



# 课程签到管理系统

## 设计文档

组 长：侯培中

组 员：黄楠绚 解晶璇 侯潇芮 谢轩 何嘉敏

指导老师：衣杨

# 第一章 需求分析

## 1.1 问题陈述

目前,我国的高校普遍实行考勤制度,而“点名”则成为了主要的考勤方式。对于老师而言,可以通过点名考勤来获取学生的出勤率,将考勤结果作为期末总评成绩的部分参考,同时对同学而言,则能够起到监督作用,提高学习的积极性,可谓是教学中的重要环节。

传统的点名方式往往是老师在课堂上逐一点名,学生举手喊到,或是派发签到表,同学签名。这种方式既浪费课堂时间,也不能完全避免学生代签到/签到的情况,同时,纸质签到还存在着数据录入过程繁杂、难以统计等缺点,考勤逐渐成为了教师和学生的负担。

在科技水平快速发展的今天,我们希望通过已有的资源和工具来简化、优化这一过程。因此,我们为教师和学生设计了一个课堂签到管理系统,让考勤从此变得简单、有条理!

基于此,本项目组将设计和开发一个课堂签到管理系统。该系统前端基于vue框架,后端基于koa2框架。该系统使用者面向管理员、教师和学生这三类人,在实现基本的签到功能的同时,我们还围绕实际课堂需求添加了其它相关功能。具体功能如下:

管理员:登陆系统,学生信息管理,教师信息管理,课程信息管理,具有该系统的最高管理权限。

教师:登录系统;发起签到二维码供学生签到;查询签到结果及导出签到信息。

学生：扫描二维码进行签到。

## 1.2 用例析取

用例析取是从待完成的软件项目中获取用户需求的主要方式。其主要目的是：从用户的预期中获得系统基本原型；便于与用户确认系统的功能；定义所有外部可见的行为。经过分析，本系统用例图如图 1-1 所示。

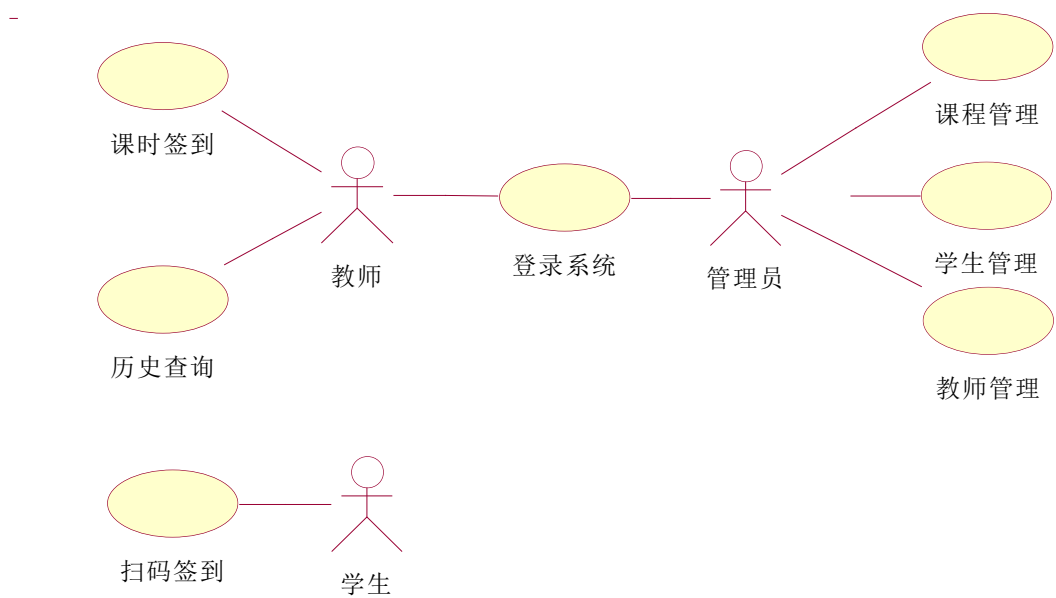


图 1-1 课程签到管理系统用例图

## 1.3 用例规约

### 1 教师管理

#### 1.1 简述

本用例描述管理员添加可供教师登录系统的账号密码以及删除教师的过程。

## 1.2 参与者

管理员。

## 1.3 事件流

### 1.3.1 基本事件流

本用例开始于管理员已登录进本系统。系统于首页引导管理员进行管理教师账号，以下的一条子事件流将被执行：

如果选择的是“添加教师”，添加教师子事件流将被执行；

如果选择的是“删除教师”，删除教师子事件流将被执行。

#### a. 添加教师

1) 管理员选择添加教师；

2) 管理员填写要导入教师信息，包括教师的工号（以下称为教师账号），账号密码，教师的姓名。

3) 管理员点击确认；

4) 系统检查课程信息；

5) 系统提示成功导入教师，并显示账号密码。

#### b. 删除教师

1) 管理员点击教师对应的删除教师按钮；

2) 管理员点击确认；

3) 系统提示成功删除教师。

### 1.3.2 备选事件流

(1) 系统提示该教师账号已存在, 给出错误提示, 管理员确认错误, 系统返回输入信息界面, 核对信息后重新填写或关闭页面。

(2) 系统提示数据库操作失败, 给出错误提示, 管理员确认错误, 系统返回首页。

(3) 系统提示未登录, 给出错误提示, 管理员确认错误, 系统返回登录输入界面, 用户重新登录。

(4) 系统检测到输入信息不全, 给出错误提示, 管理员确认错误, 系统返回输入信息界面, 核对信息后重新填写或关闭页面。

#### 1.4 特殊需求

无。

#### 1.5 前置条件

管理员使用管理员内部机密账号成功登录了本系统。

#### 1.6 后置条件

如果用例成功, 管理员成功添加或删除了教师账号。系统数据库同步更新, 教师可以凭管理员分发的教师账号及密码登录系统和使用系统。

#### 1.7 活动图

添加教师用例的活动图如图 1-2 所示。

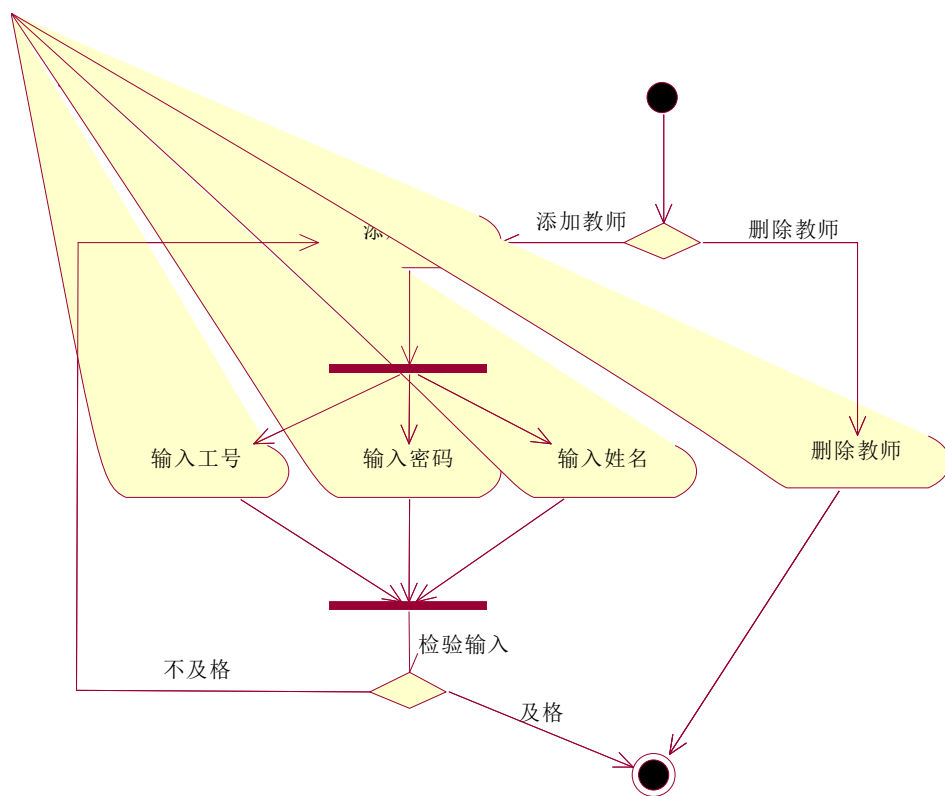


图 1-2 教师管理活动图

## 2 课程管理

### 2.1 简述

本用例描述管理员如何管理课程信息。

### 2.2 参与者

管理员。

### 2.3 事件流

#### 2.3.1 基本事件流

本用例开始于管理员登录系统后，选择课程管理。

(1) 系统于首页列出教师列表。

(2) 管理员点击某一教师，页面跳转到该教师页面，显示该教师所有课

程。管理员对该教师课程进行操作，以下的一条子事件流将被执行：

如果选择的是“导入课程”，导入课程子事件流将被执行；

如果选择的是“删除课程”，删除课程子事件流将被执行。

a. 导入课程

1) 管理员选择导入课程；

2) 管理员填写要导入课程的信息，包括填写课程 ID，课程名称，  
填写课程总课时和上课时间；

3) 管理员点击确认；

4) 系统检查课程信息；

5) 系统提示成功导入课程，并显示课程信息。

b. 删除课程

1) 管理员选择某一课程，点击删除课程；

2) 系统提示管理员确认删除课程及课程信息；

3) 管理员确认删除；

4) 系统提示删除课程成功。

2.3.2 备选事件流

(1) 导入课程失败

当课程信息输入不全或课程 ID 已存在时，给出错误信息，管理员确认错误信息，本用例结束。

(2) 删除课程失败

系统显示删除课程失败信息，管理员确认错误信息，本用例终止。

2.4 特殊需求

无。

## 2.5 前置条件

本用例开始前管理员必须已经凭账号和密码登录进系统。

## 2.6 后置条件

如果用例成功，管理员创建了一门课程或删除了一门课程，更新课程列表。否则系统状态不变。

## 2.7 活动图

课程管理用例的活动图如图 1-3 所示。

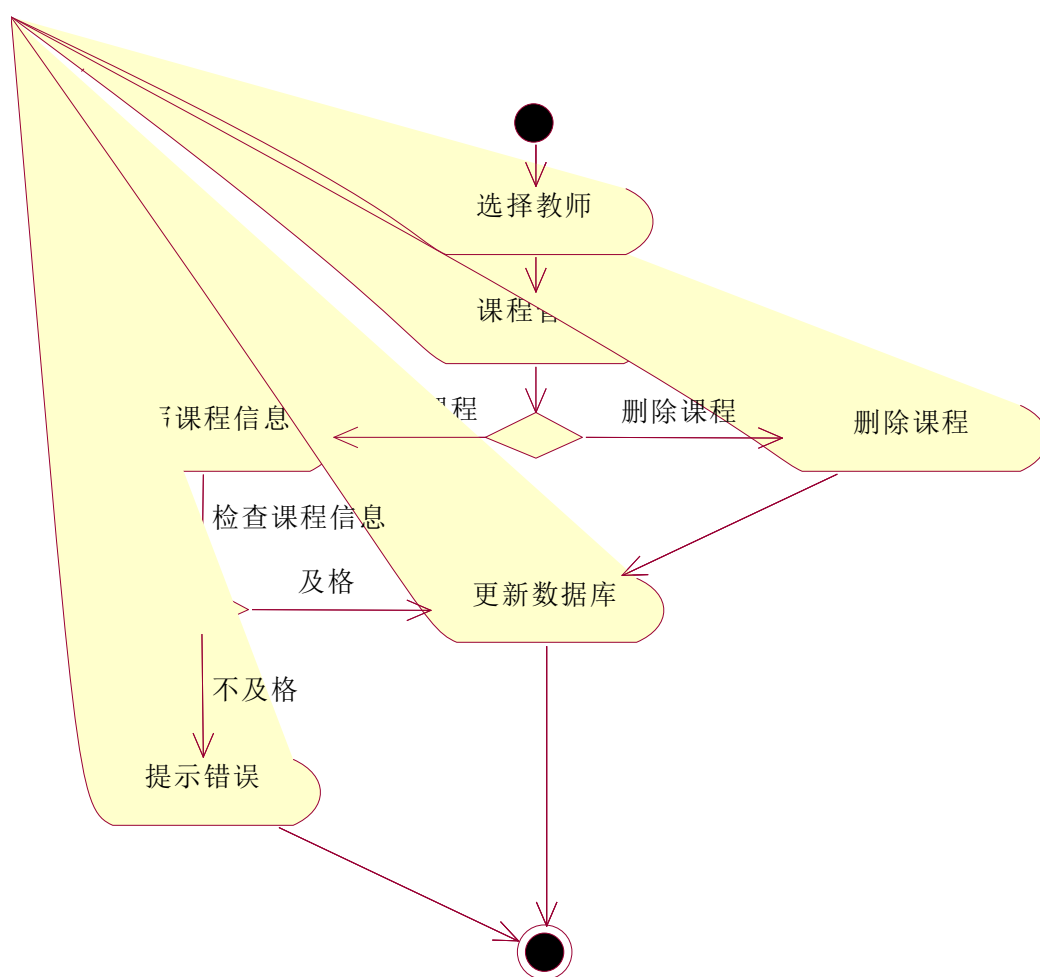


图 1-3 课程管理活动图



## 3 学生管理

### 3.1 简述

本用例描述管理员如何管理教师已有的课程中的学生信息,包括导入和删除学生信息。

### 3.2 参与者

管理员。

### 3.3 事件流

#### 3.3.1 基本事件流

本用例开始于管理员登录系统后并进入到课程,管理该课程的学生。

(1) 系统于课程页中列出学生管理选项,要管理员指出要执行的操作。

(2) 一旦管理员提供了所需要的信息,以下的一条子事件流将被执行:

如果选择的是“导入学生”,导入学生子事件流将被执行;

如果选择的是“删除学生”,删除学生子事件流将被执行。

#### a. 导入学生

1) 管理员选择“从文件导入学生”,上传学生名单文件至系统。

2) 管理员选择“手动导入学生”,则要填写导入的学生的信息,包括学生的学号和姓名。

3) 管理员确认导入学生信息及总人数。

4) 系统检查学生信息,系统同步更新学生的课程列表

5) 系统提示成功导入学生,并显示本次导入人数及该课程总人数。

#### b. 删除学生

- 1) 系统显示出该课程的学生名单列表；
- 2) 管理员选择删除某一名学生，或勾选全选。
- 3) 系统提示管理员确认从课程学生名单中删除该学生。
- 4) 管理员确认删除。
- 5) 提交给系统，系统同步更新学生课程列表。
- 6) 系统提示成功删除学生，并显示该课程当前学生人数。

#### 2.3.2 备选事件流

##### (1) 导入学生失败

系统检测到要导入的学生信息不全，或学生学号已存在该课程学生名单中时，给出错误提示并显示错误的学生信息，管理员确认错误，返回课程界面。

##### (2) 删除学生失败

系统提示删除学生失败，系统显示删除失败错误信息。管理员确认错误，返回课程界面，本用例终止。

#### 3.4 特殊需求

无。

#### 3.5 前置条件

本用例开始前管理员必须已经凭账号和密码登录进系统，管理员已进入到系统中某教师的某门课程。

#### 3.6 后置条件

如果用例成功，管理员导入或删除了该门课程的学生名单，学生用户的

课程列表也同步更新改变。否则系统状态不变。

### 3.7 活动图

学生管理用例的活动图如图 1-4 所示。

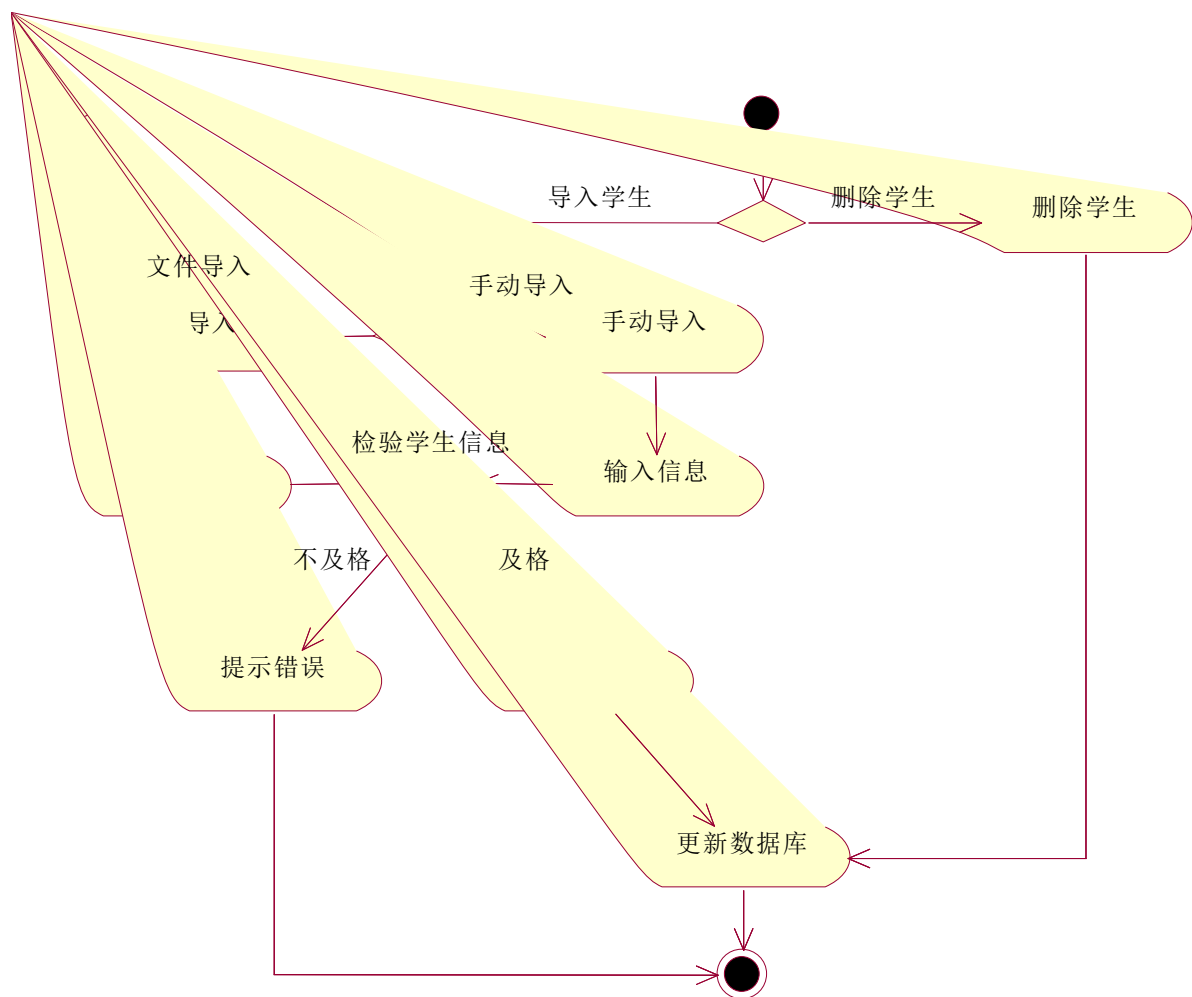


图 1-4 学生管理活动图

## 4 课时签到

#### 4.1 简述

本用例描述教师发布签到二维码供学生签到的过程。

#### 4.2 参与者

教师用户。

#### 4.3 事件流

##### 4.3.1 基本事件流

本用例开始于教师登录系统并进入到课程，对学生进行课时签到。

( 1 ) 系统于课程操作界面给出课时签到指引。

( 2 ) 教师选择发布二维码签到。

( 3 ) 学生扫描二维码输入个人信息进行签到，二维码右侧实时更新并显示成功签到的人数。

( 4 ) 教师点击结束签到，系统显示出成功签到的学生列表。

( 5 ) 系统更新数据库签到信息。

##### 4.3.2 备选事件流

( 1 ) 系统提示数据库操作失败，给出错误信息，教师确认信息，系统返回到课程界面。

( 2 ) 系统提示发布二维码失败，给出错误信息，教师确认信息，系统回到课程界面。

#### 4.4 特殊需求

无。

#### 4.5 前置条件

本用例开始前教师必须已经凭账号和密码登录进系统，教师已进入到系

统中的某门课程。

#### 4.6 后置条件

如果用例成功，教师成功对该门课程的学生进行了课时签到。课程签到信息同步到数据库中。否则系统状态不变。

#### 4.7 活动图

课时签到用例的活动图如图 1-5 所示。

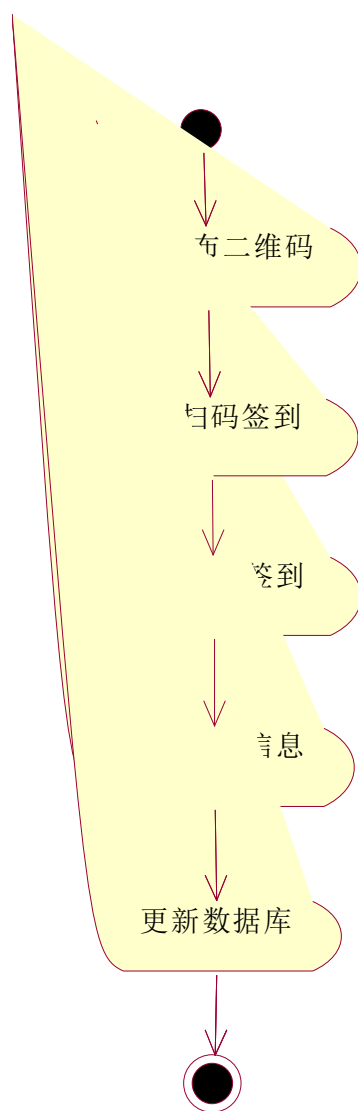


图 1-5 课时签到活动图

## 5 扫码签到

### 5.1 简述

本用例描述学生扫描教师提供的签到二维码进行签到的过程。

### 5.2 参与者

学生用户。

### 5.3 事件流

#### 5.3.1 基本事件流

本用例开始于学生打开手机二维码扫描软件并保持联网状态。

- (1) 教师选择课程，选择发布签到二维码。
- (2) 学生使用二维码软件扫描二维码。
- (3) 扫描完毕，跳转到签到信息填写界面。学生填写学号和姓名。
- (4) 学生点击确认提交，页面提示签到成功。
- (5) 系统同步更新签到信息数据库。

