软 件 学 院

课程设计任务书

**软件学院 软件工程1603班**

**课程设计名称： 面向对象分析与设计**

**课程设计题目： 3）系统用例图、用例的事件流**

**（完成起止日期：2018年12月5日—2018年12月26日共3周）**

**学 生 姓 名 赵冠华 学号 U201617059**

**指 导 教 师 方少红**

**系主任 20 年 月 日审查**

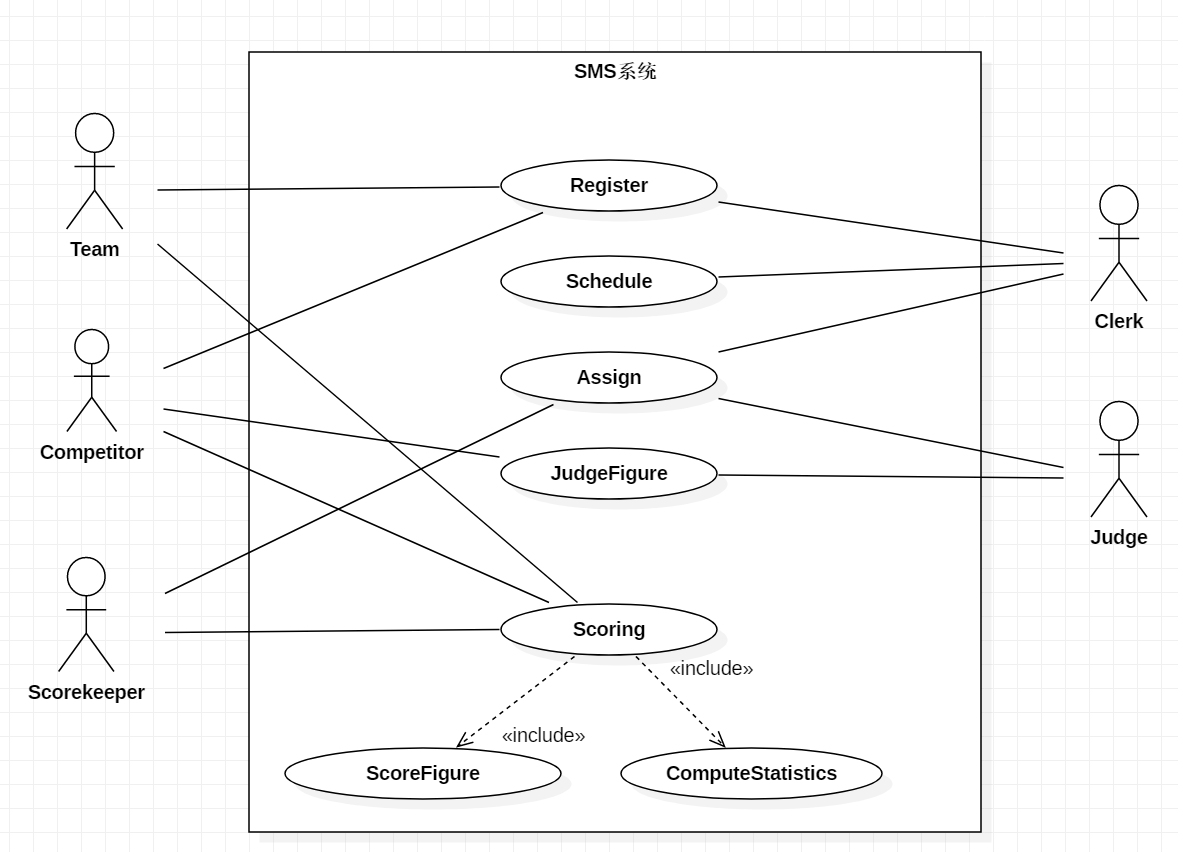
|  |
| --- |
| 设计依据：  案例说明：  某花样游泳运动会评分系统（SMS）的需求简要描述如下，其中部分类的类名和属性名在括号内表示。  花样游泳运动会(Meeting)为本地的游泳运动参赛队(Team)举办比赛(Event)，需要开发一个评分系统对完成运动员注册、安排项目日程、记载评分和累计总分。  运动会有若干个场地(Station)进行比赛，并指定若干个规定项目(Figure)，规定项目是指特定的游泳动作，每个规定项目按其复杂程度具有不同的难度系数(difficultyFactor)；每组比赛在一个场地上完成，由若干个队员(Competitor)参加，一个队员可以报名参加多个规定项目的比赛；每个报名参加某个规定项目的队员在比赛中表演一次，这样的表演称为预赛(Trial)；若干个队员组成一个参赛队，每个队员只能代表一个参赛队参加比赛；若干个裁判(Judge)考察每个队员的每次预赛分别给出原始分(rawScore)，每个场地上由若干个记分员(ScoreKeeper)记载每个裁判的原始分并以此原始分和规定项目的难度系数计算出每个预赛的净得分(netScore)。运动会对每个参赛队和每个参赛队员记总分，总分由净得分总计。 |
| 设计要求及主要内容（可另加附页）：  要求：  （2）系统用例图：完整覆盖系统需求，简要说明（Summary）每个用例和每个参与者；写出一半以上的用例的正常成功场景的事件流。 |
| 参考文献（由指导教师选定）  1 UML用户指南  2 UML面向对象建模与设计  3 UML面向对象分析与设计 |
| 同组设计者  软工1603班 刘俊傲 U201617047  软工1603班 刘杨 U201617048 |

