

目录

一. 项目简介	1
1. 项目背景	2
2. 项目目标	2
3. 项目进展	2
4. 术语表	2
5. 引用	3
二. 架构设计	4
1. 系统级分析	4
2. 应用层	5
3. 业务层	5
a. 账户管理子系统	5
b. 社区交流	6
c. 房间管理子系统	6
d. 约球子系统	7
e. 赛事子系统	7
4. 持久层	8
5. 数据库层	8
三. 设计模型	9
1. 用户	9
a. 类图	9
b. 时序图	10
c. 通讯图	11
2. 房间的创建与管理	12
a. 类图	12
b. 时序图	13
c. 通讯图	13
3. 约球/约战	14
a. 类图	14
b. 时序图	15
c. 通讯图	15
4. 赛事发布与报名	16
a. 类图	16
b. 时序图	17
c. 通讯图	18
四. UI 界面设计	19
1. 登录界面:	19
2. 主界面:	20
3. 创建房间:	20
4. 房间信息页面:	21
5. 修改个人信息:	21
6. 约战:	22
7. 发布比赛:	22
8. 加入比赛:	23
五. 部分用例模型的更新	23
1. 更改约战用例模型	23
2. 细化社区交流模型	24
六. 团队分工	25

一. 项目简介

1. 项目背景

当下高校学生具有较高的运动需求，但是往往难以在短时间内约到能够一起运动的伙伴。针对于 网球这项活动，为了让同学们不再因个人交际圈有限而无法打球，能方便地与陌生的球友进行快速线上约球，我们开发了这款小程序。本项目是设计与开发一款校园网球线上约球、赛事发布、排行榜、线上约课的微信小程序，构建校园网球平台，针对校园内全体师生，让校园内师生了解校园网球文化、了解校园网球圈，让更多的人了解网球，参与网球运动，爱上网球。让师生能够更快、更有效的了解到校园网球队友信息，不再因个人交际圈而受限；与认识、不认识的球友进行快速线上约球和约战；让网球小白能找到学习网球的渠道，建立校园网球教练圈，打造校园网球生态圈，让校园网球迸发出蓬勃的生机，从而更好的在校园内传播网球文化。

2. 项目目标

本系统的目标是为一同济大学为代表的上海高效建立活跃的线上网球平台，激发大学生锻炼的动力。本系统提供约球、约课、赛事管理三大核心功能。短期目标是解决上海高校——同济大学校内网球爱好者互相约球、举办比赛的便利性不足的问题、为更多想学习网球的学生和老师提供学球渠道的问题；长期目标是在短期目标实施运营顺利后，将此平台继续向全上海高校推广、向以省/市为单位的全国高校进行推广运用，为约球、约比赛、相关教学提供便利。未来在纵向推广的同时增加横向内容，即增加其他运动项目，使此小程序成体系地服务于校内外运动市场。

3. 项目进展

项目进度按计划正常进行，前期已经确定基本用例，完成了项目用例的抽象，绘制了用例图和活动图，填写了各用例的详细说明表格，完成了基本事件流的构建。

本次任务进展：

- a. 更新用例模型，对于之前的用例作了进一步的更改细化，同时对之前一些不合理的流程进行修改，使其更加符合实际情况。
- b. 进行类图的构建，为每个类确定基本属性，并对每个类划分功能点，承担对应的职责。
- c. 绘制用户与各个类之间通信的交互图，明确在每个通信行为中需要调用的方法。
- d. 进行项目架构分析，绘制系统分层架构图。

4. 术语表

术语	定义
登入	用户通过同济统一身份认证进入应用程序的行为
密码	用于登录网站或程序时的一串符号系统加密口令

同济大学统一身份认证	师生登录学校各主要业务系统时所采用的统一的账号（学号和密码）
惯用手	在进行乒乓球运动时的常用握拍手
论坛	网络上供网民通过互联网发布自己观点交换信息的平台
帖子	在论坛或贴吧用户所发布的表达自己观点或看法的多媒体内容
数据库	长期存储在计算机内的、有组织的、可共享的、统一管理的大量数据的集合
头像	网站及社交平台上用于仅做表示的图片
房间	一个可容纳多位用户的虚拟空间，通过房间号唯一进行标识
房主	创建房间的用户
解散房间	将所有房间内的用户与该房间的联系删除，并删除该房间的信息
约球	两位或者多位用户相约进行娱乐性质的打球，不进行比赛
约战	两位选手相约进行比赛，决出胜负并且需要上传成绩
排行榜	对各位用户约球时长和约战结果进行排名并且公示的榜单

5. 引用

李波等编著.UML2 基础、建模与设计实战.北京：清华大学出版社，2014

该书详细介绍了 UML2.0 所有的图形的定义与使用方法，同时理论和实例相结合，分析了 UML 综合实战案例，让读者在实际操作中举一反三、迅速理解 UML 的逻辑以及掌握 UML 的运用方法。

本次作业参考了原书第五章“类图和对象图”以及第六章“顺序图和通信图”。从第五章中我们学习到了类图和接口的概念以及类图的设计原则和方法步骤，在实践中我们也一直遵从书中的设计方法来进行建模，大大提升了我们设计类图的效率；从第六章中我们学习了时序图和通讯图的具体设计原则以及相互转化，以及两者之间的比较，为我们提供建模理论和方法的同时深化了我们对两种图内涵的理解。

John W. Satzinger 等编著.系统分析与设计敏捷迭代方法.机械工业出版社，原书第 6 版

本书主要讨论系统分析、系统设计和项目管理三大主题，以一个完整的系统开发案例贯穿始终。更加注重敏捷和迭代方法，并通过 UML 建模使面向对象方法的讲解更加细致和深入。

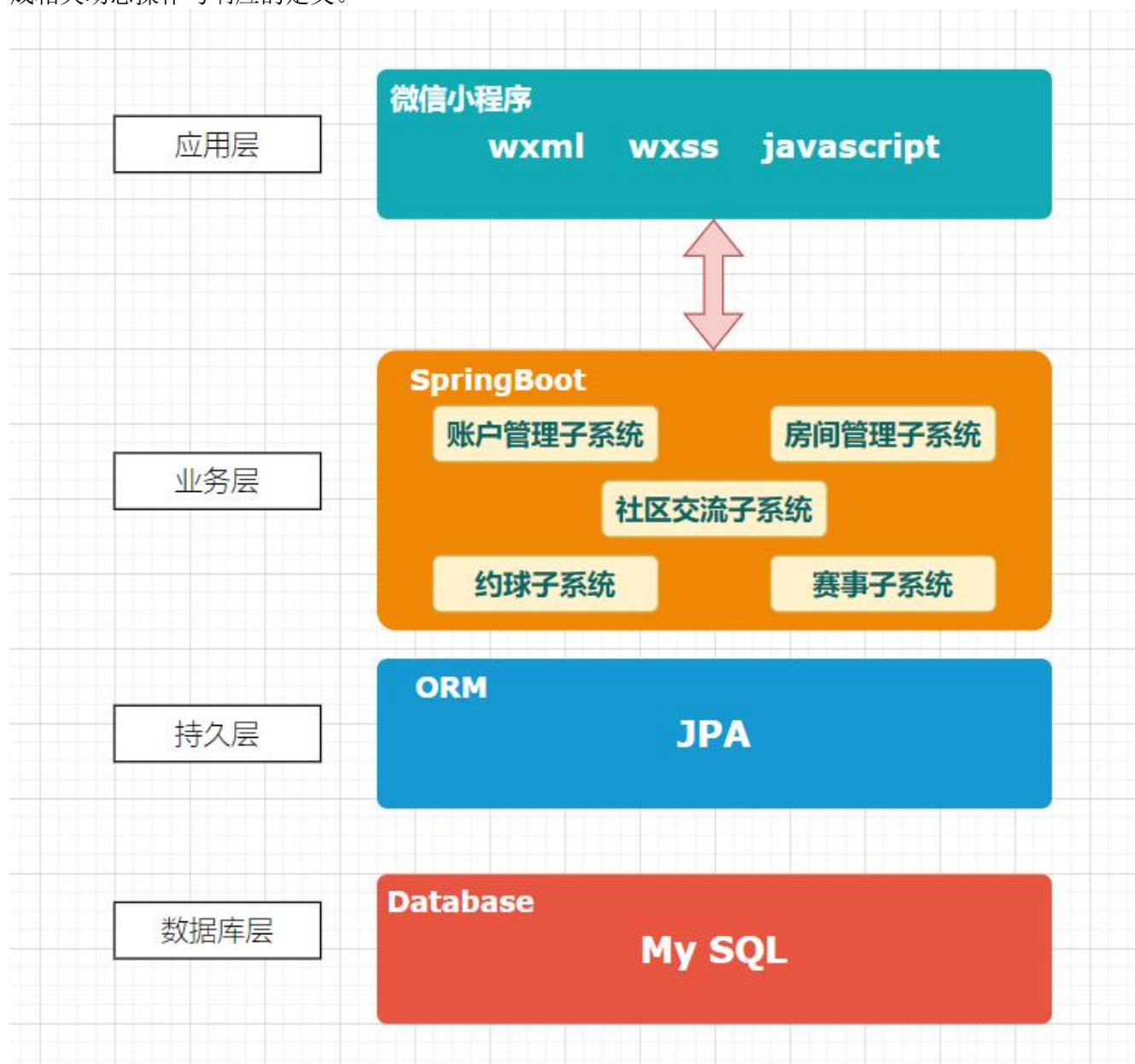
本次作业参考了原书第四章“域建模”以及第五章“需求模型的延伸”，在第四章中我们学习了头脑风暴法以及名词技术，为我们类模型以及成员函数的设计与命名提供了可遵循的模式；在第五章中我们学习了开发时序图的基本方法，以及确认其输入输出的有效方法，让我们的交互图设计更加严谨科学，其中关于用例描述的部分也帮助我们找到了修改用例的方向。

二. 架构设计

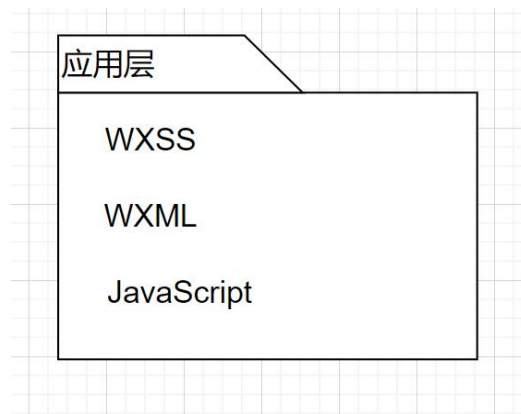
1. 系统级分析

系统整体采用分层架构，整体上分为应用层、业务层、持久层、数据库层。每一层功能相对独立。

- 数据库层采用 MySQL，具有轻量、便于操作的特点
- 持久层 ORM 映射采用 JPA，可以一键在后端生成实体类，并调用 JPA 提供的相关接口完成更加建议高效的数据操作
- 业务层采用 Spring Boot 框架，并划分五个子系统：账户管理子系统、社区交流子系统、房间管理子系统、约球子系统、赛事子系统。
- 前端采用微信小程序的形式，通过 wxml,wxss 文件修改内容样式，通过 javascript 完成相关动态操作与响应的定义。



