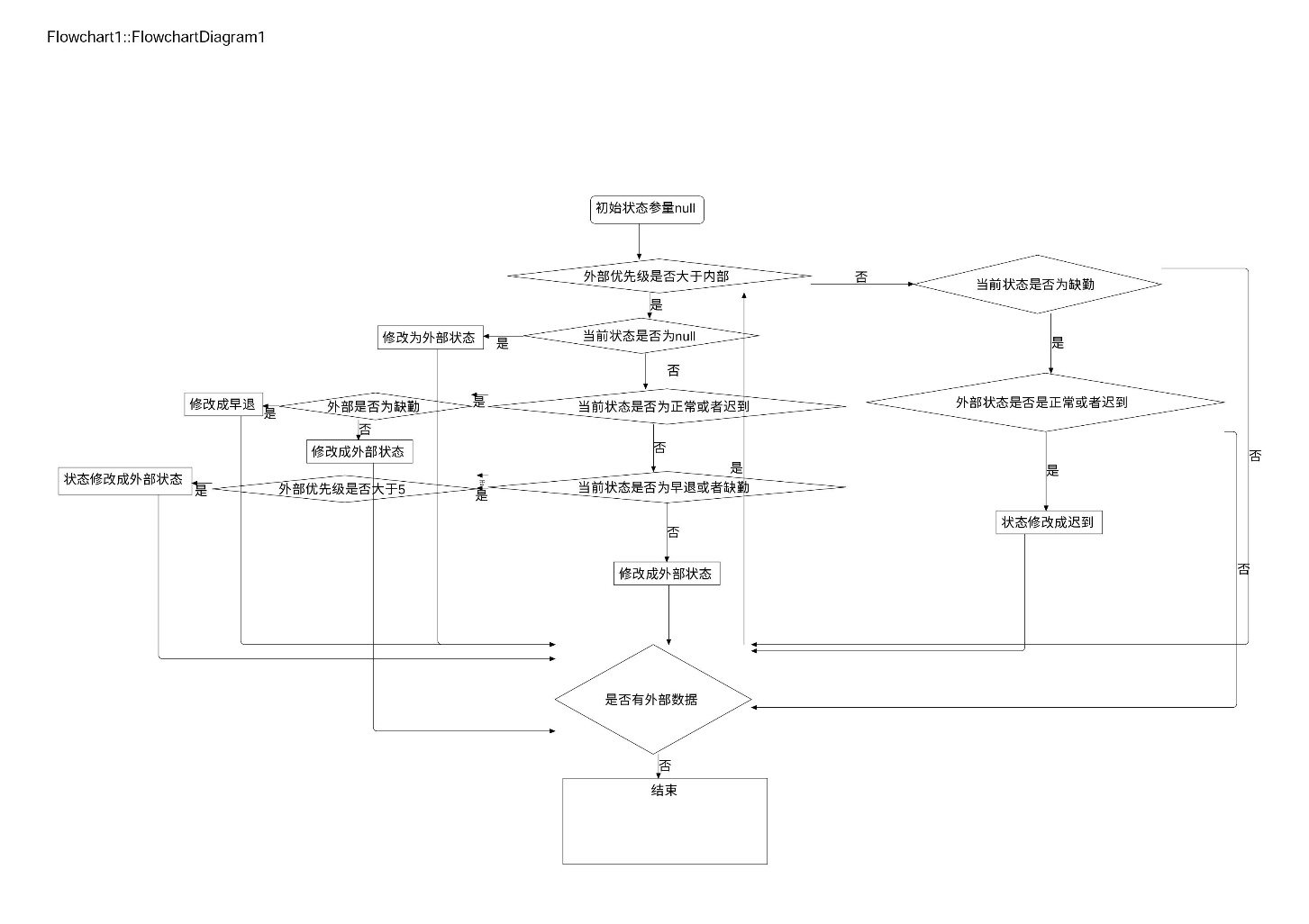
设置考勤状态优先级表

{'null': 0, 'normal': 1, 'Late': 2, 'leaveEarlier': 3, 'Absence': 4, 'Submitted': 5, 'approve': 6}

