**校园时间管理工具**

需求分析报告

**概要设计说明书**

目录

[1. 引言 2](#_Toc299498209)

[1.1 编写目的 2](#_Toc2112213742)

[1.2背景 2](#_Toc2071676884)

[1.3 定义 3](#_Toc1521020577)

[1.3 定义 3](#_Toc147503751)

[1.4 参考资料 3](#_Toc899414419)

[2. 系统结构模型 4](#_Toc320748900)

[2.1 初始对象表 4](#_Toc642808330)

[2.2 初始类图 5](#_Toc1836857900)

[2.3 细化类图 5](#_Toc1993299675)

[2.3.1 属性识别 5](#_Toc642744525)

[2.3.2 操作识别 8](#_Toc764487265)

[2.4 三层精化类图 10](#_Toc342802854)

[3.2 交互模型 14](#_Toc1936425924)

[3.2 状态模型 15](#_Toc395834383)

# 引言

## 1.1 编写目的

需求说明书目的是使用户和软件开发者双方对该软件有一个共同的理解，使之成为整个开发工作的基础。便于用户、开发人员进行理解和交流，反映出用户问题的结构，可以作为软件开发工作的基础和依据，并作为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. 待开发的软件系统的名称：校园时间管理系统
2. 本项目的任务提出者： 软件工程课程设计
3. 本项目的任务开发者： 许羽晟
4. 用户：需求更棒的时间管理应用的师生；
5. 实现该软件的计算中心或计算机网络：任意装有JAVA环境的PC机或云服务器；

## 1.3 定义

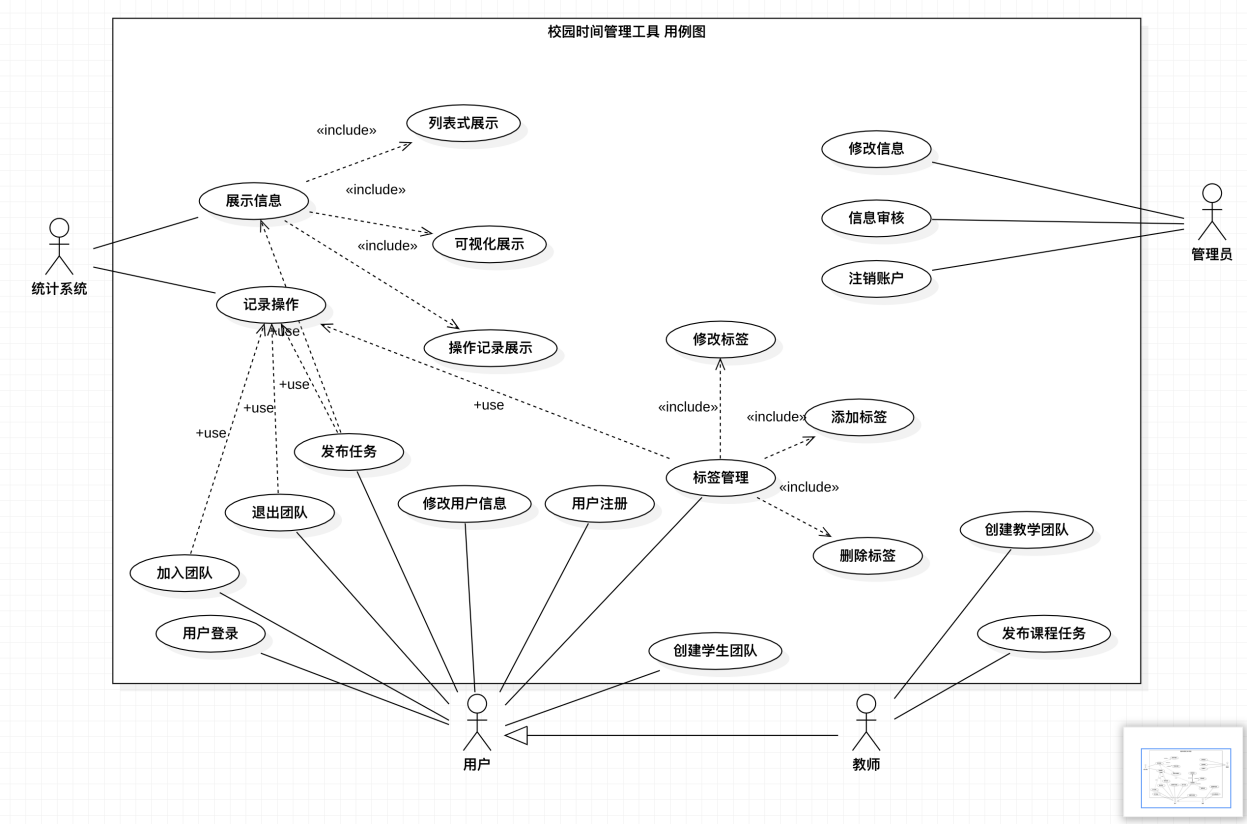
|  |  |
| --- | --- |
| 名称 | 说明 |
| 学生账号 | 每个学生专属的账号，和学生信息绑定 |
| 学生信息 | 学生的基础信息，学号，班级，专业，联系方式，教师、同团队可查看 |
| 任务团队 | 一个老师和多个学生组成的一个组，老师可以向学生发布任务，  或是多个学生之间形成的组，学生之间可以互相发布任务 |
| 发布任务 | 老师或者互为团队学生向指定或所有成员发布一项任务 |
| 完成任务 | 被发布任务者认为自己完成了任务，将它移入已完成分组 |
| 任务评论 | 所有此任务参与者可以在下面评论，或是队内根据分配关系互评 |
| 删除任务 | 发布者认为不妥，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 |

