# 校园时间管理工具 需求获取报告

目录

[校园时间管理工具 需求获取报告 1](#_Toc1937069304)

[1. 引言 1](#_Toc471703808)

[1.1 编写目的 1](#_Toc1563759979)

[1.2 背景 2](#_Toc1209095067)

[1.3 定义 2](#_Toc1770523155)

[1.4 参考资料 2](#_Toc1649253253)

[2. 任务概述 3](#_Toc1427991342)

[2.1 目标 3](#_Toc2120729769)

[2.2 用户的特点 3](#_Toc1319138324)

[2.3 约定和假设 3](#_Toc136639840)

[3. 需求规定 3](#_Toc845772237)

[3.1 对功能的规定 3](#_Toc699727766)

[3.1.1 确认参与者 4](#_Toc704112190)

[3.1.2 确定用例 5](#_Toc1378682360)

[3.1.3 用例说明 5](#_Toc165873390)

[3.2 对性能的规定 5](#_Toc400291924)

[3.2.1 精度 5](#_Toc1787584264)

[3.2.2 时间特性要求 6](#_Toc632503518)

[4. 运行环境规定 6](#_Toc442574376)

[4.1设备 6](#_Toc1611667871)

[4.2 支持软件 6](#_Toc1090668286)

[4.3接口 6](#_Toc2088955657)

[4.4控制 7](#_Toc2015066043)

# 校园时间管理工具 需求获取报告

# 引言

## 1.1 编写目的

本需求获取报告目的在于：分析终端用户的需求，将其系统化、精确化、全面化。作为开发工作的基础。便于终端用户、开发人员的相互理解和交流，反映出课程设计所需要解决的问题，作为软件开发工作的基础和依据，以及最终确认测试和验收的依据。

预期读者：软件设计者和测试者。

## 1.2 背景

说明：

1. 待开发的软件系统名称：校园时间管理工具
2. 本项目的任务提出者：软件工程课程设计教师
3. 本项目的任务开发者：许羽晟
4. 用户：希望进行时间管理的师生
5. 实现该软件的计算中心或者计算机网络：个人PC或是云服务器

## 1.3 定义

|  |  |
| --- | --- |
| 名称 | 说明 |
| 学生账号 | 每个学生专属的账号，和学生信息绑定 |
| 学生信息 | 学生的基础信息，学号，班级，专业，联系方式，教师、同团队可查看 |
| 任务团队 | 一个老师和多个学生组成的一个组，老师可以向学生发布任务，  或是多个学生之间形成的组，学生之间可以互相发布任务 |
| 发布任务 | 老师或者互为团队学生向指定或所有成员发布一项任务 |
| 完成任务 | 被发布任务者认为自己完成了任务，将它移入已完成分组 |
| 任务评论 | 所有此任务参与者可以在下面评论，或是队内根据分配关系互评 |
| 删除任务 | 发布者认为不妥，主动删除 |
| 加入团队 | 根据唯一标识团队码加入团队 |
| 课程团队 | 课程类型团队，所有课程组的任务发布页面在同一页面 |
| 学生团队 | 学生之间的团队，每个团队的任务发布页各不相同 |
| 账户管理 | 管理员对所有账户进行管理 |
| 完成统计 | 可视化地显示所有任务共同参与者的完成情况 |
| 创建团队 | 创建一个任务团队，同时获得唯一团队标识码可以邀请其他用户加入 |

## 1.4 参考资料

|  |  |  |
| --- | --- | --- |
| 编号 | 资料名 | 作者 |
| 1 | 《软件工程（第三版）》 | 齐治昌等 |
| 2 | 基于JAVA的博客需求获取报告 | 佚名 |
| 3 | 基于PYTHON的博客需求获取报告 | 佚名 |

# 任务概述

## 2.1 目标

任务分配，时间管理，是人生大事之一，Microsoft To-Do，Trello等等工具都专注于提升用户的专注力，但是他们并不是很适合学生使用，尽管提供了丰富的模版，但是它们本质都是一堆人处于一个团队，难以形成一个一对多的关系，也难以将多个课程的任务集结在一个页面上，将它和学生团队，该课设旨在提供一种校园任务管理的可行化解决方案，将团队氛围协作团队(学生团队互助)和课程团队(教师像学生发布任务)，让学生能够在学习和生活更合理的安排自己的生活，减少信息混乱对学生学习积极性的影响。

## 2.2 用户的特点

本系统的终端用户分为：

教师和学生，其中教师可以创建课程团队，作为团队中的主要任务发布者，学生可以创建学生团队，所有学生都可以在团队中互相发布任务，

管理员，可以对学生账号信息进行编辑，权限分配，它负责管理请求数据库中的各种变更请求。

## 2.3 约定和假设

开发经费：无需费用

开发期限：课设进行期间(2020.10 ～ 2020.12)

