

Class Management System

需求分析说明书

版 本： V7

发行日期： 2017/12/18

作 者：需求分析标准小组

修改历史

[illegible]

目 录

1. 愿景.....	1
1.1 应用背景.....	1
1.2 解决问题.....	1
1.3 系统主要功能.....	1
2. 用例.....	2
2.1 用户目标分析.....	2
2.2 用例图.....	3
2.2.1 Teacher 用例图	3
2.2.2 Student 用例图	6
2.3 用例详述.....	9
2.3.1 教师用例详述.....	9
2.3.2 学生用例详述.....	11
3. 业务流程分析.....	12
3.1 业务流程分析.....	12
3.1.1 课前活动图.....	12
3.1.2 课堂活动图.....	13
3.1.3 课后活动图.....	14
3.2 对象状态机分析.....	15
3.2.1 学生状态机图.....	15
3.2.2 小组状态机图.....	16
4. 非功能性需求.....	17
4.1 性能需求.....	17
4.2 可靠性需求.....	17
4.3 可用性需求.....	17
4.4 安全性需求.....	17
4.4.1 身份认证.....	17
4.4.2 权限控制.....	17
4.4.3 数据库安全.....	18
4.5 外部接口.....	18
4.6 可支持性需求.....	18
4.6.1 兼容性.....	18
4.6.2 分辨率.....	18

1. 愿景

1.1 应用背景

翻转课堂是一种新的学习模式，是对传统模式的一种颠覆，在诞生以后，就表现出强大的优势。翻转课堂目前变得越来越普遍。然而，翻转课堂的管理工作琐碎又耗时，如课堂点名、主题发布、学生分组、学生每次讨论的展示成绩汇总和计算、老师对每次课后作业的分数统计等事务，如果完全采用人工方式处理，那么不仅浪费老师和学生的时间和精力，在工作处理中也存在效率低，失误难以避免等问题。针对这些问题，我们需要解放人力，减轻老师和助教工作压力，进行课堂管理系统的开发，以期促进课堂管理的高效、便利进行。

1.2 解决问题

为了使课堂高效运转，减轻老师和助教工作负担，简化人工繁琐又不必要的工作，方便师生进行教学活动，保证课堂的高效运转，我们开发了这套课堂管理系统，开发系统的主要解决的问题如下：

教师和学生不再采取消耗大量时间的传统点名方式，而是能通过系统快速而有效地完成课堂点名。

代替了以往传统的通过现场分配小组以及发放纸张进行话题选择、角色选择、评分打分等方式。老师可以通过系统快速随机分组，学生能通过系统提前组队，选择话题，小组打分，查看自己的成绩分组等信息等。

