2018 年春季

图书馆选座系统

需求分析与设计

指导老师：衣杨

组长：　黄敏怡　15331116

组员：　胡子昂　15331111

黄燕蓝　15331121

李沁航　15331159

林彬彬　15331194

汪睿琪　15331293

联系方式：sysuhmy@163.com

目录

一、需求分析

1 图书馆选座系统问题陈述

高校图书馆有着良好的学习环境，但座位却是有限。而在期中期末等时候，随着对座位的需求增大，甚至出现了占座等浪费资源的行为，导致图书馆中座位供不应求的恶行循环，扰乱图书馆管理秩序。本项目以微信小程序为载体，采用C/S 系统，与学生信息对应，为高校图书馆设计了一个图书馆选座系统，实现图书馆自习资源的有效共享，实现预约，选座等功能，能够提高座位的使用效率。

- **学生权限：**

（1）注册登录：学生用户可以在系统上注册一个账号，帐号信息包括学号、姓名、专业，预约历史纪录，违规计数次数。注册帐号后学生可登录进入系统。

（2）查看座位预约情况：登录帐号后学生能够查看图书馆当前或未来空余座位信息。

（3）预约座位：学生可以预约未来某一时间段的空余图书馆座位。一个学生一天内最多可预约 4 次。

（4）查看个人预约记录：学生可以查看个人所有预约记录以及履约情况。但学生无法查看其他用户信息。

（5）修改或删除预约：学生在预约开始时间前可以修改或删除预约。修改或删除预约需要提前 15 分钟，否则将被记录违规一次。

（6）签到：系统通过二维码登录信息来记录用户履约情况。学生在当天预约时间进入图书馆扫二维码，视为签到完成，预约成功。若在预约开始时间后 10 分钟，学生仍未签到，则预约失败，无法履约学生将会被记录违规一次。

（7）签退：学生学习完毕，可在预约结束时间前或预约时间到达时进行签退。一旦超过预约时间，学生需再次预约申请空余座位。

- **管理员权限：**

（1）管理座位信息：管理员能够查看图书馆所有座位的预约情况。同时，对于图书馆内可用座位资源，管理员能够对其进行增加、删除(如撤掉了一排桌椅)，保证座位资源信息的及时更新。

（2）管理学生信息：管理员能够查看所有学生的帐号信息。同时，管理员通过不定时查看图书馆座位占用信息，若发现有学生提前离开且未签退，则对此学生记录违规一次。一旦一个学生在一个月内违规 3 次，则在之后一周内该学生将无法预约图书馆座位。

2 需求分析

2.1 图书馆选座位系统用例析取

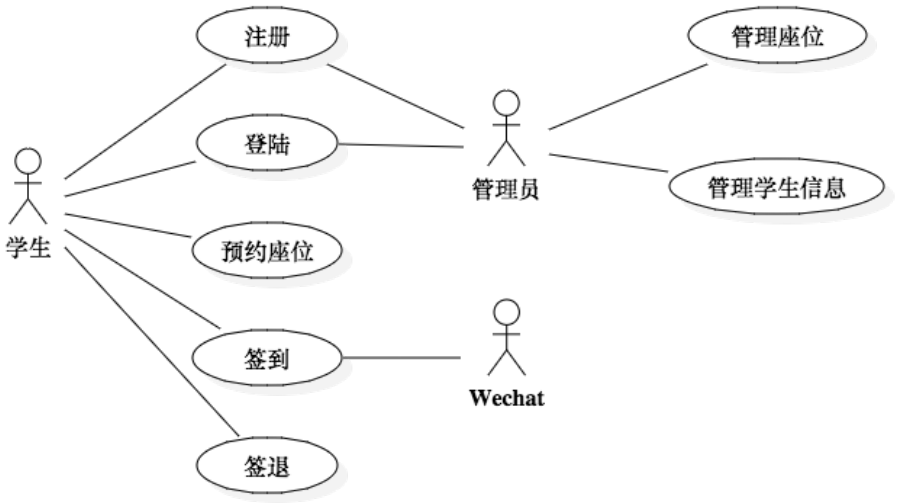

图 1.1 用例图

2.2 图书馆选座位系统用例规约

➢ 注册用例的用例规约

1. 简要说明

本用例允许学生在图书馆选座系统注册个人信息帐号。

2. 事件流

2.1. 基本事件流

用例开始于学生选择注册帐号。

（1） 系统获取学生的 OpenID，进行账号绑定

（2） 系统要求学生填写相关信息(NetID、密码、学校)

（3） 一旦学生提供了所需要信息，系统提交用户信息并保存

2.2. 备用事件流

2.2.1. 学生填写信息中不符合规定

如果学生填写的信息不符合规定，本用例将被重新开始。

2.2.2. 学生帐号已被注册

如果学生已被注册，系统提示用户进入登录界面，本用例将被取消。

3. 特殊要求

无

4. **前置条件**

学生为本校学生。

5. **后置条件**

如果用例成功，系统添加该学生信息会改变，否则系统状态不变。

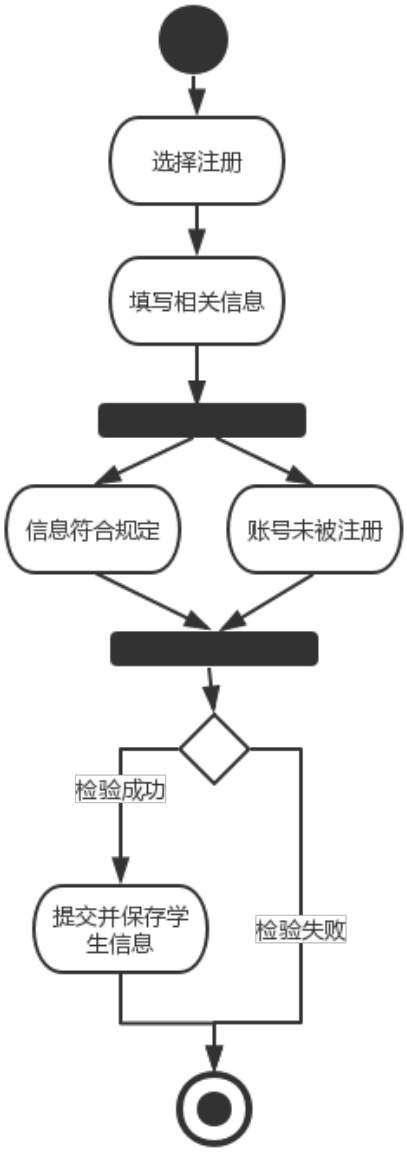

图 1.2 注册用例活动图

➢ 登录用例的用例规约

1. **简要说明**

本用例允许学生或管理员登录进入图书馆选座系统。

2. **事件流**

2.1. **基本事件流**

用例开始于学生或管理员登录进入系统。

系统自动提交 OpenID 信息，系统通过 OpenID 判断用户类型：

（1） 如果用户类型是管理员，用户以管理员身份登录进入系统。

（2） 如果用户类型是学生，用户以学生身份登录进入系统。

2.2. 备用事件流

如果 OpenID 信息不存在，注册用例将被重新开始。

3. 特殊要求

无

4. 前置条件

无

5. 后置条件

如果用例成功，用户成功登录系统并获得相应权限进行操作，否则无法进入系统。

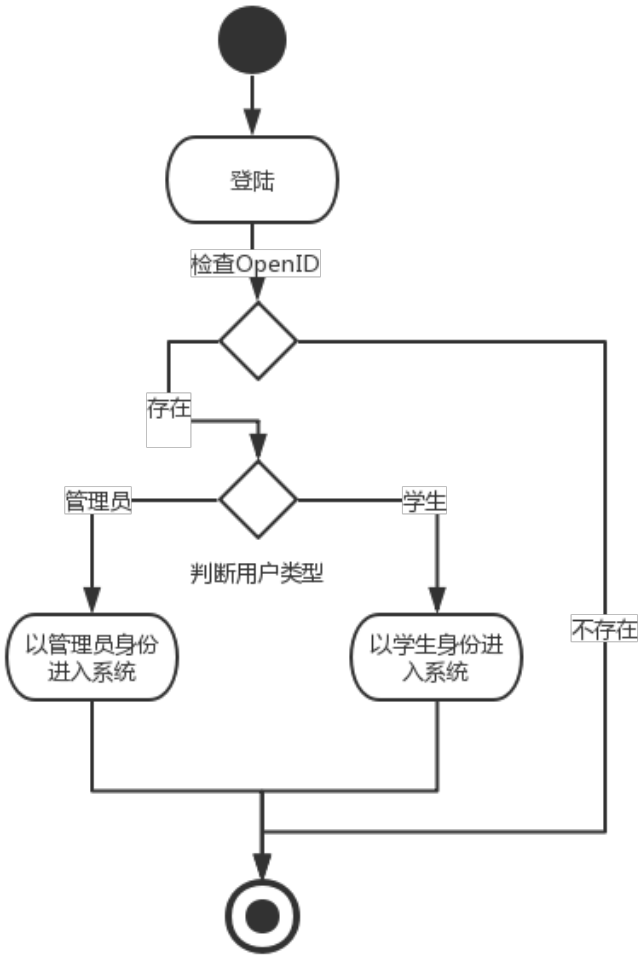

图 1.3 登录用例活动图

➢ 预约座位用例的用例规约

1. 简要说明

本事例允许学生预约/取消预约未来 2 天以内的图书馆座位。学生可以自己选择预约未来 2 天以内任何时间段内，未被其他人预约的图书馆座位，也可以取消预约之前预订过的座位。系统提供未来 2 天以内的图书馆座位预约情况。