#### 5.3.2 备选事件流

(1) 如果未输入学号或姓名，学生无法点击确认提交，系统提示信息不完善，要求学生确认并补充信息后再次提交。

(2) 如学生扫描结果显示二维码过期，签到网页将不允许输入签到信息，系统提示学生二维码过期，并关闭页面。

(3) 如果系统判定学生定位在教室范围之外，签到网页将不允许输入签到信息，系统提示学生距离教室距离过远无法签到。

(4) 如果在本次签到中，系统判定该 IP 地址已有过一次签到，系统提示学生已签到成功，请勿重复签到，并关闭页面

#### 5.4 特殊需求

无。

#### 5.5 前置条件

学生使用能扫码的 app 进行扫码，学生的移动设备打开定位；教师发布了签到二维码。

#### 5.6 后置条件

如果用例成功，学生成功签到，教师能查看签到结果，系统数据库同步更新。否则，系统状态保持不变。

#### 5.7 活动图

扫码签到用例的活动图如图 1-6 所示。

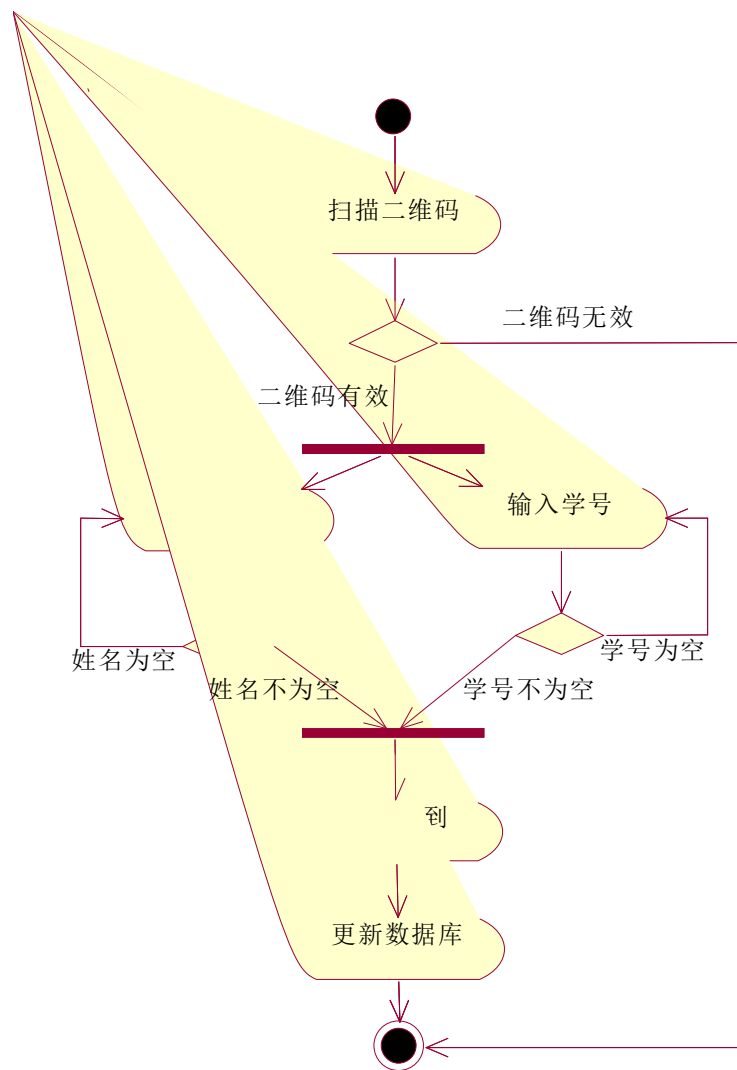


图 1-6 扫码签到活动图

## 6 登录系统

### 6.1 简述

本用例描述学生、教师和管理员登录使用本系统的过程。

### 6.2 参与者

教师用户，管理员。

### 6.3 事件流



### 6.3.1 基本事件流

本用例开始于教师和管理员进入到本系统的登录页面。

(1) 系统于登录界面引导用户选择身份进行登录；

(2) 一旦用户提供了所需要的信息，以下的一条子事件流将被执行：

如果选择的是“教师登录”，教师登录子事件流将被执行；

如果选择的是“管理员登录”，管理员登录子事件流将被执行。

#### a. 教师登录

1) 教师使用工号作为账号进行登录，初始密码由管理员交付，登录后可在个人中心修改密码。

2) 教师输入账号密码。

3) 确认登录。

4) 教师登录成功，进入到系统首页。

#### b. 管理员登录

1) 管理员使用管理员内部机密账号进行登录，登录后可在个人中心修改密码。

2) 管理员输入账号密码。

3) 确认登录。

4) 管理员登录成功，进入到系统首页。

### 6.3.2 备选事件流

(1) 如果未输入账号或密码，系统提示信息不完善，无法登陆，要求用户确认并补充信息后再次提交。

(2) 如果账号不存在，系统提示账号不存在。用户确认后，返回回到

登录界面。

(3) 如果密码错误，系统提示账号与密码不匹配，返回登录界面。

(4) 如果学生或教师忘记密码，联系管理员为其账号重置初始密码。

#### 6.4 特殊需求

无。

#### 6.5 前置条件

教师使用工号作为账号进行登录，初始密码由管理员交付；管理员使用  
管理员内部机密账号进行登录。

#### 6.6 后置条件

如果用例成功，用户能成功登录到系统并使用系统提供的功能。

#### 6.7 活动图

登录系统用例的活动图如图 1-7 所示。

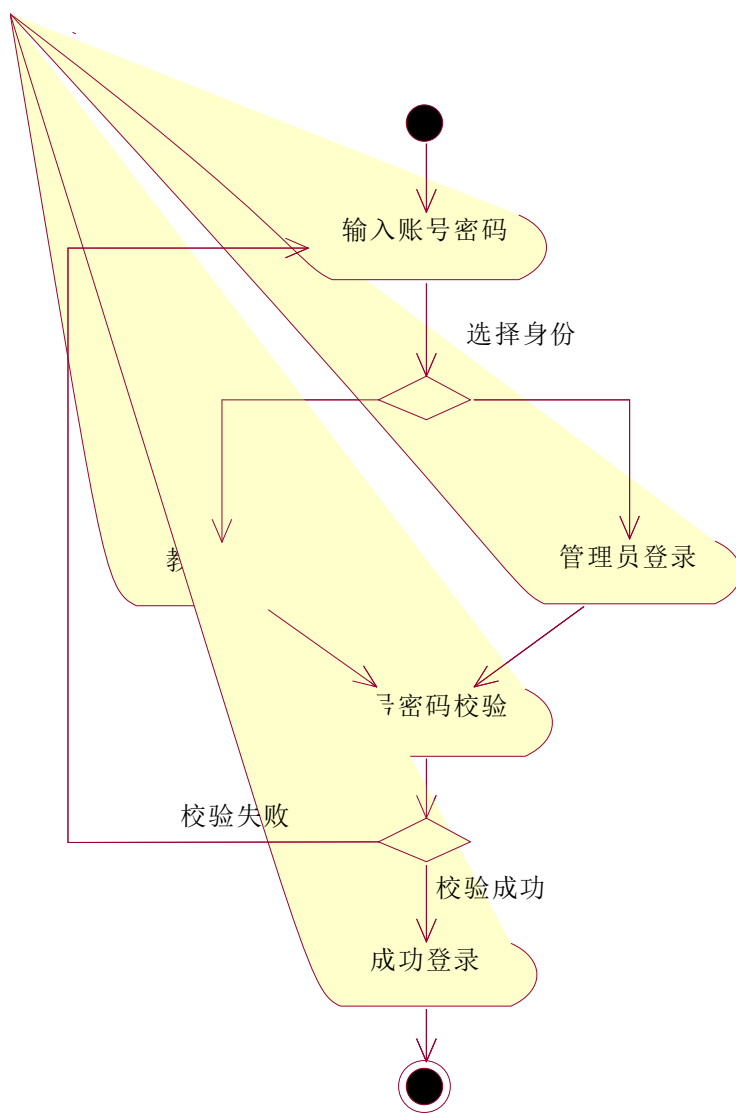


图 1-7 登录系统活动图

## 7 历史查询

### 7.1 简述

本用例描述教师使用系统查看历史课程签到及导出签到结果的情况。

### 7.2 参与者

教师用户。

## 7.3 事件流

### 7.3.1 基本事件流

本用例开始于教师已登录本系统。

(1) 系统于首页呈现出课程列表，教师直接查看课程；

(2) 当教师进入到某门课程，系统于课程页面提供了查看历史签到结果选项；

(3) 系统默认时间段或教师自行选择时间段；

(4) 系统以网格形式显示该门课程时间段内的每次签到的签到情况统计；

(5) 教师选择导出数据；

(6) 系统显示导出的签到结果的时间段、存放路径及文件名，提示教师导出成功。

### 7.3.2 备选事件流

(1) 系统提示数据库操作失败，给出错误提示，教师用户确认错误，系统返回首页。

(2) 系统提示未登录，给出错误提示，教师用户确认错误，系统返回登录输入界面，用户重新登录。

## 7.4 特殊需求

无。

## 7.5 前置条件

教师用户已经登录进本系统。

## 7.6 后置条件

如果用例成功，教师能成功查看每门课程的签到情况，获得某一时间段

下某课程的签到结果 txt 文件。用例成功，系统状态不发生改变。

## 7.7 活动图

历史查询用例的活动图如图 1-8 所示。

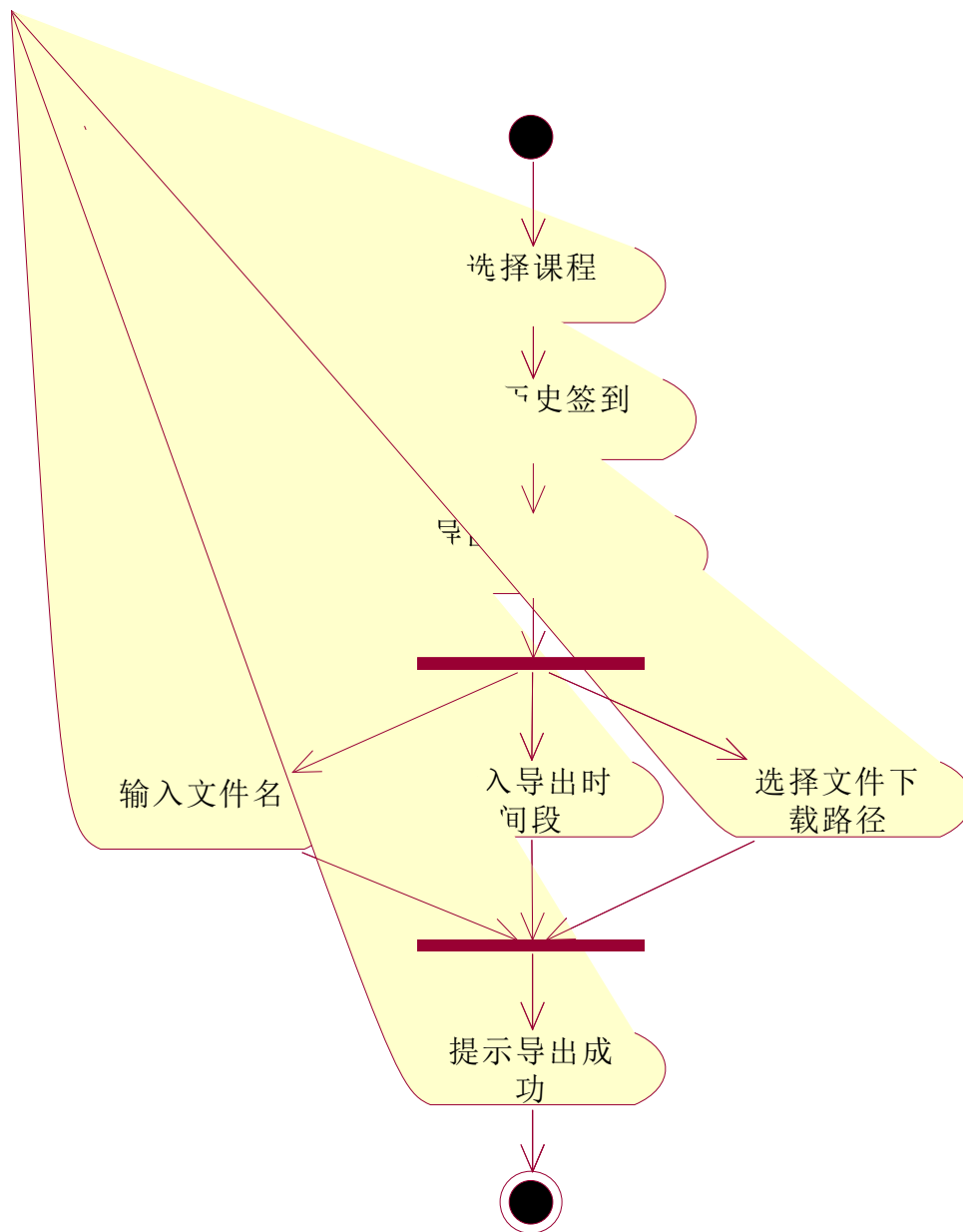


图 1-8 历史查询活动图

## 1.4 补充规约

### 1.4.1 可靠性

本系统可以保证 7\*24 小时连续运行，一周维护时间不会超过 3 小时。准确统计签到人数。

### 1.4.2 安全性

本系统面向中山大学全体在校学生。使用学号作为账号和默认密码。姓名、学号、学院、校区、入学年份由学校相关部门提供，在本系统中不得更改。密码以密文存储在数据库中。

### 1.4.3 可用性

本系统目标群体为中山大学在校教师。对于教师而言，系统页面简单易理解、易操作。教师可以简单、快捷的扫码签到，不会影响正常上课，并获得签到结果。

### 1.4.4 性能

本系统满足 100 个人同时使用，页面反应时间不能超过 3 秒，允许大量的并发访问。

### 1.4.5 可支持性

本系统支持能够扫描二维码的软件，如微信。本系统支持主流浏览器，如 chrome 浏览器，firefox 浏览器等。

### 1.4.6 可扩展性

本系统能接受对现有功能合理的改进，增加一些新的功能。

# 1.5 术语表

术语	含义
教师用户	中山大学在职教师。
学生用户	中山大学在校学生。
管理员	可添加老师、查看后台数据。
教师工号	教师身份的唯一标识 ,是教师登陆系统的 账号 ,管理员添加课程时使用此号码为课 程设置教师。
定位	移动终端的地理位置信息 ,需要学生用户 开启定位服务权限。
可扫码的 app	例如微信、支付宝等提供扫二维码功能并 可进行页面跳转的应用。