教师不用通过回收学生的纸张来统计例如分组情况、签到情况、评分等课堂信息，通过系统就能进行信息统计与计算。

教师可以提前对课堂内容进行设置，方便学生提前获知课堂内容，打破了师生交流堵塞状态。

传统的纸张管理数据，容易遭到信息的修改，通过系统管理数据，对于角色的权限不同，访问的数据也不同，提高数据安全性。

1.3 系统主要功能

- 课程、班级、讨论课的创建管理及选课退课
- 设定讨论话题及分组
- 课堂点名
- 组队课堂展示的选题、打分
- 课后报告的提交、打分

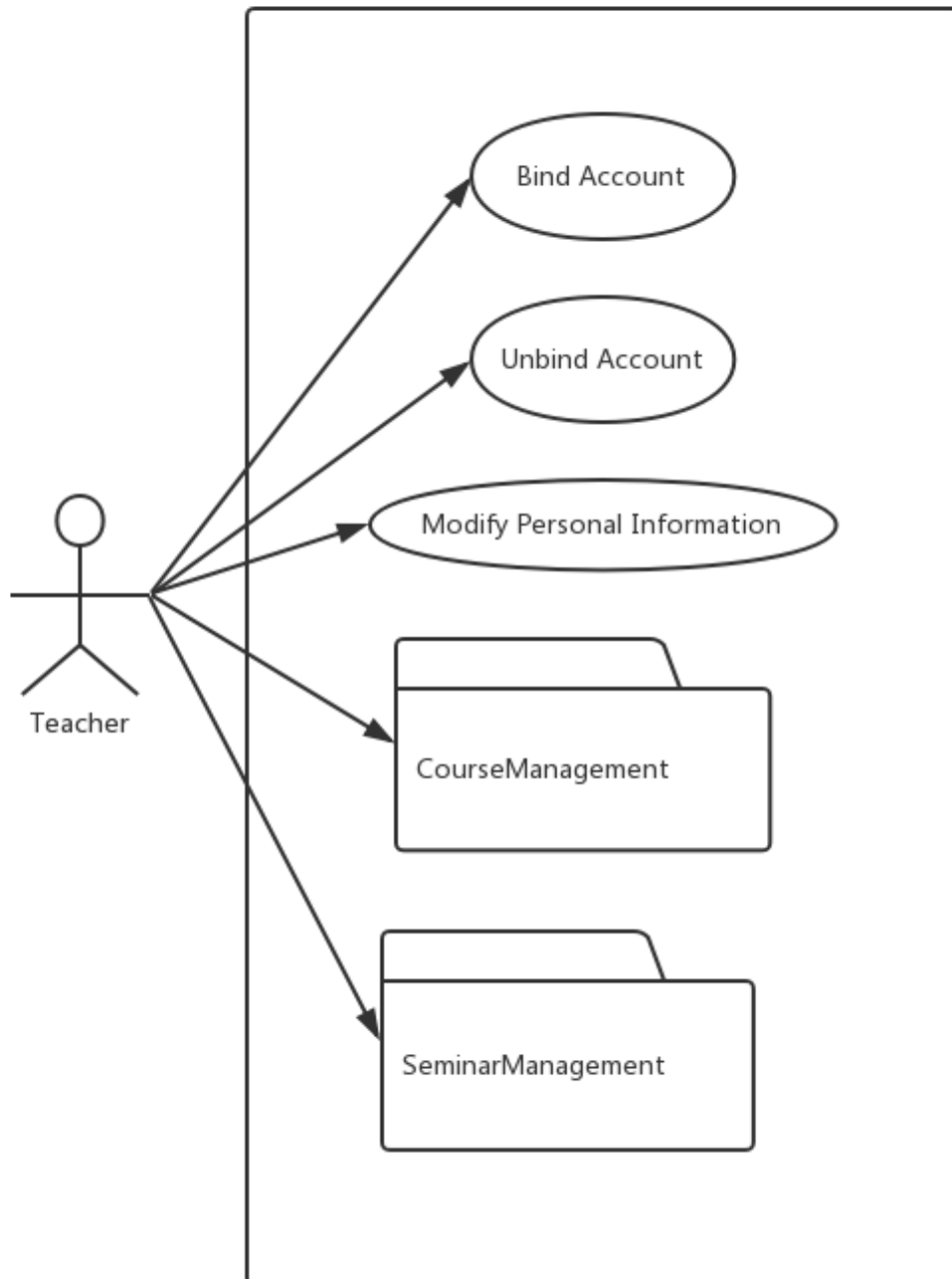
2. 用例

2.1 用户目标分析

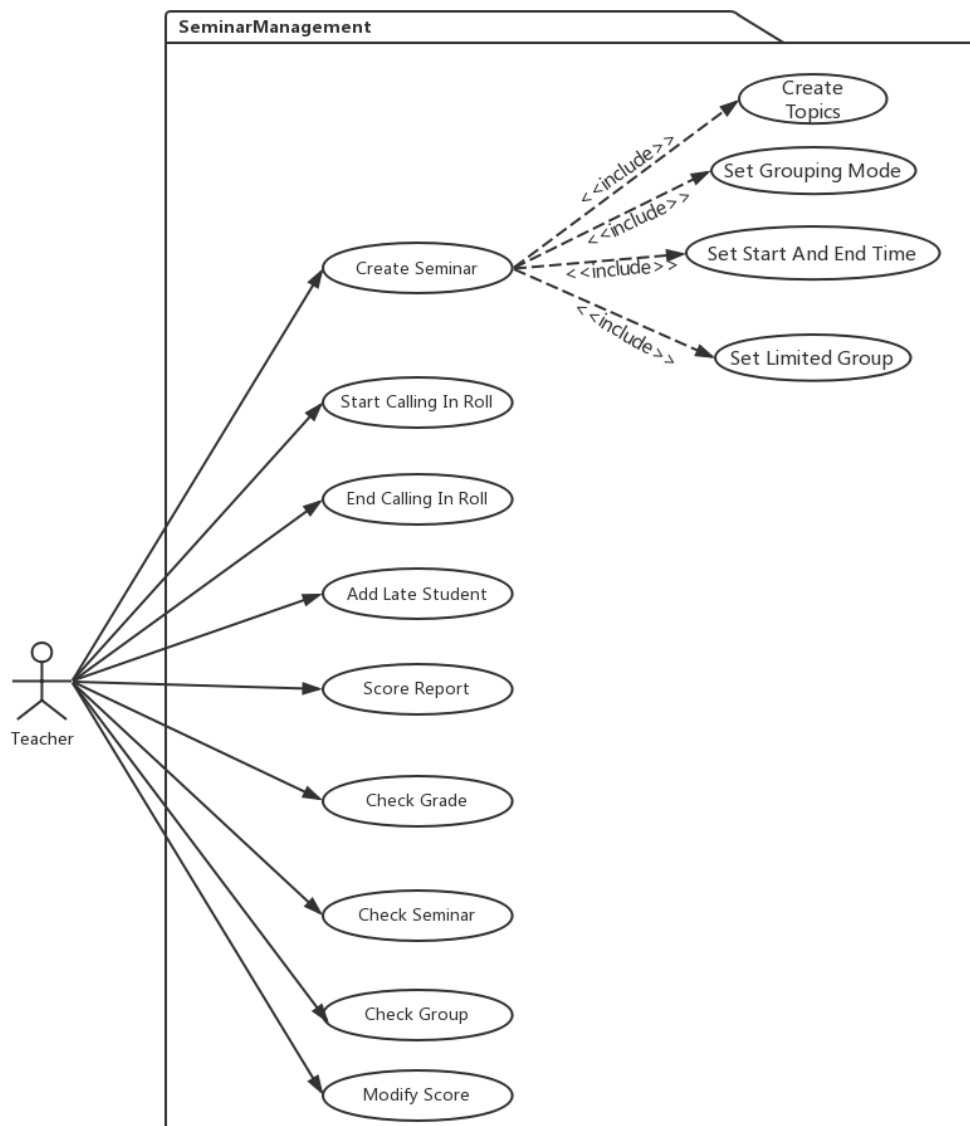
主要参与者	目标
学生	<p>主要分为参与课程和参与课堂以及其他功能三个主要模块。</p> <p>（1）参与课程模块主要达到的目标为：选课、退课，管理固定分组名单，查看课程信息。</p> <p>（2）参与课堂模块主要达到的目标为：签到、查看讨论课信息、查询分组信息、选择主题、给展示打分、上传报告、查询成绩。</p> <p>（3）其他功能模块主要达到的目标为：绑定账号、修改个人信息、解绑账号。</p>
老师	<p>主要分为管理课程和管理课堂以及其他功能三个模块。</p> <p>（1）管理课程主要达到的目标为：创建和取消课程、设置更新课程信息、导入学生名单、创建和删除班级、查看课程班级信息、更新班级信息、创建讨论课。</p> <p>（2）管理课堂主要达到的目标为：发起结束签到、添加迟到学生、给报告评分、查看分组、成绩、讨论课信息。</p> <p>（3）其他功能主要达到的目标为：绑定账号、修改个人信息，解绑账户，创建学校。</p>