# 需求规定

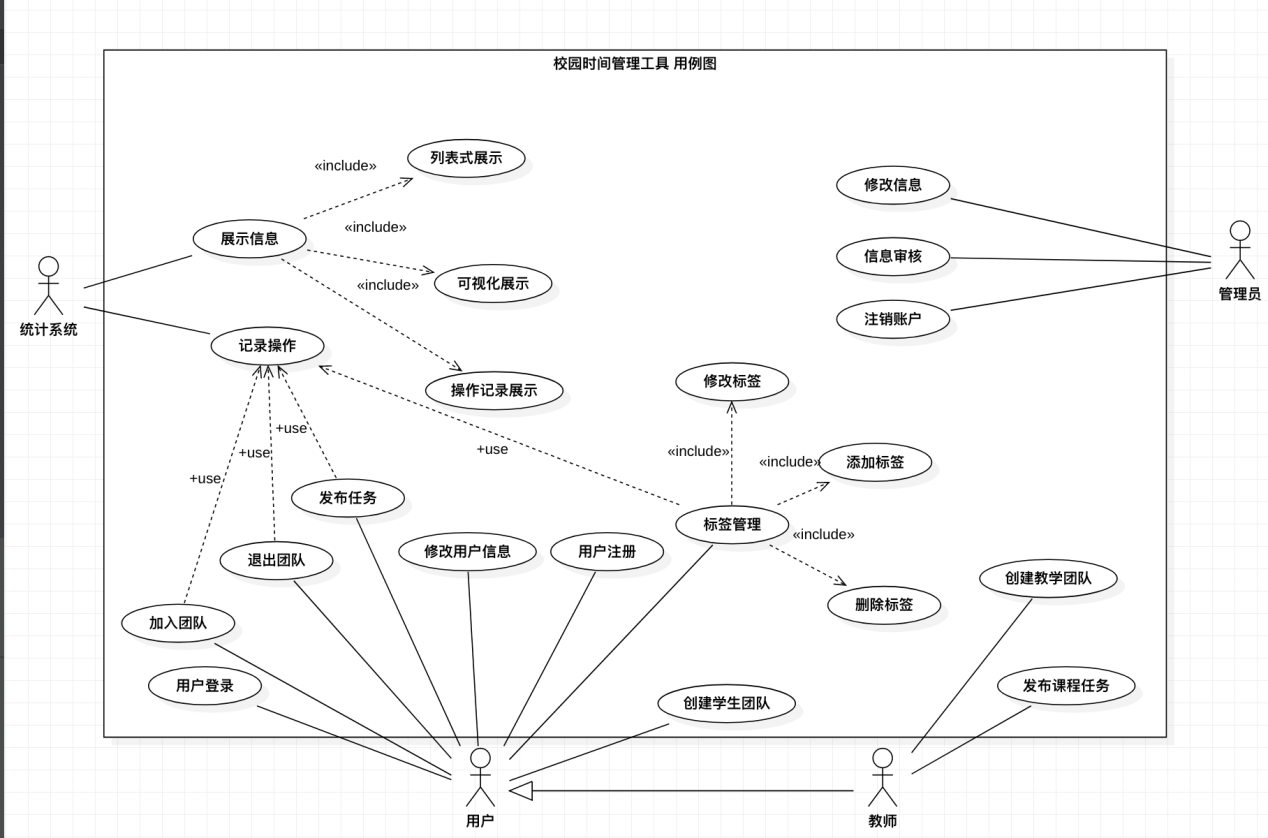
## 3.1 对功能的规定

1. 用户信息管理模块
   1. 用户注册 学生依据学号进行注册，（初始密码为身份证后6位之类的默认密码，由学院导入）同时设定密码，选择学院，之后等待管理员审核，审核通过之后学生才能使用该系统，教师也通过学生的方式进行注册后让管理员进行手动提权，设定为教师组。
   2. 用户登录 终端用户通过学号和密码进行登录，系统校验账号密码是否正确，登陆失败予以错误提示，登陆成功则进入任务看板。
   3. 修改用户信息 学生可以编辑个性签名，头像等基础信息，如果修改院、学号等关键信息需要通过管理员审核。
   4. 注销账号 提起请求后由管理员批准后删除用户账号。
2. 团队管理模块
   1. 创建团队 创建一个任务团队，同时将团队附属于创建者的“我的团队”中，同时可以通过邀请码邀请其他用户加入，邀请码唯一标识，同一个团队的人数存在上限，
   2. 退出团队 退出某个任务团队，同时用户从团队名单中删除，退出信息不会被任何人看到，所有的任务完成情况都会被清空。
3. 任务管理模块
   1. 发布任务 在团队中，有权限者(课程团队->教师，学生团队->所有学生)可以发布任务，发布任务后所选择的所有用户都会收到这条任务，任务中也可以打上一些简单的标签。
   2. 查看任务 查看任务的细则，同时可视化地显示所有共同参与这项任务的用户的完成情况，以及一些简单的统计数据。
   3. 完成任务 参与任务者标记自己已经完成了这项任务，所有其他参与者能从查看任务中看到最新的完成情况。
   4. 修改任务 发布者发现有语法、用词错误时可以修改任务，修改不会显式呈现。
   5. 提交任务 被接受任务者可以在完成任务的同时提交数据，他只能是一段简单的文字，可以用来提交一个链接或是一个答案，但本身并不能再予以反馈，否则这个系统的功能过于冗余，暂时不考虑太多的功能糅合。
4. 评论管理模块
   1. 编写评论 学生团队中的成员可以在其他学生的内容下面评论。
   2. 删除评论 被评论者可以在自己的任务下删除某人的评论。
5. 标签管理模块
   1. 添加标签 增加一项常用标签，可以用以标记任务的性质
   2. 删除标签 删除某一项标签（每个团队专属，课程团队无法修改，由管理员手动修改数据库才能修改，以防不同课程所用的标签混乱）。
   3. 修改标签 修改标签的配色和类名
6. 综合管理模块
   1. 信息修改 修改某个用户的信息及所属的类别，可以将学生组的成员升级为教师组。
   2. 信息审核 审核用户的关键信息修改，审核通过则用户的信息会被修改，审核不通过用户的信息将会保留原样。
7. 统计信息展示模块
   1. 列表式展示 根据所有参与者的完成情况提供一个列表式的数据，根据完成率及完 成成员生成一个列表，以供展示。
   2. 可视化展示 根据所有参与者的完成情况生成饼图，可视化的查看现在该任务的完成情况。
   3. 操作记录展示 记录团队中所有成员完成的任务以及标签修改、发布任务情况。

## 3.1.1 确认参与者

|  |  |
| --- | --- |
| 参与者 | 参与项目 |
| 学生 | 用户信息管理，团队信息管理，标签管理，评论管理，任务管理模块 |
| 管理员 | 综合管理模块 |
| 教师 | 用户信息管理，团队信息管理，标签管理，评论管理，任务管理模块 |
| 统计系统 | 统计信息展示模块 |