## 第二章 架构设计

### 2.1 架构描述

本系统采用 MVC ( Model View Controller ) 三层架构，将业务逻辑、数据和界面显示分离开来，使得开发人员在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑，同时简化了对后续程序的修改和扩展。这样的架构不仅使整个网站的开发清晰明了，开发模块化，而且大大提高了开发的效率。

#### 2.1.1 视图层

View 层又称视图层，是用户看到并与之交互的界面，在本系统中，视图层由 HTML 文件及 CSS 文件组成。它们负责界面的呈现、用户输入的获取以及显示控制层处理的结果。本网站的视图层由以下板块构成：

教师界面：

管理员界面：

#### 2.1.2 控制器层



Controller 层又称控制层 ,它收到来自视图层的请求并对实体类进行增删改查 ,当单击 Web 页面中的按钮时 ,控制层接收请求并处理请求 ,然后返回数据。

主要包括三个部分 :

第一部分 :根据用户通过浏览器请求网页的访问路径和前端向后端进行数据交互的 api ,交给下一部分的逻辑进行处理。

第二部分 :对用户请求进行处理的控制逻辑 ,如用户进行登陆 ,管理员添加老师 ,老师发布课程发布公告 ,随机点名 ,学生查看课程查看公告 ,进行签到等 ,把请求数据交给下一部分逻辑 ,获取数据库数据并进行处理后 ,再发送给请求方。

第三部分 :从上一层中获取请求数据 ,对请求数据进行判断及处理后 ,交给模型层 ,对数据进行 crud ( 数据库的增删改查 ) 操作后 ,再把数据库获得的结果处理后返回给上一层逻辑。

### 2.1.3 模型层

本网站的数据模型分为以下部分 :

- ( 1 ) 网站用户数据模型 ,其中包括管理员、教师用户、学生用户。
- ( 2 ) 网站项目的主要数据 ,包括签到信息、课程信息。

## 2.2 架构图

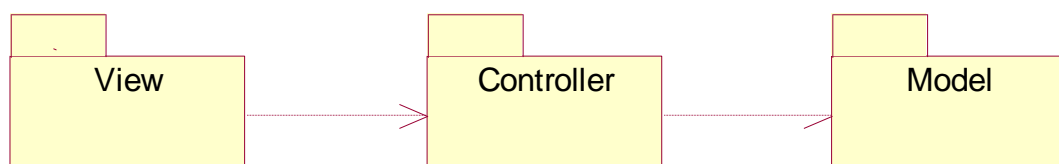


图 2-1 架构图

## 2.3 关键抽象

按照模块分类，该系统实体类的关键抽象图如图 2-2 所示：

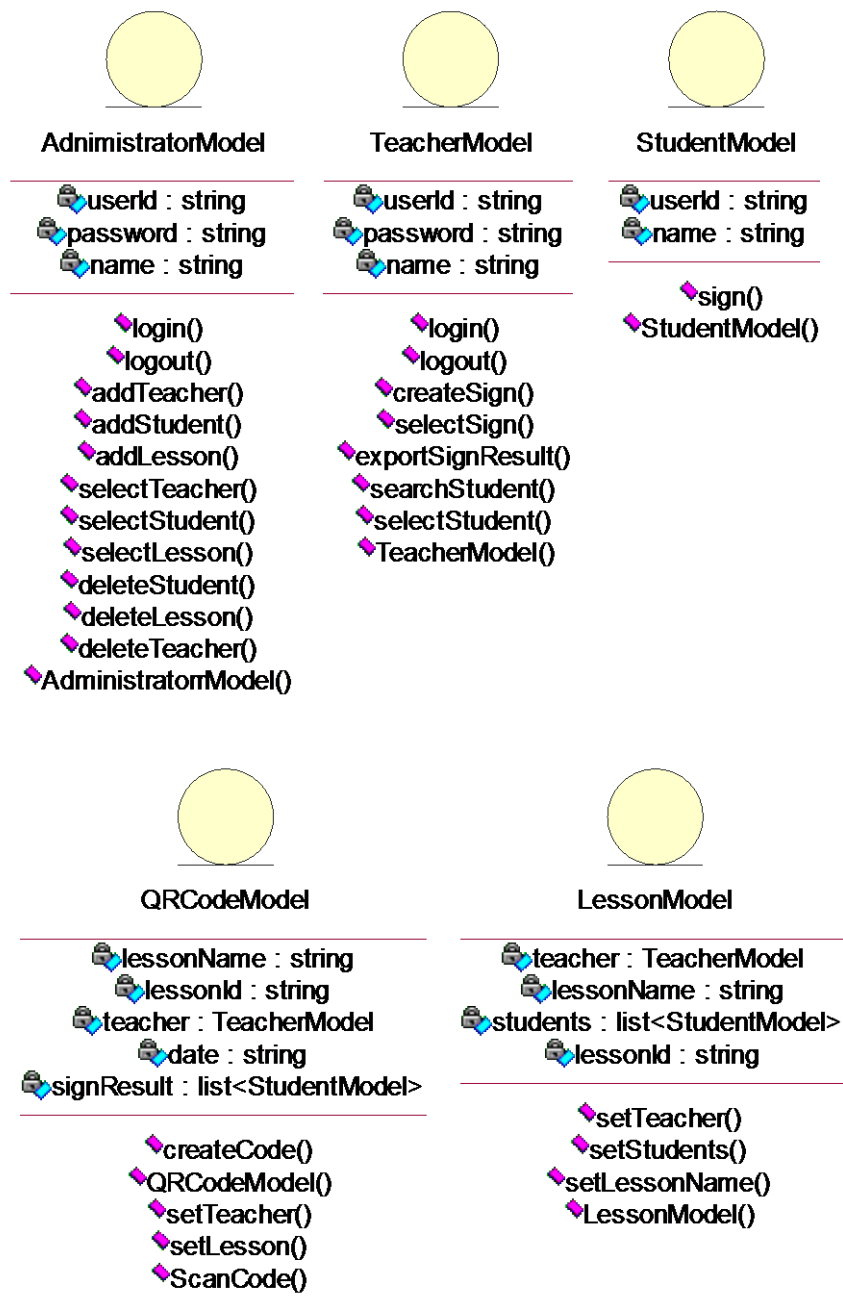


图 2-2 课程签到管理系统关键抽象图

## 第三章 用例分析

### 3.1 补充用例规约

经检查，本项目组发现用例规约比较完善，暂时无需补充。

### 3.2 用例中类的析取

#### 3.2.1 教师管理用例析取

##### 3.2.1.1 用例析取

根据第一章需求分析中的用例规约，经分析可得到该用例中的三种类。

（1）边界类：showTeacher。该边界类负责为管理员显示教师列表，提供相应的接口进入到控制类。

（2）控制类：teacherManageAction。该控制类负责处理管理教师的相关操作，包括查看教师，导入教师，删除教师。

（3）实体类：TeacherModel，LessonModel。TeacherModel 实体类表示教师对象实体，包含教师账号 ID，教师姓名，教师登录密码等信息。LessonModel 实体类包含任课教师信息，删除教师则也删除该门课程。