2. 事件流

2.1. 基本事件流

用例开始于学生选择预约座位，或者取消自己已经预约过的座位。

（1） 系统要求学生指定接下来进行的操作（预约，取消预约座位）

（2） 一旦学生选择了一个操作，以下一条事件流将被执行：

a) 如果选择的是“预约座位”，预约座位子事件流将被执行。

b) 如果选择的是“取消预约座位”，取消预约座位子事件流将被执行。

2.1.1. 预约座位

（1） 学生选择一个预约开始时间和一个预约结束时间（即预约时间段）。

（2） 系统向学生展示该预约时间段的所有座位情况，并将座位情况展示给学生。

（3） 学生选择其中一个可预约座位。

（4） 一旦学生选择了要预约的座位，系统提交预约信息。

（5） 执行更新座位状态子事件流。

2.1.2. 取消预约座位

（1） 系统向学生展示所有属于该学生的座位预约记录（座位号和预约时间），并将该信息展示给学生。

（2） 学生选择要取消的预约记录。

（3） 系统提交取消预约信息。

（4） 执行更新座位状态子事件流。

2.1.3 更新座位状态

（1） 系统收到提交的预约信息或取消预约信息。

（2） 系统更新数据库的座位信息和学生信息

2.2. 备用事件流

2.2.1. 学生被惩罚

如果学生正在处于惩罚状态，系统将拒绝该同学的预约请求，本用例将被取消。

2.2.2. 学生选择预约时间段距当前时间间距太小

如果学生在预约座位界面停留太久，使得预约时间段距离当前时间间距太小（小于 15 分钟），系统将拒绝该预约请求，本用例将重新开始。

2.2.3. 预约情况冲突

选择的座位如果不符合条件(预约已预约的座位,取消预约未预约的座位),系统将返回错误信息，本用例将重新开始。

2.2.4. 取消预约时间与预约开始时间距离少于 15 分钟

学生若需要取消预约，需提前 15 分钟取消。否则虽然依然可以取消预约座位，但该学生将被记录违规一次。

3. 特殊要求

无

4. 前置条件

用户已经登陆系统并且该用户已经进行过学生认证。

5. 后置条件

如果用例成功，系统中的座位预约情况会改变，否则系统状态不变。

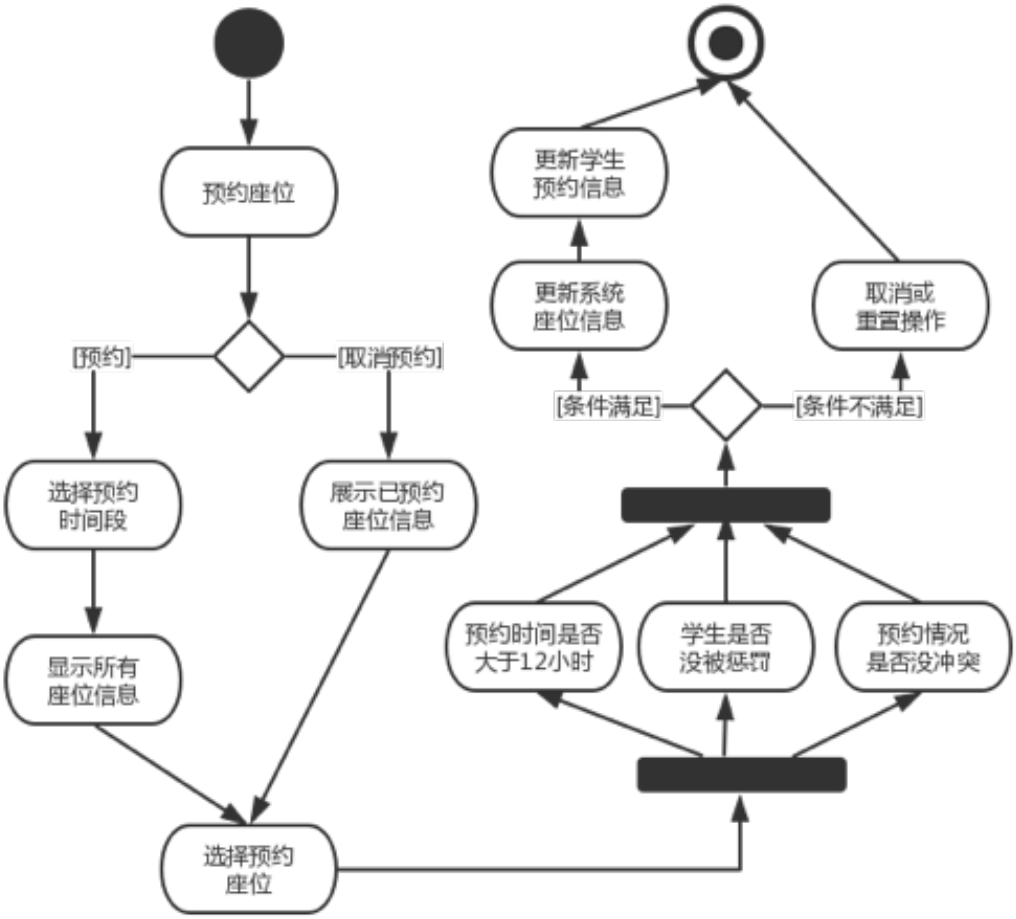

图 1.4 预约座位用例活动图

➢ 签到用例的用例规约

1. 简要说明

本用例允许学生对提前预约好的座位进行签到。

2. 事件流

2.1. 基本事件流

用例开始于学生选择预约座位，或者取消自己已经预约过的座位。

（1） 系统跳转到扫一扫界面提示学生扫描座位上的二维码进行签到
（2） 学生扫描座位上的二维码
（3） 系统根据扫描的二维码以及当前时间和学生的预约信息进行匹配
（4） 执行信息匹配子事件流