2.应用层



WXML（WeiXin Markup Language）是框架设计的一套标签语言，结合基础组件、事件系统，可以构建出微信小程序页面的结构。

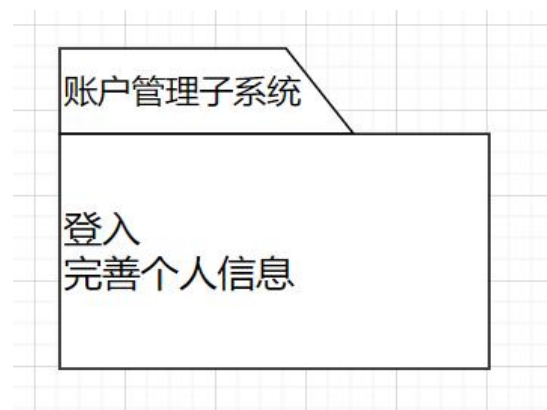
WXSS (WeiXin Style Sheets)是一套样式语言，用于描述 WXML 的组件样式。WXSS 用来决定 WXML 的组件应该怎么显示。

JavaScript（简称“JS”）是一种具有函数优先的轻量级，解释型或即时编译型的编程语言。定义网页如何相应用户的操作。

3.业务层

业务层展现了本系统的基本功能模块，包括以下五个子系统。

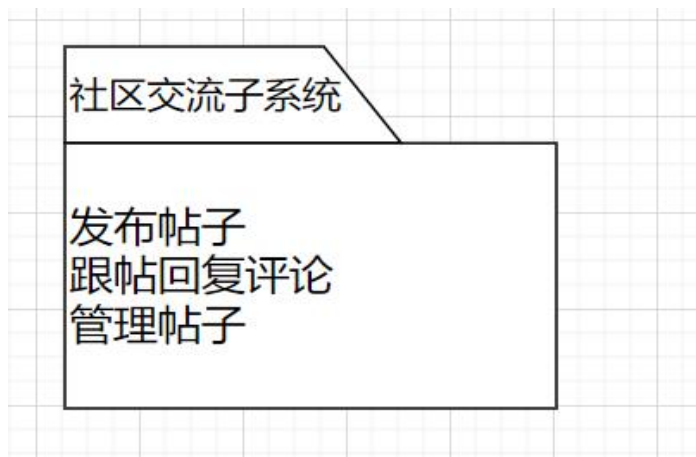
a. 账户管理子系统



登入：本功能允许用户通过同济大学统一身份认证平台，在输入正确的学号和密码后登入本系统并可以进行之后的完善个人信息、约战等操作。

完善个人信息：本功能允许用户在成功登入之后进行账户的个人信息完善。用户将输入昵称、个人简介、性别、运动水平、手机号码。这些输入的内容将被存储在系统数据库中并将展示部分信息给其他用户。

b. 社区交流

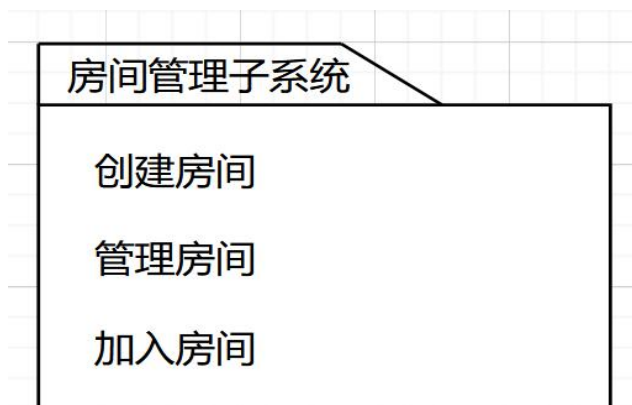


发布帖子：本功能允许所有用户在社区交流板块的任意部分创建并发布新的帖子，在帖子里可以输入文字，附上图片、链接、文件等。当帖子发布者拥有对自己帖子的管理权限，如：删除不友好回帖，修改原贴内容等。

跟帖回复评论：本功能允许用户在其他用户已发布帖子的下面进行跟帖回复。同样可以输入文字，附上图片、链接、文件等。

管理帖子：本功能允许管理员对社区交流板块内的所有帖子及其回帖进行管理，管理员有权限删除帖子、对帖子内容进行修改、对不良帖子发布者进行警告或封号处理。

c. 房间管理子系统

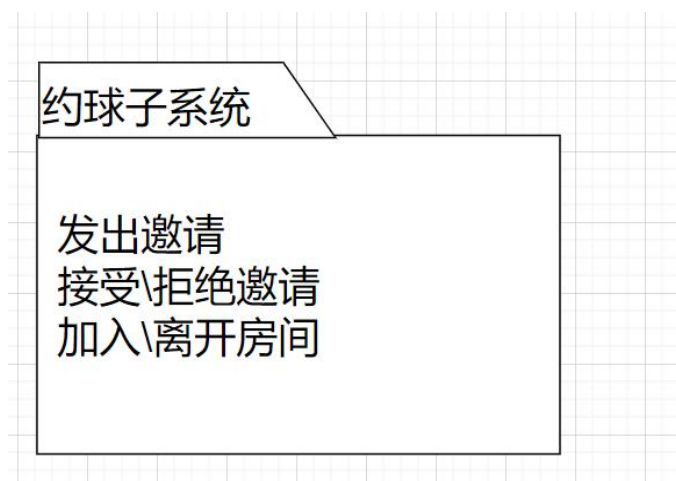


创建房间：本功能允许所有用户创建一个新房间，填写房间相关信息，设置房间基本信息。

管理房间：本功能允许房主修改房间的相关信息，对房间内的成员进行移出，对已创建的房间进行解散。

加入房间：本功能允许所有用户查看房间，可以通过相关条件进行筛选或者根据房间门和房主名进行搜索来进行，加入房间和退出房间。

d. 约球子系统

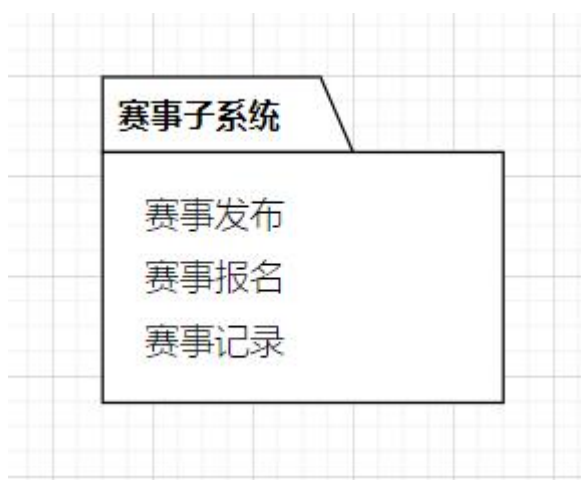


发出邀请：本功能允许房主向其他用户发出约球或者约战邀请。如果被邀请人同意接受邀请，则可以实现双人至多人约球。

接受\拒绝邀请：当用户面对他人的邀请时，可以自主选择接受或者拒绝邀请。如果接受邀请，则可以加入房间。如果拒绝邀请，则可以输入就拒绝原因。

加入\离开房间：房主和其他用户在被授权的情况下自由加入和离开房间。如果房主离开房间，则房间解散。

e. 赛事子系统

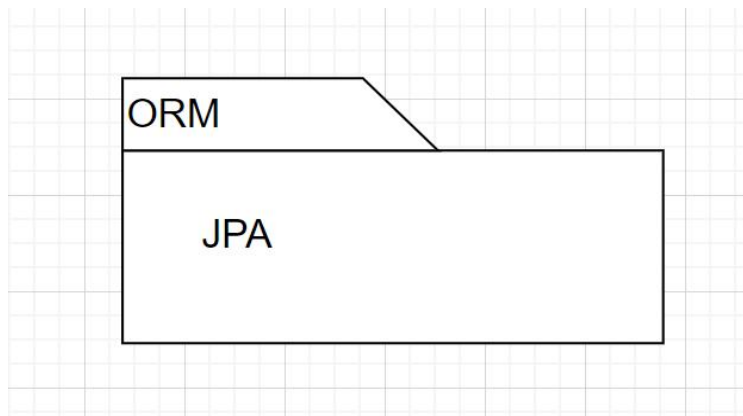


赛事发布：本功能允许拥有管理权限的用户进行赛事信息的发布，发布后的赛事信息将展示在比赛信息页面。

赛事报名：本功能允许普通用户进行赛事信息的查看与报名，报名后的个人信息将经过赛事举办者审核后加入比赛信息页面。

赛事记录：本功能主要实现了系统在赛事进行过程中对一系列实时信息（比分、比赛双方个人信息）以及比赛结果的记录。

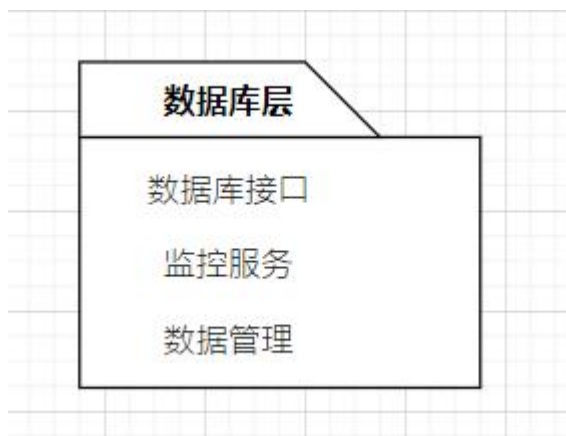
4.持久层



ORM: Object Relational Mapping 的缩写，为“对象关系映射”框架。ORM 框架是为了解决面向对象与关系型数据库中数据类型不匹配的技术，通过描述 Java 对象与数据库表之间的映射关系，自动将 Java 应用程序中的对象持久化到关系型数据库的表中。