# 系统结构模型

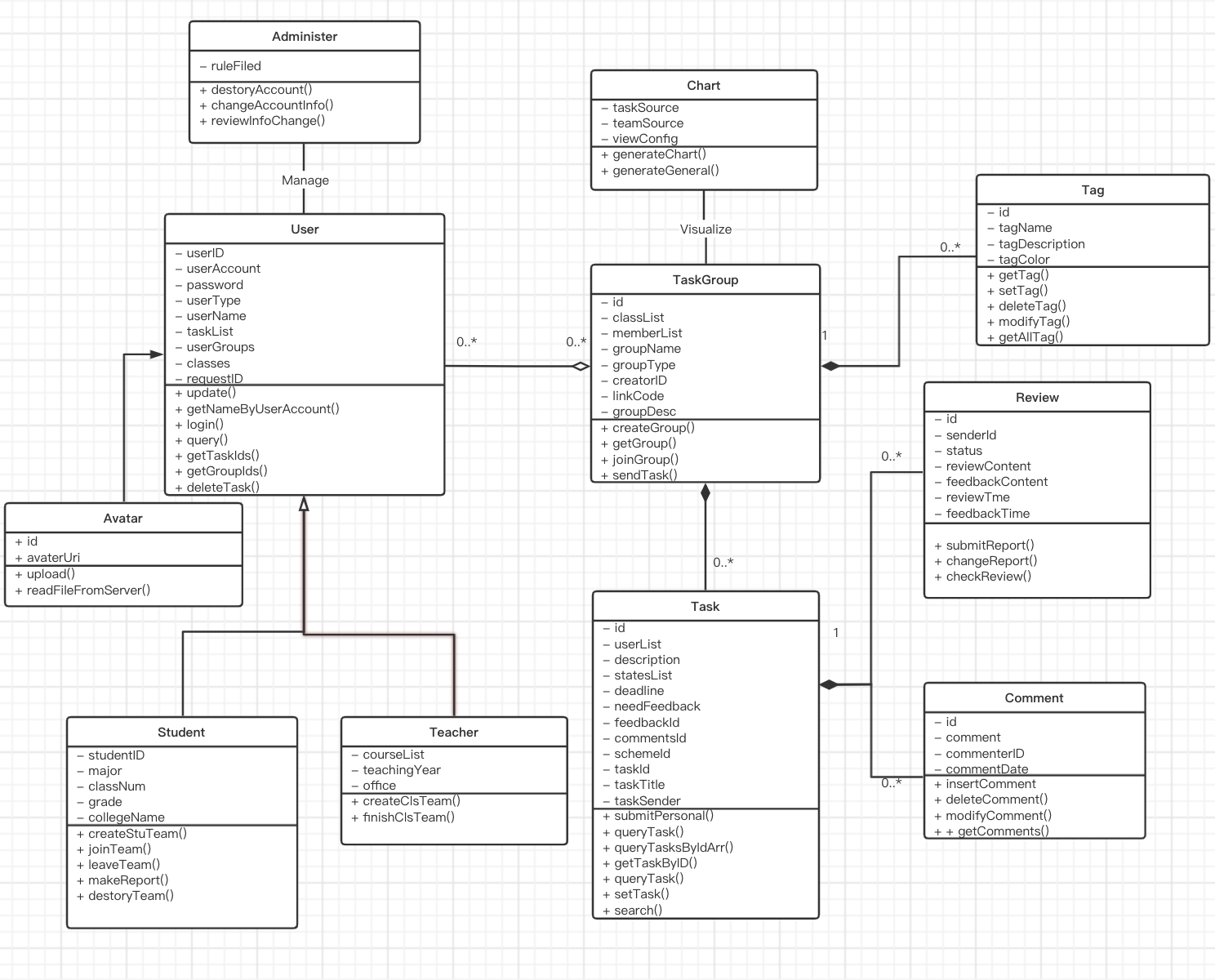
## 2.1 初始对象表



初始对象/类表：

|  |  |  |
| --- | --- | --- |
| 可能的类 | 英文类名 | 简单描述 |
| 用户 | User | 该类是学生和教师的基类，提供两个类的通用方法，如注销账户，修改信息，修改密码 |
| 学生 | Student | 继承用户，该类用于对学生提供一系列时间管理的相关操作 |
| 教师 | Teacher | 继承用户，该类用于对教师提供一些列时间管理的相关操作 |
| 管理员 | Administrator | 不继承用户，仅提供简单的信息审核的一系列操作 |
| 任务 | Task | 用来管理发送的任务请求及一些列操作 |
| 标签 | Tag | 用来管理团队中的任务标签 |
| 团队 | TaskGroup | 用来关系团队中人员增减的一些列操作 |
| 统计 | Chart | 用来分析数据，并且生成一些可视化/文字式的统计结果 |
| 评论 | Comment | 用来进行评论的一系列操作。 |
| 汇报 | Review | 用来处理需要汇报结果并审核的任务的一些列相关操作 |