2.1.1. 信息匹配

系统根据二维码代表的座位序号以及当前时间与学生端的所有预约信息进行匹配，匹配成功，返回正确信息，更新座位状态，更新学生预约信息状态。

2.2. 备用事件流

2.2.1. 学生的预约信息中无该座位的相关预约信息

如果学生的预约信息中没有其签到的座位，即扫码信息不匹配，本用例将被取消。

2.2.2. 学生签到时间提前超过 10 分钟

如果学生提前签到且签到时间距离预约开始时间超过 10 分钟，系统将拒绝该请求，本用例将被取消。

2.2.3. 学生签到时间迟于 30 分钟

如果学生迟到，且时间超过 30 分钟，系统将拒绝该签到请求，增加该用户的惩罚次数一次，并将此次预约记录取消，本用例将被取消。

3. 特殊要求

无

4. 前置条件

用户已经登陆系统。

5. 后置条件

如果用例成功，系统中的座位状态以及学生预约信息会改变，否则系统状态不变。

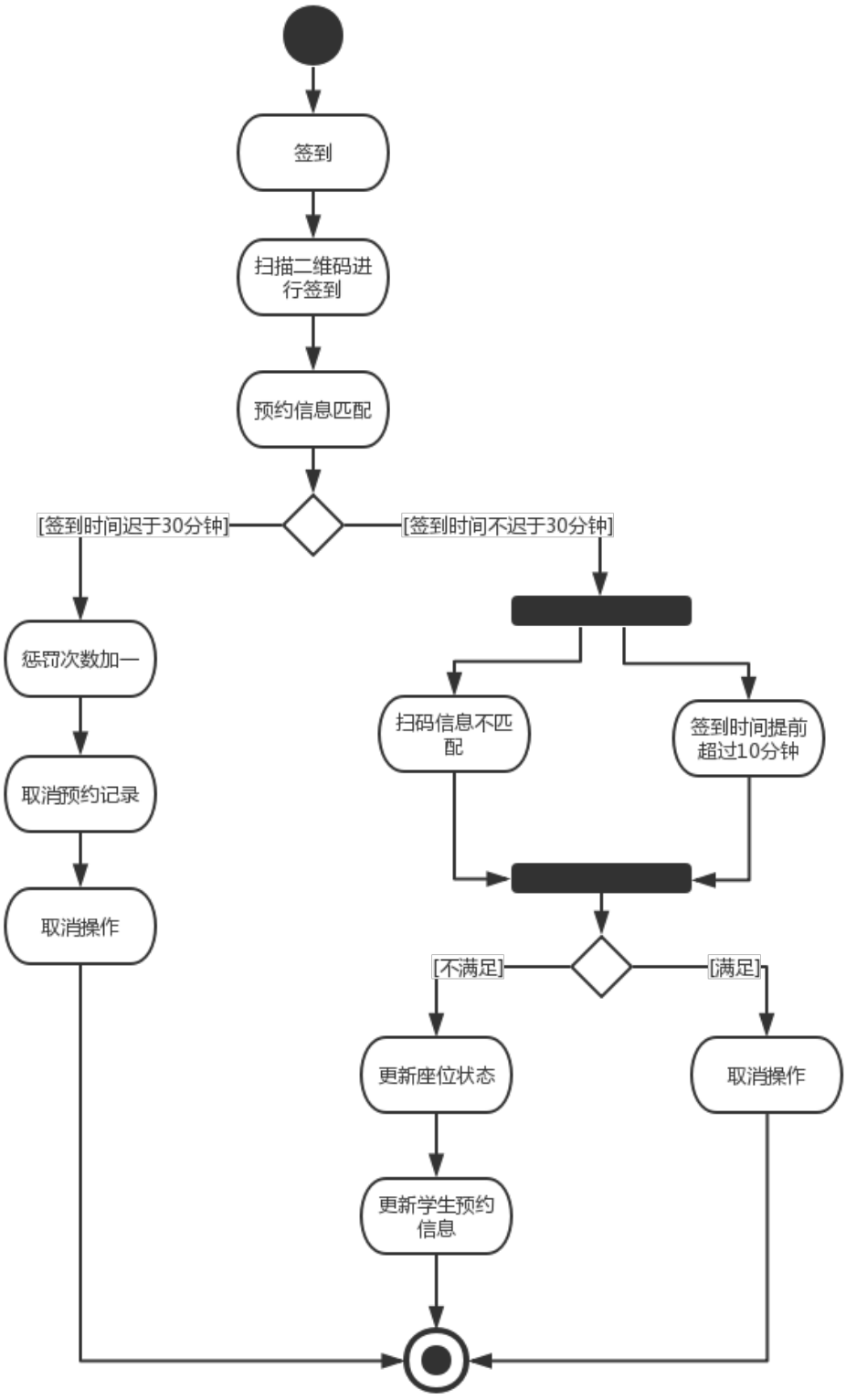

图 1.5 签到用例活动图

➢ 签退用例的用例规约

1. 简要说明

本用例允许学生对当前座位（即提前预约且已经签到过的座位）进行签退。

2. 事件流

2.1. 基本事件流

用例开始于学生对已预约且已签到的座位进行签退：

（1） 系统根据时间自动帮学生签退

（2） 系统自动判断座位信息，更新座位状态，并更新学生端预约信息状态。

2.2. 备用事件流

无

3. 特殊要求

无

4. 前置条件

用户已经登陆系统。

5. 后置条件

如果用例成功，系统中的座位状态以及学生预约信息会改变，否则系统状态不变。

图 1.6 签退用例活动图

- 管理座位用例的用例规约

1. 简要说明

本用例允许管理员管理图书馆内可用座位资源，可以在系统中管理图书馆座位的数量、位置，改变座位的可被预约时间。

2. 事件流

2.1. 基本事件流

用例开始于管理员选择管理座位，此时管理员可以看到图书馆的座位表。系统要求管理员指出要执行的操作：管理座位分布，管理座位可预约情况。一旦管理员选择了一个操作，以下一条子事件流将被执行：