JPA : Java Persistence API: JAVA 持久层应用接口。定义了一组可以由任何工具或框架实现的概念。

5.数据库层



数据库接口: 为实现业务逻辑以及调用前端提供后端数据库的接口。

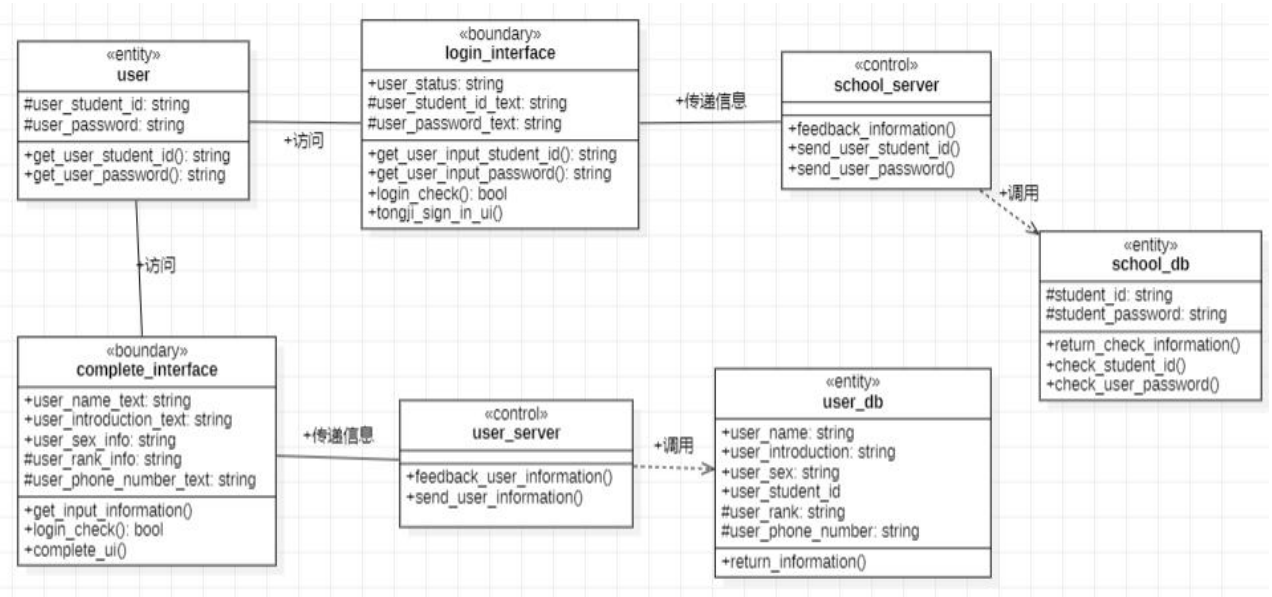
监控服务: 提供实时的业务性能以及安全监控。

数据管理: 主要负责数据库的访问与数据库的修改。

三. 设计模型

1. 用户

a. 类图



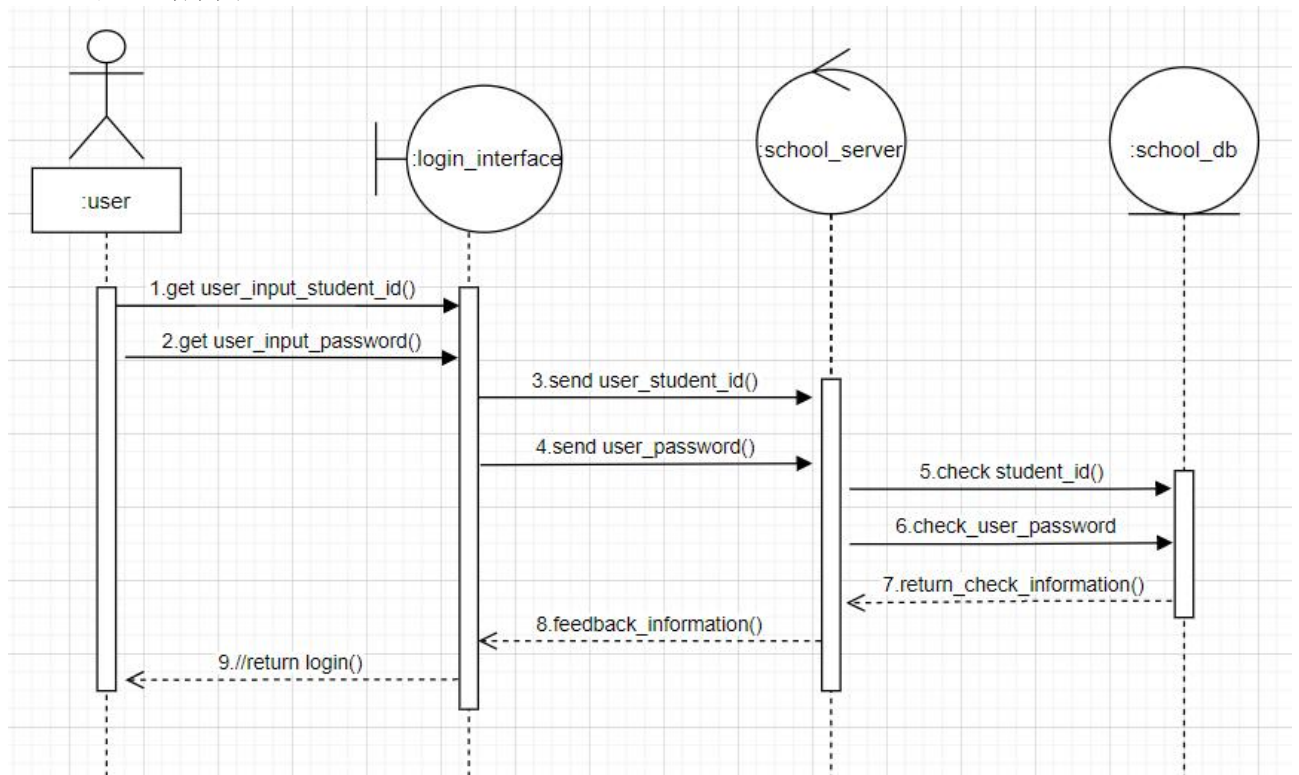
A. **user** 类中有用户的学号、用户密码这两个属性。学号由纯数字组成，密码由数字和字符组成，因此采用 **string** 数据类型。且为了保障用户的账号安全，学号和密码只能有其子类直接访问，所以将访问权限设置为 **protected**。**user** 实体可以分别访问登入和完善个人信息界面。

B. 其中 **login_interface** 类将直接调用同济大学统一身份认证平台，利用 **school_server** 即学校的服务器进行调用学校数据库进行判断用户输入的学号和密码是否正确。如果正确，则用户登录成功，并可以进行完善个人信息的操作。

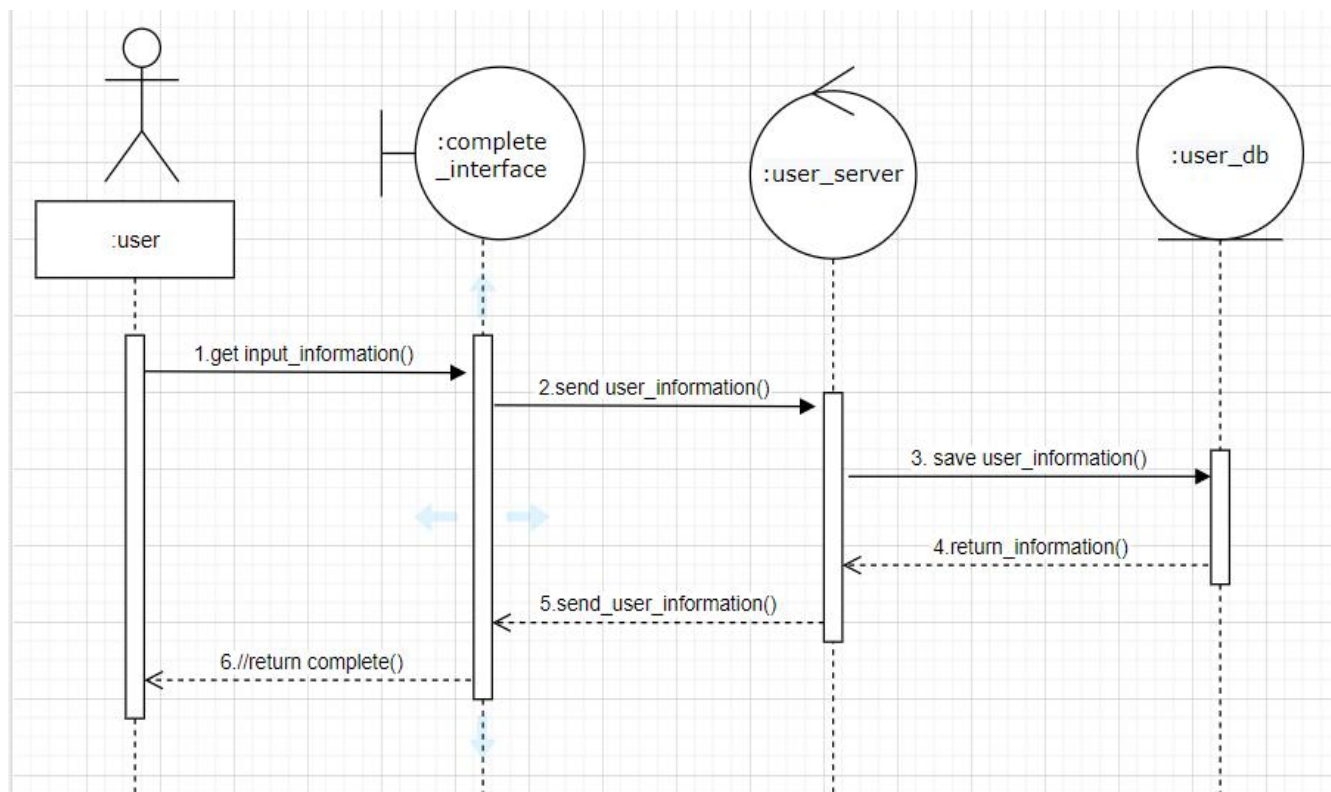
C. 在用户成功登入之后将可以访问完善个人信息界面对应的 **complete** 类，用户可以依次填入自己的昵称、个人介绍和性别并对外公开。可以通过 **user_server** 即用户服务器向用户数据库中存储相关信息。在 **user_db** 类即程序数据库中采用 **string** 数据类型进行存储昵称、自我介绍等，访问权限为 **public**。而对于较为隐私的个人排位情况以及电话号码访问权限设为 **protected**。