2.2 用例图

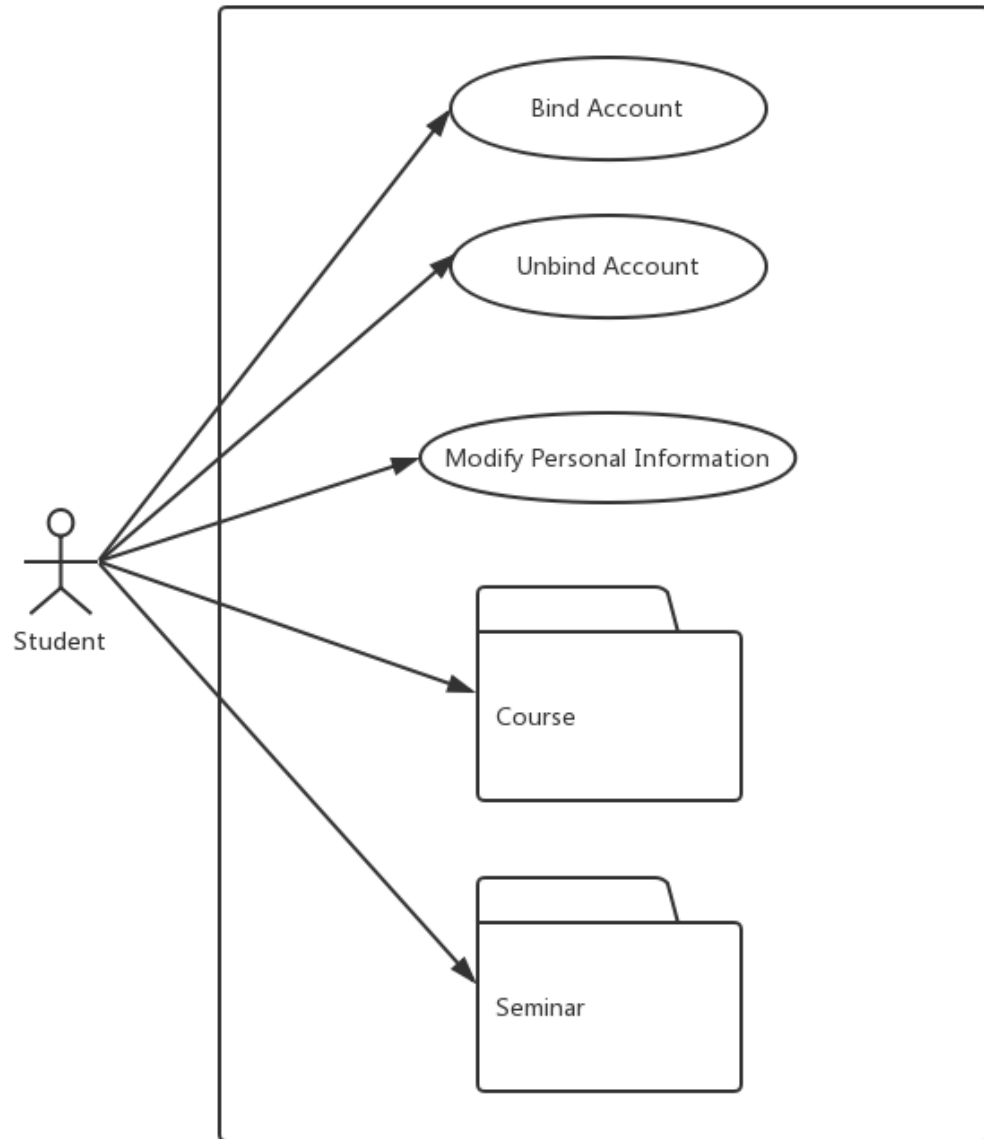
2.2.1 Teacher 用例图

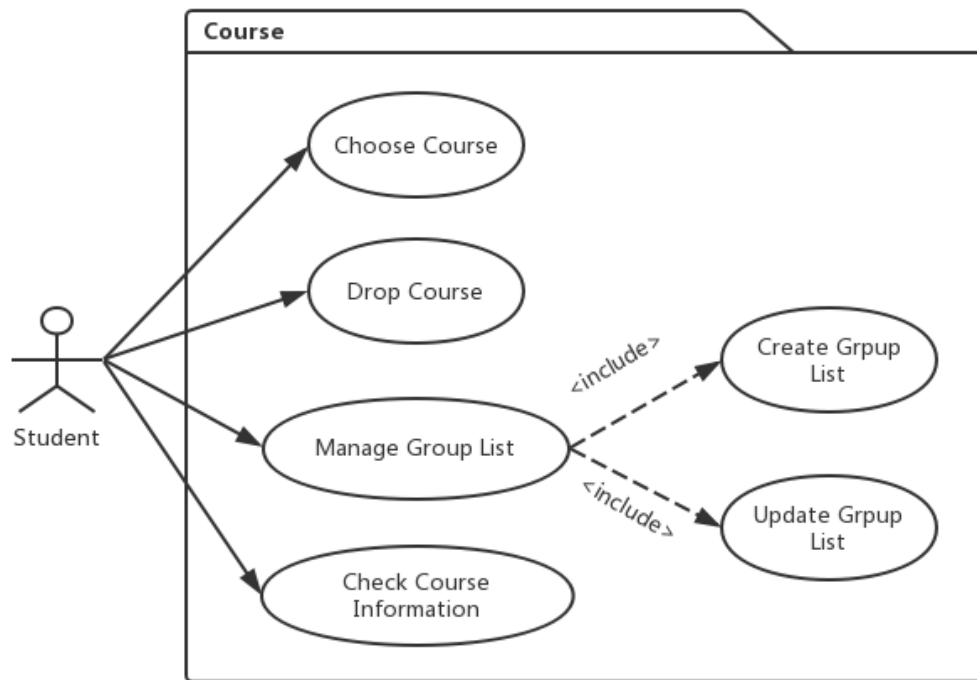


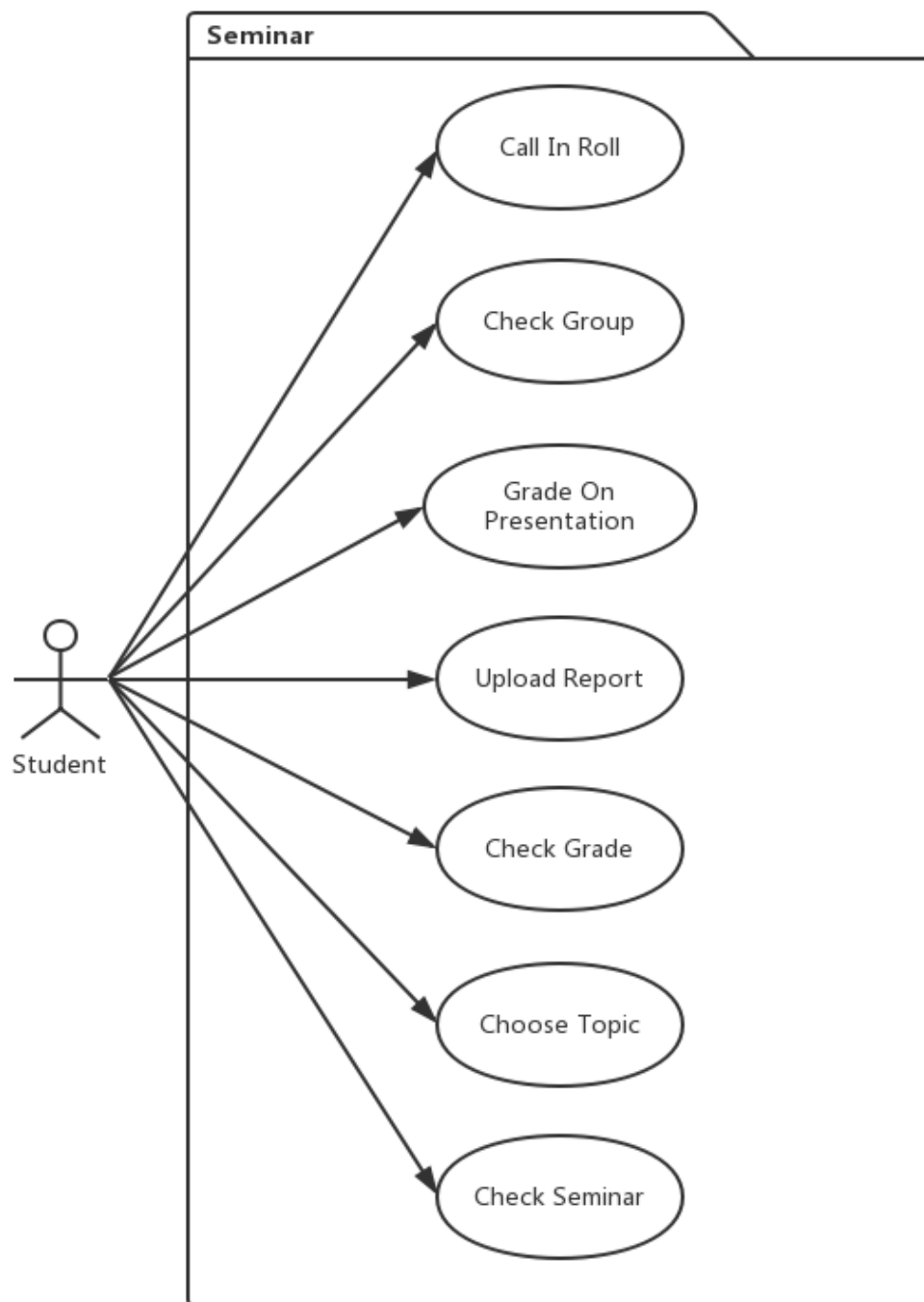




2.2.2 Student 用例图







2.3 用例详述

2.3.1 教师用例详述

(J2EE 组可通过微信扫码或输入账号密码登陆网页, 而 .Net 小组通过输入账号密码登陆网页。)

1. **Bind Account(绑定账号)**: 教师进入微信小程序, 输入教工号, 完善个人信息, 如姓名、学校 (注: 完善个人信息时, 教师需要选择学校。若系统中没有教师所在学校的记录, 则允许教师自行创建学校), 将微信号与自己的教工号绑定。(.Net 无此用例, 需要注册账号, 在注册账号的时候输入电话号码和教工号, 其中电话号码作为注册名/账号, 在后期无法修改)。

2. **Unbind Account(解绑账号)**: 教师将自己的教工号与微信号解除绑定, 此后就可以重新绑定账号。(.Net 无此用例)

3. **Modify Personal Information(修改个人信息)**: 教师可以在系统中编辑修改自己的个人信息 (只能修改 E-mail、职称, 其它都不能修改, 尤其是教工号、名字、学校、(.net)手机绝对不能修改)

4. **Create Course(创建课程)**: 教师选择创建新课程, 并设置一些与该课程相关的信息(Set Course Information)。注: 在 Set Course Information 用例中, 若一门课程的各个班级有相同的规则 (如评分规则), 则教师可以在课程中设置该规则。