## 3.1.2 确定用例



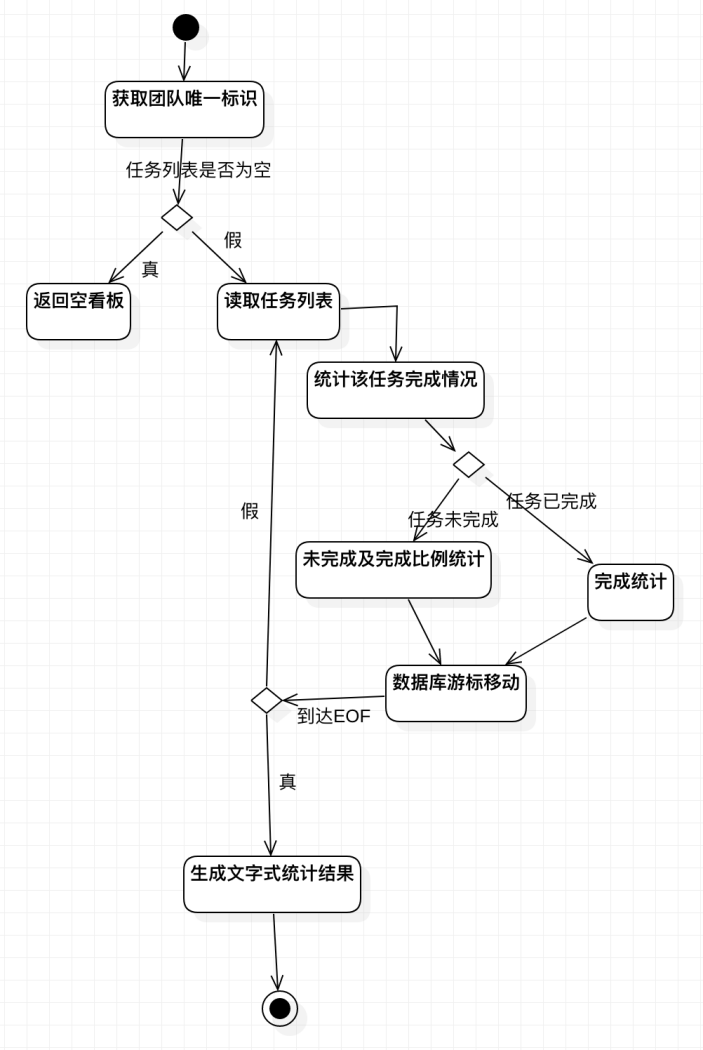
校园时间管理用例图 （UML不支持<<use>>，使用+use表示使用关系）

## 3.1.3 用例说明

1. **列表式展示**

描述：列表式展示任务的完成情况，团队的情况

参与者：统计系统

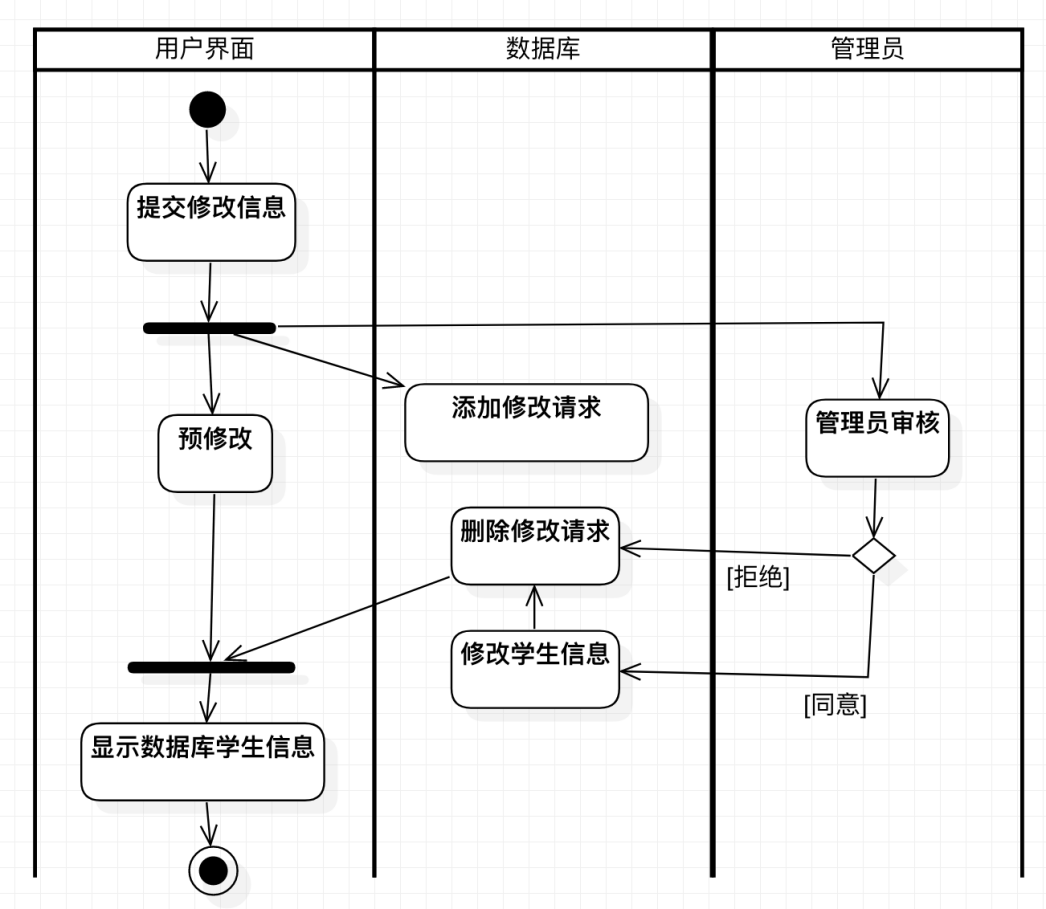
前置条件：团队存在

1. **修改信息**

描述：师生选择修改自己的个人信息(院，年级，班级等)

