啪啪打卡设计文档

1 引言

* 1. 系统概述

教学实验管理系统（啪啪打卡）是实现老师管理成绩实现现代化和信息化的重要系统。该教学实验管理系统能够为老师提供充足的信息和快捷的查询登记的手段，并支持学生老师简单交互的功能。该系统充分考虑到各种小的功能需求，面对实验项目的不断丰富、学生人数不断增多、潜在的推广价值，完整地、无bug地、高效率地实现了所有的系统功能。

* 1. 文档概述

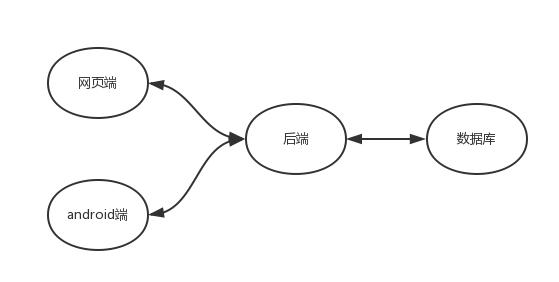
本文档按照模块划分的顺序编写，分为网页端，android端，后端与数据库三个模块，每模块中详细列举了细化功能。

本文档旨在让读者更好了解啪啪打卡系统的设计结构，方便系统的二次开发和维护。使用者具体操作流程详见用户指南。

2 引用文件

devise文档 <https://github.com/plataformatec/devise>

3 系统级策略设计

1. 前后端解耦合，网页端、android端分别同后端接口进行交互。  
   设计如下：  
   
2. 整套系统的交互基于HTTP协议。  
   网页端、后端、android 端预留了 SSL 的接口。今后可以轻松支持 HTTPS 协议传输。
3. 用户登录使用token和cookie系统同时管理，分别对应android端和网页端。用户登录使用devise库管理密码加密等操作。
4. 用户上传文件需要UTF-8编码。

4 系统体系结构设计

4.1 运行环境

* 网页端使用chrome、Safari、firefox、360、搜狗等主流浏览器访问。
* 后端要求使用 x86 架构 CPU， 1G 以上内存，需要有Internet访问。
* Android端运行环境：最低Android 4.0，体验好要求Android 4.4以上，最佳要求Android 6.0以上。Android 4.4以下可能发生显示异常，但是可以使用。  
  需要配备 Wi-Fi 模块以连接到数据库。  
  签到功能需要使用 GPS。  
  拍摄实验照片/录像的功能需要使用摄像头。

4.2 设计思想与关键技术

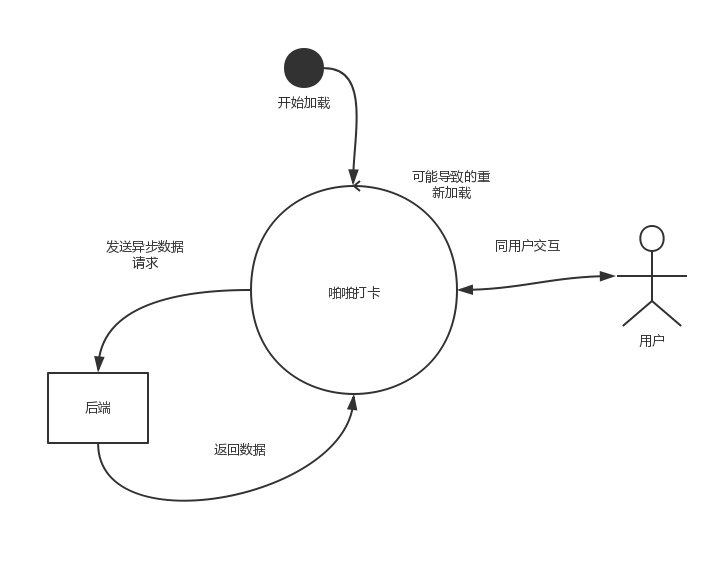
* 异步通信  
  后端同网页端或android端通过异步通信的方式传输。网页端所有数据都是通过ajax操作从约定的接口地址处获取JSON解析得到。
* 防止过度重绘  
  网页使用ajax实现异步刷新，Android通过convertView实现不重绘已有图形。提高效率。
* 自动获取手机号  
  使用 android 内置的获取手机号的接口，加速用户的登录过程。
* 防止GPS漂移  
  由于硬件限制，一次GPS定位难免会出现位置不准确的情况。为避免之，我们要以某个时间间隔连续进行若干次定位，当前仅当这些位置相差不超过某个阈值时，我们才认可这一次的定位结果，并将其发送至服务器。