5. **Create Class(创建班级)**: 教师在已被创建的课程中创建班级, 并设置该班级的上课时间(Set Time)和上课地点(Set Location)和针对该班级的规则(Set Rules) (注: 若一门课程的各个班级有相同的规则 (如评分规则), 则教师可以在课程中设置该规则, 无需在班级中重新设置)。

6. **Import Student List(导入选课学生名单)**: 教师往系统中导入选课的学生名单。

7. **Create Seminar (创建讨论课)**: 教师在课程下创建讨论课, 并为该次讨论课创建要讨论的话题 (Create Topics), 设置分组方式 (随机分组还是课前固定分组) (Set Grouping Mode), 设置每个话题的限选组数和每组人数上限 (Set Limited Group) (注: 无需对每个班级分别设计组数上限, 各个班级上限一样), 并设定讨论课的开始时间和截止时间。

8. **Delete Course(删除课程)**: 教师可以选择删除某门已创建的课程。

9. **Delete Class(删除班级)**: 教师可以选择删除某门课程的某个班级。

10. **Update Course Information(更新课程信息)**: 教师可以更新课程的相关信息。

11. **Start Calling in Roll(开始点名)**: 教师在系统中选择开启点名功能。在实行随机分组策略时, 点名是必须的 (要把点名的学生进行随机分组); 在实行固定分组策略时, 点名与否取决于教师。

12. **End Calling in Roll(结束点名)**: 教师在系统中选择结束点名功能。若是随机分组, 结束点名时系统会将点名的学生进行随机分组, 组长确定后就可以抢题, 所有人可以实时看到选题情况, 点名结束后 3 分钟内组长没抢题, 系统随机分配; 若是固定分组, 结束点名后不会。