教师管理用例析取图如图 3-1 所示。

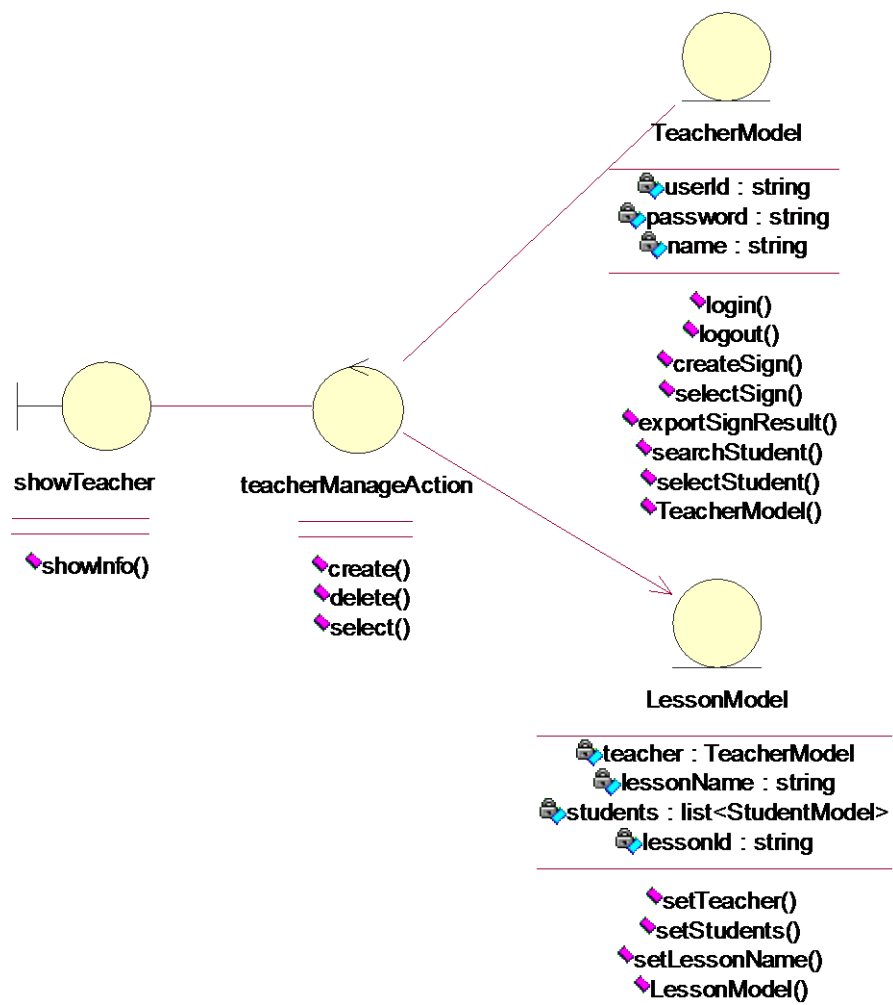


图 3-1 教师管理用例析取图

### 3.2.1.2 时序图

根据该用例类的操作顺序，教师管理用例时序图如图 3-2 所示。

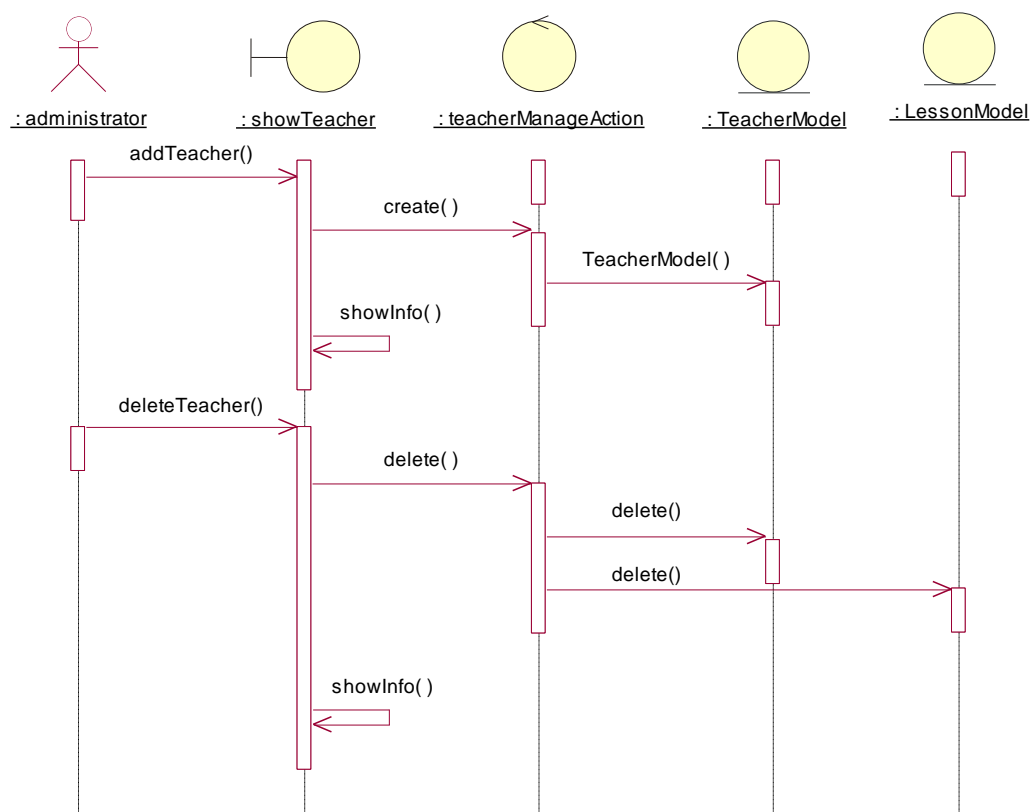


图 3-2 教师管理时序图

## 3.2.2 课程管理用例析取

### 3.2.2.1 用例析取

根据第一章需求分析中的用例规约，经分析可得到该用例中的三种类。

(1) 边界类：showLesson。该边界类负责为管理员显示某教师的课程列表，并提供相应的接口进入到控制类。

(2) 控制类：lessonManageAction。该控制类负责处理管理课程的相

关操作，包括查看课程，导入课程，删除课程。

( 3 )实体类 :LessonModel。该实体类表示课程对象实体 ,包含课程 ID ,  
课程名称，任课教师，课程学生名单等信息。

课程管理用例析取图如图 3-3 所示。

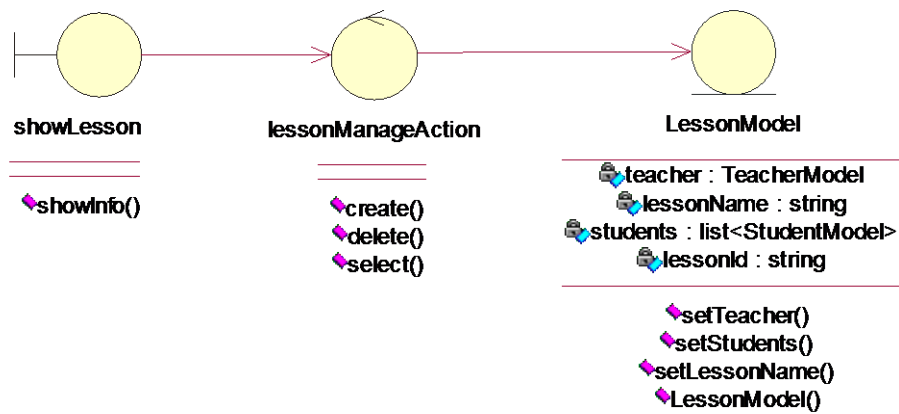


图 3-3 课程管理用例析取图

3.2.2.2 时序图

根据该用例类的操作顺序，课程管理用例时序图如图 3-4 所示。

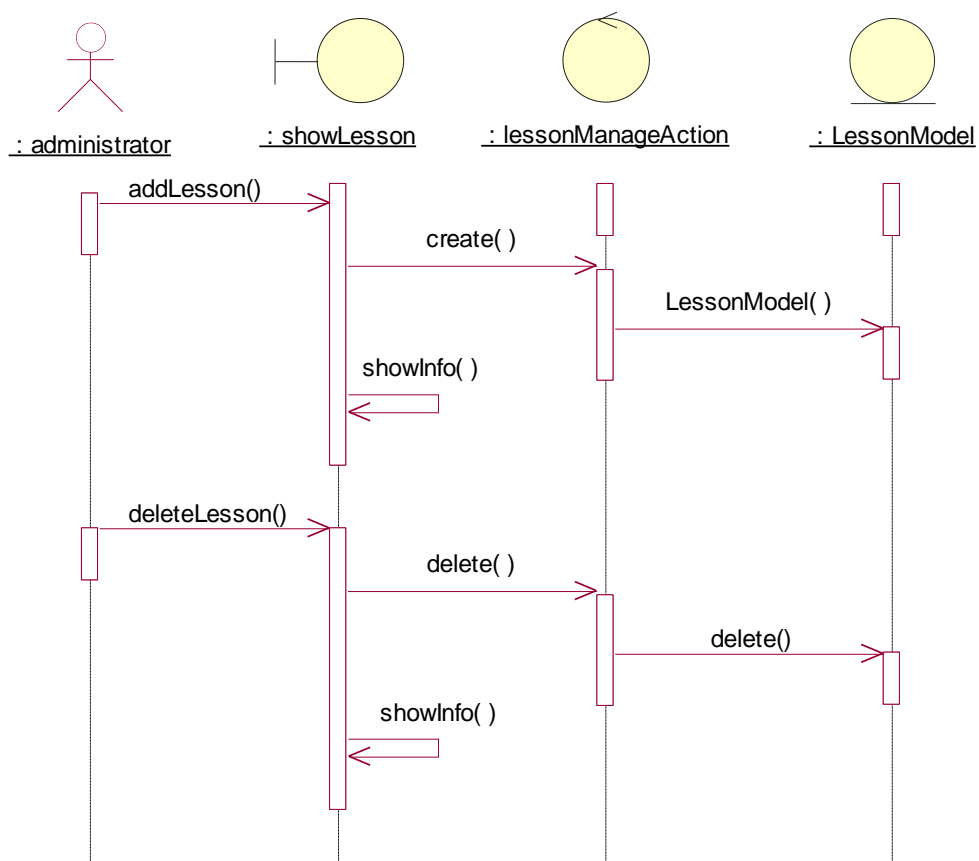


图 3-4 课程管理时序图

## 3.2.3 学生管理用例析取

### 3.2.3.1 用例析取

根据第一章需求分析中的用例规约，经分析可得到该用例中的三种类。

- ( 1 ) 边界类：showStudents。该边界类负责为管理员显示某门课程的学生名单，并提供相应的接口进入到控制类。

( 2 ) 控制类：studentManageAction。该控制类负责处理管理学生的  
相关操作，包括查看学生，导入学生，删除学生。

( 3 ) 实体类：StudentModel，LessonModel。StudentModel 实体类  
表示学生对象实体，包含学生 ID 和学生名字。LessonModel 中含  
有学生名单。