b. 时序图

登入时序图

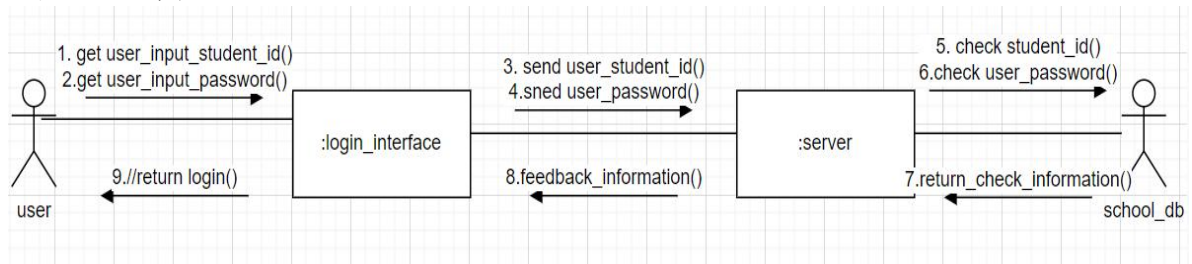


完善个人信息时序图

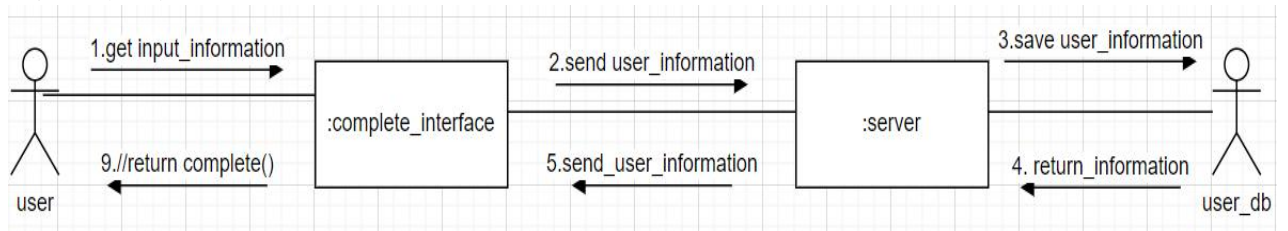


c. 通讯图

登入通讯图

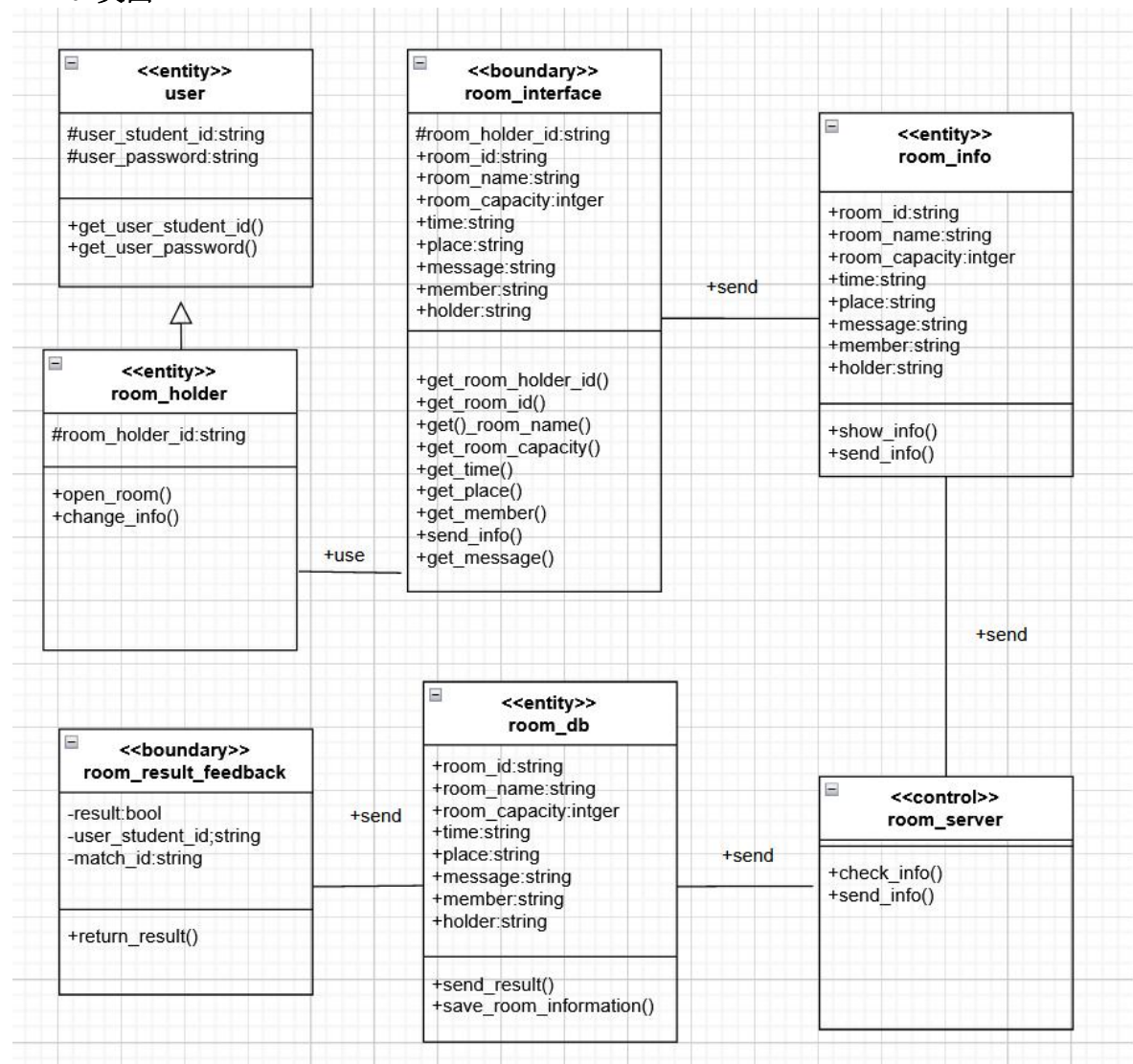


完善个人信息通讯图



2. 房间的创建与管理

a. 类图



A. user 类表示所有用户个体（包括房主），其中三个属性：user_student_id 表示学号，user_name 为用户名，user_password 为用户密码，为保护隐私，前两者均设为 protected；成员函数 get_user_student_id 和 get_user_password 分别用于获取用户学号和密码用于登录操作。

B. room_holder 类继承自 user 类，代表房主，拥有 room_holder_id 的唯一标识，可以创建房间和管理房间，对应的函数为 open_room(), change_info()。

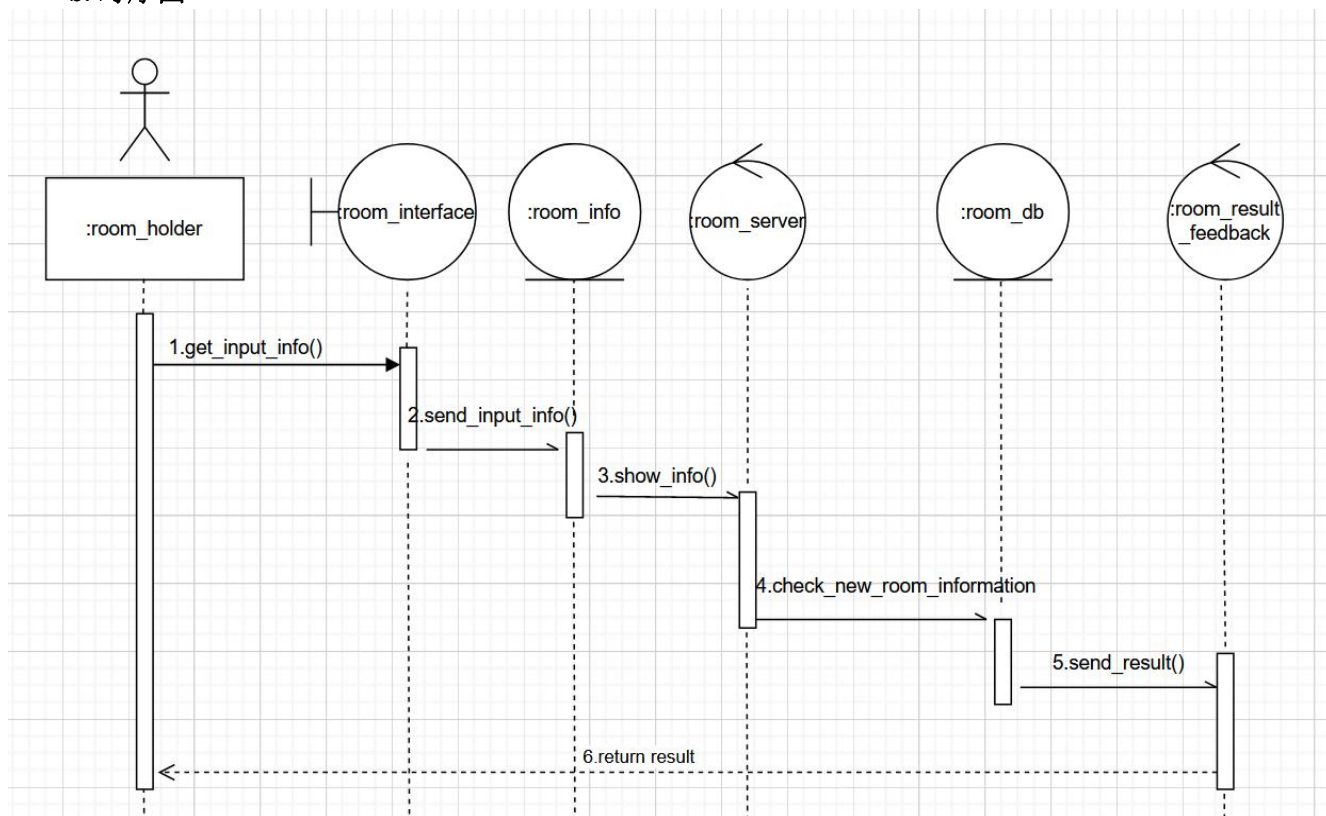
C. room_interface 类为房主进行操作的接口，操作类型通过 operation_kind 进行区别是管理房间或创建房间。该接口可以通过 get_input_info 获取房主输入的比赛信息，并通过 send_input_info 进行信息传递。