13. **Add Late Students**(分配迟到学生): 在讨论课实行随机分组的机制时, 教师可以将随机分组结束后才到达教室的学生加入在课堂上已经随机分好的组中, 让其参与讨论。

14. **Score Report**(给报告打分): 教师对学生提交到系统中的报告打分。

15. **Update Class Information**(更新班级信息): 修改针对某个班级的打分规则。

16. **Check Group**(查看分好的组): 教师查看当前分好的所有组。

17. **Check Grade** (查询分数): 教师可以查询讨论课每个学生的得分, 包括课堂展示得分、课后报告得分以及总分。

18. **Check Seminar** (查看讨论课信息): 教师可以查看已创建的讨论课的信息。

19. **Check Class Information**(查看班级信息): 查看自己创建的班级的信息。

20. **Check Course Information**(查看课程信息): 查看自己创建的课程的信息。

21. **Modify Grade** (修改分数): 老师可以修改报告得分。

2.3.2 学生用例详述

(J2EE 组可通过微信扫码或输入账号密码登陆网页, 而.Net 小组通过输入账号密码登陆网页。)

1. **Bind Account(绑定账号)**: 学生进入微信小程序, 输入学号, 将微信号与自己的学号绑定, 同时, **学生还需要选择所在学校** (.Net 无此用例, 需要注册账号, 在注册账号的时候输入电话号码和学工号, 其中电话号码作为注册名/账号, 在后期无法修改, 但也需完成选择所在学校的功能)。

2. **Unbind Account(解绑账号)**: 学生将自己的学号与微信解除绑定, 此后就可以重新绑定账号。(.Net 无此用例)

3. **Modify Personal Information(修改个人信息)**: 学生可以在系统中编辑修改自己的个人信息 (只能修改 E-mail、学历, 其它都不能修改, 尤其是学工号、名字、学校、(.net)手机绝对不能修改)

4. **Choose Course (选课)**: 学生选择课程 (具体到班级)。该用例的情况是学生在课程开始前没有选某门课, 但在课程开始后想选, 就可进行选课, 而一开始就在老师导入的选课名单里的学生不用进行此操作。

5. **Drop Course (退课)**: 学生退选课程。

6. **Manage Group (管理固定分组名单)**: 学生在某一时刻 (无论有没有讨论课被创建) 上传小组成员名单, 上传名单的学生默认为小组的组长。当讨论课的分组方式是固定分组的时候, 系统会默认使用固定小组的名单作为讨论课分组, 这时候需要组长提前在讨论课模块选择主题。修改团队成员需要组长来负责。

7. **Check Course Information (查询课程信息)**: 学生查看课程信息 (所分配到的班级信息)

8. **Call in Roll (签到)**: 在教师开启签到功能后, 结束签到功能之前, 学生可以在系统中进行签到。(注: 签到需要学生的位置信息)

9. **Check Group (查询分组名单)**: 学生可以在系统中查看当前课堂自己所在随机分组和组内的组员名单以及固定分组的名单及其组内成员。

10. **Grade on Presentation (给课堂展示打分)**: 学生按照本班级的打分规则给其他小组的展示打分。

11. **Upload Report (上传报告)**: 课后学生可以上传本组的课堂讨论报告。

12. **Check Grade (查询分数)**: 学生可以查询讨论课的得分, 包括换算后的课堂展示得分、课后报告得分以及总分。

13. **Choose Topic (选择主题)**: 当讨论课为随机分组模式时, 组长确定后就可以抢题, 所有人可以实时看到选题情况, 到截止时间组长没抢题, 系统随机分配; 当讨论课为固定分组模式时, 学生需要在讨论课前进行选择主题 (只能由组长选题)。

14. **Check Seminar(查看讨论课)**: 可以查看讨论课的信息 (讨论课所包含的主题及其内容, 讨论课需要的分组方式, 讨论课的开始结束时间, 每个主题在每个班要求的上限组数)