注：1、此任务书应由指导教师填写。

2、此任务书最迟必须在课程设计开始前一周下达给学生。

系统用例图

## 2.1 系统用例图



**//完整覆盖系统需求**

## 2.2 参与者说明

|  |  |
| --- | --- |
| Clerk（操作员） | ①Schedule：安排场地和赛程  ②Assign：给比赛场地分配裁判和记分员  ③Register：参赛队注册队员，队员注册规定项目 |
| Scorekeeper（记分员） | ①ScoreFigure：  规定项目评分，记分员收到原始分，计算净得分  ②ComputeStatistics：  运动会总计分，分别汇总参赛队和队员的总分 |
| Judge（裁判） | ①JudgeFigure：  规定项目评判，裁判观看队员对规定项目的预赛并给出原始分 |
| Competitor（队员） | ①Register：参赛队注册队员，队员注册规定项目 |
| Team（参赛队） | ①Register：参赛队注册队员，队员注册规定项目 |

//简要说明（Summary）每个用例和每个参与者

## 2.3 用例简要说明（每个用例1页，共6个用例）

用例1 Register注册

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 1 | **名称** | Register |
| **创建者** | 赵冠华 | **最后一次更新者** | 赵冠华 |
| **创建日期** | 2018/12/28 | **最后更新日期** | 2019/1/3 |
| **参与者** | SMS系统用户Clerk、Team、Competitor，目标是准确地注册参赛队账号、参赛者用户名、参赛者用户密码，记录用户信息 | | |
| **触发条件** | 用户首次使用SMS系统 | | |
| **前置条件** | 用户是同一参赛队的队员，但尚未成功登录过SMS系统 | | |
| **后置条件** | 系统存储用户登录信息，包括参赛者用户名、参赛者用户密码 | | |
| **优先级** | 高 | | |
| **正常流程** | 1. Team选择注册Competitor 2. 系统提示Team输入**队伍名**、**参赛者用户名**、**参赛者用户密码**等信息 3. 系统验证注册信息的合法性，提示注册成功，存储注册信息，更新数据库 4. Competitor选择登录 5. 系统提示Competitor输入**队伍名**、**参赛者用户名**、**参赛者用户密码** 6. Competitor输入**队伍名**、**参赛者用户名**、**参赛者用户密码** 7. 系统验证登录信息（teamUser），提示登录成功，进入SMS系统 8. Competitor勾选自己要参加的项目，点击注册 9. 系统验证注册信息的合法性，提示注册成功，存储注册信息，更新数据库 | | |
| **扩展流程** | 1. 用户已经注册过系统账号  ——无需注册，直接登录  2. 参赛者用户名输入错误或参赛者用户密码不符合格式  ——提示错误信息，要求重新输入 | | |
| **特殊需求** | 1. Team数据库至少能够存储Team个参赛队的信息 2. Competitor数据库至少能够存储5000名Competitor的信息 | | |

//根据第一题类图中的属性做出事件流

①

|  |  |  |
| --- | --- | --- |
| **Admin** | 注册登陆 | content 注册登录信息 |

②

|  |  |  |
| --- | --- | --- |
| **Competitor** | 队员 | name 姓名  age 年龄  address 地址  telephone 联系方式 |
| **Team** | 参赛队 | name 名称 |