（1） 如果选择的是“增减座位”，增减座位子事件流将被执行。

（2） 如果选择的是“修改可预约时间”，修改可预约时间子事件流将被执行。

（3） 如果选择的是“查看座位预约信息”，查看座位预约信息子事件流将被执行。

2.1.1. 增减座位

（1） 系统将图书馆的座位编号列表显示给管理员。

（2） 如果要删除座位，选择编号列表中的编号，删除该编号，即对该座位进行删除，该座位不再可预约。

（3） 如果要增加座位，在编号列表中添加编号，即增加一个新的座位。

（4） 增加座位的同时，在座位表中添加该座位信息，包括其位置（即楼层）和默认可预约时间（默认可预约时间为图书馆开馆时间-闭馆时间）。

（5） 执行更新座位表子事件流。

2.1.2. 修改可预约时间

（1） 管理员得到当前的座位表的具体信息

（2） 管理员修改可预约时间，选择执行新的可预约时间的座位编号范围，进行座位信息更新

（3） 执行更行座位表子事件流。

2.1.3. 查看座位预约信息

（1） 管理员得到当前的座位表的具体信息

（2） 管理员选择一个座位，可以查看该座位的座位号，所有可预约时间中已被预约的时间段以及该各个时间段中预约人的学工号，未被预约的时间段。

2.1.4. 更新座位表

系统将修改后的座位具体信息保存，并在原有的座位表基础上更新座位表信息，在修改后的第三天后执行新的座位表信息。

2.2. 备用事件流

2.2.1. 修改的可预约时间不合法

则修改失败，无法更新座位表信息，重新开始本用例。

2.2.2. 未找到座位编号

如果在修改可预约时间和查看座位预约信息子事件流中，系统无法返回学座位表的具体信息，将显示错误消息。管理员确认错误，这时本用例重新开始。

3. 特殊要求

无

4. 前置条件

本用例开始前，管理员必须已经登陆系统，获得管理权限。

5. 后置条件

如果用例成功，图书馆的座位信息被更新，在第三天后使用使用新的数据信息。

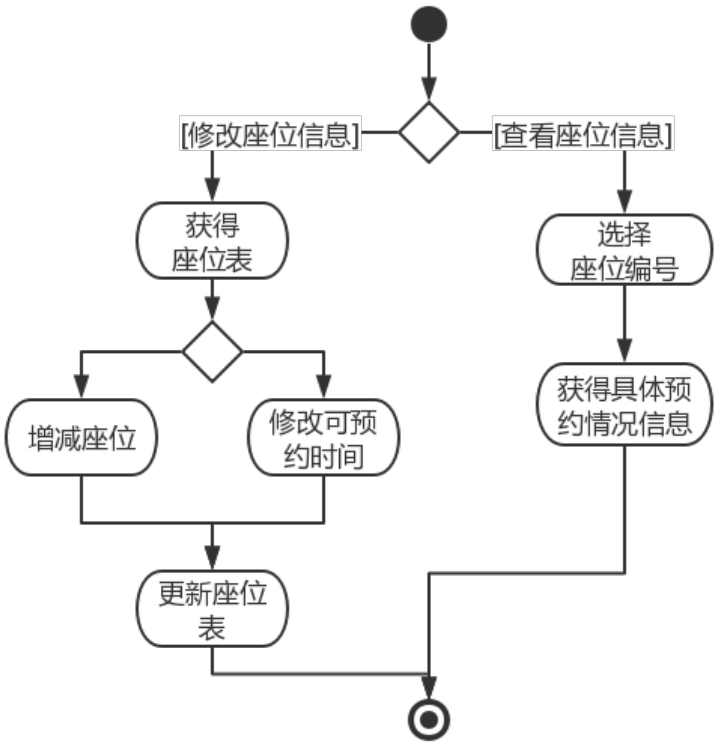

图 1.7 管理座位用例活动图

➢ 管理学生用例的用例规约

1. 简要说明

本用例允许管理员管理学生信息，对学生进行权限修改。

2. 事件流

2.1. 基本事件流

用例开始于管理员在人工巡逻的时候，查到被预约的位置没有人坐，查看座位预约信息，获得学号

（1） 输入学号，获得该学生在系统中的信息

（2） 如果该学生不在惩罚状态，则对其进行违规计数，计数超过三次，则对该学生进行惩罚，修改其在系统中的权限，在惩罚状态中不可进行座位预约。

2.2. 备用事件流

如果某一学生被误设为惩罚状态，证明是管理员操作失误或是系统出错时，可以对学生状态进行修改，解除对其的惩罚。

3. 特殊要求

无

4. 前置条件

管理员必须已经登陆系统，获得管理权限，并且知道学生的学号。

5. 后置条件

本用例结束后，如果有学生信息权限有被修改过，则该学生的信息在系统中被更新。

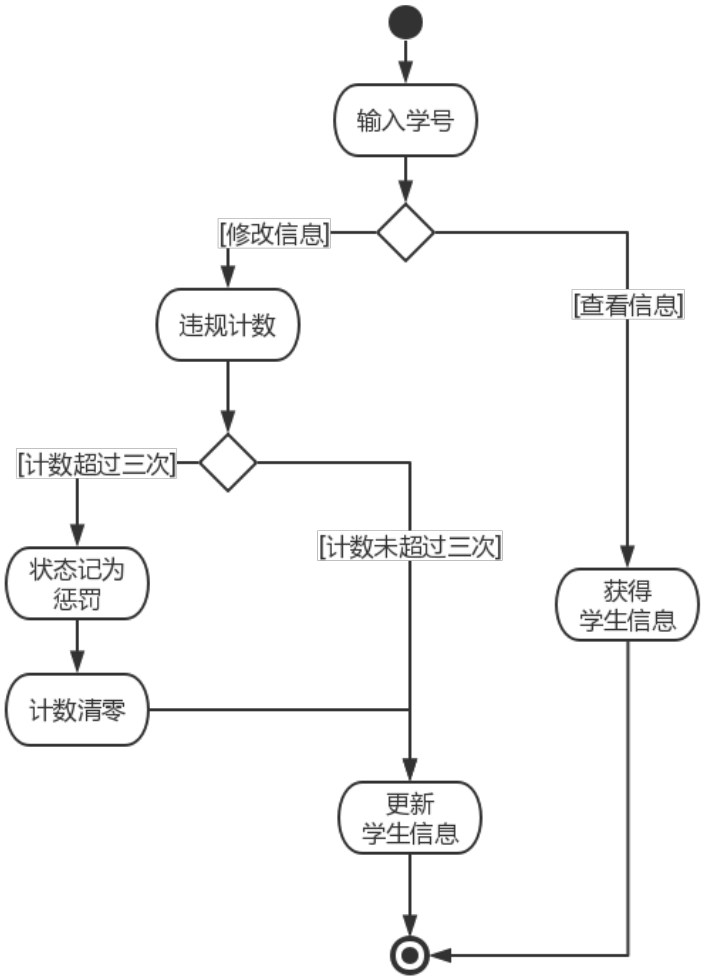

图 1.8 管理学生信息用例活动图

3 补充归约

本部分为系统的整体非功能性补充要求。

（1） 兼容性

系统以微信小程序实现，面向所有微信用户

（2） 可靠性

保证图书馆座位信息的实时可靠性，预定的可靠性，并提供不间断的服务

（3） 性能

系统可支持 1000 个在线用户；用户每个操作的响应时间都在 5 秒以内

（4） 易用性

系统的界面简洁易懂，操作逻辑简单合理，操作效率高

（5） 安全性

系统对于用户的操作记录以及个人信息不被非法获取，或丢失

系统确保学生与管理员只能在各自允许的权限范围内操作

（6） 可拓展性

支持多个学校使用本系统

（7） 设计约束

使用 wechat web 进行前端开发

使用 golang 语言进行后台开发

4 术语表

名词术语	定义	英文
学生用户	已经在系统中注册了并标记为本校学生的用户，具有一般用户权限。	student
管理员	已经在系统中注册了并标记为管理员的用户，具有管理员权限。	administrator
帐号信息	用户在注册时填入的信息，学生用户帐号信息包括学号、姓名、密码、专业、个人预约记录、违规记录次数。管理员用户帐号信息包括工号和密码。	info
违规计录次数	学生用户帐号信息之一，记录学生违规使用系统次数。	violation
个人预约记录	学生用户帐号信息之一，记录学生预约图书馆座位的所有历史情况，包括预约时间，预约座位，是否履约。	reservation
座位信息	图书管所有座位的相关信息，包括所有时间段每个座位预约记录。	seat_info
空余座位信息	图书馆座位信息之一，记录所有时间段内空余的座位信息。	available_seat
预约座位信息	图书馆座位信息之一，记录所有时间段内被预约的座位信息。	reservated_seat
惩罚状态	当学生违规记录次数达到三次，则进入惩罚状态，一旦进入惩罚状态一周内该学生用户无法预约图书馆座位。一周后违规记录次数清零，跳出惩罚状态。	punishment

表 1.9 术语表

二、架构设计

1 架构描述

1.1 简要描述

在本系统中，使用 MVC 作为系统的架构。

1.2 设计框架图

图 2.1 设计架构图

1.3 系统架构描述

(1) View

在小程序中通过使用 POST 和 GET 的方式发出请求，使得用户通过使用移动设备借助网络进行操作。在本应用中，用户在 view 层进行操作，发送请求，并且接收来自 controller 层所反馈的信息。