初始化学生的数据，以学生的学号为键，对应值为信息字典，该字典包括姓名和考勤状态，初始时把考勤状态设置为null(0)即最低。

1. 参量迭代

通过外部数据不断更新内部的考勤状态，更换的条件就是优先级和结果认定方法。流程图如下：



图六 算法流程图

**算法描述如下：**

初始化时若一个学生的任意一次考勤置为null

若当前状态大于外部权限

若当前为空　状态直接修改为外部状态

若当前为正常或者迟到　若外部是缺勤　则修改为早退　否则覆盖

若当前为早退　除了　５，６　其他的都不能修改‘

若当前为缺勤　遇见５，６就修改为请假

其他若大于直接覆盖

否则

若当前为缺勤　存在正常或者迟到就修改为迟到

1. . 格式化结果数据

调用\_\_calculation方法对计算后的数据格式化并输出。

**stufunction.py: StudentFun类是一个学生功能类，包含了查看历史消息，当前考勤状态和提交假条功能。**

**方法 insert\_leave\_record(key)**

该方法用来向考勤细节表提交假条,参数key包含的有请假的必要信息。学生请假的原则是只能向一个考勤窗口申请一次请假。首先验证学生插入假条的考勤细节表的存在性，若不存在则直接返回False。在验证是否是第一次请假，若不是直接返回False，最后调用DataManage.update更新考勤细节表增加假条信息。

**方法 view(filename, stu\_id, teacher\_id, num, counter=None)**

该方法用来统计文件中学生某一次考勤记录，若num不为0则代表进行的是查看第num此记录，对学生考勤结果进行直接显示后退出。若num为0代表的是对历史记录的统计并把参量存入counter中进行统计汇总。该方法可用来分析给定的是细节表文件还是汇总表文件，判断的依据就是键值。若’checkin’+num不存在与文件的表头之中则代表是细节表文件。调用AuxiliaryFunction().statistics\_calculation(stu\_info,\_type)方法对该学生的考勤状态进行计算并显示，若给定的文件是细节表文件\_type为‘checkinResult‘，否则为’checkin’+num，代表不同文件需要计算的键值。counter代表的是统计参量,该方法可用来计算学生的历史考勤状态，作为统计信息的载体。

**方法 real\_view(self, key)**

该方法用来查看学生最近一次的考勤状态。先从课程信息表中查询到关于此学生班级对应的所有课程信息预计对应的教师信息。再查询seq表中满足条件的最后一条信息即为最近一次的考勤信息。查询状态优先查看汇总表文件，当汇总表不存在时查询考勤细节表文件，然后调用view方法查询该学生的考勤状态。

**方法 history(self, key)**

该方法用来查询学生的历史记录，key代表的是学生的个人信息。先从课程信息表中查询到关于此学生班级对应的所有课程信息预计对应的教师信息。再依次以这些参数格式化成文件名，若文件名不存在代表教师尚未对该次考勤进行统计则无法显示考勤信息返回False。设置参量counter作为学生各种考勤状态的统计，并把文件名,counter,学生个人信息作为参数传给view通过迭代更新counter，最后格式化显示counter的信息即为学生的历史考勤信息。

**view.py: View类继承了AuxiliaryFunction类。用来帮助老师查看某次课程的考勤信息，某课程的汇总考勤信息，所有课程下所有学生的汇总考勤记录信息显示，以及查看随堂考勤记录，绘制考勤汇总表，批准请假。**

**方法 historical\_statistics(self)**

该方法用于检测细节表中是否存在学生申请的假条并对其进行认证。利用DataManage.target\_info方法读取细节表所有信息迭代查找是否有考勤状态为Submitted的学生，以该学生的请假信息作为参数若调用take\_leave\_absence()函数进行认证。

**方法 take\_leave\_absence(self, stu\_id)**

该方法用于对请假信息进行认证。首先向教师显示提交时间，请假证明，以及学生学号和姓名，若老师批准则修改考勤细节表中该学生的考勤状态为approve，否则修改成Absence。

**方法 init(self, key)**

该方法用于初始化考勤汇总表，当汇总表第一次创建时，通过参数key查询到所有的学生学号加入到考勤细节表中。

**方法 create\_sum(self, key, info='')**

该方法用来创建考勤汇总表，函数执行的必要信息都存储在参数key中,若需要计算的考勤细节表不存在则返回False。若该教师第一次生成汇总表文件，则调用init()方法进行初始化把课程下所有学生的学号追加到汇总表文件中，后序的调用该方法对同一汇总表统计会覆盖之前的信息。调用historical\_statistics()方法对考勤细节表中的请假信息进行认证并更新考勤细节表，调用父类的statistics\_calculation方法对考勤结果进行统计，最后调用DataManage(DataManage.update, args=(self.sum\_filename, 'dl', old\_data)).run()方法删除原有的汇总表数据，利用DataManage(DataManage.update, args=(self.sum\_filename, 'w', old\_data)).run()更新汇总表数据。