3. 业务流程分析

3.1 业务流程分析

3.1.1 课前活动图

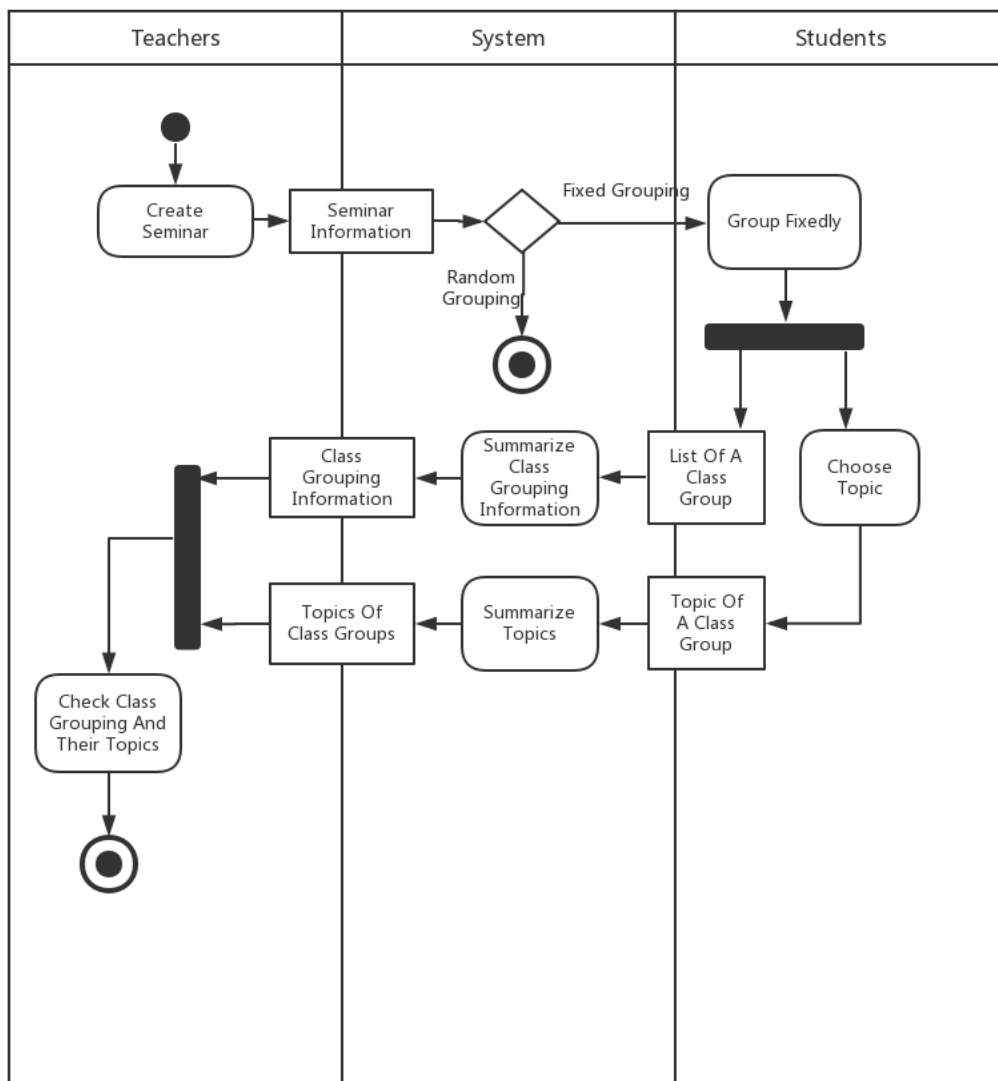
课前活动图是讨论课之前要进行的准备活动。

首先老师要创建讨论课，并且设置好讨论课信息（包括分组方式等），话题的创建在讨论课下完成。

如果分组方式是固定分组，则学生在课前完成分组任务和选题任务，上传分组选题结果，系统汇总后给老师，老师查看分组选题结果；

如果分组方式是随机分组，则分组选题任务在课上进行。

至此，课前的任务全部完成。



3.1.2 课堂活动图

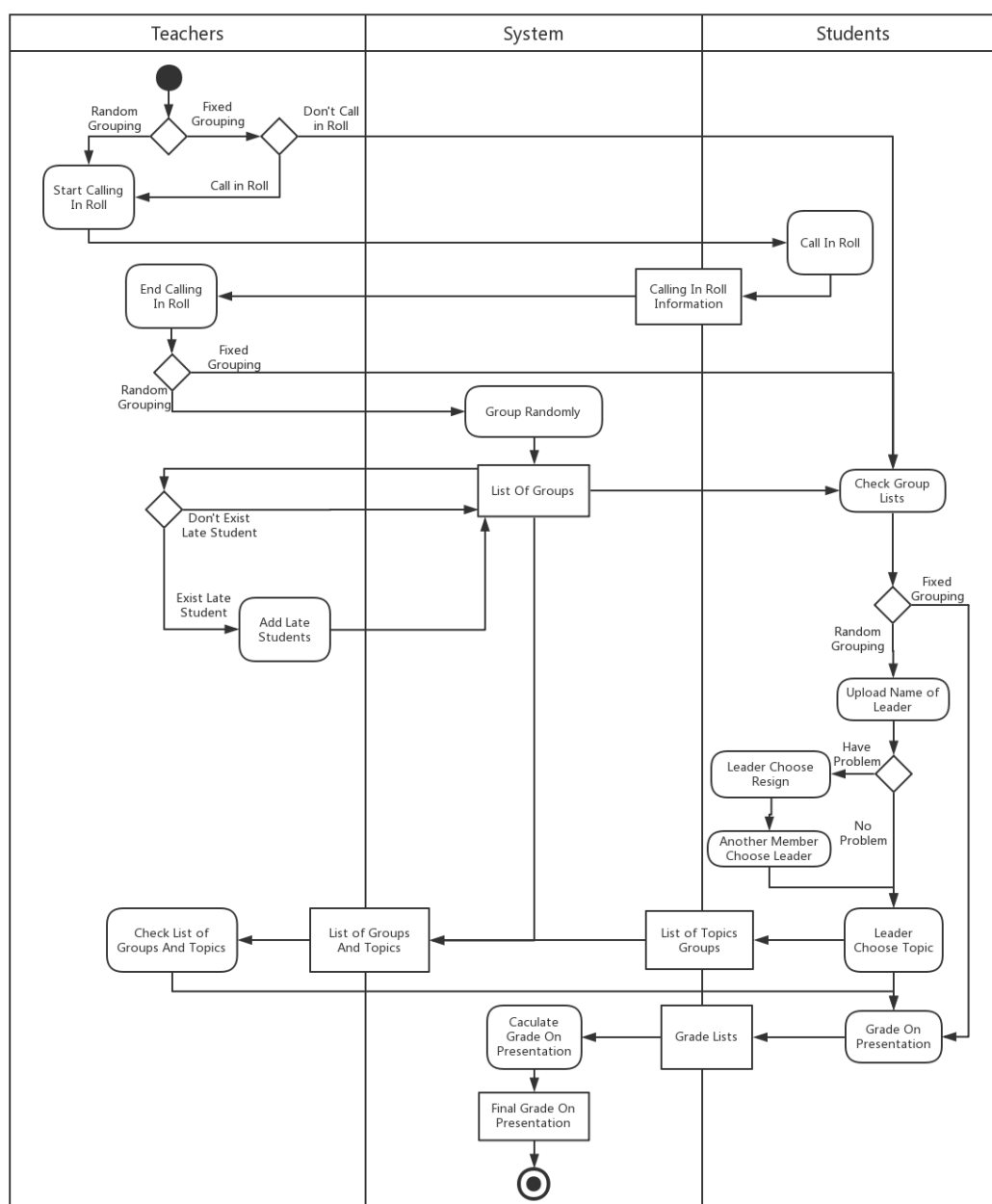
课堂活动图表示了课堂过程中的任务流程。

如果分组策略为随机分组，则老师需要发起点名，学生参与点名，老师结束点名后，系统进行随机分组，学生可以查看分组情况。每个组需要选出一个组长，组长在系统上进行选题，系统进行汇总后给老师查看，小组选题后即可进行讨论。若有学生迟到，老师可以手动将其加入到某个小组中。