(2) Controller

通过接收 HTTP 请求，并且根据请求调用相应的逻辑，并且将结果返回给客户端。

(3) Model

在逻辑层处理所有的逻辑，其中包括数据的逻辑是否符合要求等，在本层仅仅注重逻辑的处理，而不注重数据的存取的细节。

(4) DAO

作为数据源层，用于进行数据的交换，主要用于从 DB 中获取数据然后传递给 Model 层进行逻辑的处理，并且接收数据交给 DB 进行数据的修改等操作。

(5) DB 数据库

用于储存所有的用户信息，座位信息，此处注重数据的读写存储的细节方法。

2 关键抽象

2.1 学生信息

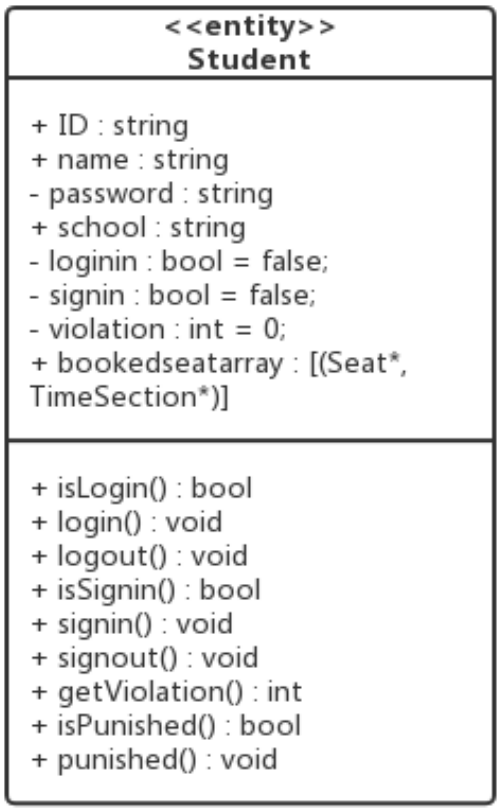

图 2.2 学生类

2.2 管理员信息

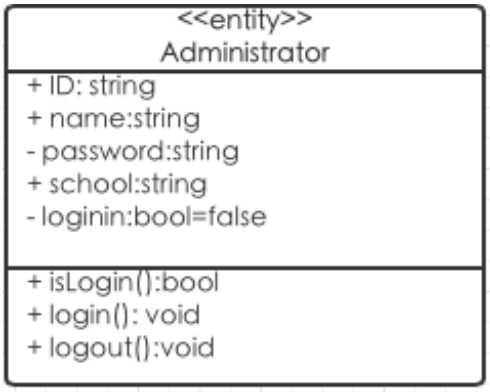

图 2.3 管理员类

2.3 座位信息

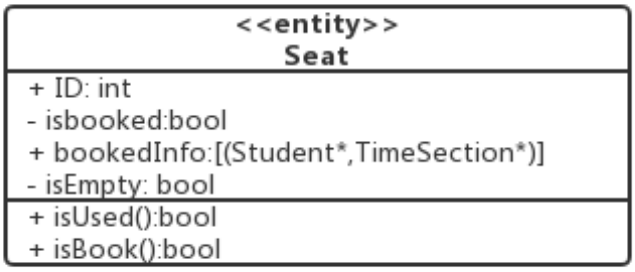

图 2.4 座位类

2.4 时间范围

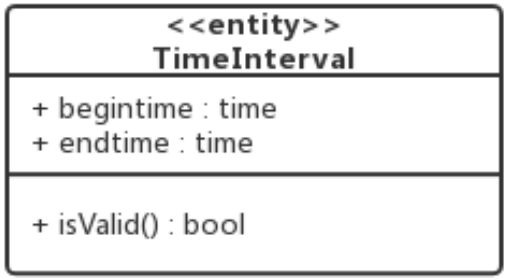