**方法 view\_time(self, key)**

该方法用来查看最近一次的考勤结果。先读取seq表获取最后一次的考勤记录，若不存在则返回False，判断当前汇总表是否有该次的考勤记录若不存在则查找考勤细节表，调用DataManage(DataManage.target\_info, args=(filename,)).run()方法读取文件的数据。调用父类的dis\_play()方法对数据进行临时查看。

**方法 get\_att\_rate(self, key)**

该方法返回汇总表中某一次考勤的出勤率若该次考勤不存在返回-1。调用父类的statistics\_calculation()方法中返回的出勤率。

**方法 get\_all\_rate(self, key)**

该方法返回某一课程的全部出勤率统计结果，若该课程的汇总表不存在则返回False。调用get\_att\_rate()统计每一次的考勤出勤率若出勤率为-1则跳过此次的计算，最后对总出勤率进行输出。

**方法 dis\_summary\_sum(self, key)**该方法用来统计教师所有课程的出勤率。通过查询课程信息表获取所有课程和教师的映射关系，利用映射关系生成汇总表文件名。调用get\_all\_rate()方法依次对每个汇总表的出勤率进行统计，若汇总表文件名不存在则显示错误信息。

**方法 class\_sum\_info(key)**

该方法用来统计每个课程中每个学生的出勤状况的详细出勤信息。开始每个学生学生的信息进行初始化，早退数，缺勤数，迟到数，请假数，早退数为零，对统计汇总表中每一次的考勤记录对应学生的考勤状态数量进行迭代，最后显示汇总数据。

**方法 dis\_all\_sum** **(self, key)**

该方法用来显示所有课程下对应学生的出勤状况出勤统计信息。读取课程信息表找到课程与教师的映射关系枚举所有对应的汇总文件名，舍弃不存在的文件名，对存在的文件调用class\_sum\_info()方法进行统计输出。

**bashcheckin.py : BashCheckin类是一个考勤对象的基类包含了考勤对象的必要方法，包括接收考勤信息，产生随机数量的学生，对时间的把握，对考勤学生数据的初始化，获取考勤次序号，生成当次考勤次序信息，写入考勤次序信息，以及对考勤信息的计算。调用了DataManage接口类和Maintain类**

**构造方法 \_\_init\_\_(self, key)**

用于初始化考勤对象，设置一个存储器记录每个学生的考勤次数，设置一个状态参量记录当前是否开启自助考勤，设置一个缓冲时间参量通过Maintain类的read\_rule()方法读取该老师的缓冲时间设置信息，设置一个当前时间参量记录当前开启的时间，设置一个变量记录上课的结束时间，设置一个时间信息字典记录配置文件的上下课时间，设置一个变量记录当前对象记录的考勤细节表。

**方法 get\_time(self)**

判断当前时间是否属于某个上课区间，若找到把当前时间赋给考勤对象的考试时间，结束时间赋值给考勤对象的结束时间，若找不到显示当前是非上课时间。但是用户确定一定考勤后将会开启并把当前时间设置为考试时间，当前时间加100分钟设置为结束时间，若没有任何可以考勤的时间返回False。

**方法 init\_data(self, stu\_info\_list, \_type)**

对给定的学生信息进行初始化，把考勤类型设置为\_type,，初始化每一个学生的考勤次数为五，考勤结果为缺勤，考勤状态为假。若\_type为收‘man’则由用户自定义选择考勤状态。

**方法 get\_seq\_num(self)**

该方法用于获取当前的考勤次序号。读取seq文件获取最后一次的次序号，若不存在则返回1。

**方法 get\_seq\_info(self)**

该方法用于创建本次的考勤次序信息。包括考勤时间，考勤教师，考勤课程，以及考勤次序号,并返回创建的信息。

**方法 write\_seq(self)**

该方法用于更新seq文件。若seq不存在则自动创建，否则追加到文家末尾。

**方法 random\_stu\_list(self)**

该方法用户生成满足百分比的学生信息。从课程下的班级中找到所有的学生，创建一个空的列表。每次产生一个随机数,判断以随机数为下标的学生是否已经于列表之中，若不存在则添加，直到列表的长度满足百分比的输入终止返回产生的学生信息。