D. room_info 类展示所有已有房间的信息，包括 id，名称，容量、时间地点、简介、人员、创建者等。

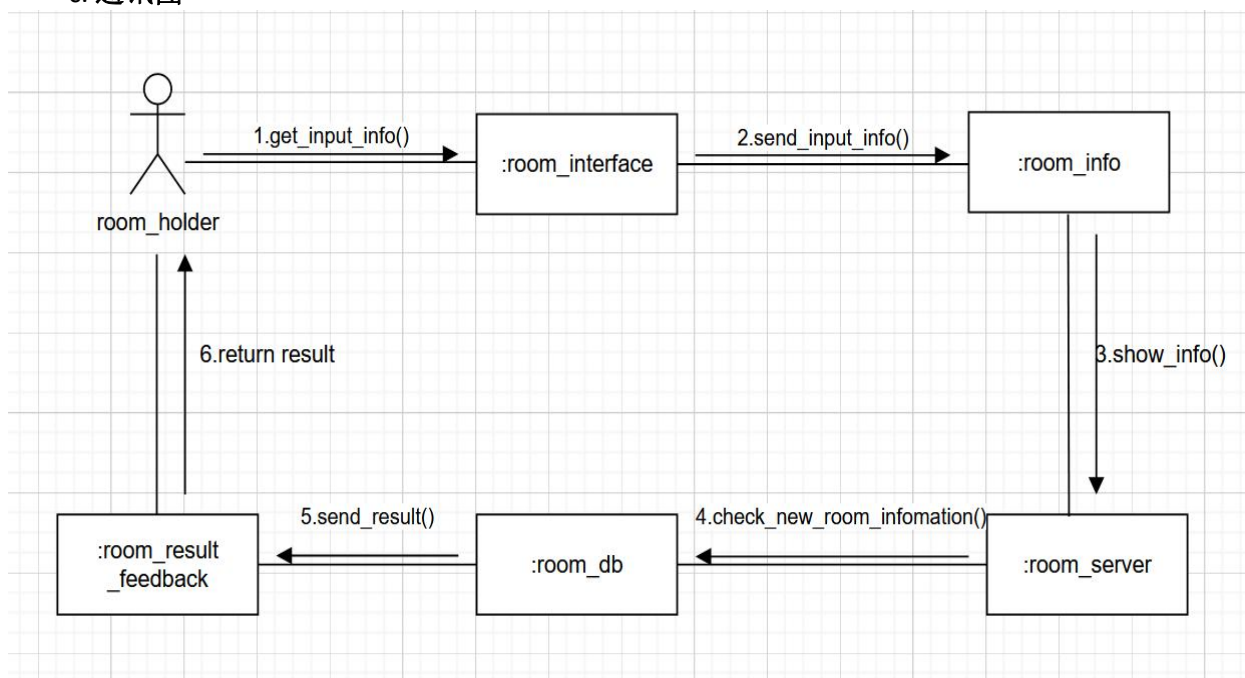
E. room_server 类为控制类，包含三个函数 check_info(), send_info(), send_result()。功能分别为检查输入信息是否与 room_db 信息矛盾，向 room_db 传输信息，向 result_feedback 传递结果。

F. result_feedback 类为操作结果，向房主显示操作成功与否。

b. 时序图

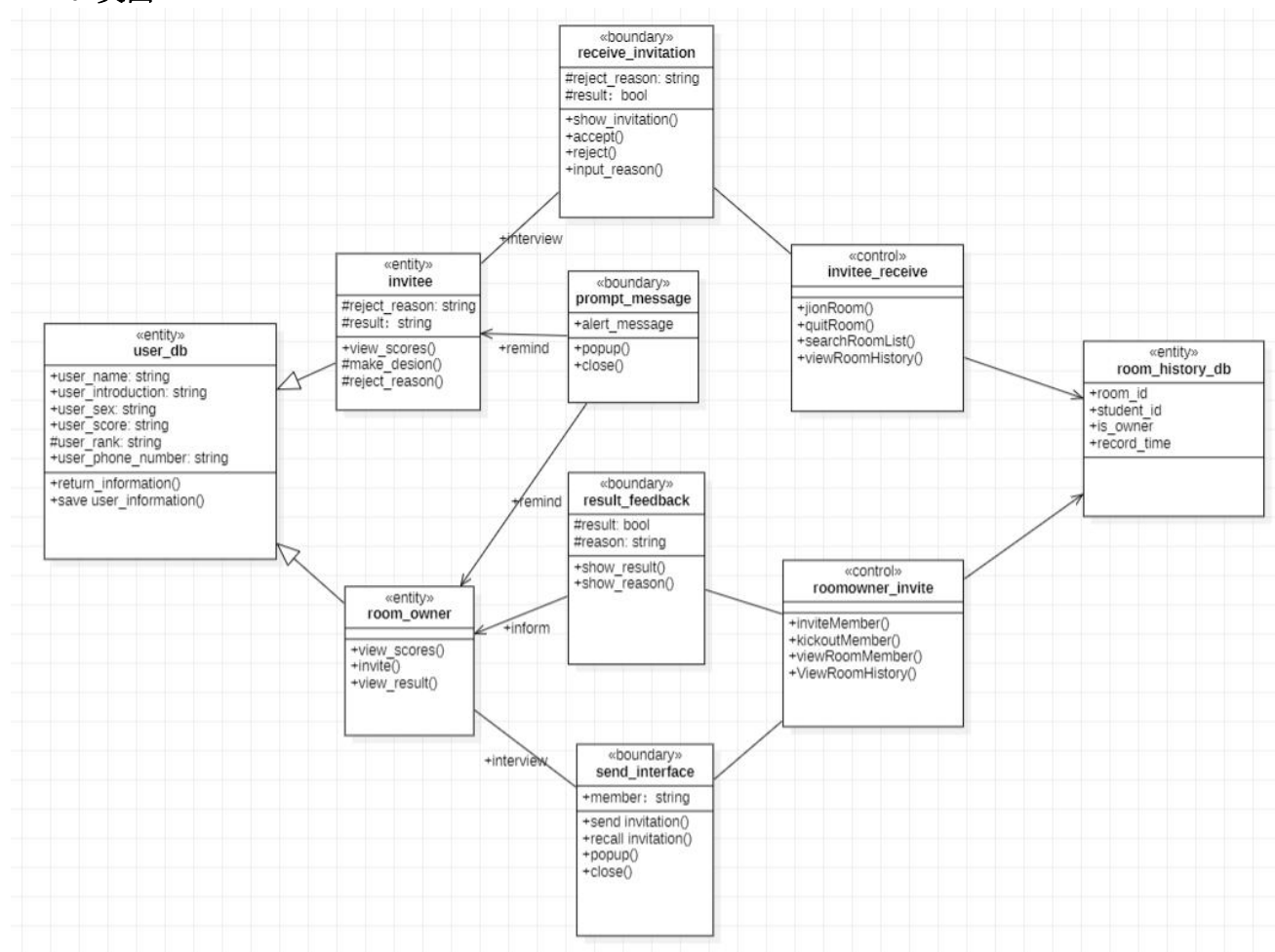


c. 通讯图



3. 约球/约战

a. 类图



A. `user_db` 类中有用户的姓名、简介、积分、排名等这属性。姓名由数字、字符等构成，介绍由文字和数字构成，因此均采用 `string` 数据类型。且为了保障用户的隐私，排名只能有其子类直接访问，所以将访问权限设置为 `protected`。

B. `room_owner` 类表示房主，继承自 `user_db` 类。`room_owner` 类中有 `view_scores`、`invite`、`view_result` 三个成员函数。房主可以通过调用 `view_scores` 函数来查看其他人的积分，依次来评估其他人的实力。房主可以调用 `invite` 函数来对其他人发出邀请。房主还可以通过 `view_result` 函数来查看邀请的结果是否成功。

C. `invitee` 类表示被邀请人，同样继承自 `user_db` 类。被邀请人可以查看他人的积分，决定是否接受邀请，如果选择拒绝可以向房主发送拒绝的原因。因为保护个人隐私，被邀请人的决定以及拒绝原因都是被设定为 `protected`，只有邀请人和管理员可以查看。

D. `send_interface` 类为邀请发出界面，房主在邀请过程中会访问该界面。房主可以通过这个界面进行发出邀请以及撤回邀请的操作。

E. `receive_interface` 类为收到邀请界面。被邀人收到其他人的约战邀请时该界面会弹出。被邀人在该界面上选择是否接受邀请并输入原因。

F. `prompt_message` 类为一个提醒弹窗，系统检查到两人积分相差较大时将会 `send_interface` 和 `receive_interface` 界面之上弹出提醒，建议房主取消邀请和受邀人拒绝邀请。

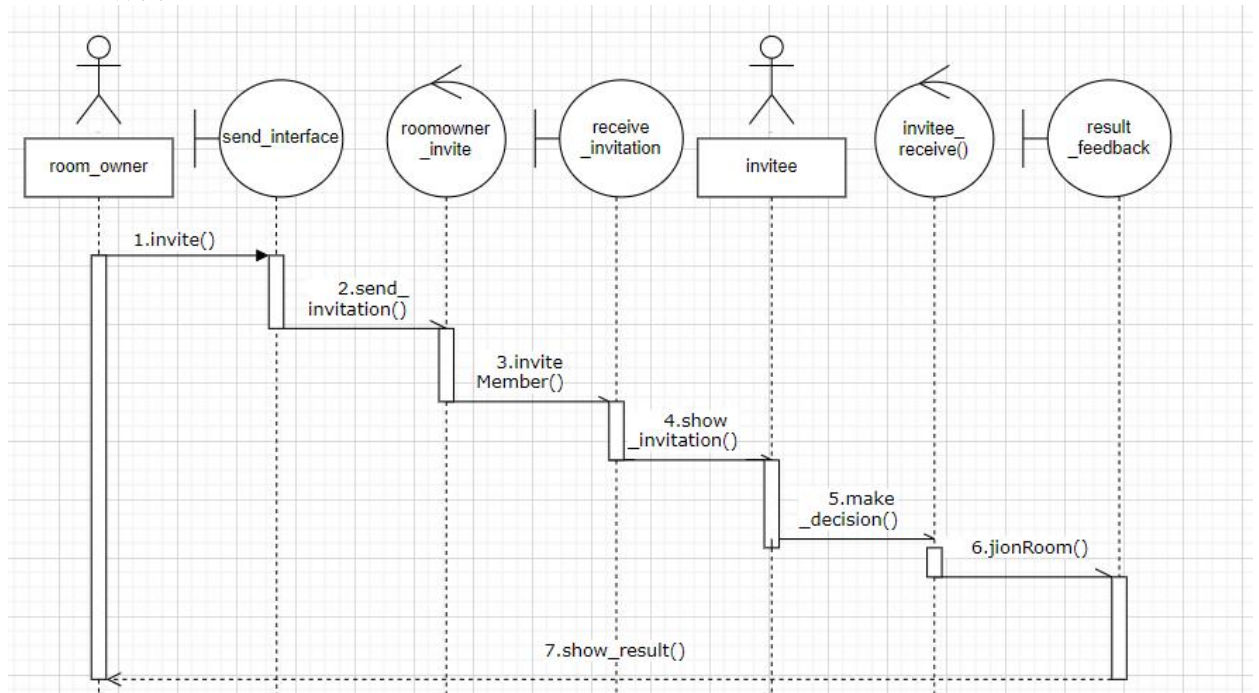
G. `result_feedback` 类为邀请结果界面。当邀请结出来后，房主可以调用该界面查询邀请结果，如果邀请失败还可以查看失败原因。

H. invitee_receive 类为一个 control 类。包含了一共四个函数：
jionRoom(),qiutRoom(),searchRoom(),viewRoomHistory()函数。jionRoom()和 qiutRoom()
分别为加入房间和退出房间。searchRoom()可以推现有的房间进行搜索。通过可以
viewRoomHistory()查看房间历史信息。