参与者：用户 管理员

前置条件：已经注册过账号

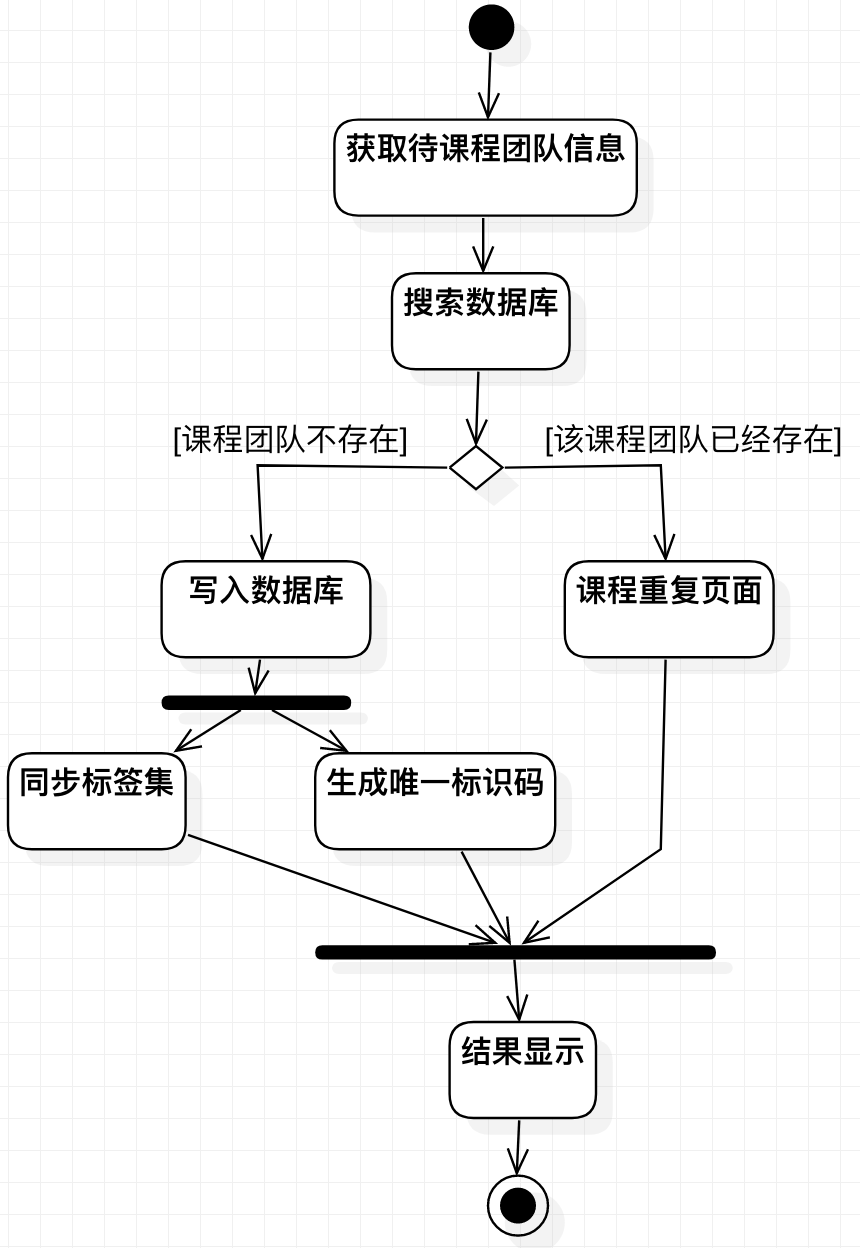


1. **创建课程团队**

描述：教师选择创建一个新的课程团队，一个团队是否唯一的标识是课程名和教师名

参与者 教师

前置条件：拥有创建课程团队的资格

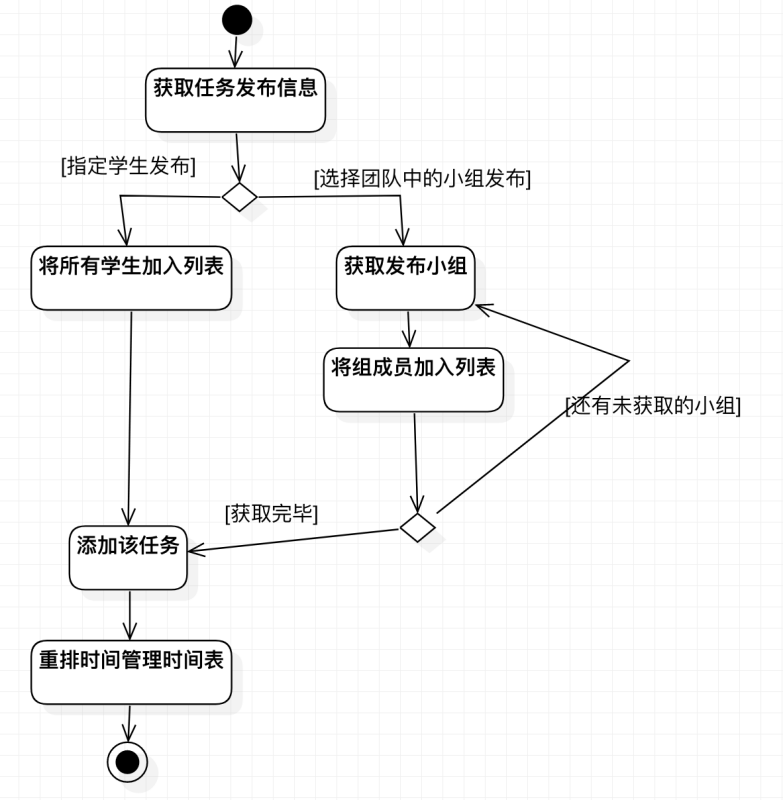


1. **发布课程任务**

描述：教师选择它所属的一个课程团队，同时将任务下发给指定的学生

参与者：教师

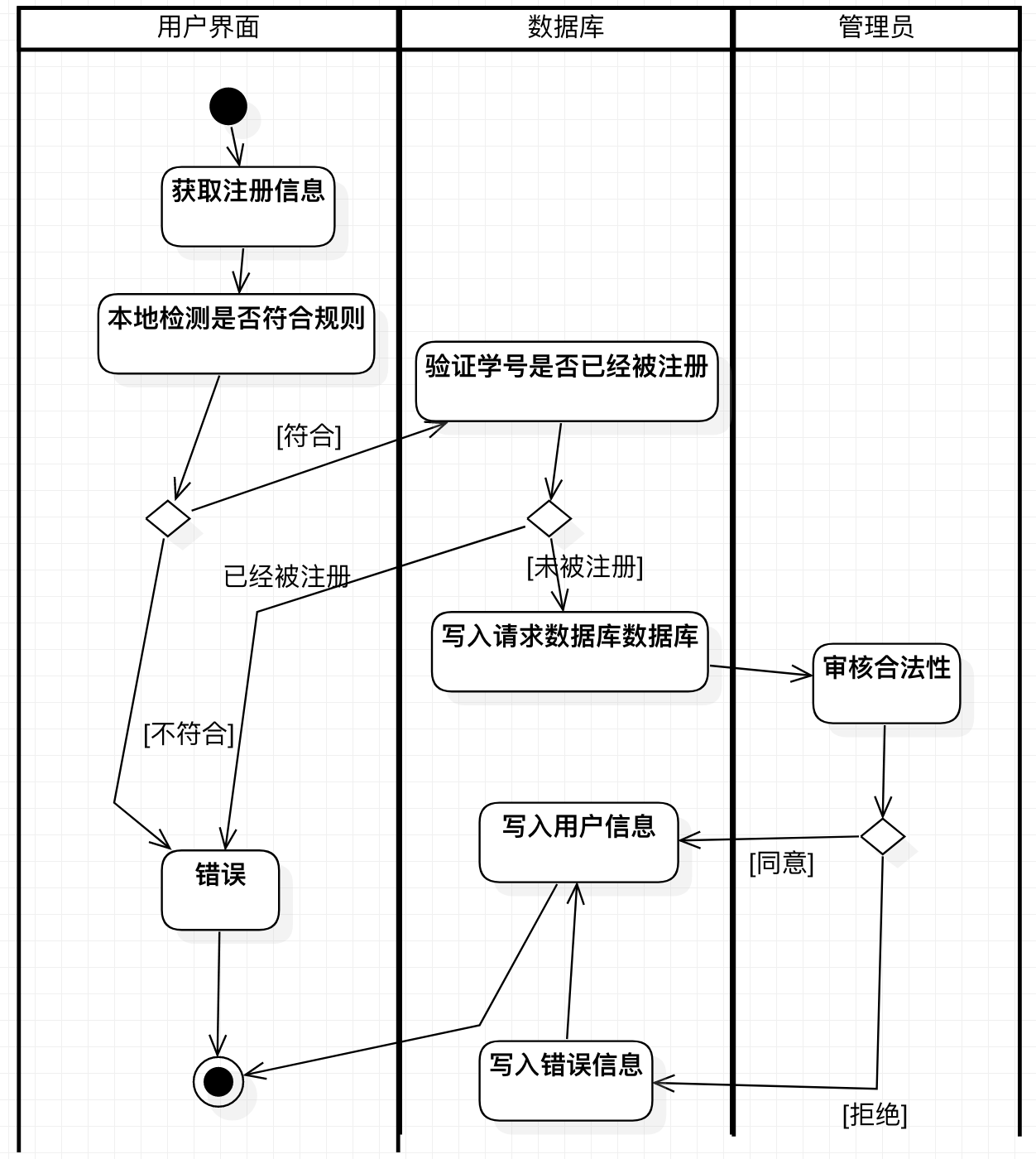
前置条件：是课程团队的创建者



1. **用户注册**

描述：学生选择新建用户，待管理员审核后即可使用，合法性失败时也会写入数据库，这样下次登录时就会得知账号注册失败。

参与者 学生、管理员

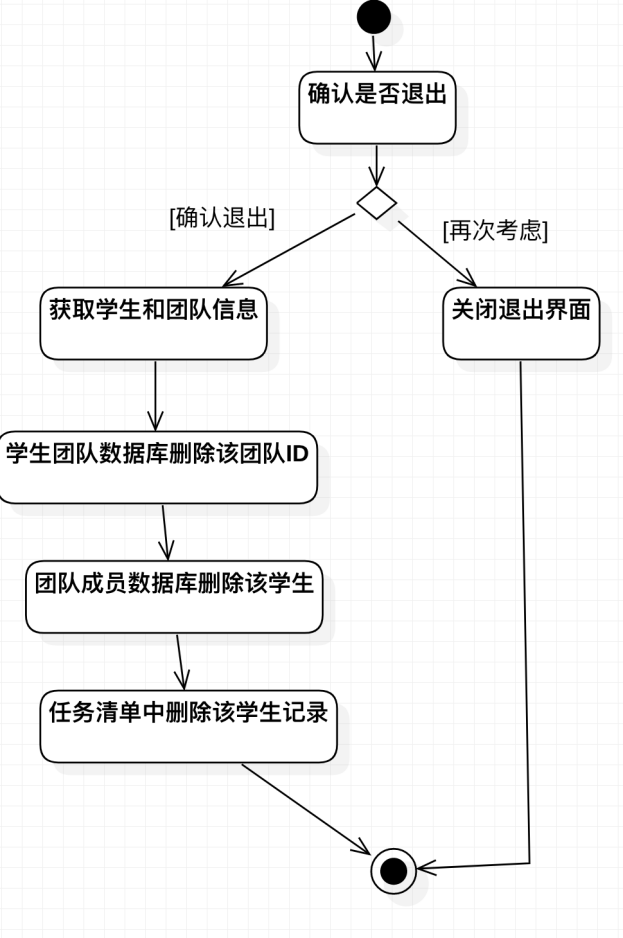


1. **退出团队**

描述：学生选择退出这个团队，学生数据库中将会移除这个团队信息，同时该团队用户名单和任务名单中都会移除和这个学生有关的所有信息

参与者：学生

前置条件： 存在于这个团队之中

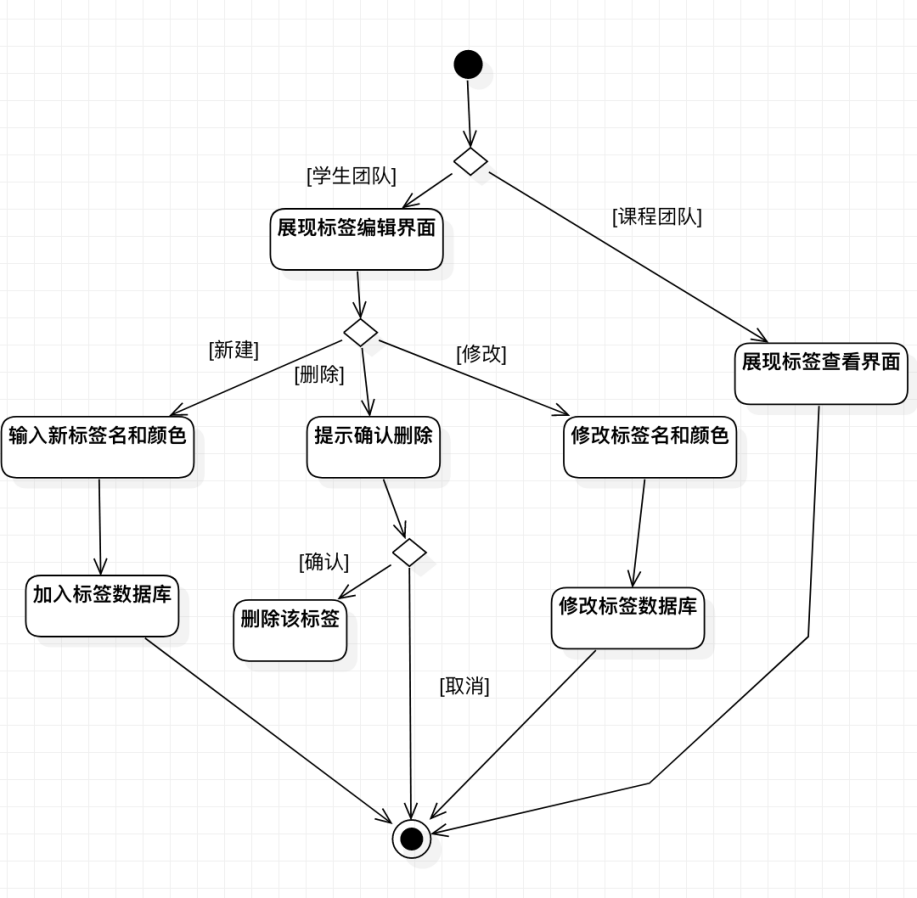