用例2 Schedule，安排场地和赛程

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 2 | **名称** | Schedule |
| 创建者 | 赵冠华 | 最后一次更新者 | 赵冠华 |
| 创建日期 | 2018/12/28 | 最后更新日期 | 2019/1/3 |
| **参与者** | SMS系统用户Clerk，目标是分配好比赛的场地，确定比赛时间 | | |
| **触发条件** | Register结束 | | |
| **前置条件** | 参赛队已经成功注册队员，队员已经成功注册规定项目 | | |
| **后置条件** | 场地和赛程安排完毕，运动会进入赛前准备阶段 | | |
| **优先级** | 高 | | |
| **正常流程** | 1. Clerk查看所有的比赛项目和每个项目的参赛人数以及场地列表place 2. Clerk确定运动会的起始和结束日期 3. Clerk根据每个项目的参赛人数分配合适的场地 4. Clerk根据前面确定的各项分配，安排合理的赛程（时间） 5. Clerk将系统生成的日程表做出秩序册（**项目**、**时间**、**场地**、**参赛队员**） 6. Clerk将秩序册发放给各参赛队 | | |
| **扩展流程** | 2.比赛时间和场地有冲突  ——系统提示错误信息，提醒Clerk重新进行分配  2.比赛结束时间超过每天的限制  ——系统提示错误信息，提醒Clerk重新进行分配 | | |
| **特殊需求** | 无 | | |

//根据第一题类图中的属性做出事件流

①

|  |  |  |
| --- | --- | --- |
| **AssignController** | 赛程管理 | sessionTime会话时间 |
| **CompetitorController** | 运动员管理 | sessionTime会话时间 |

②

|  |  |  |
| --- | --- | --- |
| **Station** | 场地 | location 地点  StationName 场地名称 |

用例3 Assign，给比赛场地分配裁判和记分员

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 3 | **名称** | Assign |
| **创建者** | 赵冠华 | **最后一次更新者** | 赵冠华 |
| **创建日期** | 2018/12/28 | **最后更新日期** | 2019/1/3 |
| **参与者** | SMS系统用户Clerk，目标是安排好Scorekeeper和Judge，保证比赛正常进行 | | |
| **触发条件** | Register结束 | | |
| **前置条件** | 参赛队已经成功注册队员，队员已经成功注册规定项目 | | |
| **后置条件** | Scorekeeper和Judge分配完毕，运动会进入赛前准备阶段 | | |
| **优先级** | 高 | | |
| **正常流程** | 1. Clerk用户查看schedule表，查看judge和Scorekeeper的人数  2. Clerk通过系统将judge和Scorekeeper设置为“忙碌”和“空闲”两种状态  3. Clerk将judge根据项目进行分组  4. Clerk根据分组留出预备的judge和Scorekeeper，无异常情况不参与项目  5. Clerk根据项目的队员和时间安排人数合适的judge和Scorekeeper  6. Clerk查看judge和Scorekeeper的状态，再根据分组安排下一个项目  7. 系统生成Assign表  8. Clerk将Assign表发给judge和Scorekeeper | | |
| **扩展流程** | 1.若Clerk安排的judge和Scorekeeper为“忙碌”状态  ——系统提示错误信息，提醒Clerk重新安排  2. 若judge的分组和比赛项目不一致  ——系统提示错误信息，提醒Clerk重新安排  3.若比赛中judge和Scorekeeper出现异常情况  ——Clerk查看系统，启用预备的judge和Scorekeeper | | |
| **特殊需求** | 1.预备的judge人数不得少于30人  2.预备的Scorekeeper人数不得少于100人 | | |

//根据第一题类图中的属性做出事件流

①

|  |  |  |
| --- | --- | --- |
| **Clerk** | 操作员 | name 姓名  C\_ID 编号 |
| **ScoreKeeper** | 记分员 | name 姓名  S\_ID 编号 |
| **Judge** | 裁判 | name 姓名  J\_ID 编号 |
| **Station** | 场地 | location 地点  StationName 场地名称 |
| **Event** | 比赛 | startTime 开始时间  EventName 比赛名称 |

用例4 JudgeFigure，规定项目评判

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 4 | **名称** | JudgeFigure |
| **创建者** | 赵冠华 | **最后一次更新者** | 赵冠华 |
| **创建日期** | 2018/12/28 | **最后更新日期** | 2019/1/3 |
| **参与者** | 裁判judge，目标是对每一位competitor的表演给出原始分 | | |
| **触发条件** | 项目入场时间到 | | |
| **前置条件** | 场地和赛程已经安排好，比赛场地已分配好裁判和记分员 | | |
| **后置条件** | 场地空，准备下一项目 | | |
| **优先级** | 高 | | |
| **正常流程** | 1. 运动员入场，judge和Scorekeeper就位 2. judge宣布比赛开始 3. 系统根据入场的competitor向judge提供难度系数 4. judge观看competitor表演 5. 表演完毕 6. judge根据选手的表现给出原始分 7. judge将原始分交给Scorekeeper 8. 下一位competitor入场，重复上述步骤 9. 所有competitor全部表演完毕，该项目结束 10. 进行下一个项目，重复上述步骤 11. 所有项目结束，JudgeFigure结束 | | |
| **扩展流程** | 无 | | |
| **特殊需求** | Judge不得在competitor未表演完时就给出净得分 | | |