图 2.5 时间类和时间范围类

三、类的析取

1 预约座位用例的用例析取

➢ 类图：

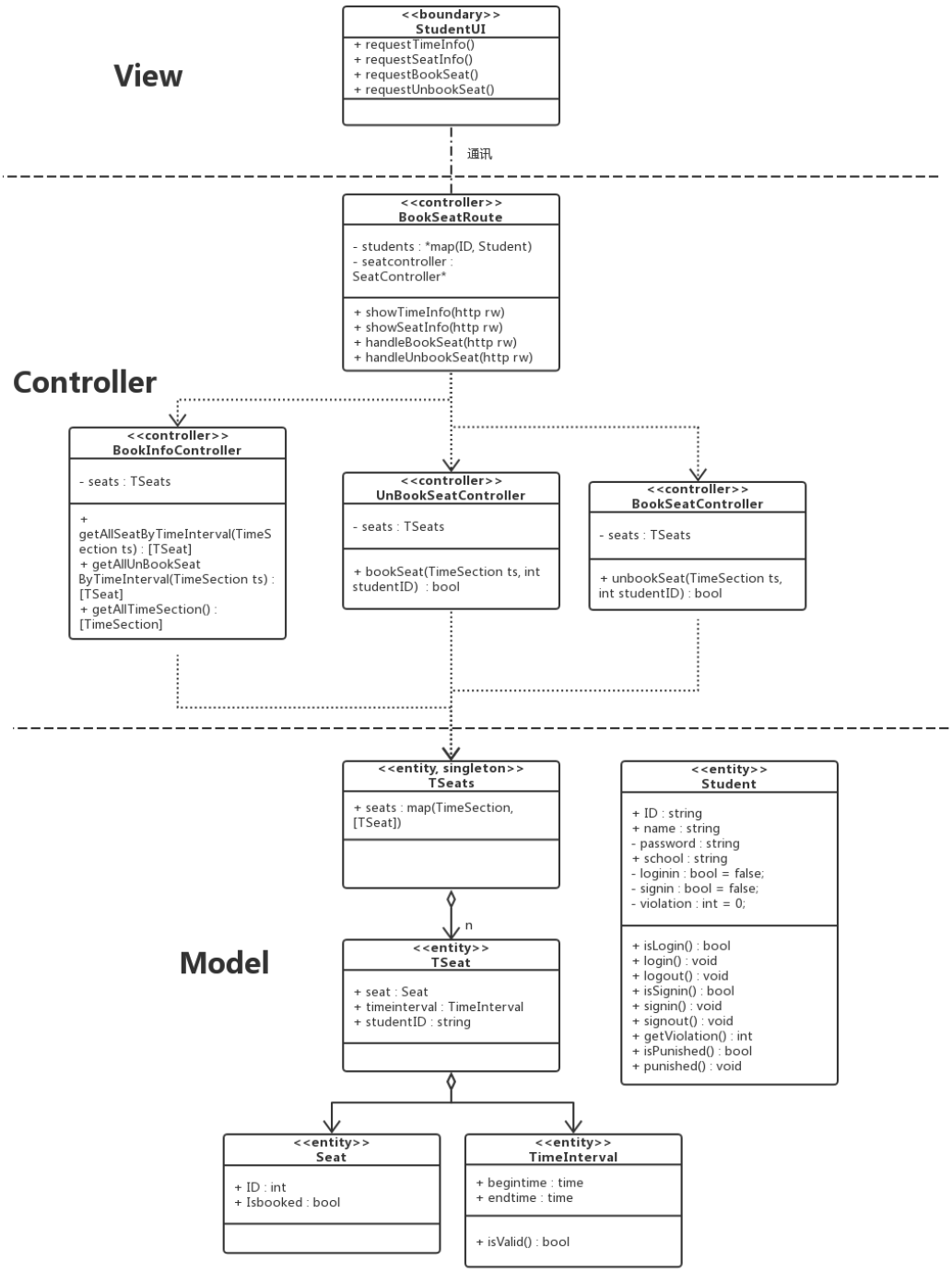

图 3.1 预约座位用例类图

- **时序图：**

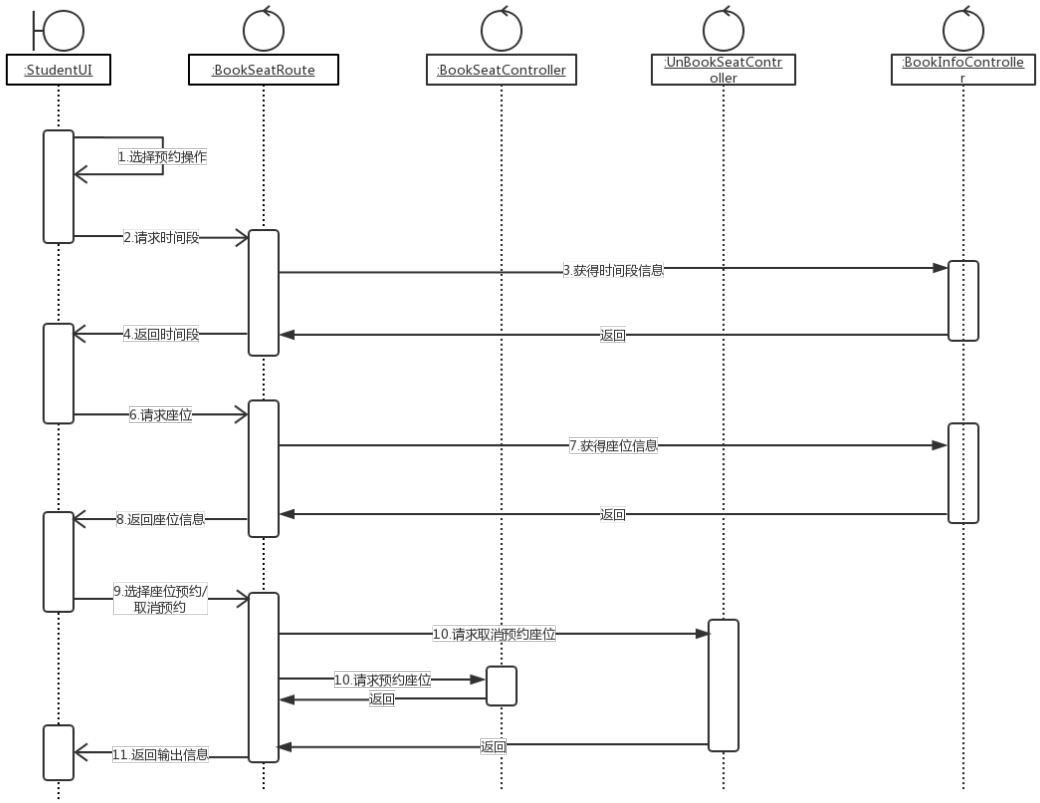

图 3.2 预约座位用例时序图

- **协作图：**

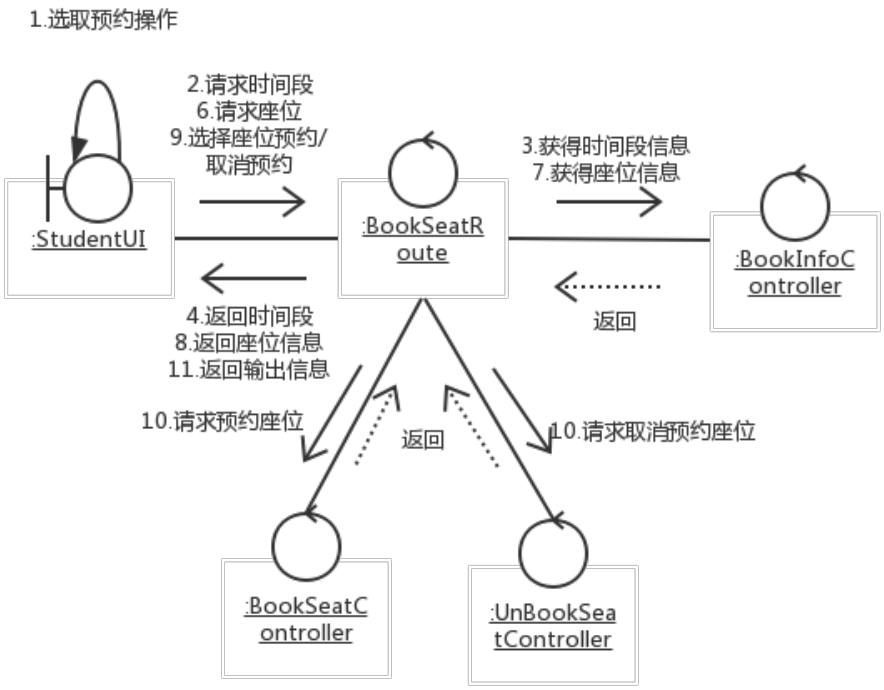

图 3.3 预约座位用例协作图

2 签到用例的用例析取

➢ 类图：

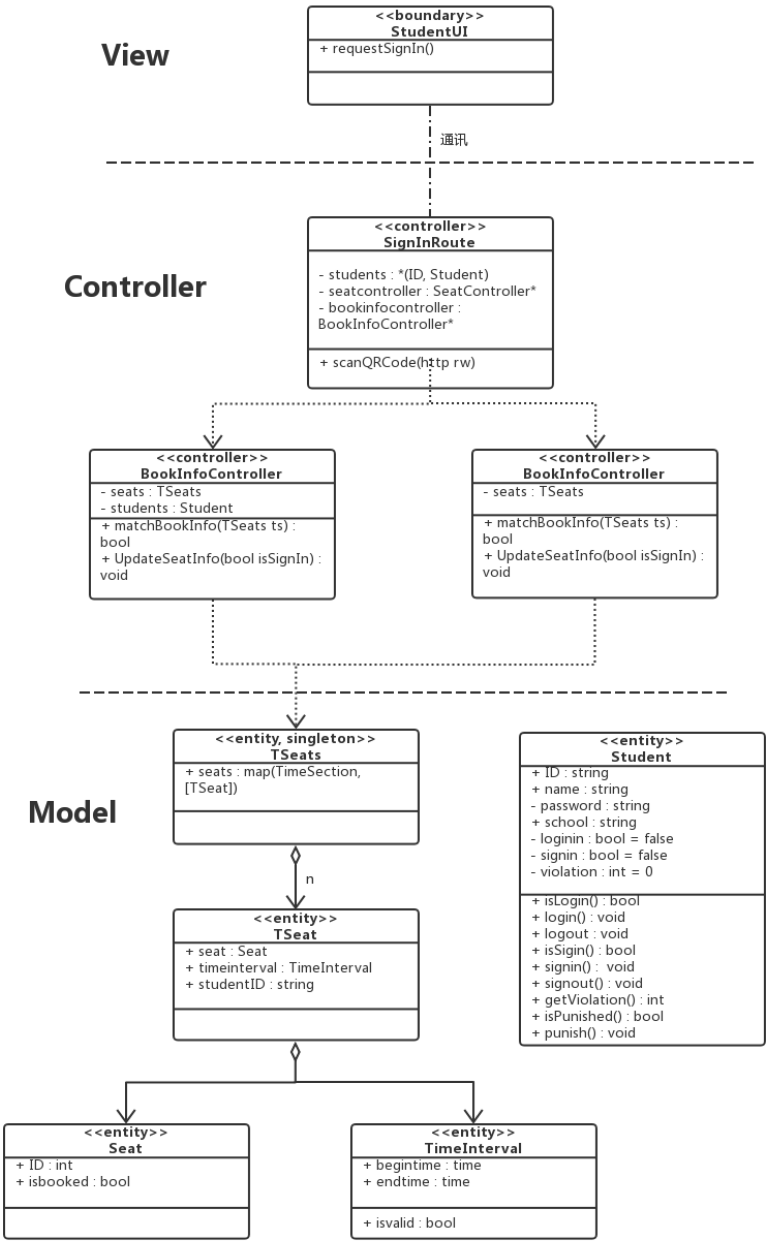

图 3.4 签到用例类图

- **时序图：**

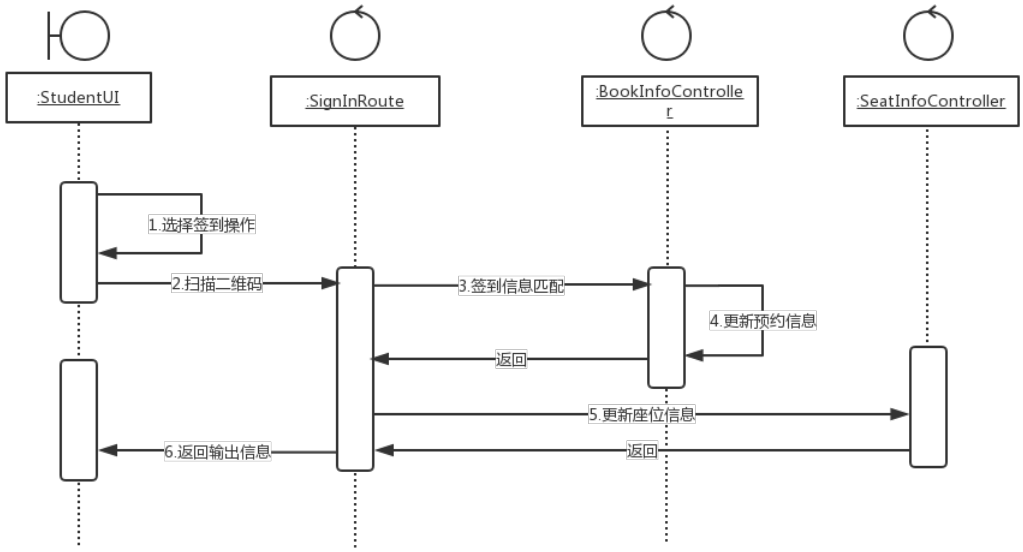

图 3.5 签到用例时序图

- **协作图：**

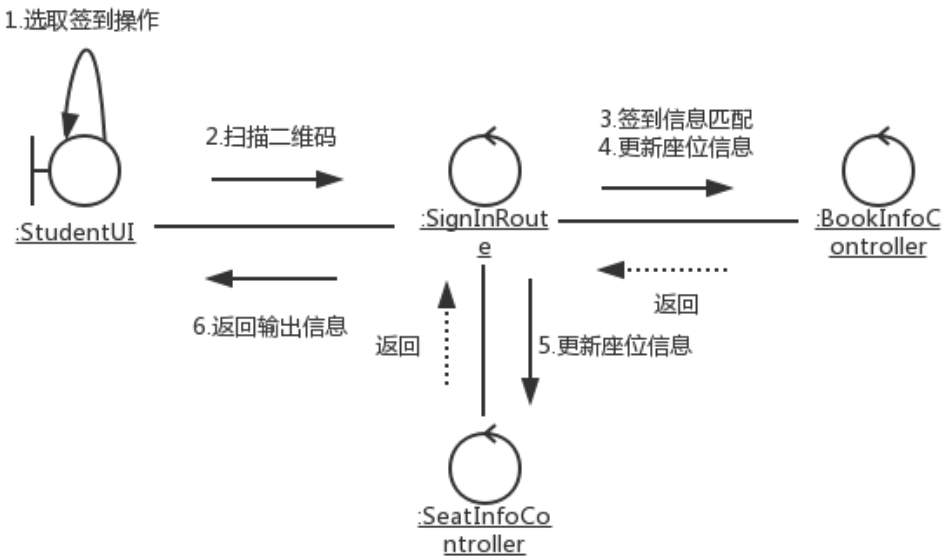

图 3.6 签到用例协作图

3 管理座位用例的用例析取

➢ 类图：