如果分组策略为固定分组，老师可以点名，也可以不点名。若要点名，点名结束后直接进入讨论阶段。若不点名，则直接进入讨论阶段。

讨论结束后，小组进行展示，每个同学要对各组进行打分，将分数上传到系统中，系统进行汇总后给老师查看。

至此，课上任务全部完成。



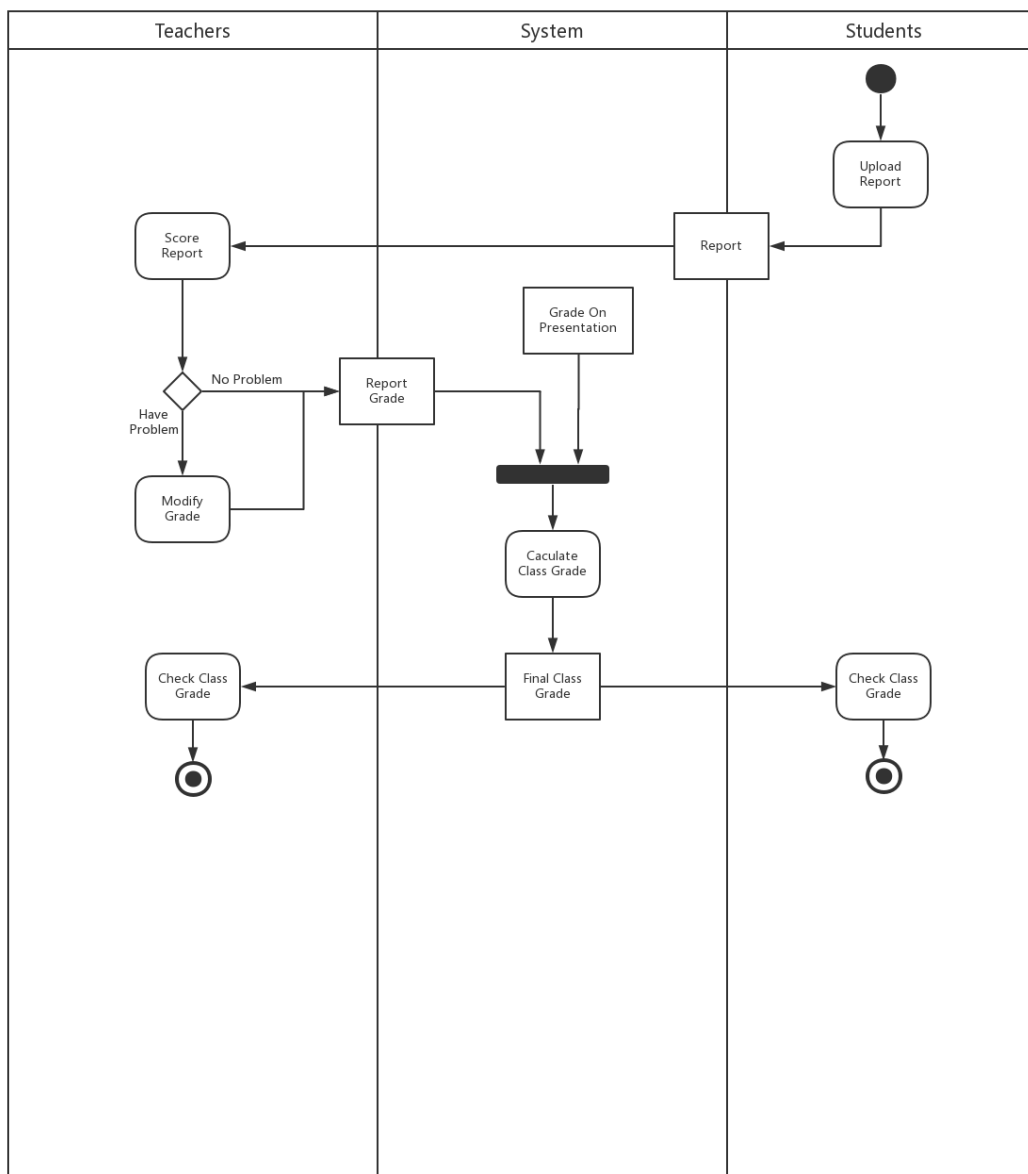
3.1.3 课后活动图

课后活动图是讨论课结束之后的任务流程。

主要流程为：学生上传报告，老师拿到报告后对其进行评分，并可以对讨论课分数（包括学生打分和报告分数）进行多次修改。

系统自动将报告和课程的分数按标准计算，并将总成绩返回给老师和学生进行查看。

至此，讨论课之后的活动结束。

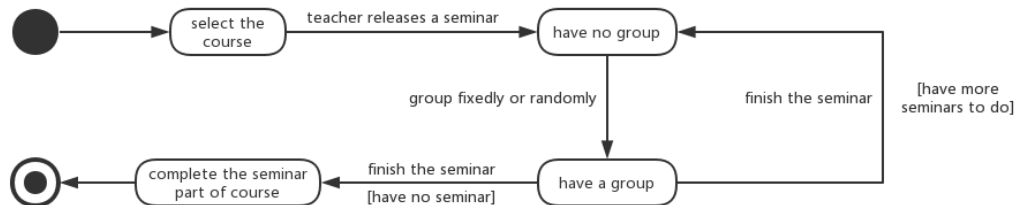


3.2 对象状态机分析

3.2.1 学生状态机图

该图展现了学生在一个课程从开始到结束的整个状态变化，是顶层(top-level)的状态机图。

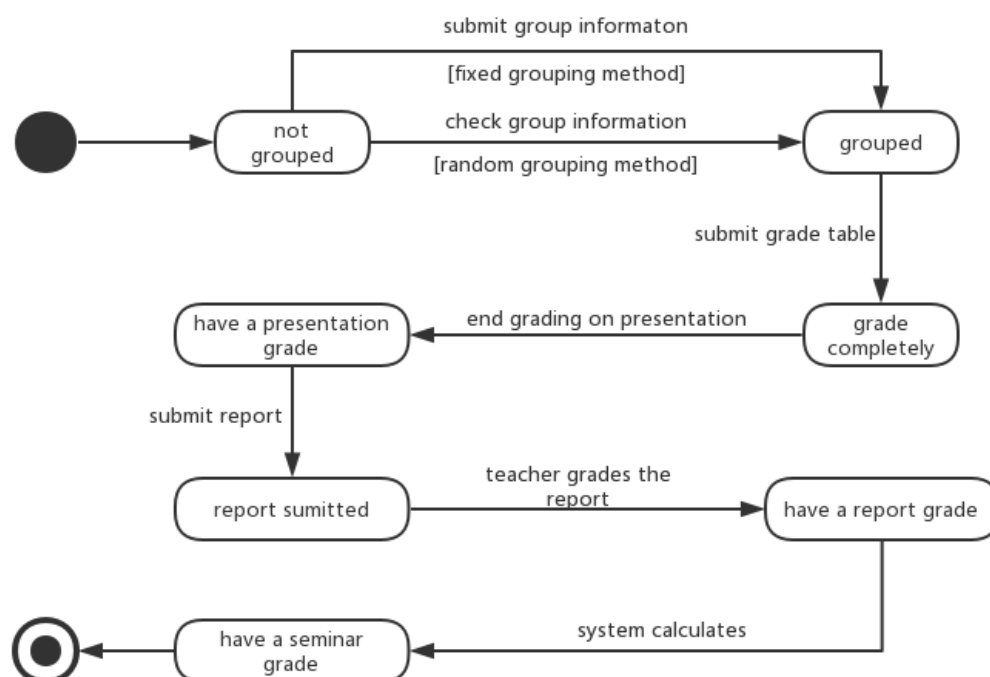
刚开始时学生选择了一门课程，当老师发布第一个讨论课时，学生会转入没有分组状态。当学生经过随机分组或固定分组后，学生的状态由没有分组状态转变为已分组状态。当一次讨论课的流程结束时，如果这不是本门课的最后一次讨论，则学生又转为没有分组状态；如果本门课没有讨论课需要讨论时，则学生转变为完成该门课所有讨论的状态。



3.2.2 小组状态机图

该图展现了小组在一次讨论的整个流程的状态变化。

刚开始时，小组为未组成状态，当经过随分组或固定分组后，小组状态转变为已组成状态。当该小组提交了给其它小组的打分时，小组转变为完成打分状态。当所有小组都打完分后，系统会计算出每个小组的展示分，小组转变为有展示分状态。当小组在课后提交了讨论报告时，小组转变为已提交报告状态。当老师给小组的报告打完分后，小组状态转变为有报告分状态。当老师给所有组都打完报告分后，系统根据展示分和报告分计算出该次讨论课的总分/最终成绩。



4. 非功能性需求

4.1 性能需求

响应速度：一般的 web 端操作不会需要太多响应时间，速度自然是越快越好，用户可以容忍一至二秒的延迟。而以高并发的签到为例，学生希望在三至五秒内得到自己签到情况的反馈，老师则希望在结束签到的半分钟内得到所有学生的签到情况。

用户数量：假设一所高校中有 20 位老师希望借助该系统来辅助教学，每位老师教授 50~200 名学生不等，若在全国范围内有 50 所高校引进了该系统，则需要支持的用户数量为十万左右。

准确性：主要是在签到时所使用的地理位置的准确性。系统判定学生的状态的准确性达到可包容范围。

4.2 可靠性需求