//根据第一题类图中的属性做出事件流

①

|  |  |  |
| --- | --- | --- |
| **Judge** | 裁判 | name 姓名  J\_ID 编号 |
| **Station** | 场地 | location 地点  StationName 场地名称 |
| **Event** | 比赛 | startTime 开始时间  EventName 比赛名称 |
| **Trial** | 预赛 | netScore 净得分 |
| **Figure** | 规定项目 | title 项目名称  difficultyFactor  难度系数  Description 具体描述 |
| **Competitor** | 队员 | name 姓名  age 年龄  address 地址  telephone 联系方式 |

用例5 ScoreFigure，规定项目评分，记分员收到原始分，计算净得分

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 5 | **名称** | ScoreFigure |
| **创建者** | 赵冠华 | **最后一次更新者** | 赵冠华 |
| **创建日期** | 2018/12/28 | **最后更新日期** | 2019/1/3 |
| **参与者** | Scorekeeper记分员，目标是计算出每一位competitor的净得分 | | |
| **触发条件** | judge将原始分交给Scorekeeper | | |
| **前置条件** | JudgeFigure（规定项目评判）结束 | | |
| **后置条件** | 得出净得分排名 | | |
| **优先级** | 低 | | |
| **正常流程** | 1. Scorekeeper收到judge给出的原始分  2. Scorekeeper将competitor的原始分录入系统  3. Scorekeeper将计算净得分的规则输入系统  2.系统通过计算给出净得分 | | |
| **扩展流程** | 无 | | |
| **特殊需求** | 1.净得分数据要精确到小数点后3位 | | |

//根据第一题类图中的属性做出事件流

①

|  |  |  |
| --- | --- | --- |
| **ScoreKeeper** | 记分员 | name 姓名  S\_ID 编号 |
| **Judge** | 裁判 | name 姓名  J\_ID 编号 |
| **Station** | 场地 | location 地点  StationName 场地名称 |
| **Event** | 比赛 | startTime 开始时间  EventName 比赛名称 |
| **Trial** | 预赛 | netScore 净得分 |

②

|  |  |  |
| --- | --- | --- |
| **ScoreController** | 积分管理 | sessionTime会话时间 |
| **AssignController** | 赛程管理 | sessionTime会话时间 |
| **CompetitorController** | 运动员管理 | sessionTime会话时间 |

用例6 ComputeStatistics，运动会总计分，分别汇总参赛队和队员的总分

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 6 | **名称** | ComputeStatistics |
| **创建者** | 赵冠华 | **最后一次更新者** | 赵冠华 |
| **创建日期** | 2018/12/28 | **最后更新日期** | 2019/1/3 |
| **参与者** | Scorekeeper记分员，目标是计算出参赛队和队员的总分 | | |
| **触发条件** | 每一个项目的每一名competitor的净得分全部给出 | | |
| **前置条件** | 所有项目的ScoreFigure和ScoreFigure结束 | | |
| **后置条件** | 得出参赛队和队员的总分排名 | | |
| **优先级** | 低 | | |
| **正常流程** | 1. 所有项目的JudgeFigure结束 2. 所有项目的ScoreFigure结束 3. Scorekeeper汇总所有的净得分 4. 系统计算出每位competitor的总得分 5. 系统计算出每个Team的总得分 | | |
| **扩展流程** | 无 | | |
| **特殊需求** | 1. 总分数据要精确到小数点后3位 | | |

//根据第一题类图中的属性做出事件流

①

|  |  |  |
| --- | --- | --- |
| **Clerk** | 操作员 | name 姓名  C\_ID 编号 |
| **ScoreKeeper** | 记分员 | name 姓名  S\_ID 编号 |
| **Competitor** | 队员 | name 姓名  age 年龄  address 地址  telephone 联系方式 |
| **Team** | 参赛队 | name 名称 |

②

|  |  |  |
| --- | --- | --- |
| **ScoreController** | 积分管理 | sessionTime会话时间 |