学生管理用例析取图如图 3-5 所示。

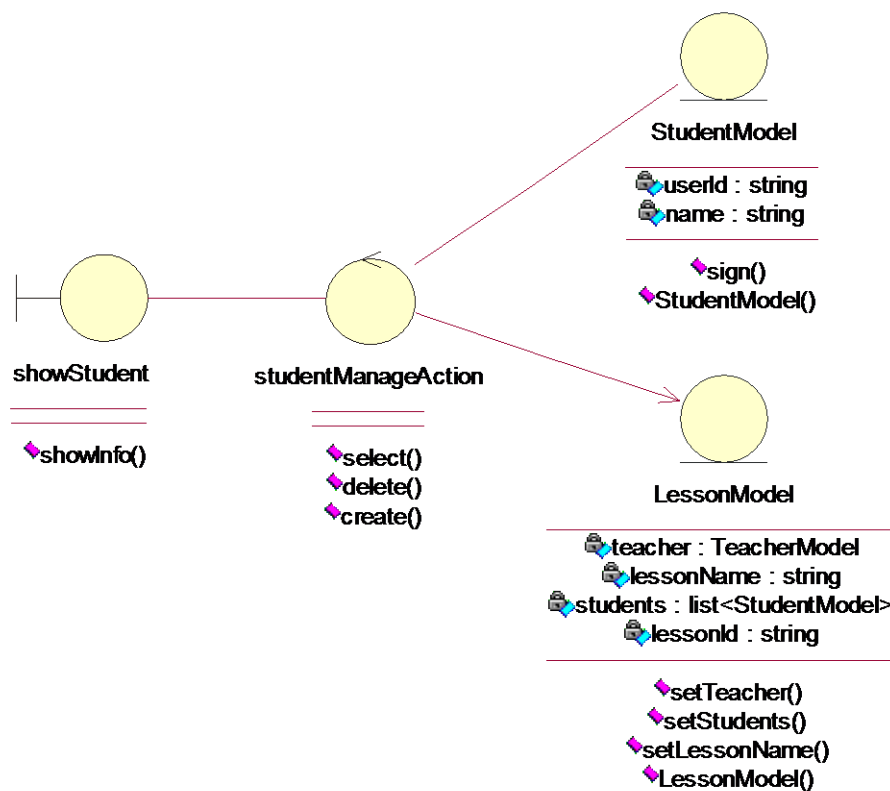


图 3-5 学生管理用例析取图

### 3.2.3.2 时序图

根据该用例类的操作顺序，学生管理用例时序图如图 3-6 所示。



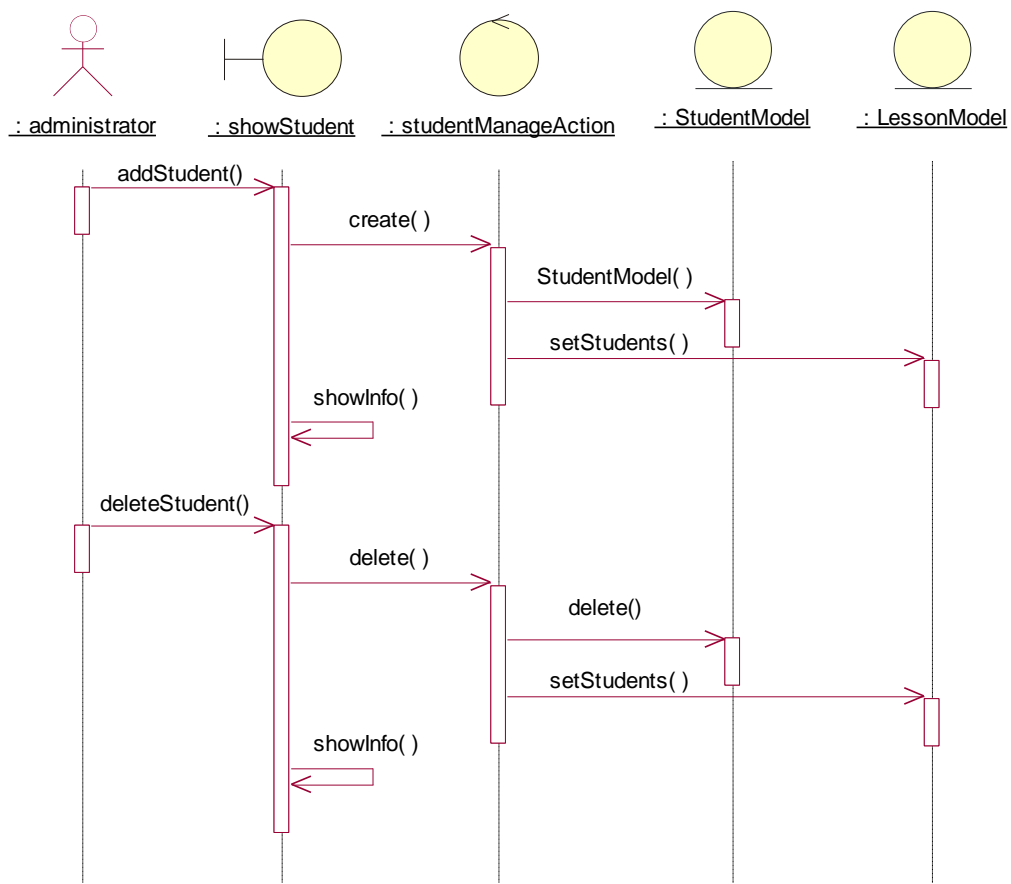


图 3-6 学生管理时序图

## 3.2.4 课时签到用例析取

### 3.2.4.1 用例析取

根据第一章需求分析中的用例规约，经分析可得到该用例中的三种类。

(1) 边界类：showSignIn。该边界类负责为教师显示某课程当前已签到成功的人数，并提供相应的接口进入到控制类。

(2) 控制类：signInAction。该控制类负责处理签到的相关操作，包括

发布签到、结束签到。

( 3 ) 实体类：QRCodeModel。该实体类表示签到信息对象实体，包含学生 ID 和学生名字。

课时签到用例析取图如图 3-7 所示。

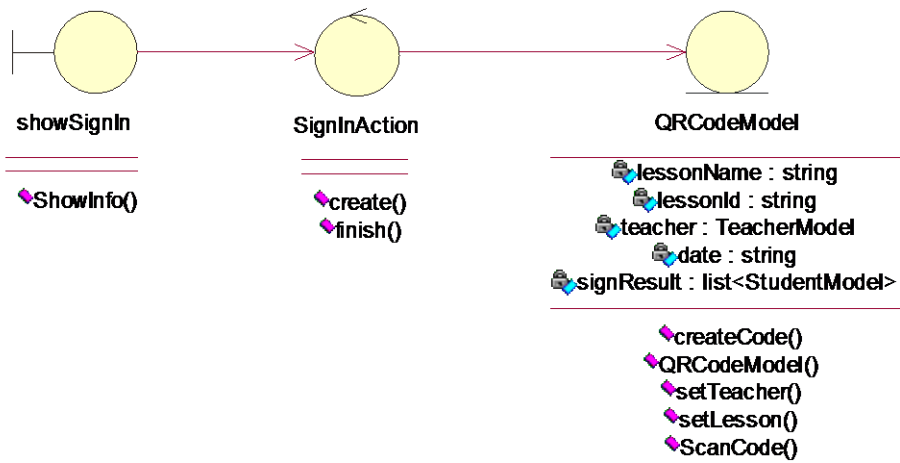


图 3-7 课时签到用例析取图

3.2.4.2 时序图

根据该用例类的操作顺序，课时签到用例时序图如图 3-8 所示。

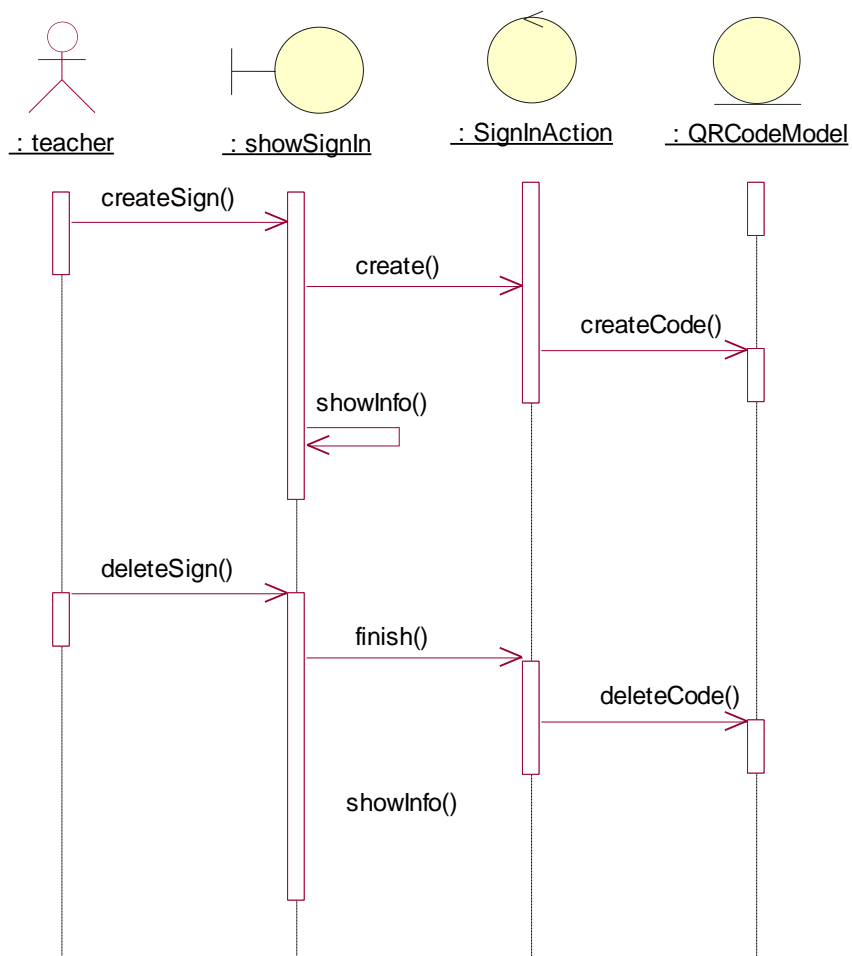


图 3-8 课时签到时序图

## 3.2.5 扫码签到用例析取

### 3.2.5.1 用例析取

根据第一章需求分析中的用例规约，经分析可得到该用例中的三种类。

(1) 边界类：showStudentSign。该边界类负责为学生显示签到提示信息，并提供相应的接口进入到控制类。

(2) 控制类：studentSignAction。该控制类负责处理学生输入的签到

信息的正确性，包括验证学生 ID 及密码。

( 3 ) 实体类：StudentModel。该实体类表示学生对象实体，包含当前签到结果学生列表，课程信息以及日期。

扫码签到用例析取图如图 3-9 所示。

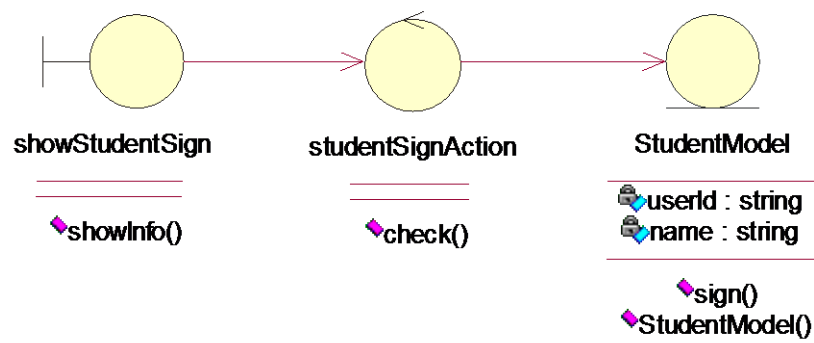


图 3-9 扫码签到用例析取图

3.2.5.2 时序图

根据该用例类的操作顺序，扫码签到用例时序图如图 3-10 所示。

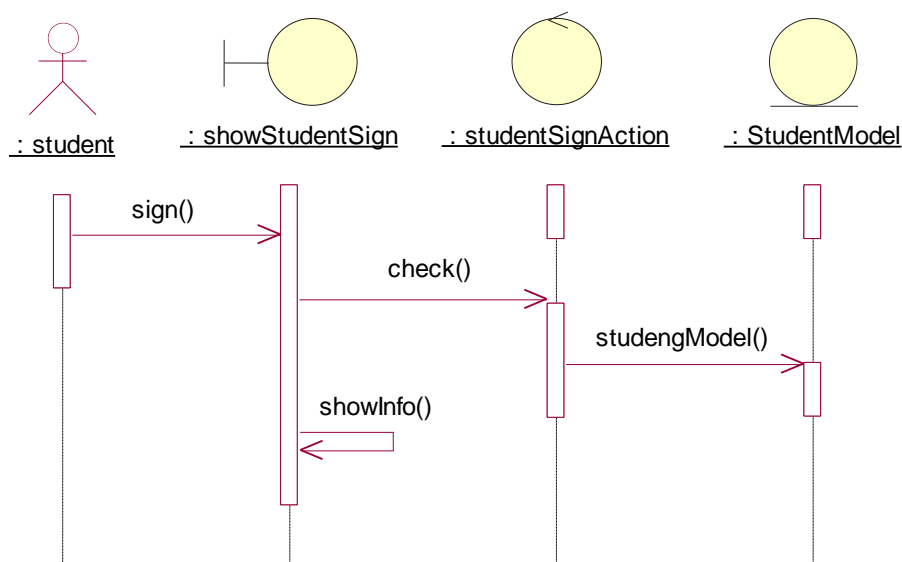


图 3-10 扫码签到时序图

## 3.2.6 登录系统用例析取

### 3.2.6.1 用例析取

根据第一章需求分析中的用例规约，经分析可得到该用例中的三种类。

- ( 1 ) 边界类：showLogin。该边界类负责为用户（管理员/教师）显示登录提示信息，并提供相应的接口进入到控制类。
- ( 2 ) 控制类：loginAction。该控制类负责处理用户输入的相关信息的正确性，包括验证管理员 ID 及密码、教师 ID 及密码。
- ( 3 ) 实体类：TeacherModel、AdministratorModel。其中 TeacherModel 实体类表示教师对象实体，包含教师账号 ID、教师登录密码等信息。AdministratorModel 实体类包含管理员账号 ID、管理员登录密码等信息。