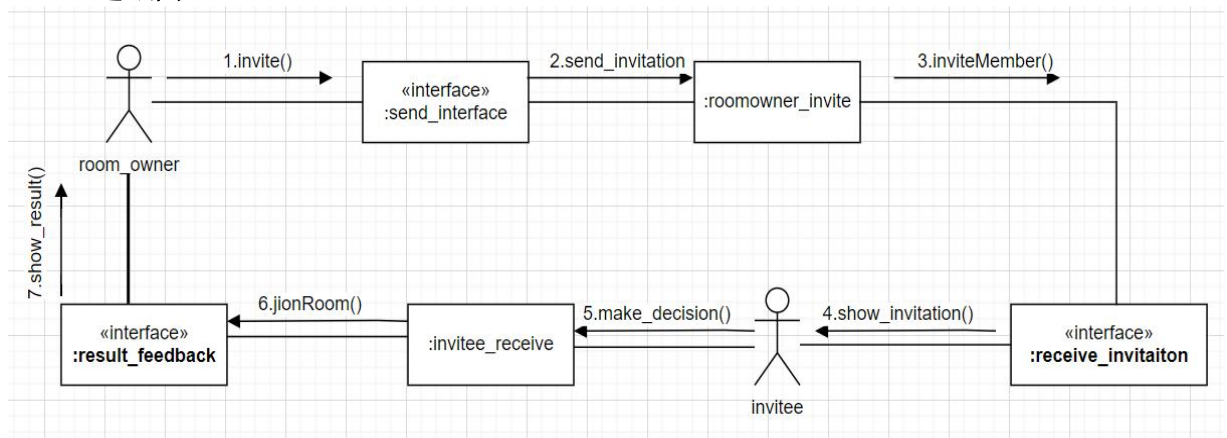
I. roomowner_invite 也是一个 control 类。公包含四个函数：inviteMember()、
kickoutMember()、viewRoomMember()、ViewRoomHistory()。inviteMember()和
kickoutMember()分别为邀请成员和踢出成员。viewRoomMember()和 ViewRoomHistory()
分别为查看房间成员和查看房间历史信息。

J. room_history_db 是 room_db 的继承类。主要对房间信息的变化进行存储。

b. 时序图

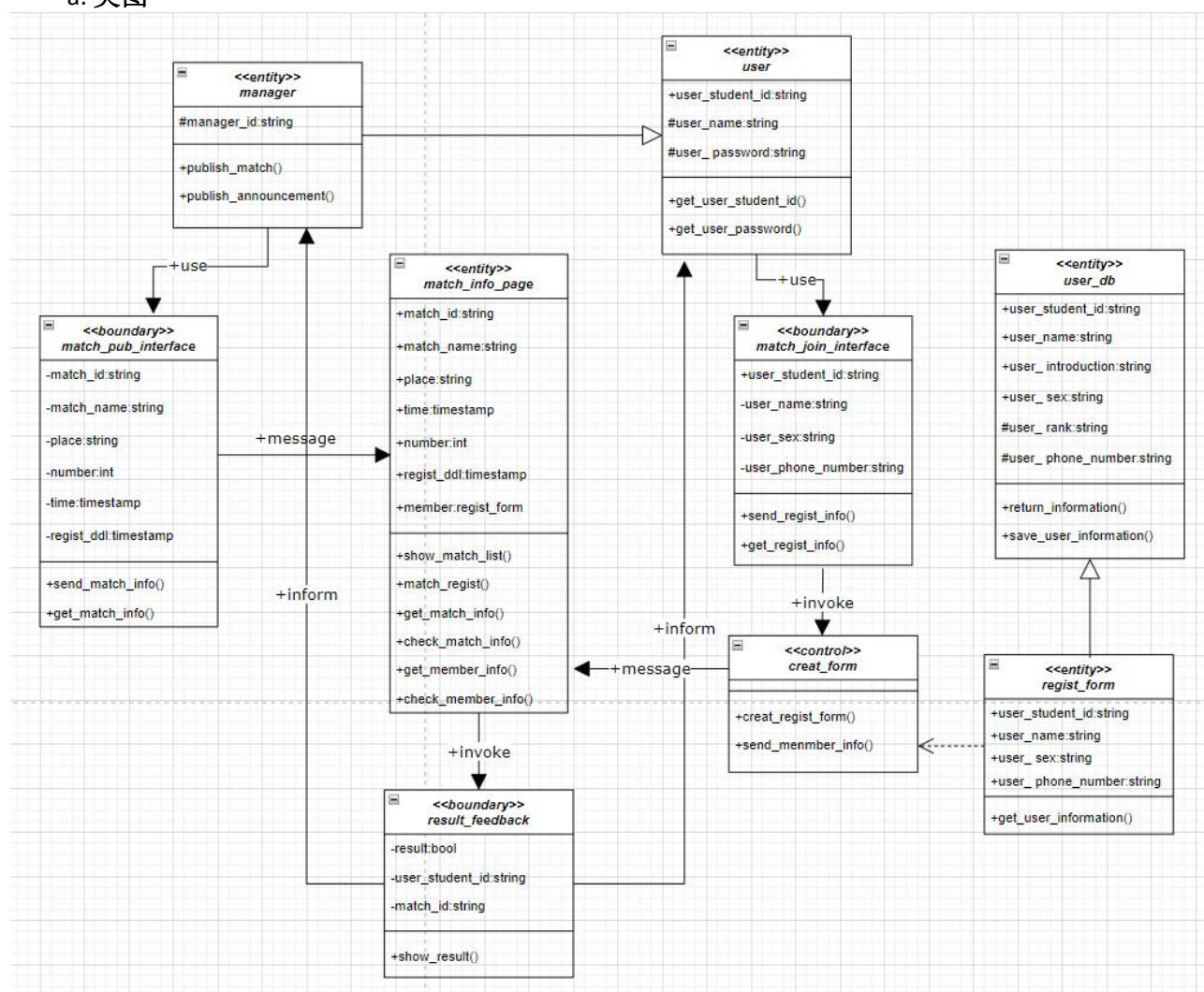


c. 通讯图



4. 赛事发布与报名

a. 类图



A. **user** 类表示所有用户个体（包括管理员），其中有三个属性：**user_student_id** 表示学号，**user_name** 为用户名，**user_password** 为用户密码，为保护隐私，前两者均设为 **protected**；成员函数 **get_user_student_id** 和 **get_user_password** 分别用于获取用户学号和密码用于登录操作。

B. **manager** 类继承自 **user** 类，代表管理员，拥有 **manager_id** 唯一标识，可以进行发布通知、发布赛事的操作，对应的成员函数为 **publish_match** 和 **publish_announcement**。

C. **match_pub_interface** 类为管理员发布赛事的接口，每一场赛事用 **match_id** 进行区分，赛事信息包括赛事名称、时间、地点、报名截止日期、赛事可容纳人数，该接口可以通过 **get_match_info** 获取管理员输入的赛事信息，并通过 **send_match_info** 进行信息传递。

D. **match_join_interface** 类为用户报名赛事的接口，用来存储用户的报名信息，包括学号、用户名、性别、电话号码，该接口可以通过 **get_regist_info** 获取用户输入的信息，并通过 **send_regist_info** 进行信息传递。

E. **creat_form** 类为控制类，负责把用户报名信息聚合为元组进行登记，依赖于 **regist_form** 类，该类继承自 **user_db** 类，聚合了其中的部分属性用来描述赛事报名信息。

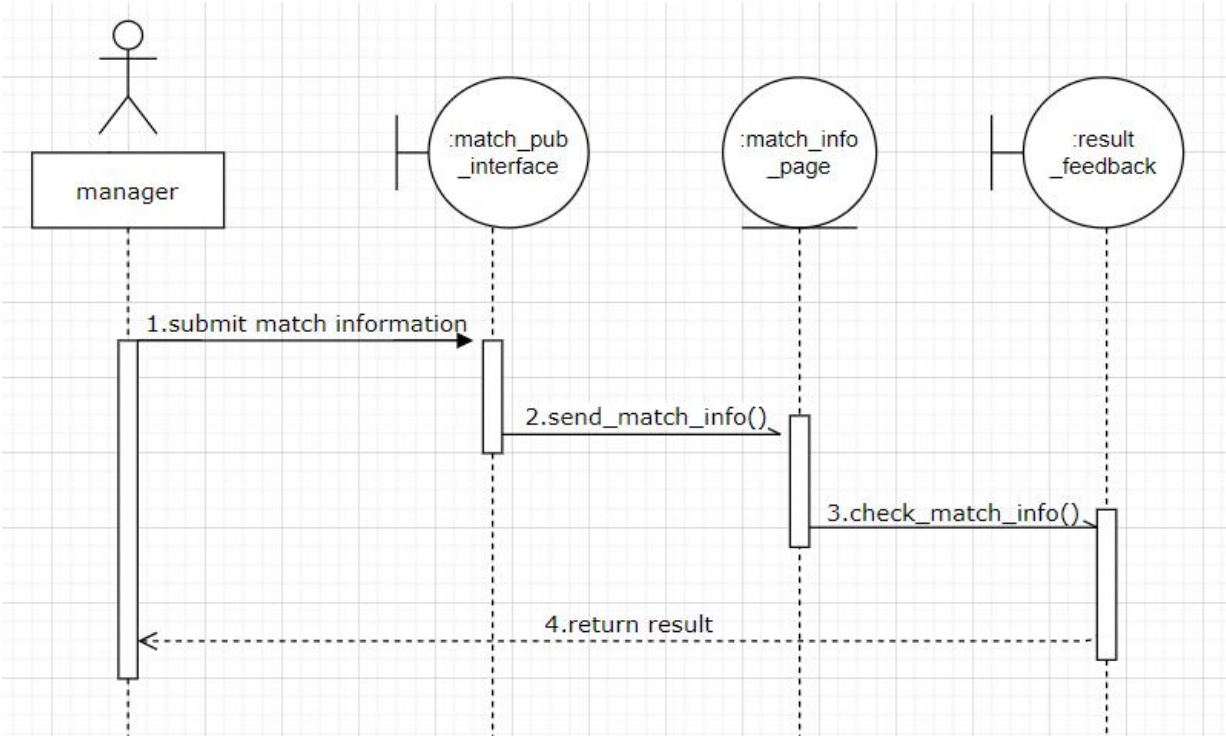
F. **match_info_page** 类为赛事信息页面实体，包含了一项赛事的基本信息以及参赛人员