## 2.2 初始类图



## 2.3 细化类图

### 2.3.1 属性识别

类名：User（用户类）

属性表：

表2.1.1 User类属性表

|  |  |  |
| --- | --- | --- |
| 中文名 | 英文名 | 数据类型 |
| 用户ID | ID | String |
| 账户 | userAccount | String |
| 密码 | password | String |
| 权限 | userType | Integer |
| 用户名 | userName | String |
| 团队清单 | userGroups | List<Integer> |
| 任务清单 | tasks | List<Integer> |
| 请求ID | requestId | Integer |

类名：Student（学生类） 继承User

属性表：

表2.1.2 Student类属性表

|  |  |  |
| --- | --- | --- |
| 中文名 | 英文名 | 数据类型 |
| 学号 | studentID | Integer |
| 专业 | major | String |
| 班级 | classNum | Integer |
| 年级 | grade | Integer |
| 学院 | collegeName | String |

类名：Teacher（教师类）继承User

属性表：

表2.1.3 Teacher类属性表

|  |  |  |
| --- | --- | --- |
| 中文名 | 英文名 | 数据类型 |
| 开设课程 | courseList | List<Integer> |
| 教龄 | tachingYear | Integer |
| 办公楼 | office |  |

类名：Administrator（管理员类）

属性表：

表2.1.4 Administrator类属性表

|  |  |  |
| --- | --- | --- |
| 中文名 | 英文名 | 数据类型 |
| 管辖学院 | ruleFiled | List<Integer> |

类名：Task（任务类）

属性表：

表2.1.5 Task类属性表

|  |  |  |
| --- | --- | --- |
| 中文名 | 英文名 | 数据类型 |
| 任务名 | TaskTitle | String |
| 参与人员ID | userList | List<Integer> |
| 截止日期 | deadline | Date |
| 完成人员 | statesList | List<Integer> |
| Id号 | id | Integer |
| 详细 | description | String |
| 需要反馈 | needFeedback | Boolean |
| 反馈ID | feedbackId | String |
| 颜色 | schemeId | Integer |
| 评论列表 | comments | List<Integer> |
| 任务发起人 | taskSender | String |

类名：Tag（标签类）

属性表：

表2.1.6 Tag类属性表

|  |  |  |
| --- | --- | --- |
| 中文名 | 英文名 | 数据类型 |
| 标签名 | tagName | String |
| 标签颜色 | tagColor | String |
| 标签描述 | TagDescription | String |
| ID | id | Integer |

类名：TaskGroup（团队类）

属性表：

表2.1.7 TaskGroup类属性表

|  |  |  |
| --- | --- | --- |
| 中文名 | 英文名 | 数据类型 |
| 团队ID | id | Integer |
| 班级编号 | classList | List<Integer> |
| 团队名单 | memberList | List<Integer> |
| 团队任务 | taskList | List<Integer> |
| 团队类型 | groupType | Integer |
| 团队编码 | linkCode | Integer |
| 团队描述 | groupDescription | String |
| 创建者 | creatorId | Integer |

类名：Chart（展示类）

属性表：

表2.1.8 Chart类属性表

|  |  |  |
| --- | --- | --- |
| 中文名 | 英文名 | 数据类型 |
| 任务数据源 | taskSource | Task |
| 团队数据源 | teamSource | Team |
| 展示附加条件 | viewConfig | ViewConfig |

类名：Review（报告类）

属性表：

表2.1.9 Report类属性表

|  |  |  |
| --- | --- | --- |
| 中文名 | 英文名 | 数据类型 |
| id | id | Integer |
| 发送者 | senderId | Integer |
| 状态 | Status | Integer |
| 评论内容 | reviewContent | String |
| 回馈内容 | feedbackContent | String |
| 反馈时间 | reviewTime | Date |
| 审核时间 | feedbackTime | Date |

类名：Comment（评论类）

属性表：

表2.1.10 Comment类属性表

|  |  |  |
| --- | --- | --- |
| 中文名 | 英文名 | 数据类型 |
| Id | Id | Integer |
| 评论者 | commentId | String |
| 评论内容 | comment | Stng |
| 时间 | commentDate | Date |
| 时间 | time | Date |

### 2.3.2 操作识别

1.操作的分类和约定

基本操作：包括数据库检索和更新，如增加、删除、修改、选择、查询等

关键操作：必须由对象提供的、在算法上复杂的业务操作

约定：set、get方法不写出来，因为他默认存在且对于需求报告没有过多意义

参数的意思并非是数据类型,函数的参数，而是用类英文的方法表达所需要处理的数据的语义。

所有已设置XXX开头或者类似的函数表达的是对数据库的操作而非Set、Get方法

1. 具体方法识别

User

表2.2.1 User类的操作

|  |  |  |  |
| --- | --- | --- | --- |
| 中文名 | 英文名 | 返回值 | 参数 |
| 登陆账户 | login | Boolean | username,pwd |
| 修改信息 | changeInfo | Boolean | userInfo |
| 注销账户 | deleteAccount | Boolean | NULL |
| 注册账户 | registerAccount | Boolean | userInfo |
| 查询团队 | queryTasks | List<Task> | userID |
| 审核任务 | feedbackTask | Boolean | TaskID,reviewerID,revieweeID,rawText |
| 评论任务 | makeComment | Boolean | Comment |
| 发起任务 | openTask | Boolean | Task |