1. **标签管理**

描述 对于团队中所有用的标签进行管理，类似GITHUB的ISSUE TAG，课程的标签不允许进行任何修改为了保证系统UI的一致性

参与者 学生

前置条件：在学生团队中



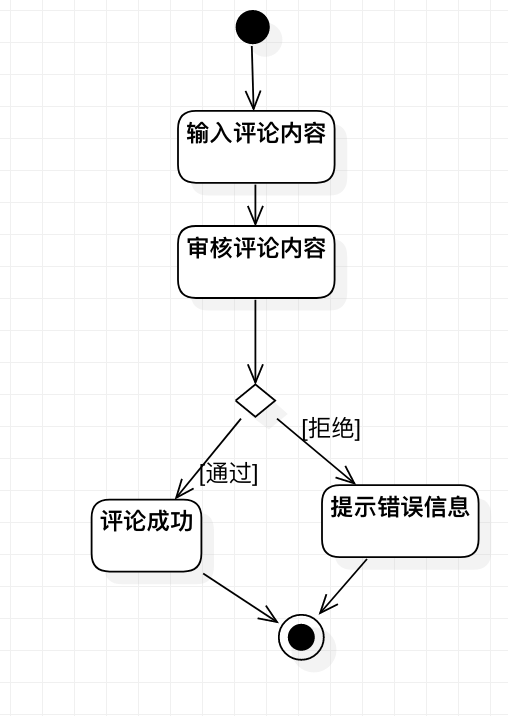
1. **编写评论**

描述 在别人的任务完成页面里对其评论，通常使用在学生团队中，审核评论内容具体是指是否存在大量无意义换行，是否存在一些违禁词汇，由于是学生自治用的团队，规则

不会太严格，主要是为了防止出现大量无意义换行导致UI显示混乱。

参与者 学生

前置条件：在同一学生团队中

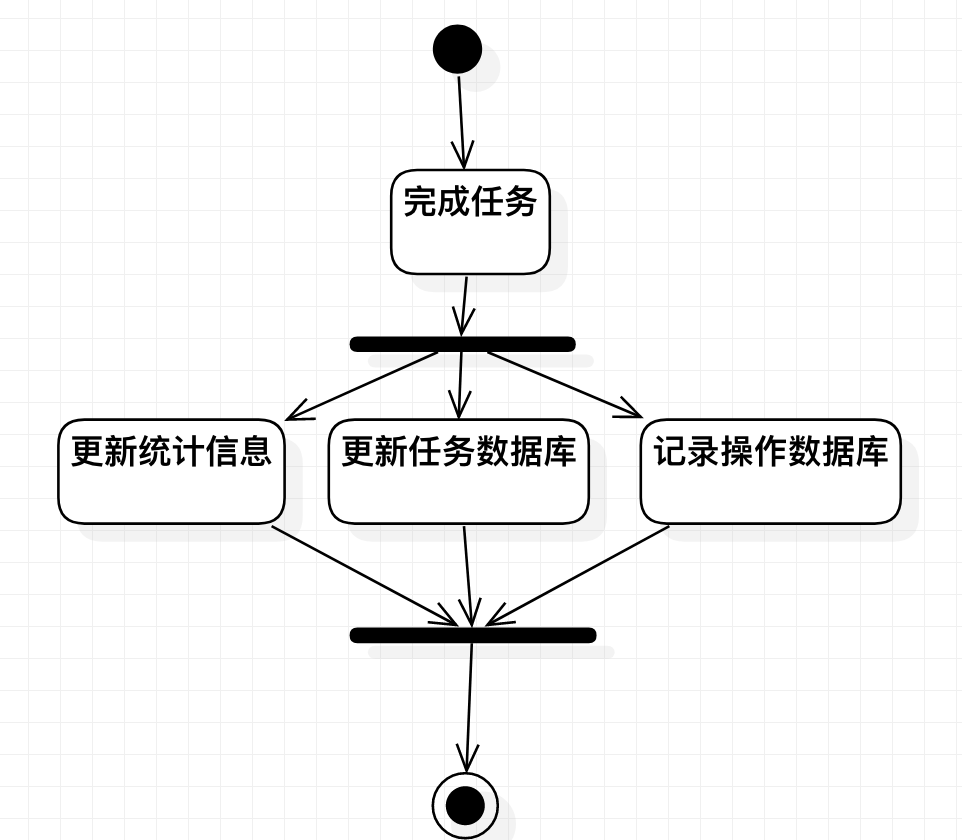


1. **完成任务**

描述 学生完成任务，同时更新任务数据库，更新统计信息，更新团队的操作记录面板

参与者： 学生

前置条件：任务存在

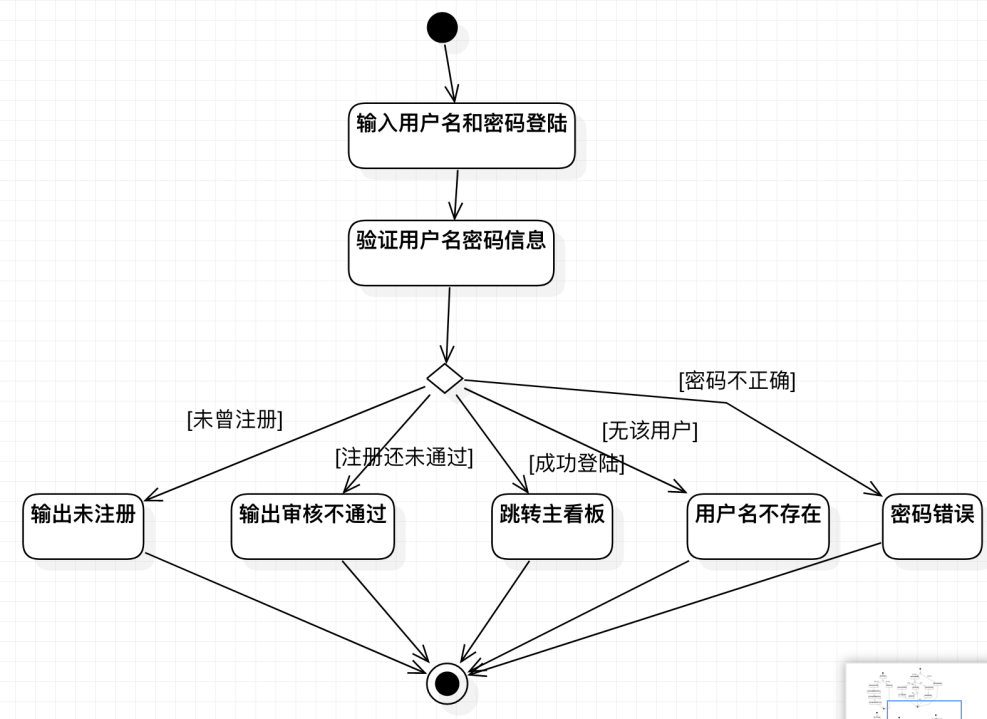


1. **用户登录**

描述 用户输入用户名和密码，并且完成登陆，由于用户名默认存储在数据库中，所以我们还需要特判该账号是否未注册还是注册了未被审核通过。

参与者： 学生

前置条件：用户已经注册

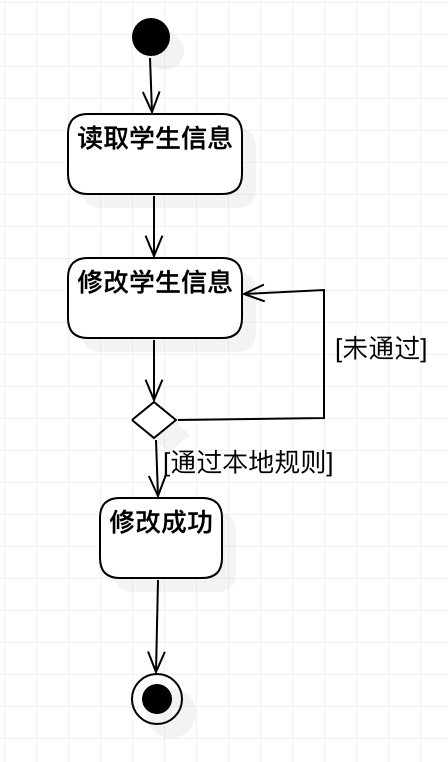