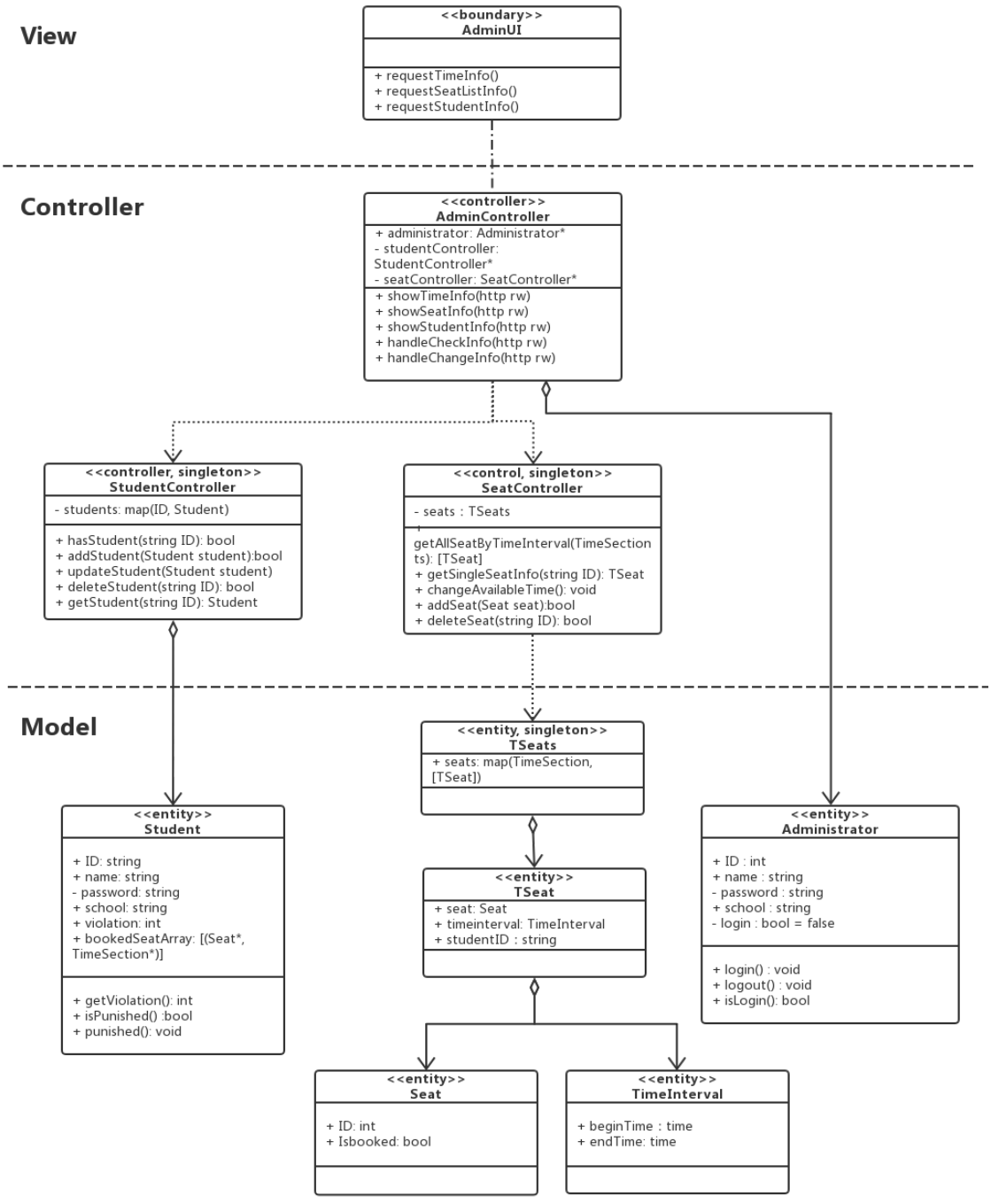

图 3.7 管理座位用例类图

- **时序图：**

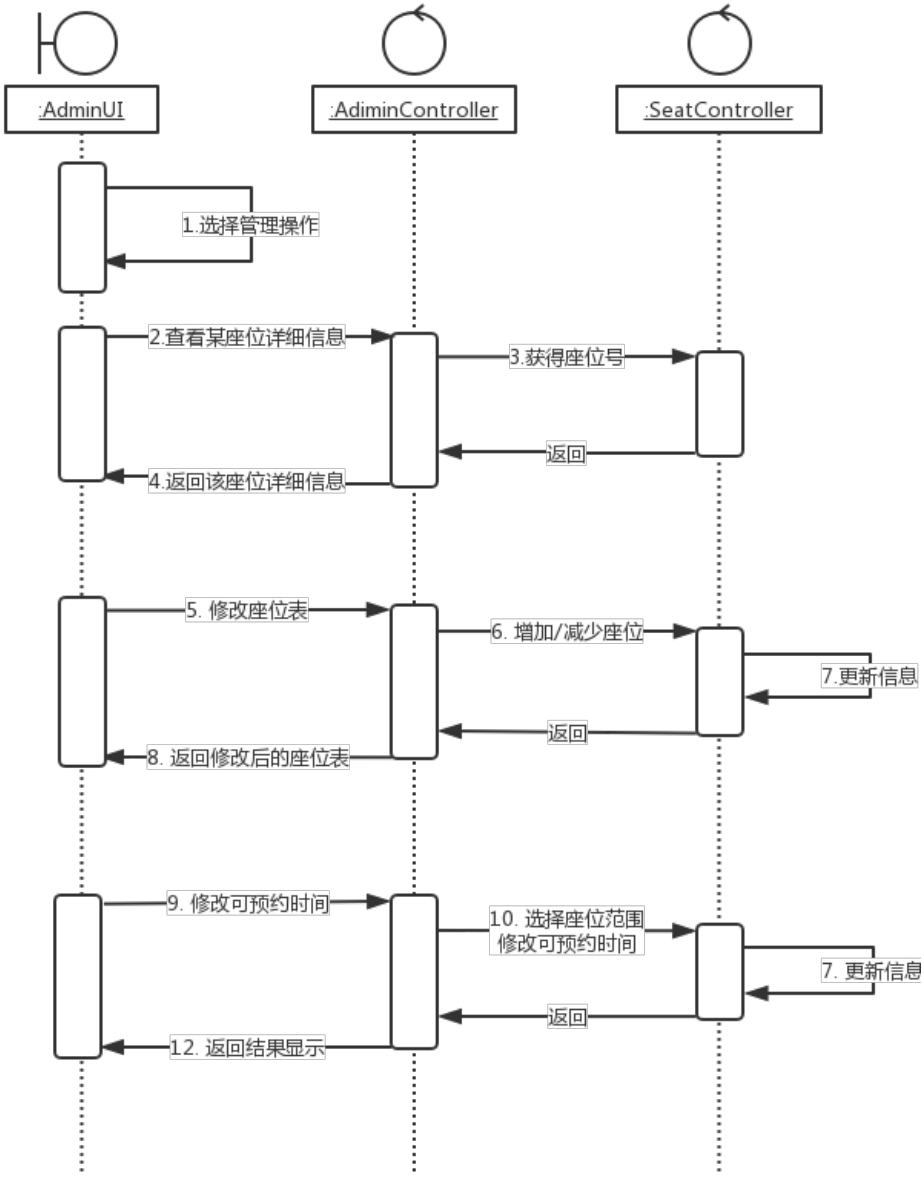

图 3.8 管理座位用例时序图

- **协作图：**

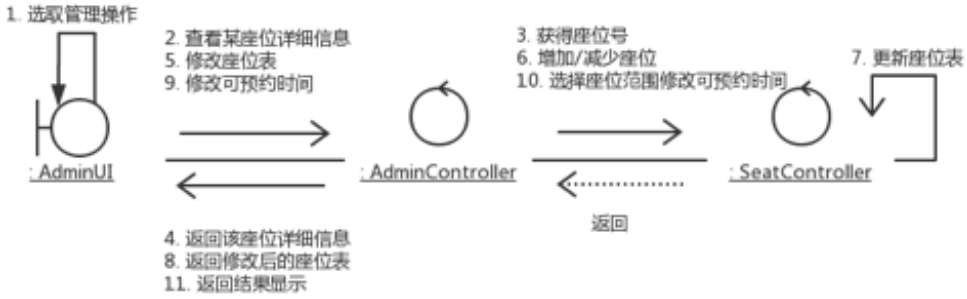

图 3.9 管理座位用例协作图

四、子系统及其接口设计

1 分析合并类

本节将析取出来的边界类、控制类、数据类进行合并整理，得到系统的合并类图。

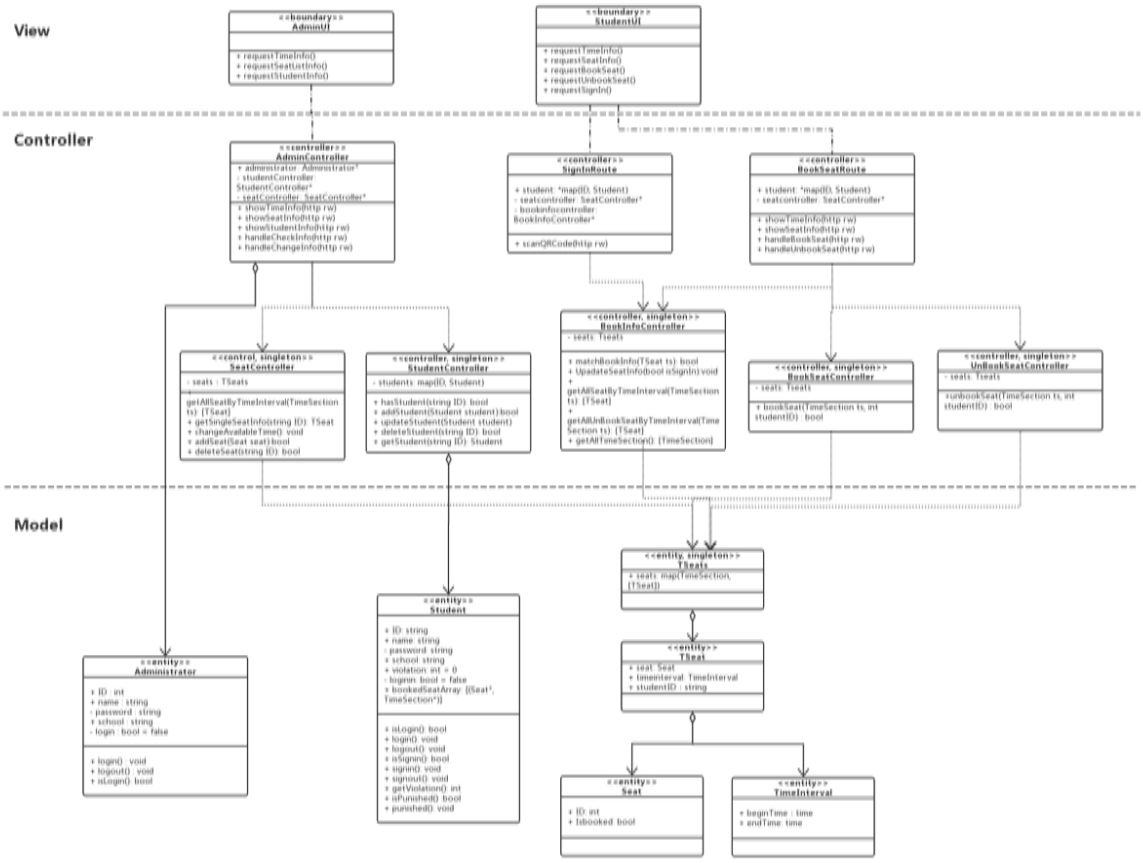

图 4.1 合并类图

2 确定设计类

本项目组对析类进行了分析与检查，以确定其是否能成为设计类。经过分析发现，所有分析类均为单逻辑，不需要进行类的分解或合并，因此不做修改。

3 划分子系统

经过本项目组分析，本系统无需进行子系统设计及其接口设计。

五、部件设计

1 分析并发需求

该系统需要满足以下三个并发需求：

1） 同一时间，同一个学校的不同学生可能会同时发出操作请求，此时需要并发进行