Student

表2.2.2 Student类的操作

|  |  |  |  |
| --- | --- | --- | --- |
| 中文名 | 英文名 | 返回值 | 参数 |
| 创建学生团队 | createStuTeam | Boolean | teamInfo |
| 加入团队 | JoinGroup | Boolean | teamID,StuID |
| 退出团队 | leaveTeam | Boolean | teamID,StuID |
| 反馈 | makeReport | Boolean | Report |
| 解散学生团队 | destoryTeam | Boolean | teamID |

Teacher

表2.2.3 Teacher类的操作

|  |  |  |  |
| --- | --- | --- | --- |
| 中文名 | 英文名 | 返回值 | 参数 |
| 创建课程团队 | createClsTeam | Boolean | teamInfo |
| 结束课程 | finishClsTeam | Boolean | teamID |

Administer

表2.2.4 Administer类的操作

|  |  |  |  |
| --- | --- | --- | --- |
| 中文名 | 英文名 | 返回值 | 参数 |
| 注销用户账户 | destoryAccount | Boolean | userID |
| 修改用户信息 | changeAccountInfo | Boolean | userID,newInFo |
| 审核用户修改 | reviewInfoChange | Boolean | accept |

Task

表2.2.5 Task类的操作

|  |  |  |  |
| --- | --- | --- | --- |
| 中文名 | 英文名 | 返回值 | 参数 |
| 获取用户全部任务 | getTaskIds | List<Task> | userID |
| 查询单个任务 | query | Task | Taskid |
| 删除任务 | delete | Boolean | TaskId |
| 更新用户任务状态 | update | None | userID |
| 获取用户任务状态 | queryTask | Status | id,taskId |
| 获取ID批的所有任务情况 | queryTasksByIdArr | List<Task> | List<Integer> |
| 提交任务 | submitPersonalTask | Stirng | Task |
| 查询 | Search | List<Task> | searchStr |

Tag

表2.2.6 Tag类的操作

|  |  |  |  |
| --- | --- | --- | --- |
| 中文名 | 英文名 | 返回值 | 参数 |
| 增加标签 | set | void | Tag |
| 删除标签 | delete | void | id |
| 修改标签 | update | void | Tag |
| 加载标签 | getAll | List<Tag> |  |
| 创建配色方案 | createTags | String | Scheme |
| 获取配色方法 | getTags | Scheme | Integer |

TaskGroup

表2.2.7 TaskGroup类的操作

|  |  |  |  |
| --- | --- | --- | --- |
| 中文名 | 英文名 | 返回值 | 参数 |
| 创建团队 | createGroup | Boolean | TaskGroup |
| 获取团队 | getGroup | List<TaskGroup > | List<Integer> |
| 删除团队 | deleteTeam | Boolean | TeamID |
| 增加人员 | addNewUser | Boolean | TeamID,UserID |
| 删除成员 | deleteUser | Boolean | TeamID,UserID |

Report

表2.2.8 Report类的操作

|  |  |  |  |
| --- | --- | --- | --- |
| 中文名 | 英文名 | 返回值 | 参数 |
| 提交汇报 | submitReport | Boolean | Report |
| 修改汇报 | changeReport | Boolean | ReportID,newReport |
| 查看反馈 | checkReview | Review | ReportID |

Comment

表2.2.9 Comment类的操作

|  |  |  |  |
| --- | --- | --- | --- |
| 中文名 | 英文名 | 返回值 | 参数 |
| 提交评论 | makeComment | String | Comment |
| 删除评论 | deleteComment | String | CommentID |
| 修改评论 | modifyComment | Boolean | CommentID,newComment |
| 获取所有评论 | getAllComments | List<Comment> | taskID |
| 获取指定ID的评论 | getComments | List<Comments> | taskID[] |

Chart

表2.2.10 Chart类的操作

|  |  |  |  |
| --- | --- | --- | --- |
| 中文名 | 英文名 | 返回值 | 参数 |
| 生成图表 | generateChart | List<Param> | ChartConfig |
| 生成文字描述 | generateGeneral | List<String> | ChartConfig |

Avatar

表2.2.10 Avatar类的操作

|  |  |  |  |
| --- | --- | --- | --- |
| 中文名 | 英文名 | 返回值 | 参数 |
| 提交头像 | upload | String | file |
| 流式获取头像 | readFileFromServer | 流式输出 | fileURI |

## 2.4 三层精化类图

在初始类图的基础上，运用三层体系结构思想对基本类模型进行划分。根据各个对象的具体情况，将对象分成前端界面、事务和数据层。

约定 采用vue开发，所有的部分都是vue Component，在运行时动态的切换如<router view>进行更换，更方便界面的统一性和开发的效率，内嵌的意思是在文件里直接<template>构建一个完整的类似组件的模块，主要是为了方便数据的获取更符合vue的开发方法，因为vue的参数传递是很复杂的，所以很有可能一个Team.vue中继承了大部分的功能，这边的功能主要按照vue页面来划分，vue作为一个单页面应用可以通过v-if等方式控制页面的layout，配合vuetify的flex布局，使得页面增减变得非常容易。

1. User层

界面层

|  |  |
| --- | --- |
| 名称 | 描述 |
| signUp.vue | 注册页面组件 |
| signIn.vue | 登陆页面组件 |
| resetInfo.vue | 修改用户信息组件 |
| 内嵌于Group.vue | 创建评论 |
| StartTaskPopup | 创建任务 |
| reviewTask.vue | 审核任务 |
| query.vue | 查询团队 |