1. **信息修改**

描述 管理员对用户的信息进行直接修改，无需任何确认即可，本地规则防止管理员手误输入了错误信息。

参与者 管理员

前置条件：用户账号存在

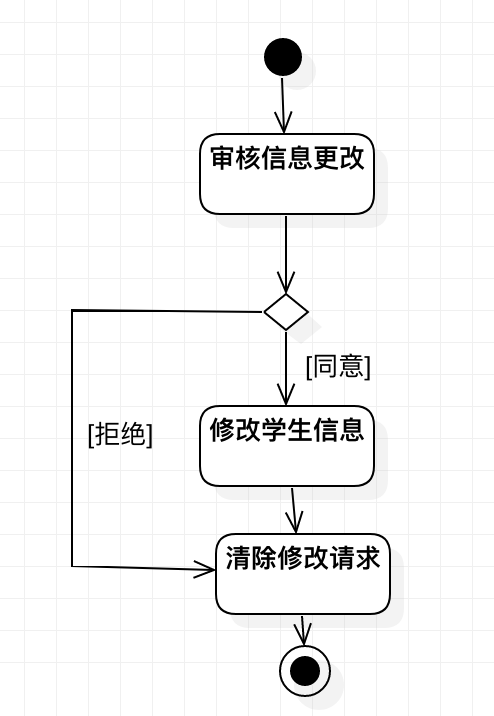


1. **审核修改**

描述 管理员收到用户的信息修改请求，对其信息修改进行审核，如果同意就完成修改，否则保持原样

参与者：管理员

前置条件：用户发起信息修改请求

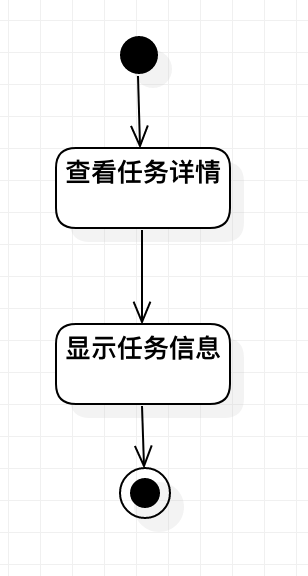


1. **查看任务**

描述 用户选择查看任务详情，从数据库中查询该任务，并且将任务的详情和统计信息读取出来，显示在任务详情面板上。

参与者：用户

前置条件：存在任务

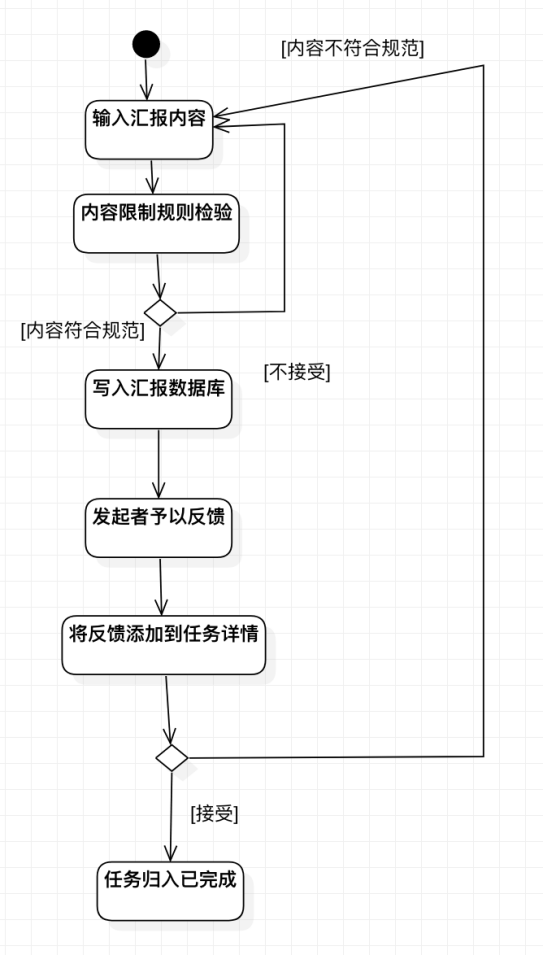


1. **提交任务**

描述 学生选择提交任务，同时输入汇报内容，由发起人接受反馈，并予以评论，再选择是否接受汇报，同时这边请注意，发起人可以是学生也可以是老师，学生团队的学生和课程团队的老师都可以是发起人，但是提交人只能是学生。

参与者： 用户

前置条件： 有发起人向用户发送任务，并且要求予以反馈