登录系统用例析取图如图 3-11 所示。

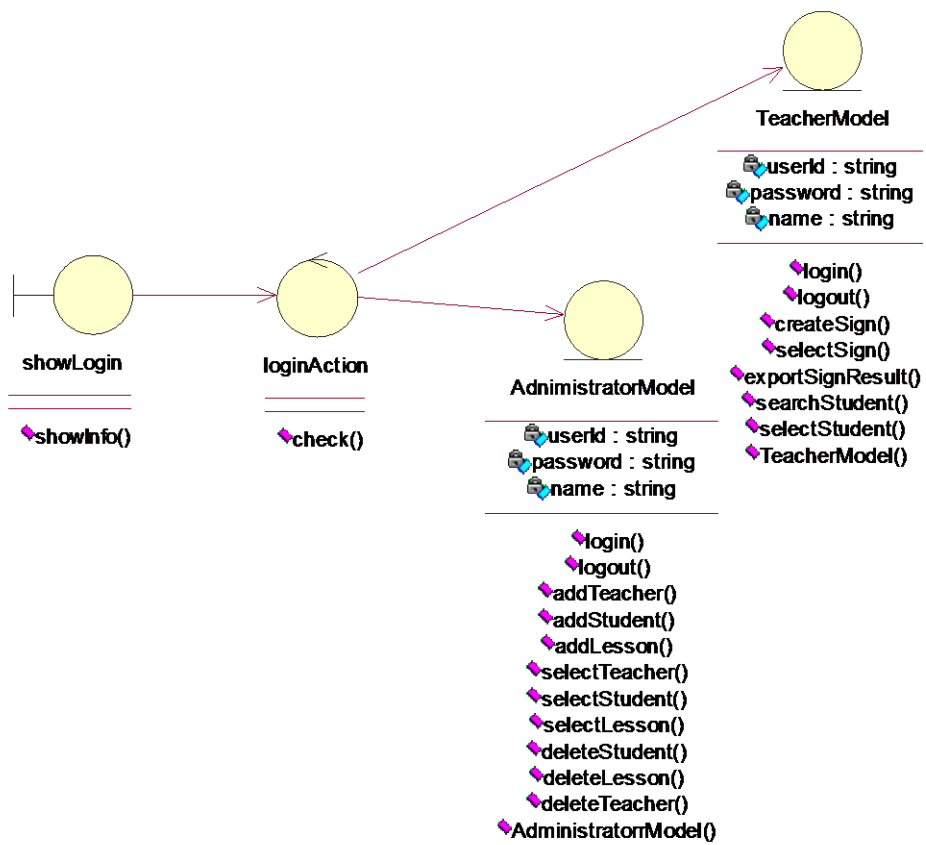


图 3-11 登录系统用例析取图

### 3.2.6.2 时序图

根据该用例类的操作顺序，登录系统用例时序图如图 3-12 所示。

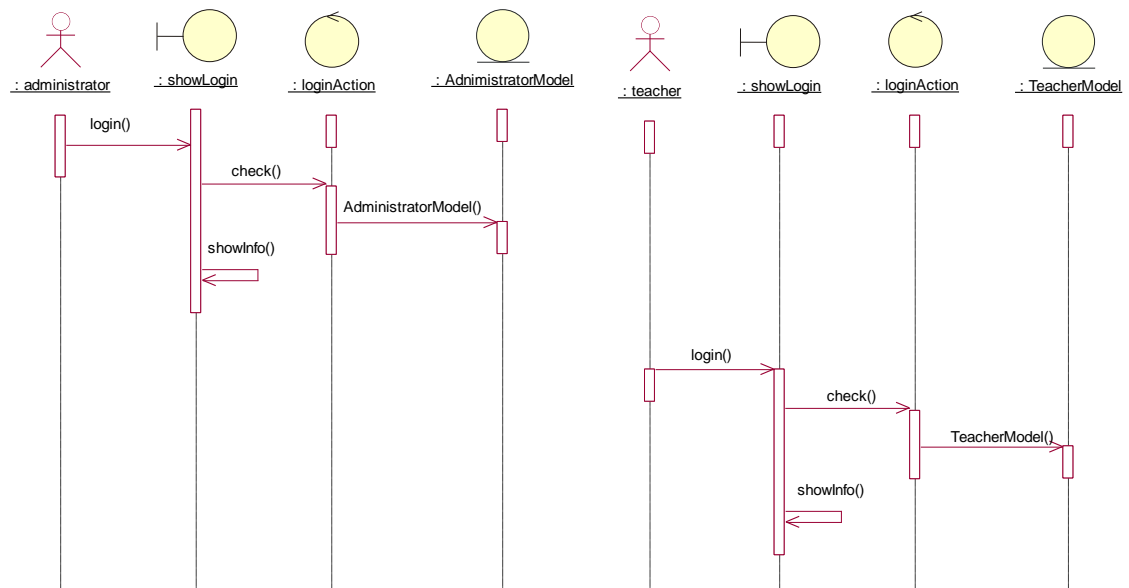


图 3-12 登录系统时序图

## 3.2.7 历史查询用例析取

### 3.2.7.1 用例析取

根据第一章需求分析中的用例规约，经分析可得到该用例中的三种类。

( 1 ) 边界类：showQuery。该边界类负责为教师显示某课程历史签到的学生名单，并提供相应的接口进入到控制类。

( 2 ) 控制类：queryAction。该控制类负责处理历史签到信息的相关操作，包括查询历史签到，导出历史签到。

( 3 ) 实体类：QRCodeModel。QRCodeModel 实体类表示签到信息对

象实体，包含某次课程某日期的学生签到结果列表。

历史查询用例析取图如图 3 所示。

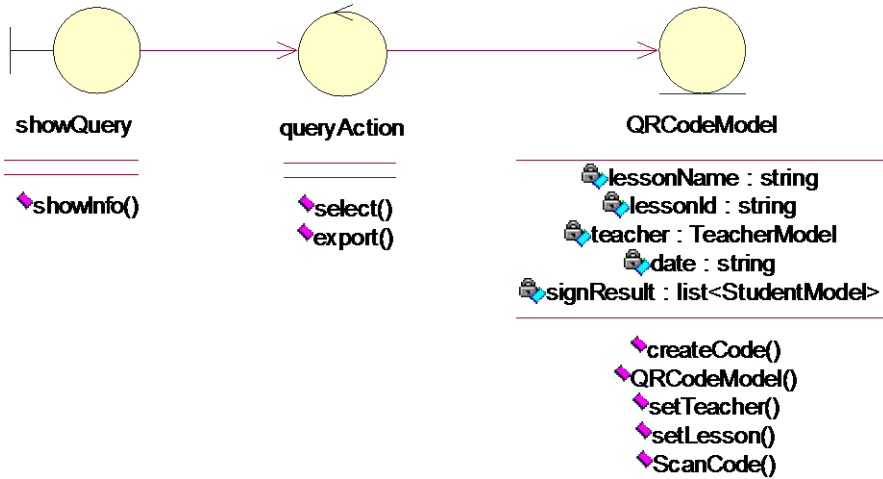


图 3-13 历史查询用例析取图

3.2.7.2 时序图

根据该用例类的操作顺序，历史查询用例时序图如图 3-14 所示。



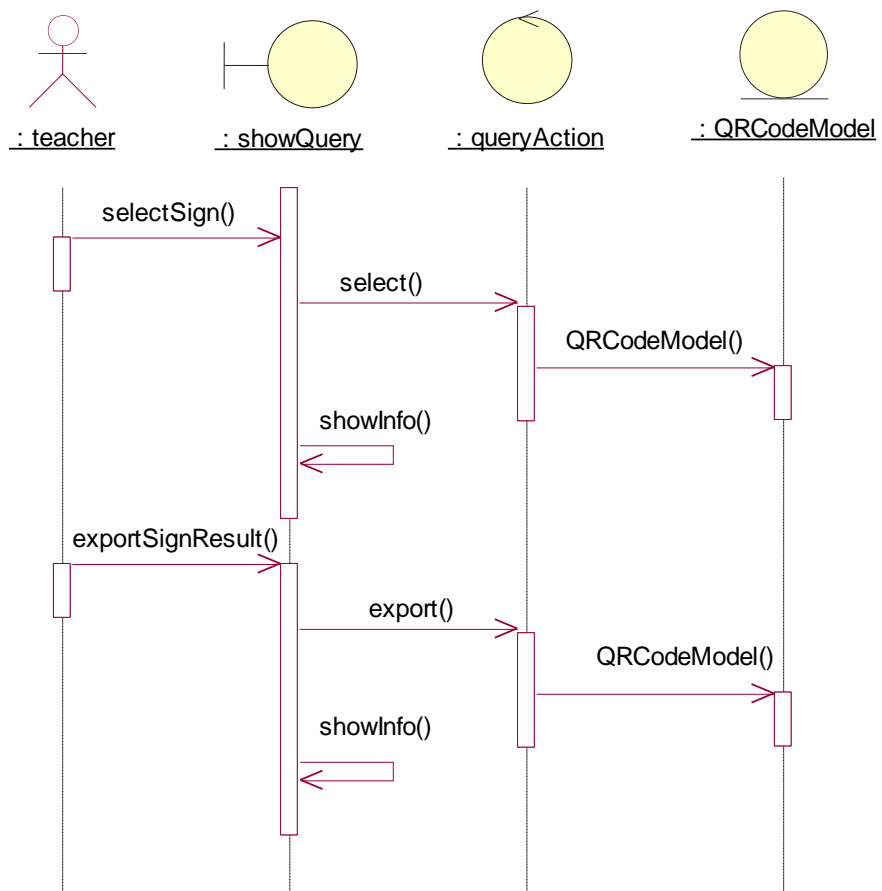


图 3-14 历史查询时序图

### 3.3 总体类的析取

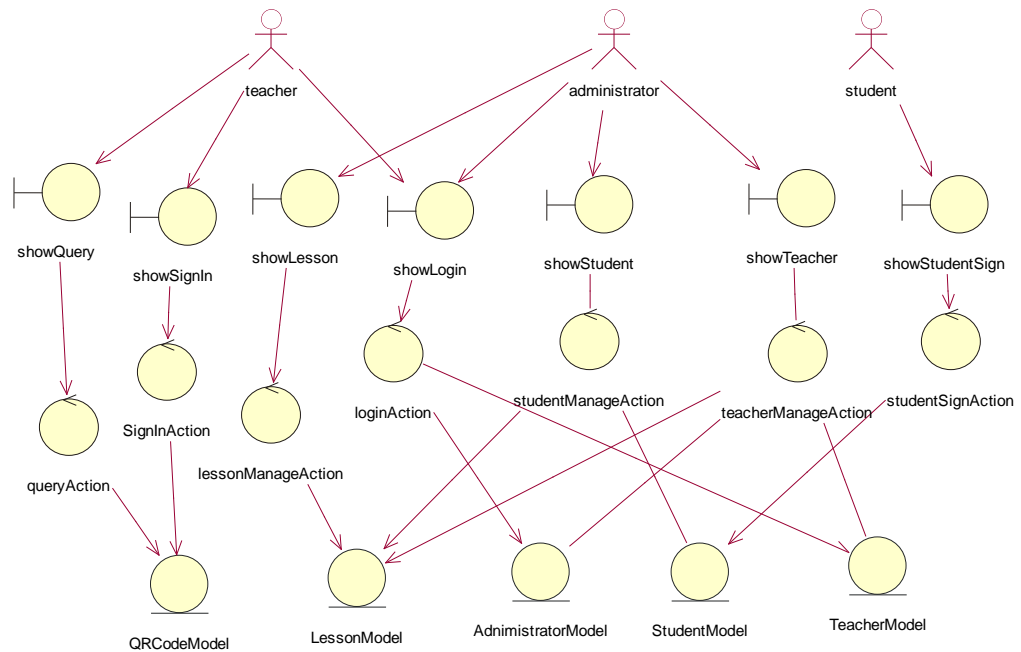


图 3-15 总体类的析取

### 3.4 分析机制

根据上述边界类、控制类、实体类需要满足的非功能性的需求，列出系统的分析机制表如下表：

课程签到管理系统分析机制表	
分析类 ( analysis classes )	分析机制 ( analysis mechanism )
AdnimistratorModel	安全性、持久性
TeacherModel	安全性、持久性
StudentModel	安全性、持久性

QRCodeModel	安全性、持久性
LessonModel	安全性、持久性

## 第四章 子系统与接口设计

### 4.1 确定设计类

根据第三章分析可知，系统的总体类设计图如下所示：

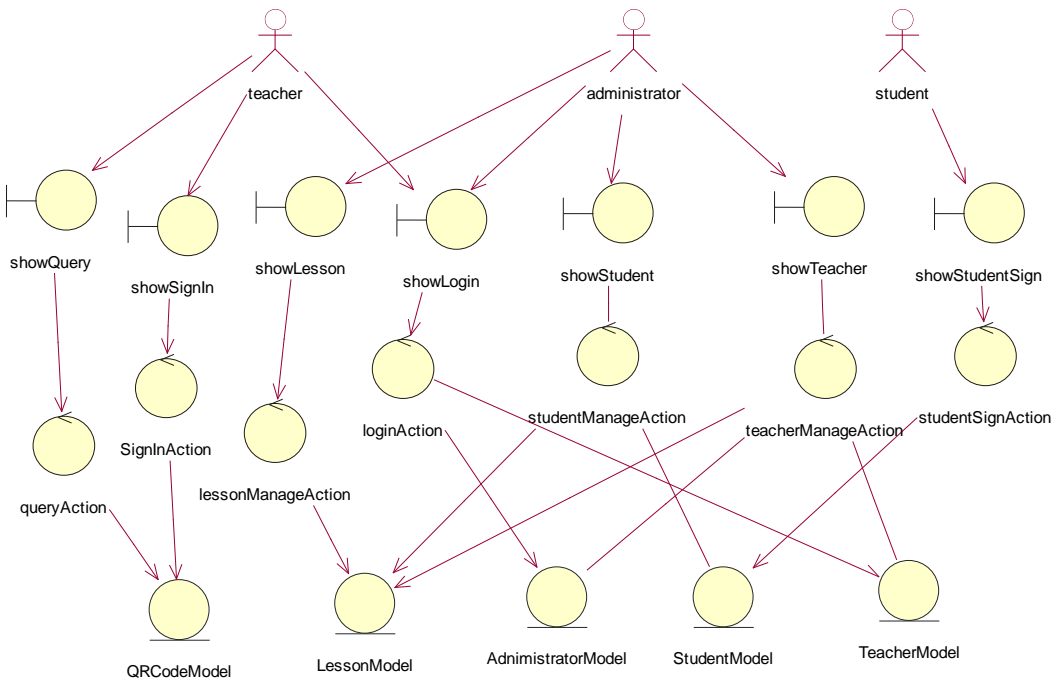


图 4.1 总体类设计

## 4.2 对系统进行子系统划分

子系统设计是将某些类封装在一个子系统中,子系统可以提供一个或者多个接口,其他类或子系统可以直接通过接口获得子系统的功能,而不需要与子系统内部的类发生联系。子系统作为一个功能单元而存在于系统中。经过分析,根据功能将课程签到管理系统的子系统划分设计如图 4-2 所示,本系统中共有三个子系统,其中 InfoManageSubsystem 子系统提供教师、学生、课程信息的查询及增加、删除、修改功能,SignManageSubsystem 子系统提供老师发起签到、学生签到以及签到历史查询的功能,LogManageSubsystem 子系统提供教师、管理员登录功能。