事务层 UserCntroller.java

数据层 user.frm

1. Student层

界面层

|  |  |
| --- | --- |
| 名称 | 描述 |
| 内嵌于Navbar.vue | 创建学生团队 |
| 内嵌于Navbar.vue | 加入团队 |
| quitTeam.vue | 退出团队 |
| makeReport.vue | 任务反馈 |

|  |  |
| --- | --- |
| dismissTeam.vue | 解散团队 |

事务层 UserController.java

数据层 User.frm

1. Teacher层

界面层

|  |  |
| --- | --- |
| 名称 | 描述 |
| 内嵌于navbar.vue | 创建团队 |
| finishClassTeam | 结束这一学期的课程团队 |

事务层 UserController.java

数据层 User.frm

1. Administer层

界面层

|  |  |
| --- | --- |
| 名称 | 描述 |
| destoryAccount.vue | 删除账号 |
| changeAccountInfo.vue | 修改账号信息 |
| reviewInfoChange | 审核数据修改 |

事务层 Administer.java

数据层 Administer.frm

1. Task层

界面层

|  |  |
| --- | --- |
| 名称 | 描述 |
| 内嵌于Task.vue | 修改任务描述 |
| 内嵌于Task.vue | 设置成员 |
| 内嵌于Tag.vue | 设置标签列表 |
| Dashboard.vue | 显示任务列表 |

事务层 TaskController.java

数据层 Task.frm

1. Tag层

界面层

|  |  |
| --- | --- |
| 名称 | 描述 |
| 内嵌于Tag.vue | 增加一个标签 |
| 内嵌于Tag.vue | 删除一个标签 |
| 内嵌于Tag.vue | 修改标签信息 |
| 内嵌于Tag.vue | 加载所有标签信息 |

事务层 TagController.java

数据层 Tag.frm

1. Team层

界面层

|  |  |
| --- | --- |
| 名称 | 描述 |
| 内嵌于 Nabbar.vue | 创建队伍 |
| 内嵌于Team.vue | 删除团队 |
| 内嵌于Team.vue | 增加成员 |
| 内嵌于Team.vue | 删除成员 |
| 内嵌于Team.vue | 修改团队详细信息学 |

事务层 taskGroupController.java

数据层 taskGroup.frm

1. Report层

界面层

|  |  |
| --- | --- |
| 名称 | 描述 |
| submitReport.vue | 提交汇报 |
| changeReport.vue | 修改汇报 |
| checkReview.vue | 查看汇报的反馈 |

事务层 ReviewController.java

数据层 Review.frm

1. Comment层

界面层

|  |  |
| --- | --- |
| 名称 | 描述 |
| 内嵌于Group.vue | 提交评论 |
| 内嵌于Group.vue | 删除评论 |
| 内嵌于Group.vue | 修改评论 |

事务层 CommentController.java

数据层 Comment.frm

1. Chart层

界面层

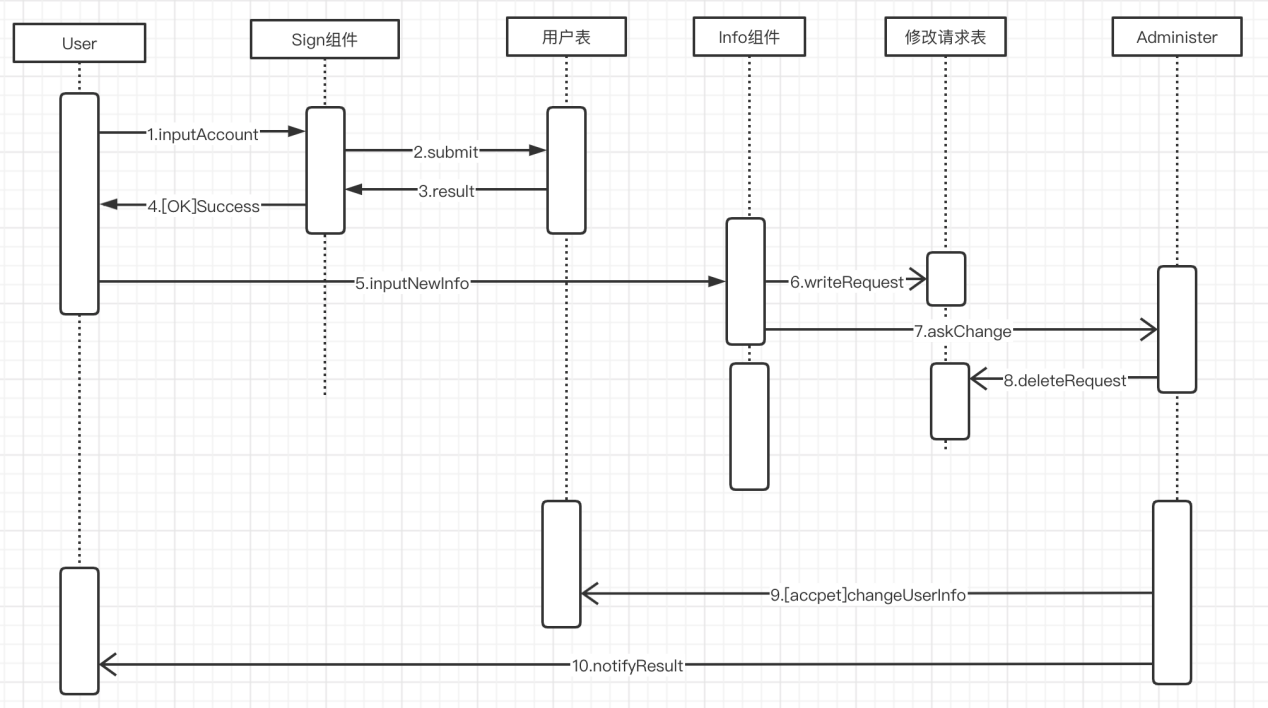
|  |  |
| --- | --- |
| 名称 | 描述 |
| generateCharts.vue | 生成可视化界面 |
| generalInfo.vue | 生成文字化描述 |