## 3.2 对性能的规定

## 3.2.1 精度

用户信息、评论信息、团队信息、任务信息、标签信息全部按照符合条件的输入进行保存，不进行舍去。

## 3.2.2 时间特性要求

1. 服务器响应时间： 小于1秒
2. 创建团队处理时间： 小于2秒
3. 发送任务处理时间： 小于2秒
4. 注册账户处理时间： 小于1秒
5. 评论任务处理时间： 小于2秒
6. 加入团队处理时间： 小于1秒
7. 统计信息呈现时间： 小于3秒
8. 完成任务处理时间： 小于1秒
9. 修改任务处理时间： 小于3秒
10. 添加标签处理时间： 小于1秒
11. 修改标签处理时间： 小于2秒
12. 查看任务处理时间： 小于2秒
13. 登陆账户处理时间： 小于2秒

### 3.2.3灵活性

1. 操作方式上的变化：可以通过电脑端或手机端的浏览器进行访问；
2. 运行环境的变化：在多种现代浏览器中运行；
3. 精度和有效时限的变化：精度变化要对系统的影响尽量减小；
4. 计划的变化或改进：能够比较容易地根据需求进行改进。

## 3.3输入输出要求

输入数据基本为：用户帐号、用户密码、待发布的任务、待发表的评论、待创建的团队、待创建的标签，以上输入均为字符串形式；还有一些请求，这些请求通过前端包装成GET/POST方法作为后端的输入；输出数据基本为时间管理工具系统界面、发布的任务、发表的评论、创建和所在的团队。