系统应具有较强的稳定性，特别是在学生集体进行签到的时候。对于突发的故障要有一定的处理能力，能在短时间内恢复正常工作。

4.3 可用性需求

系统的设计应符合美学和人体工程学，易于上手使用。

4.4 安全性需求

4.4.1 身份认证

用户初次进入时，必须绑定账号并完善相关信息才能进入系统。

每次登录时系统需对用户的身份进行认证。J2EE 组的微信小程序通过微信登录，Web 端则通过微信扫码进行登录；.net 组的移动端 Web 和 PC 端 Web 则通过用户名和密码进行登录。老师和学生用户分别拥有不同的权限，访问不同的数据。

4.4.2 权限控制

老师：①只能看到自己创建的课程，不能看到别的老师创建的课程。

②老师只能修改学生报告的成绩，不能修改课堂打分的成绩。

学生：①学生只能看到换算后的课堂打分成绩、报告成绩、总成绩。不能看到每个学生给其他小组的详细打分情况。

②学生只能看到自己小组的相关信息，不能看到别的小组的相关信息。

所有用户只能看到或操作本学校的数据。

学生只能看到学生能看到的页面、使用 Restful Controller 的 API 而不能看到老师能看的，反之亦然。

4.4.3 数据库安全

防止 SQL 注入。

4.5 外部接口

微信小程序中需要用到定位或地图的接口。可能需要上传和下载文件的接口。

4.6 可支持性需求

4.6.1 兼容性

如果该系统被推广到全国范围内使用，则要求有较好的可维护性和可移植性，能在不同平台上稳定运行，项目的浏览器兼容性要求为当前 Chrome、Firefox 与 Edge 的最新稳定版与最新 ESR 进行，即 Firefox 57、Firefox ESR 52.4.1、Chrome 62、Edge 16，并配合 Safari 11 的对于主要功能的兼容性测试。

其中相关小组至少需要实现：

- ① Web 端（J2EE）：Chrome 浏览器
- ② 微信小程序（J2EE）：华为(HUAWEI)M3 8.4 英寸平板电脑和小米 5
- ③ .NET 移动 web 系统：华为(HUAWEI)M3 8.4 英寸平板电脑和小米 5
- ④ .NET Web 端：Chrome 浏览器

4.6.2 分辨率

除了对不同的运行环境有不同的兼容性，该系统还需要满足适应不同的设备，考虑到时间与精力等限制条件，主要满足主流设备条件即可，具体为：

- ① WEB 端：
满足：1920*1080、1440*900、1366*768、1024*768
其中最小分辨率为 1024*768
- ② 手机/平板：
由于安卓手机的 DPR 数值多样，因此为了方便定义了浏览器分辨率（浏览器分辨率=设备分辨率/DPR）。
满足：宽度适应 300（4 英寸竖屏）到 800（平板竖屏），其中最小为 300