事务层 ChartController.java

数据层 Chart.frm

## 3.1 交互模型

约定：本系统大多数用例并不算复杂，从意思就可以知道需要实现的功能，其中修改信息，审核注册，提交汇报，发送任务4个用例涉及到较复杂的对象间交互及数据库操作，特此使用顺序图加以详细描述。同时，前端页面通过组件标识，采用Vue的组件式开发程式，所以我们所有的前端页面都可以称作组件，这为之后的布局和修改提供了非常良好的基础，同时为了不对数据产生理解错误，部分信息采用中文标注。

图3.1 修改信息 顺序图

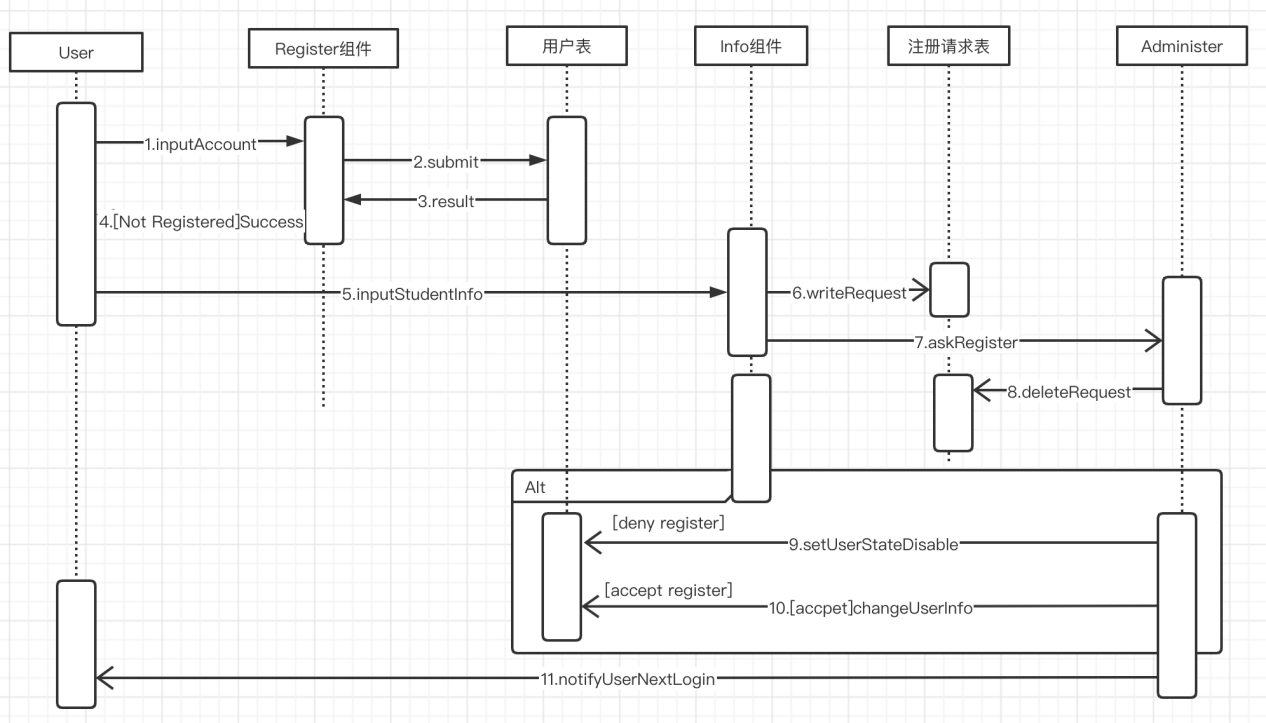


图3.2 注册账号 顺序图

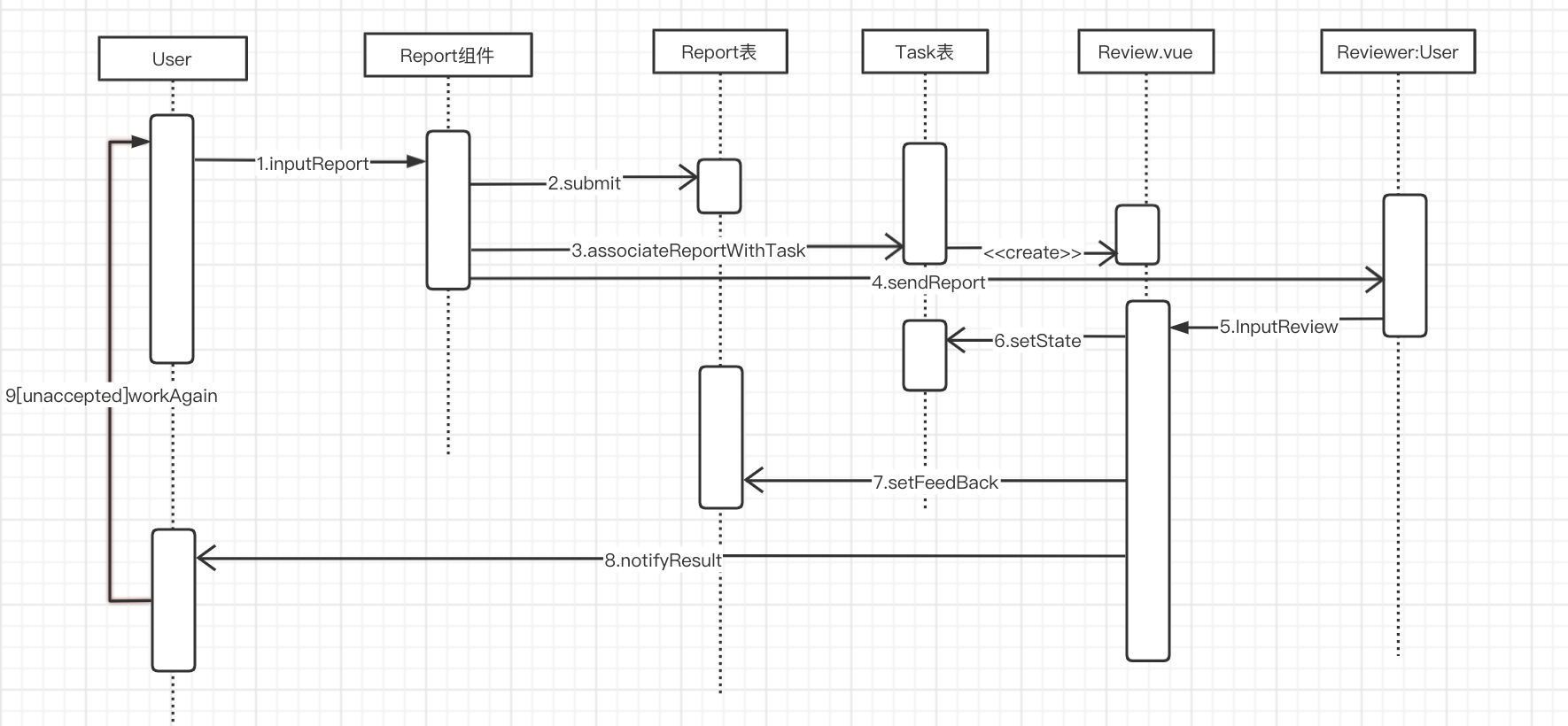


图3.3 提交汇报 顺序图

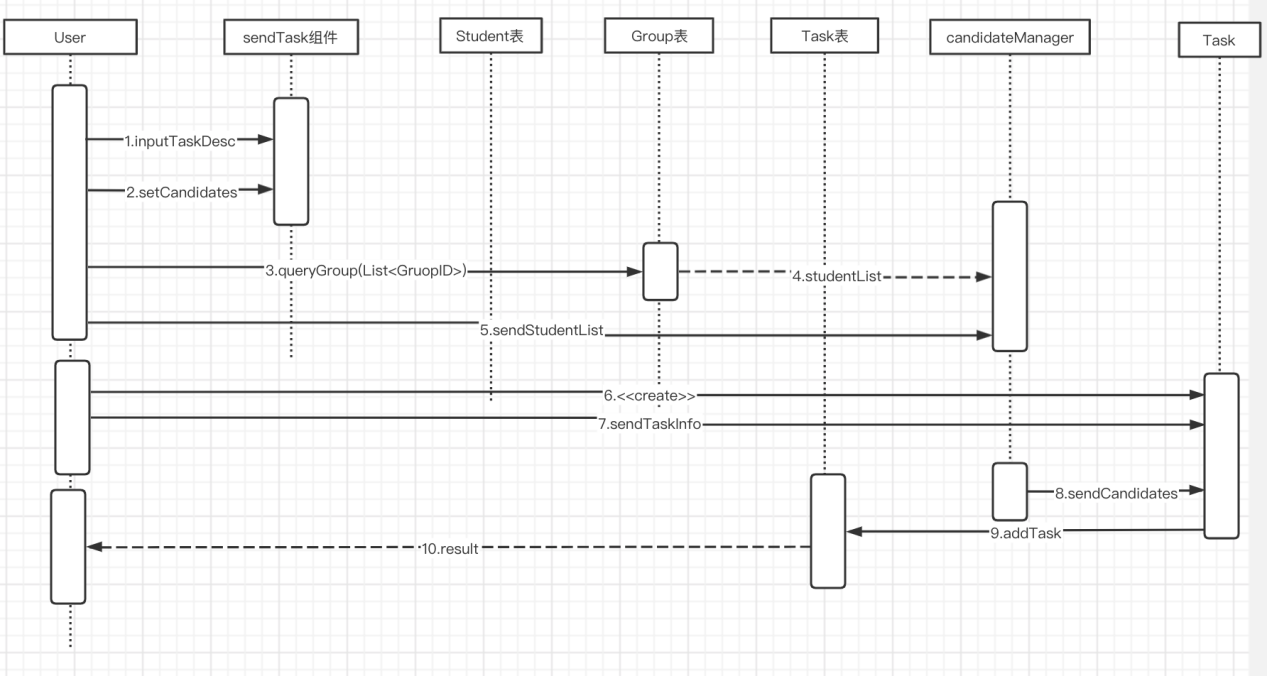


图3.4 发送任务 顺序图