4.3 模块具体设计及运行过程

4.3.1 网页端

网页端是为了教师管理课程和学生设计的，包括教师登录、修改个人信息、添加删除修改课程、管理学生及助教、查看实验课成绩和到课情况、打分并评价、推送消息和与学生私信联系等多种功能。

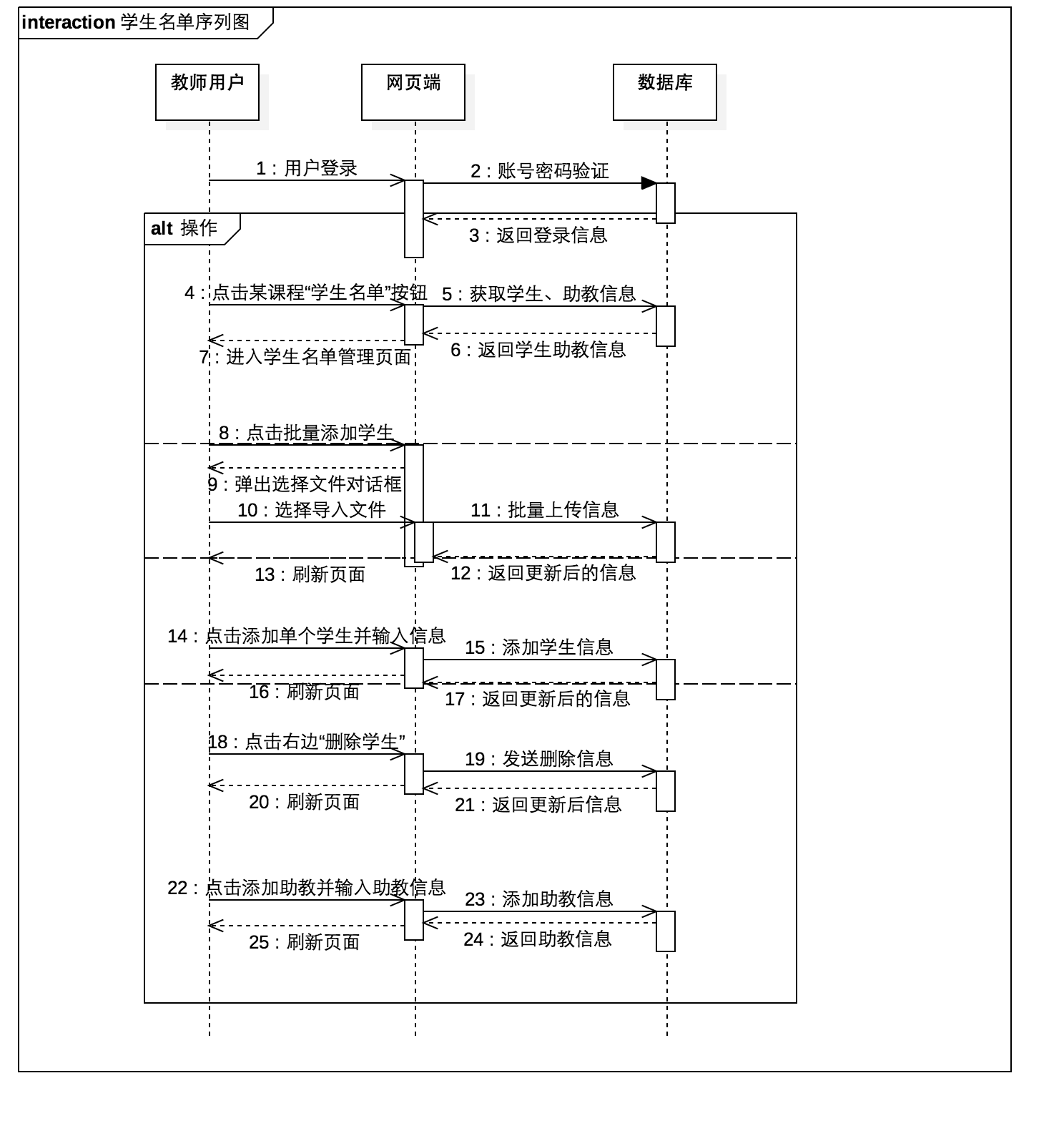
网页端的页面加载流程如下：



下面具体介绍一下各具体功能的运行序列：

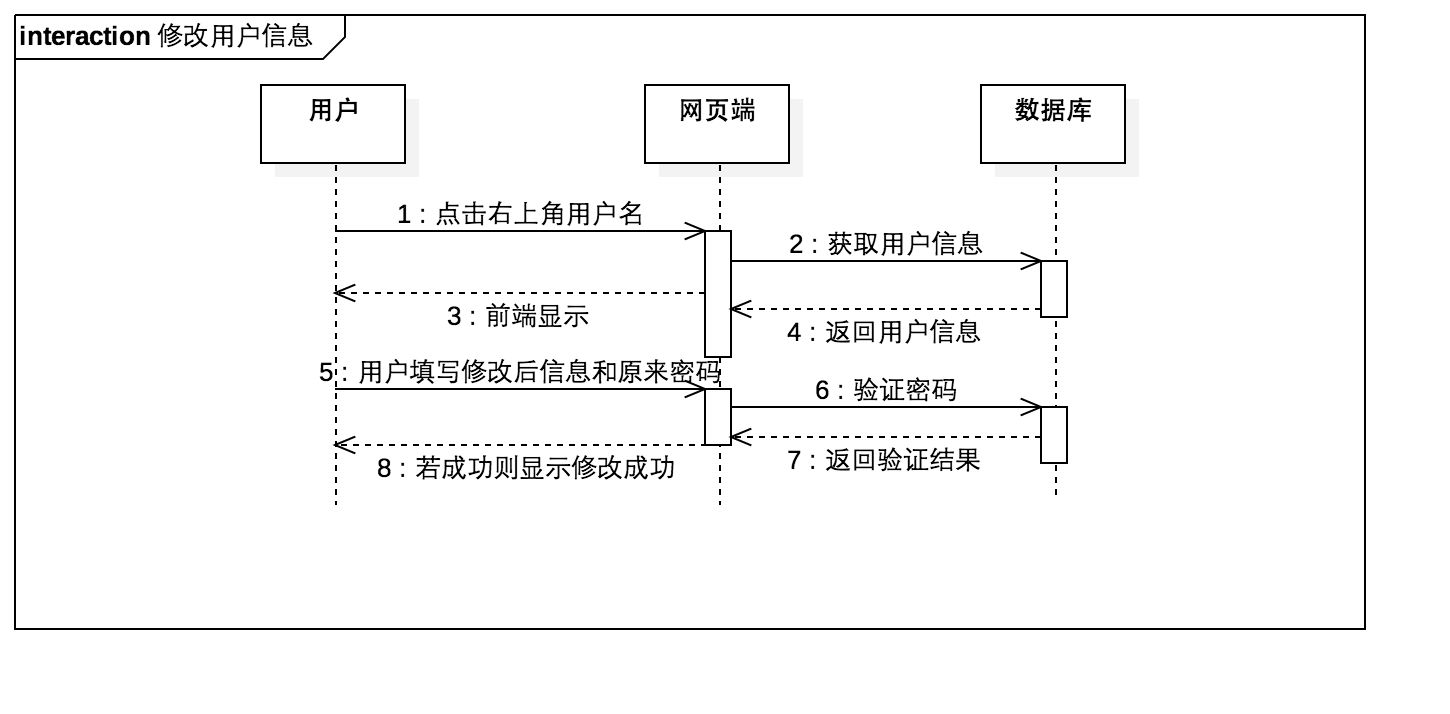