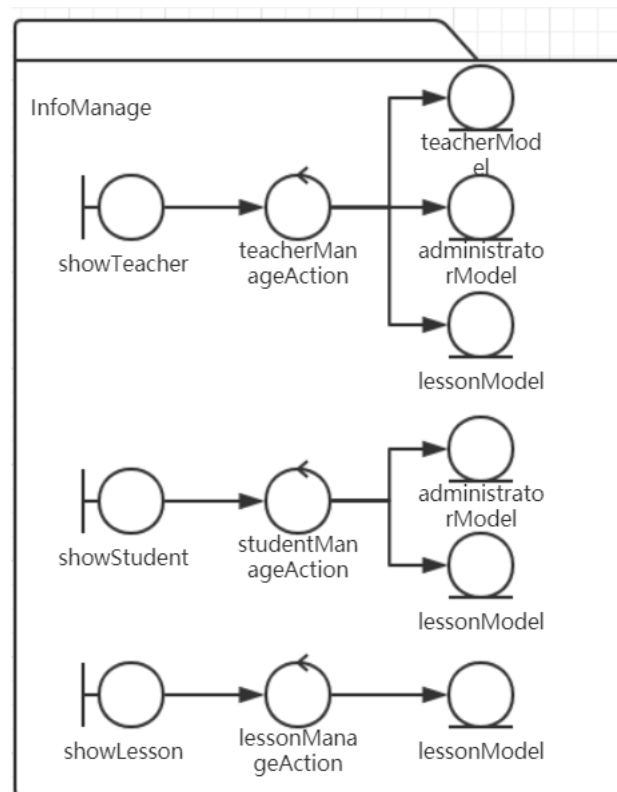


图 4.2 InfoManage 子系统

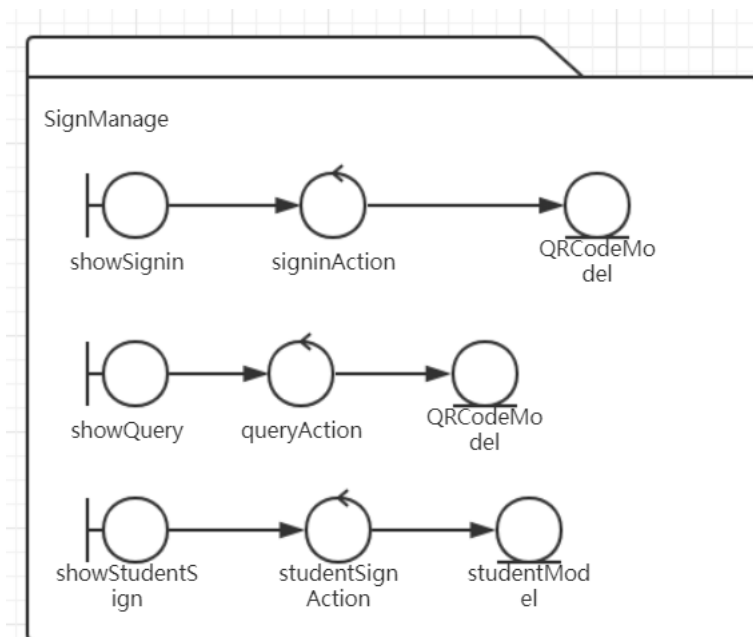


图 4.3 SignManage 子系统

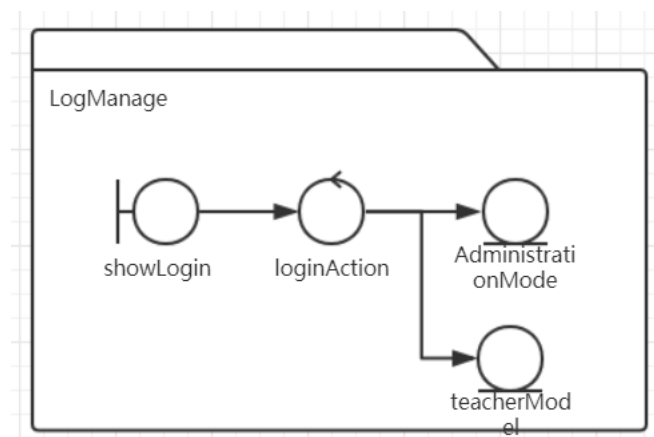


图 4.4 LogManage 子系统

### 4.3 子系统及其接口设计

经过分析，课程管理系统的子系统及其接口如下图所示：

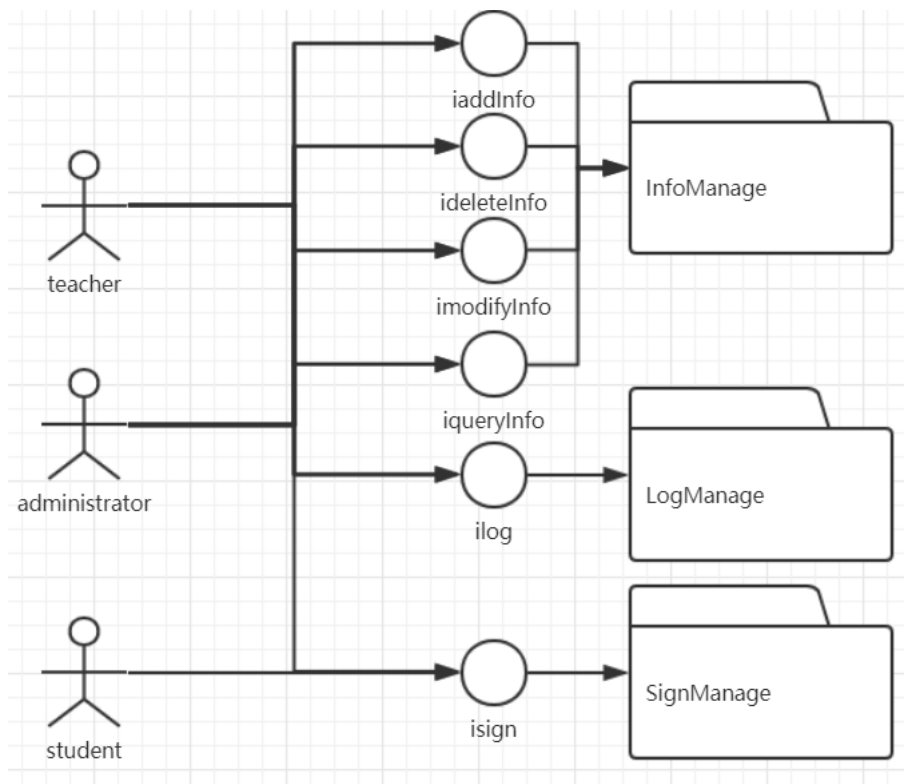


图 4.5 子系统及其接口设计图

## 第五章 运行时的架构设计

### 5.1 分析本系统的并发需求

本系统作为一个网页系统，为了满足课堂需求，系统需要支持 100 个人以上同时登录使用、并发请求加载网页的内容等，页面反应时间不能超过 3 秒，允许大量的并发访问。

假设一个用户相当于一个请求，该系统的并发需求很大程度上由需要进行课堂签到的数量决定。考虑到大多数课堂容量的实际情况，总共的用户数量大概控制在 200 人以下，并发量则会小于 200，综合考虑该系统将并发需求控制在：一个时

刻最多同时只有 100 个人同时请求。

## 5.2 识别出对应的进程和线程

由于该系统需要考虑并发需求，因此该系统包含以下进程和线程：

### 5.2.1 进程

主进程：整个系统仅由一个进程构成，教师或管理员系统的入口则为该进程的入口，该进程生命周期与系统的生命周期一致，当系统结束运行时候，进程将结束；如果进程被迫结束（例如异常），则该系统也会结束。而主进程中包括多个线程。

### 5.2.2 线程

#### 5.2.2.1 请求线程（RequestHandler）

在发起签到后，可能存在多个学生用户请求，因此需要存在一个请求线程，该线程仅用来接收所有学生用户的请求，并将请求放到对应的请求队列中去；请求线程不对用户请求做具体的响应（请求线程相当于一个生产者）。

#### 5.2.2.2 响应线程（ResponseHandler）

请求线程将用户请求存放到一个请求队列中去，需要一个响应线程对队列进行轮询操作，如果发现队列存在用户的请求，则从资源池请求分配资源进行响应（响应线程相当于一个消费者）。

#### 5.2.2.3 处理线程（HandlerThread）

处理线程由资源池进行分配，当响应线程请求一个线程处理请求的时候，如果资源池存在空闲的线程，则将该空闲线程用于处理该请求。

### 5.3 描述相应的进程和线程

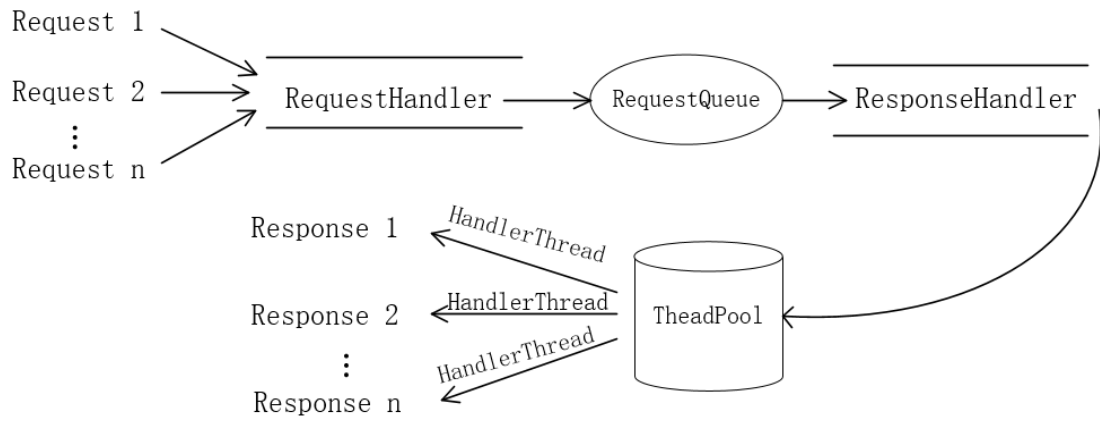


图 5.1 系统处理请求过程

系统的请求处理过程如上图 5.1 所示。用户所有的请求 Request 通过请求线程 RequestHandler 进入请求队列 RequestQueue 中；如果请求队列 RequestQueue 已满，则会阻塞 RequestHandler，直至 RequestQueue 有空余的位置，如果 RequestQueue 中为空，则会阻塞 ResponseHandler。这些请求被 ResponseHandle 取出并进行响应，最终这些请求通过 HandlerThread 线程处理，每个请求都与一个 HandlerThread 线程进行对应，HandlerThread 处理后将用户请求的结果 Response 返回给用户。