2） 学校的管理员可能会在学生操作时修改座位信息和学生信息，此时需要并发进行

3） 不同学校可能会共用同一个服务器，此时需要并发进行

2 针对某个需求的设计方案

对三个并发需求，共提出两个设计方案：

1） 针对第一个和第二个需求：当收到一个新请求，主线程创造新的线程，新的线程执行收到的请求，主线程则继续等待接收。并且在进行数据库操作前，添加读写锁，保证不会出现读写冲突。

2） 针对第三个需求：多个学校请求公用一个服务器 CPU 资源。由于不同学校的数据不互通，将不同学校的数据放入不同数据库，或同一个数据库中的不同库或不同表中。

3 生命周期

新线程的周期开始于主线程收到一个新请求，结束于线程执行函数结束并回复完毕。

4 映射到现实系统

1） 使用 negroni 包进行线程创建调度。

2） 构建 mutexmanager 类，管理数据库读写锁。

3） 封装数据库接口，使不同学校的座位信息存储到同一个数据库的不同表中。

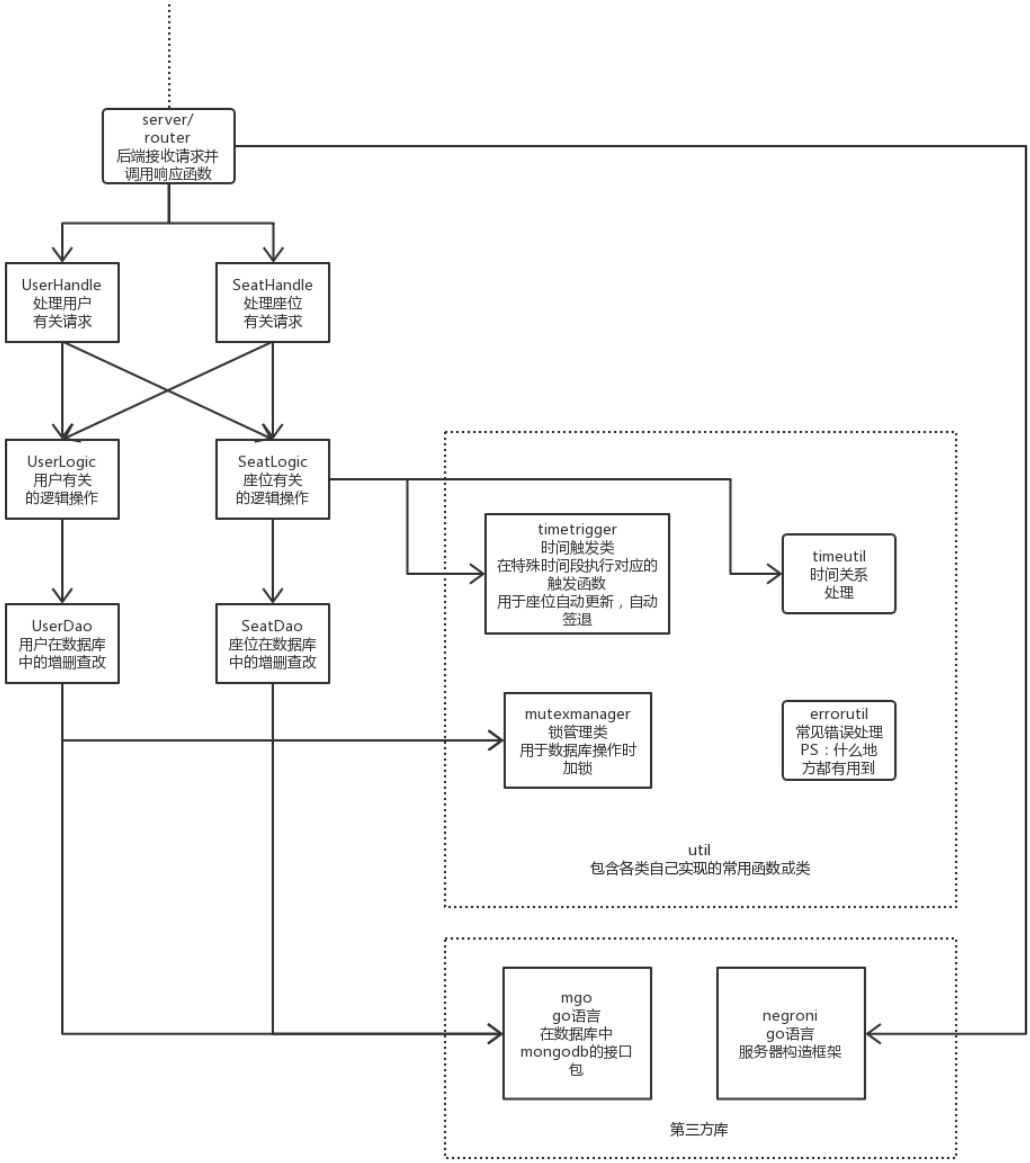

图 5.1 后端实现的逻辑结构图