**方法 auto\_cal(self, stu\_info)**

该方法用于计算学生的自助考勤信息。首先判断学生是否已经请假若请假则无法考勤，再判断是否考勤的次数是否在5次之内，若剩余次数为0则不允许考勤，最后判断是否已经完成考勤和考勤证据路径是否有效，若已经完成考勤也无法进行考勤。若考勤证据路径无效则显示考勤验证失败考勤次数减1并在考勤细节表中更新学生的考勤信息。若证据验证成功判断提交信息时间与开始考勤时间的差值是否在考勤缓冲时间之内，若在则记录为正常，否则记录为迟到，用计算结果更新考勤细节表。

**方法 random\_cal(self, stu\_info)**

该方法用于计算学生的随记考勤信息。同自动考勤一样也是要对考勤的次数，是否已经完成，是否已经请假进行验证，验证通过进行对证据路径的验证。若验证成功修改考勤细节表中该学生的考勤状态为正常，若失败考勤次数减1，显示失败信息。

**checkinnode.py :** **CheckInNode类是一个考勤对象类，包含了开始自助考勤，开始抽点考勤，开启手动考勤，接收学生的考勤信息。**

**方法 start\_auto(self)**

该方法用于开启自动考勤。首先判读当前是否已经开启考勤窗口，若存在则无法再次开启，判断当前是否有合适的时间开启考勤，若不存在则返回False。选择该课程下所有的学生信息初始化后加入到考勤细节表中，更新考勤次序表，返回True。

**方法 start\_random(self)**

该方法用于开启抽点考勤。首先判断当前是否开启了自动考勤，若没有则不允许进行抽点考勤，然后创建满足百分比的随机的学生名单更新考勤对象的抽点列表，并追加抽点学生信息初始化后加入到考勤细节表中。

**方法 man\_check\_in(self)**

该方法用于开启手动考勤，给学生选择适合的考勤状态，并生成新的考勤细节表同时更新考勤次序表。

**方法 receive(self, stu\_info)**

该方法接收学生的考勤信息，根据学生提交考勤信息的状态选择父类的auto\_cal()或者random\_cal（）对信息进行计算和写入。

**basebusiness.py: 类BaseBusiness用于创建考勤队列，负责对考勤队列的状态进行更新。**

**构造方法 \_\_init\_\_(self)**

该方法用于创建考勤队列和计时器。

**方法 stop\_check\_in(self)**

该方法用于设置每次计时器的时间。若当前队列中的考勤对象大于2，则计算第二个考勤对象剩余时间，给计时器重新定时并踢出第一个考勤对象。

否则直接踢出第一个考勤对象并销毁计时器。

**方法 time\_check(self)**

该方法负责对考勤队列的检查，每次调用时调用stop\_check\_in()方法对考勤队列进行更新，若考勤队列已经为空则什么也不做返回False。

**方法 start\_check\_time(self)**

该方法用于开启考勤队列的计时器只在考勤对象进入空的考勤队列时调用。

**businessapi.py：Business类是整个业务层的接口类并继承了BaseBusiness类，负责整个业务的逻辑操作和封装。**

**方法 \_\_init\_\_(self)**

该方法负责集成各个业务类。

**函数 stu\_info\_test(key)**

负责对学生身份进行验证，若验证成功该学生的所有信息，否则返回False。

**函数 teacher\_test(key)**

负责对教师身份进行验证，若验证成功返回教师的所有课程及课程下对应的班级以及教师本人的信息。否则返回False。

**方法 insert\_leave(self, key)**

判断学生所在的班级是否存在于考勤队列之中，若存在则接收请假信息并转交给学生申请请假的考勤对象，否则返回False。

**方法 auto\_check\_in(self, key)**

判断学生所在班级是否存在于考勤队列之中，若存在则接收自助考勤信息并转交给学生申请自助考勤的考勤对象，否则返回False。

**方法 random\_check\_in(self, key)**

判断学生所在班级是否存在于考勤队列之中，若存在则接收抽点考勤信息并转交给学生申请抽点考勤的考勤对象，否则返回False。

**方法 tips(self, key, info='')**

该方法用于考勤信息的提醒和判断当前是否可以考勤。设置状态参量count =0,判断学生所在班级是否存在于考勤队列的考勤对象之中，若存在 count += 1，若info不为缺省参数显示当前存在与您有关的自动考勤和发起者，若为缺省，不进行显示。再判断查找到的考勤对象的随机抽点名单是否有该学生，若存在，count += 2，若info不为缺省值显示存在与您有关的抽点考勤，否则不显示。最后返回count。(count作为状态参量返回值0代表不存在与您有关的考勤信息，为1代表只存在自助考勤，为3代表存在抽点考勤)。