## 3.2 状态模型

状态模型用于描述实体在事件刺激下的反应式动态行为，包含实体所有可能的状态。

对于账号的状态图。

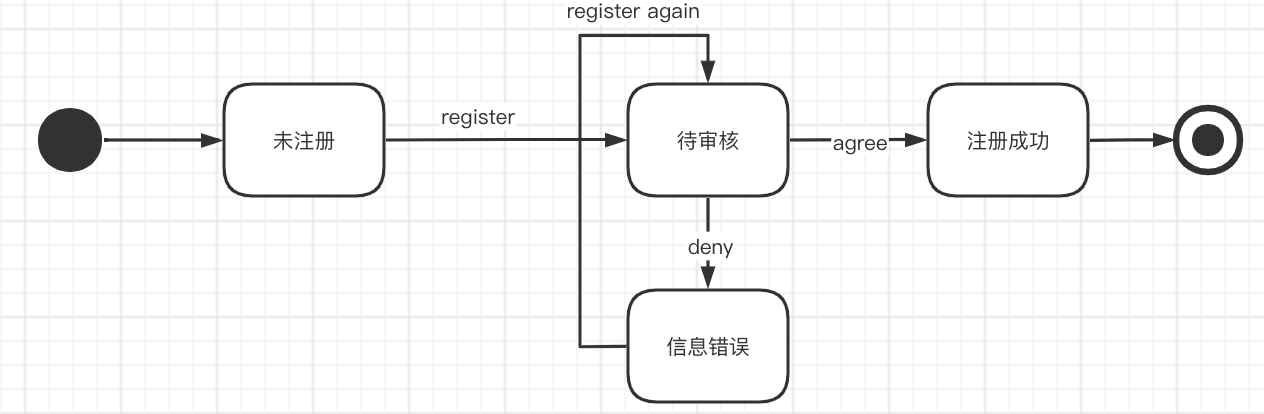


图3.5 账号 状态图

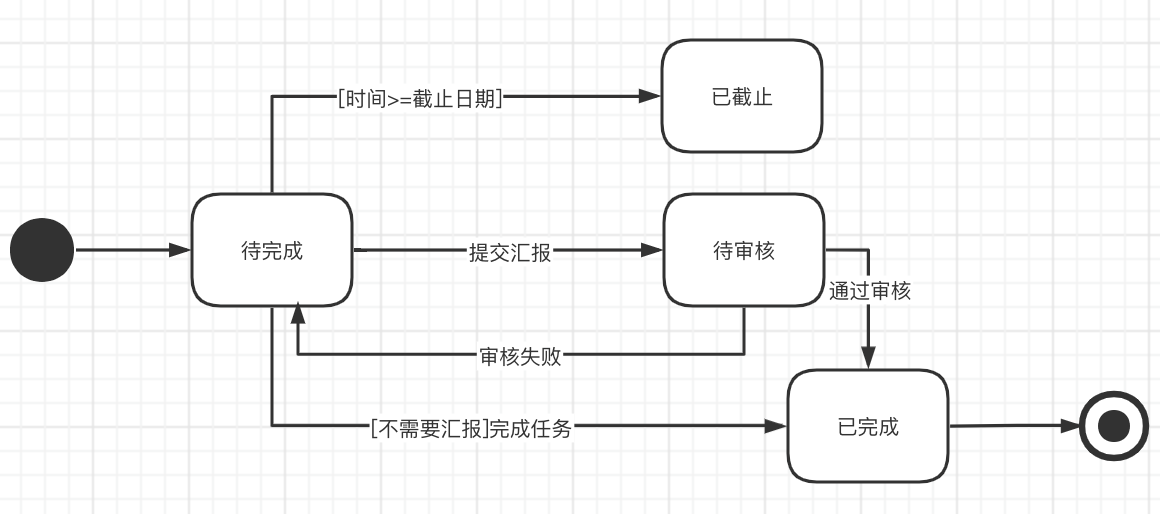


图3.6 任务 状态图

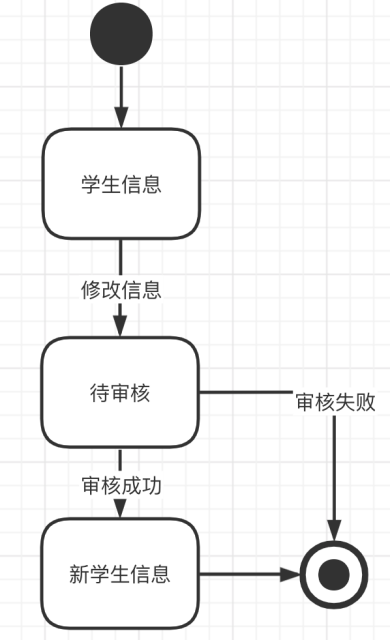


图3.7 修改学生信息 状态图