的报名信息，不仅可以展示已有赛事信息，还可以进行赛事报名，除此之外，它还被用来检查赛事信息与报名信息的合理性。

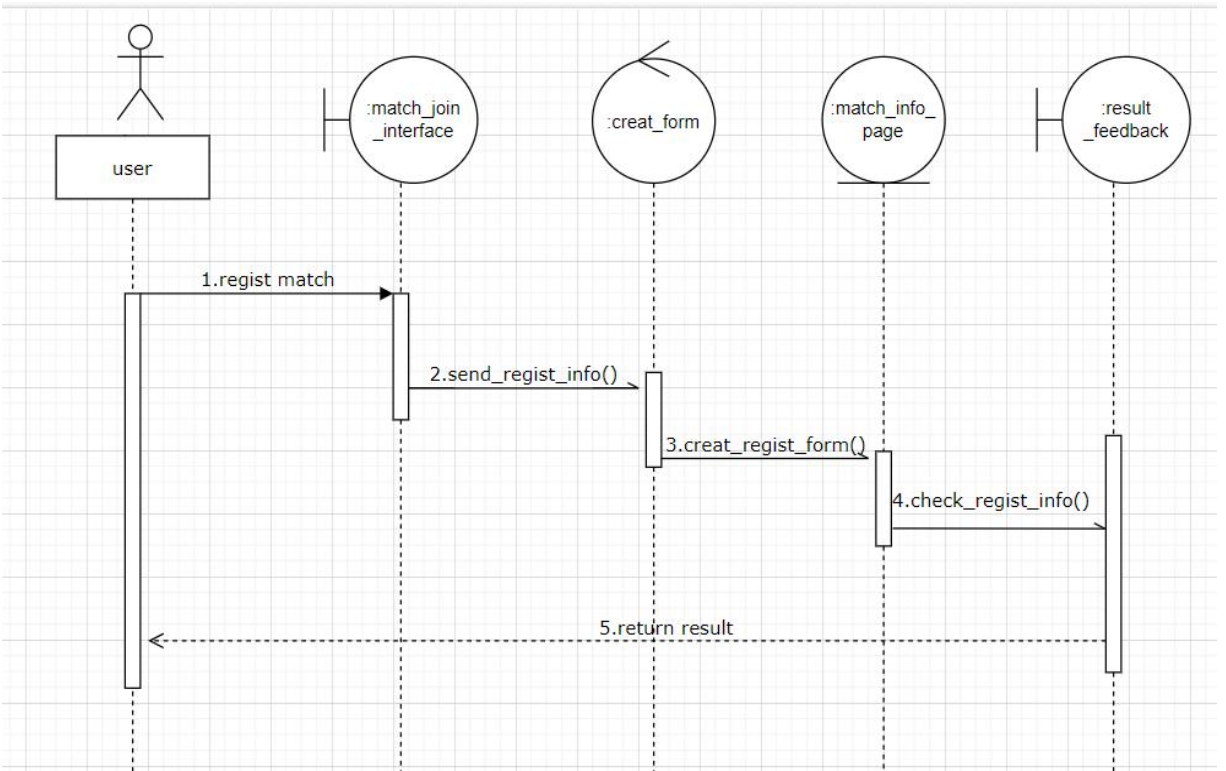
G. result_feedback 类为审核结果返回接口，系统审核后的结果通过该接口传递给相应用户/管理员。

b. 时序图

赛事发布时序图：

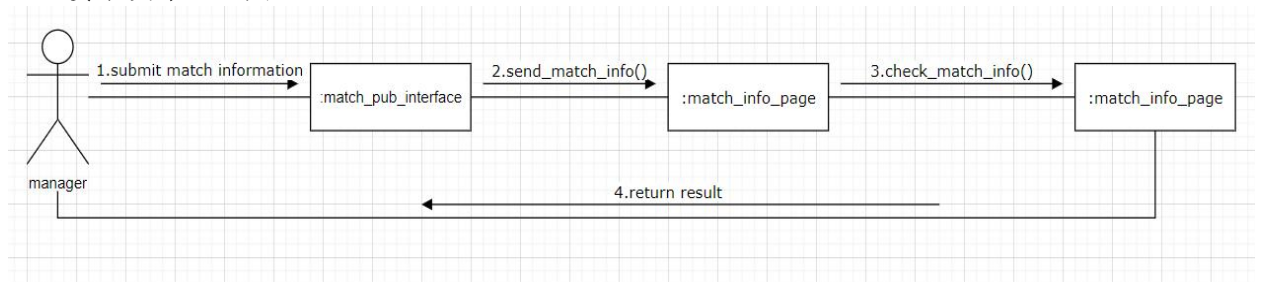


赛事报名时序图：

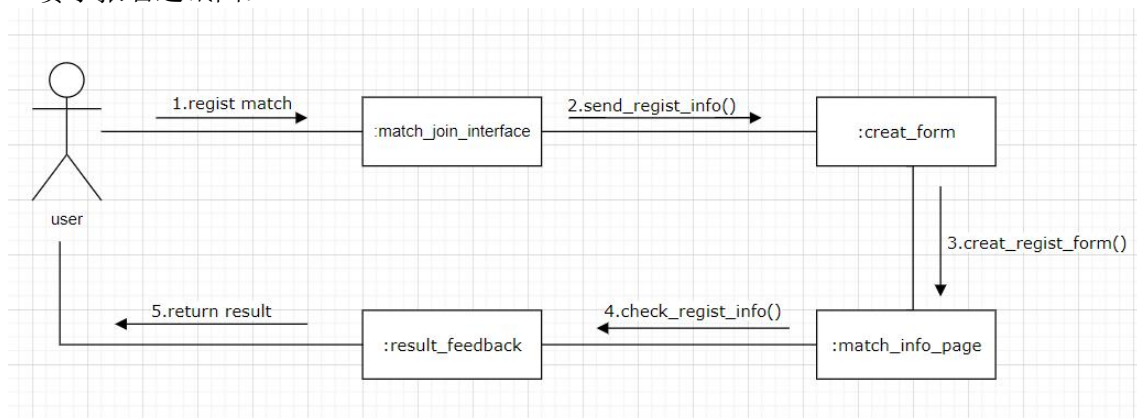


c. 通讯图

赛事发布通讯图：



赛事报名通讯图：



四. UI 界面设计

1. 登录界面:



注册

学号

验证码

新密码

确认密码

确认



学号

昵称

密码

确认密码

注册

2. 主界面：



3. 创建房间：

开始时间 日 时 分

持续时间 小时 分钟

校区 四平校区

场地 1

人数上限 2

备注

创建房间

4. 房间信息页面：

选择校区 ▾

选择日期 ▾

选择时间 ▾

筛选房间

时间: 4月 20 日9:00~10:00
地点: 嘉定校区体育馆乒乓球房
人数: 3/5
备注: 这个房主很懒, 没有留下任何备注

加入

时间: 4月 25 日3:00~4:00
地点: 四平校区体育馆乒乓球房
人数: 3/4
备注: 1缺1! 高手快来!!!

已加入

时间: 5月 3 日18:00~20:00
地点: 四平校区体育馆乒乓球房
人数: 3/5
备注: 这个房主很懒, 没有留下任何备注

加入

时间: 6月 5 日9:00~10:00
地点: 嘉定校区体育馆乒乓球房
人数: 2/2
备注: 乒乓球单挑王55场97胜率

加入

返回

5. 修改个人信息：

新昵称

上传头像

提交

返回

6. 约战:



挑战者

得分

 提交



被挑战者

 返回主界面

7. 发布比赛:

比赛名称

开始报名时间

日

▼

时

▼

分

▼

截止报名时间

日

▼

时

▼

分

▼

地点

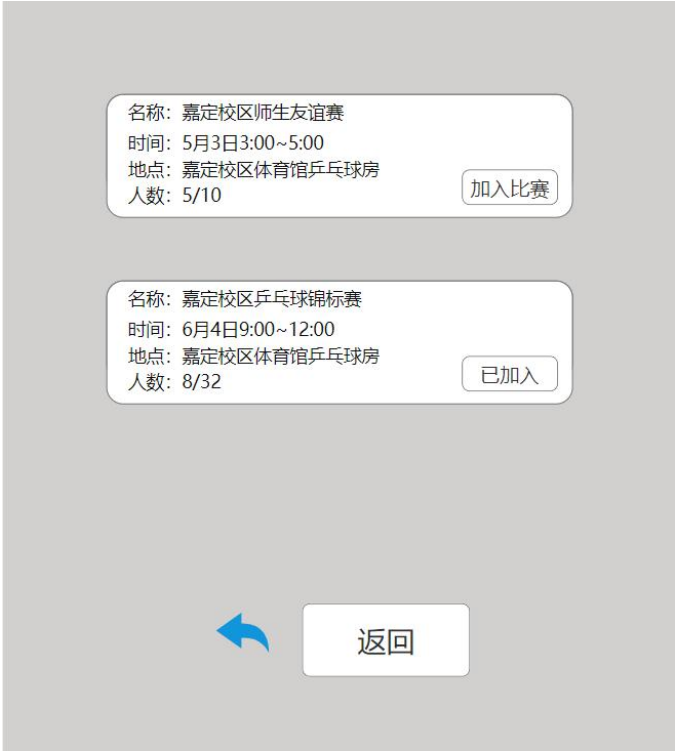
人数上限

比赛时间

发布比赛

取消

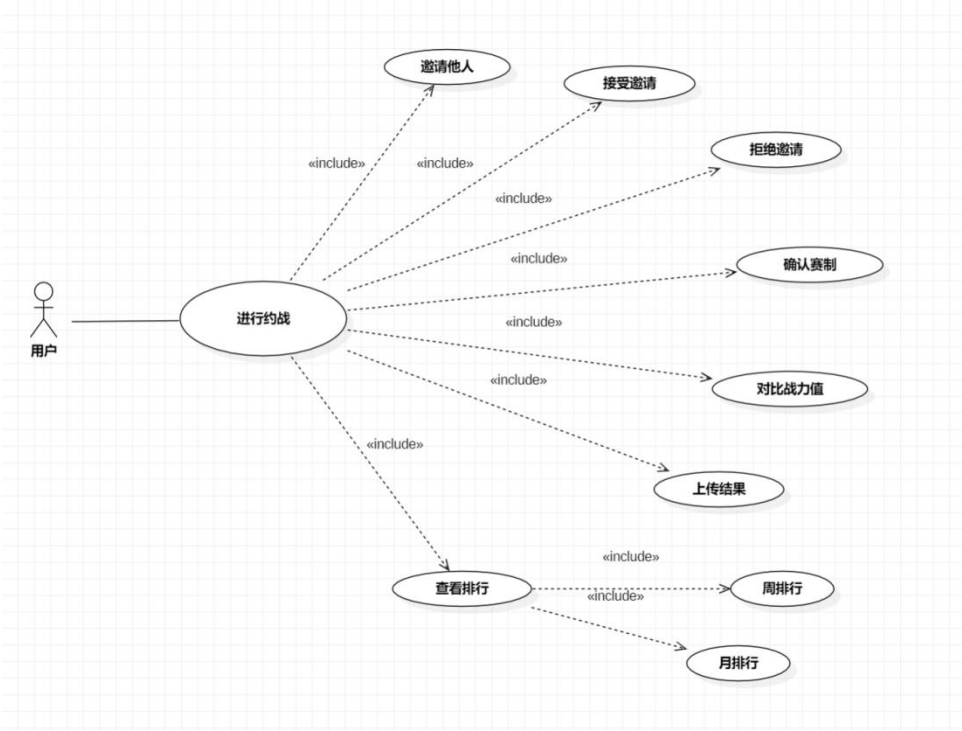
8. 加入比赛：



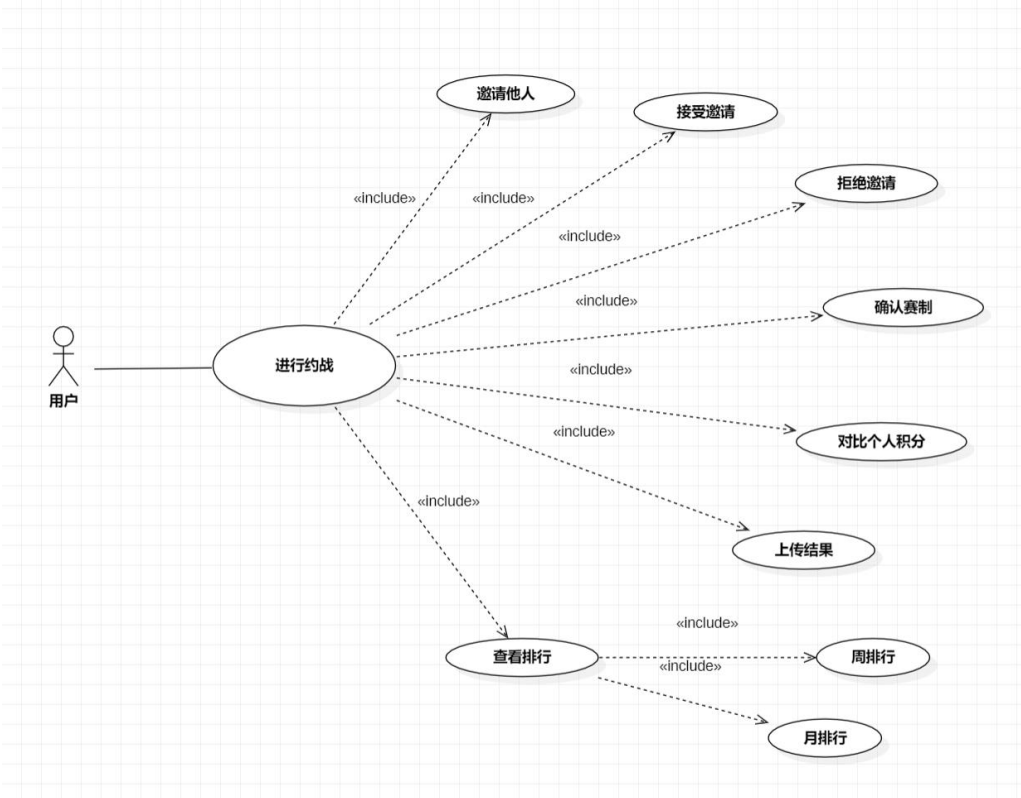
五. 部分用例模型的更新

1. 更改约战用例模型

更改内容：删去战力值的概念
更改前：



更改后：



更改原因：

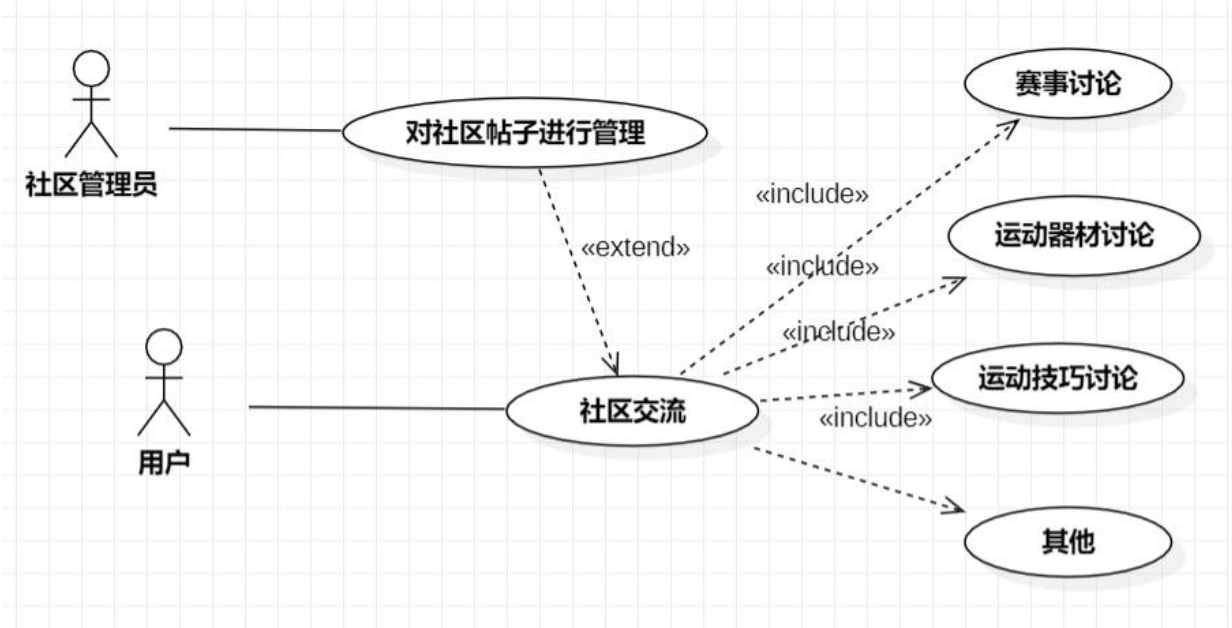
战力值与个人积分的概念产生重复，过于冗余。因此将战力值删去，只保留积分和排名，其中排名由积分决定。在约战时如果积分相差太大系统会提示不建议约战。

2. 细化社区交流模型

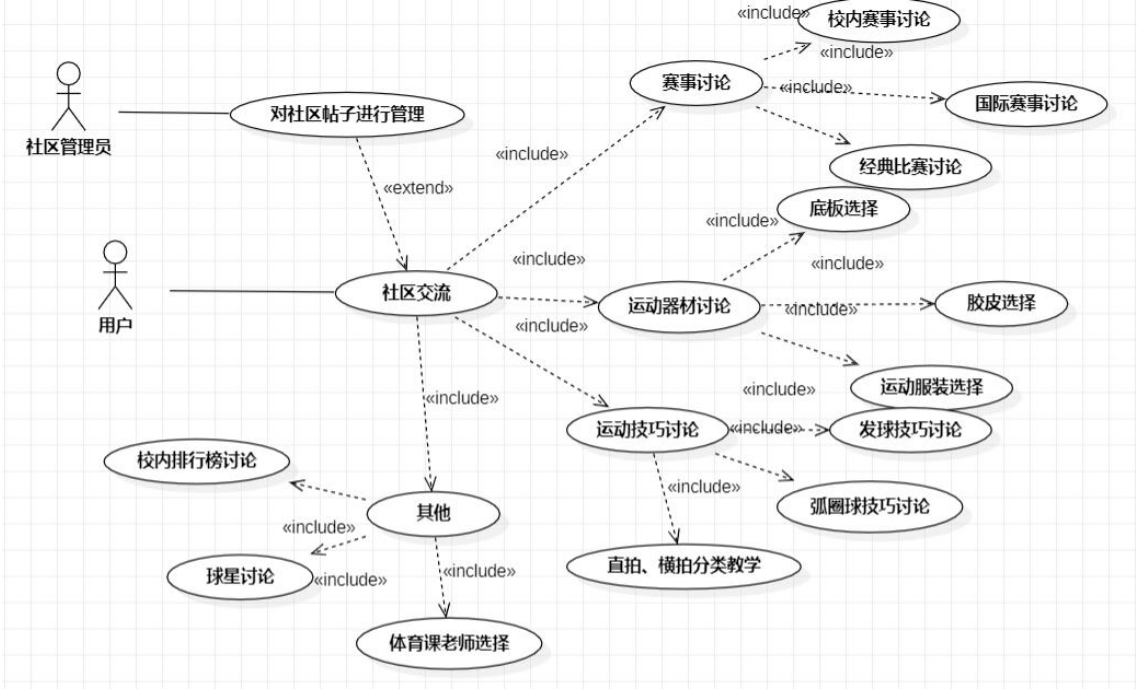
更改内容：

对社区交流用例细化，对之前四个分区模块进一步细化。

更改前：



更改后：



更改原因：

之前社区交流的内容过于单调，为了创建更加良好的社区生态，增加社区活力，促进成员活跃程度，对于社区讨论内容进行细化。

在赛事讨论模块，对于不同种类的比赛进行分类，校内赛事讨论能够让同学们互相交流比赛过程中的心得，让观看者与参赛者在讨论中共同成长；国际赛事讨论有助于扩展同学们的国际视野，在放松身心的同时感受国际一流水准的技巧；经典赛事则是选择一些十分经典的比赛单独供同学反复观摩与讨论，在经典中提升乒乓球水平。

在运动器材讨论模块，对于不同种类的运动器材进行分类。底板、胶皮、运动服装模块让同学们能够互相交流打球前的准备工作，对于刚刚入门的新手也能起到攻略的作用。

在运动技巧模块中增加发球技巧讨论、弧圈球技巧讨论、直拍横拍教学模块。增加了同学们寻找特定技巧时的方便性。

对之前的其他模块，则是扩展出校内排行榜讨论、球星讨论、体育课老师选择等模块。这是系统给出的示范内容，仅供参考，用户可以自由增加其他模块。

六. 团队分工

在完成本次任务的过程中，所有组员积极参与讨论，努力完成自己的任务。组员之间配合融洽，遇到问题及时沟通、共同解决。组内分工平均而明确，具体如下：

模块	分工
项目简介	由彭浩书写
架构设计	由所有组员共同设计与书写：系统级分析（范珑骁）、应用层（彭浩）、业务层（韩翼、杨正熹、范珑骁、胡鑫烨）、持久层（彭浩）、数据库层（范珑骁）

设计模型	由韩翼（用户）、胡鑫烨（房间）、杨正熹（约球）、范珑骁（赛事）分工完成
UI 界面设计	由彭浩绘制
用例模型更新	由韩翼、杨正熹进行用例图的修改，彭浩进行文档中相关 part 的书写