**函数 \_\_time\_check(check\_node)**

判断当前时间是否已经大于给定考勤对象的下课时间，若为真返回False，否则返回True。

**方法 start\_check\_in(self, key)**

该方法用于创建考勤对象并开启对其的考勤窗口。首先创建对象的深拷贝，key包含了该教师的所有信息我们只需要对其进行操作即可得到结果。创建参量当做考勤对象的键值，该参量包括教师的个人信息，课程号，课程名称，以及数据结构类型为集合的班级集合。

1 判断开启者是否有效

从考勤队列中依次查找是否存在考勤对象的教师与开启者相同的状况，若存在调用\_\_time\_check()函数对考勤队列中的考勤对象进行验证查看是否下课。若没有下课提示开启者已经开始一门考勤不能再次开启。若已经下课则踢出下课的班级。

1. 判断被开启者是否有效

从考勤队列中所有的考勤对象的班级集合与创建的考勤对象的班级集合是否有交集。若存在交集判断考勤对象是否已经下课，调用\_\_time\_check()函数进行验证，若正在上课则提示无法考勤，否则从考勤队列中踢出该考勤对象。

1. 踢出的原则

若被踢者不是队列首元素则直接踢出，否则调用父类的stop\_check\_in()方法踢出。

调用创建的考勤对象的开始考勤方法，若开启考勤失败则提示错误返回False，若以上操作都有效那么判断当前队列是否为空，若为空调用父类的start\_check\_time（）对考勤队列进行初始化，否则加入到考勤队列的末尾提示考勤成功返回True。

**方法 can\_statistics(self,key)**

该方法用于验证老师是否可以进行某种操作。判断依据是教师是否存在于考勤队列中的考勤对象之中，若不是则返回True,否则返回False。

**方法 man\_check\_in(self, key)**

该方法用于创建考勤对象开启手动考勤。调用can\_statistics（）查询是否运行，若为真则开启手动考勤，否则返回False。

**方法 dis\_info(self,key)**

该方法用于提示教师当前的考勤状态。从考勤队列中查找是否存在与之有关的考勤对象，若存在则显示此考勤对象的考勤信息（即是对哪个课程开启了自助考勤，对哪个课程开启了抽点考勤）返回True。若不存在与之有关的考勤对象，则提示与之有关的课头列表，包括课程号，课程名称，与课程有关的班级集合返回False。

**userinterface 模块**

**模块描述：** 完成微信考勤系统与用户的交互并接受输入信息提交给业务层。

**功能：**完成与用户的交互和数据的接收。

**交互的模块:** business

**模块设计:**

该模块的主要文件，文件中包含的主要类及其功能和其他包的交互如下：

**userinterface.py : SystemRun类是界面类完成显示各种菜单的显示和数据的接收。**

**方法 form(self)**

该方法用于用户的登入和身份的验证。创建business对象，调用business.teacher\_test（）或者business.stu\_info\_test（）对微信号进行验证。若输入的微信号不存在或者输入的选项不合法则提示错误信息。否则把调用方法的返回值和business对象传递给登入用户的菜单页面方法。

**方法 stu\_form(key, line)**

该方法用于显示学生用户的菜单页面包括在线请假，进行考勤，进行抽点考勤，查看当前考勤结果，查看历史考勤结果。调用line方法的tips判断当前是否存在与登入用户有关的考勤，若存在允许用户进行请假，考勤否则不允许执行请假和考勤。

**方法 teacher\_form(self, key, line)**

该方法用于显示教师用户的菜单页面包括开启自助考勤，开启抽点考勤，开启手动考勤，设置考勤缓冲时间，出勤情况随堂统计，生成出勤状况统计表，出勤成绩汇总输出，学生信息维护，考勤信息维护。调用line的dis\_info（）方法显示与教师有关的信息，若当前教师已经开启考勤窗口则不允许进行生成出勤状况统计表。选择出勤状况汇总输出时会调用方法att\_output（）显示汇总输出菜单。

**方法 admin\_form(line)**

该方法用于显示管理员用户的菜单页面包括教师信息导入，学生信息导入，课程信息导入。

**方法 att\_output(key, line)**

该方法用于考勤成绩的汇总输出菜单显示，包括查看某次课程的出勤率，查看某次课程汇总出勤信息，显示用户总考勤概要信息，显示用户总考勤详细信息。