## 3.4数据管理能力要求

需要管理的学生数量可以是全院学生，至多20000个，需要承载至多2000个团队同时存储于数据库中，根据实际情况，支持每秒响应数个请求，需要同时管理学生表、学院表、团队表、每个团队的任务列表、每个团队的标签表。

## 3.5故障处理要求

发生错误时保证数据完整，对于数据库发生故障时要能够及时解决故障，恢复数据，以保证数据的一致性和完整性。

## 3.6其他专门要求

无其他专门要求。

# 运行环境规定

## 4.1设备

设备的软硬件说明

1. 所有现代的电子设备皆可以正常使用该程序
2. 设备需要有互联网接入
3. 设备需要安装有现代的浏览器

## 4.2 支持软件

1. 服务器端软件选择
   1. 操作系统：任意Java支持的系统
   2. 数据库管理系统： MySQL
   3. 开发工具：Intellij idea
2. 客户端软件选择
   1. 大多数现代浏览器

## 4.3接口

所有接口符合RESTFUL风格，可以通过前端网页访问，也可以通过授权的后端通过HTTP请求直接获取源数据开发第三方程序

## 4.4控制

本系统通过网络提供服务，用户通过浏览器访问服务器，向服务器发出服务请求。因此，需要使用 TCP/IP 网络协议，作为标准的通信控制接口。