1. 查看课程学生助教名单。

序列图如下：



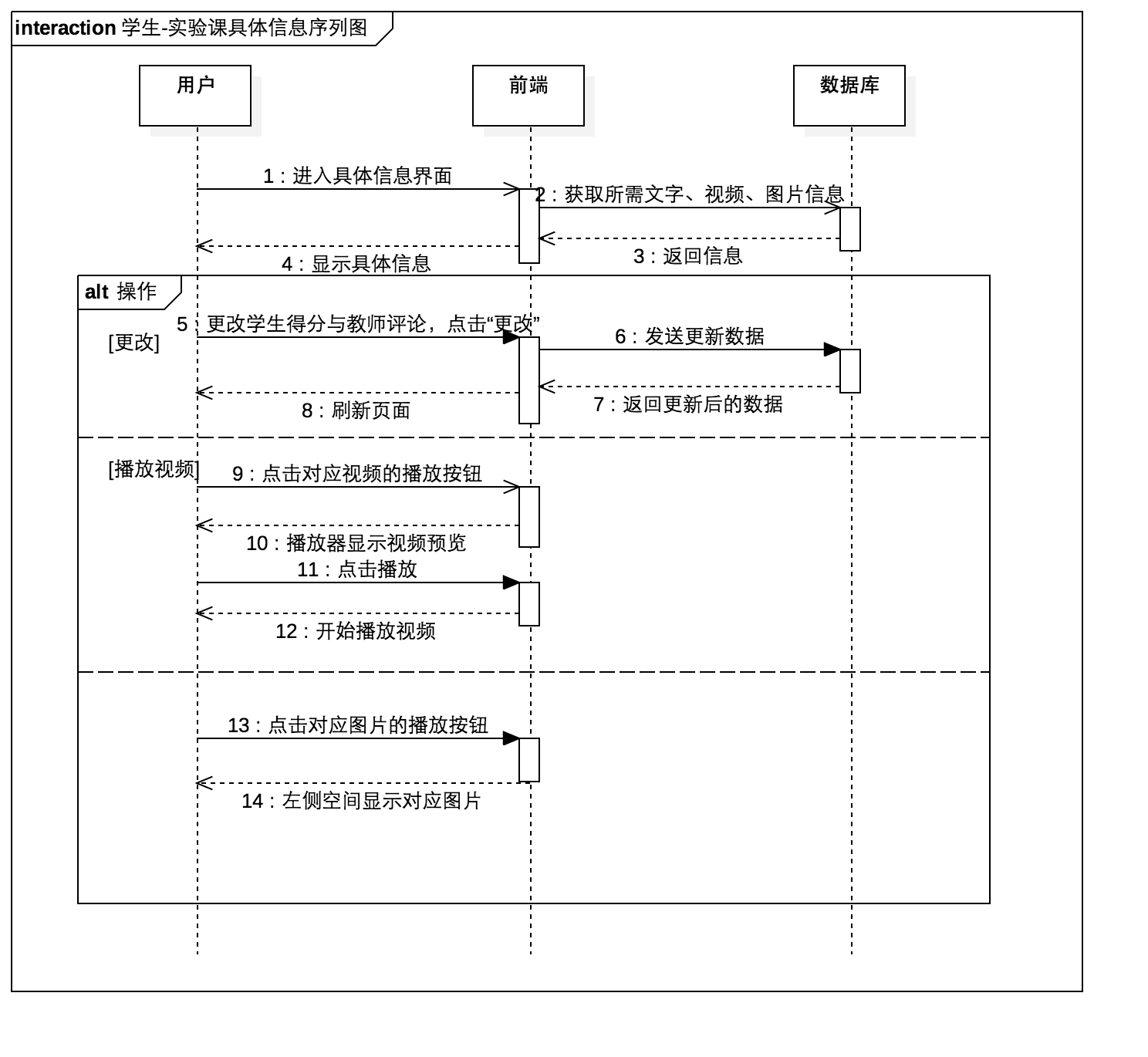
1. 修改用户信息。

序列图如下：



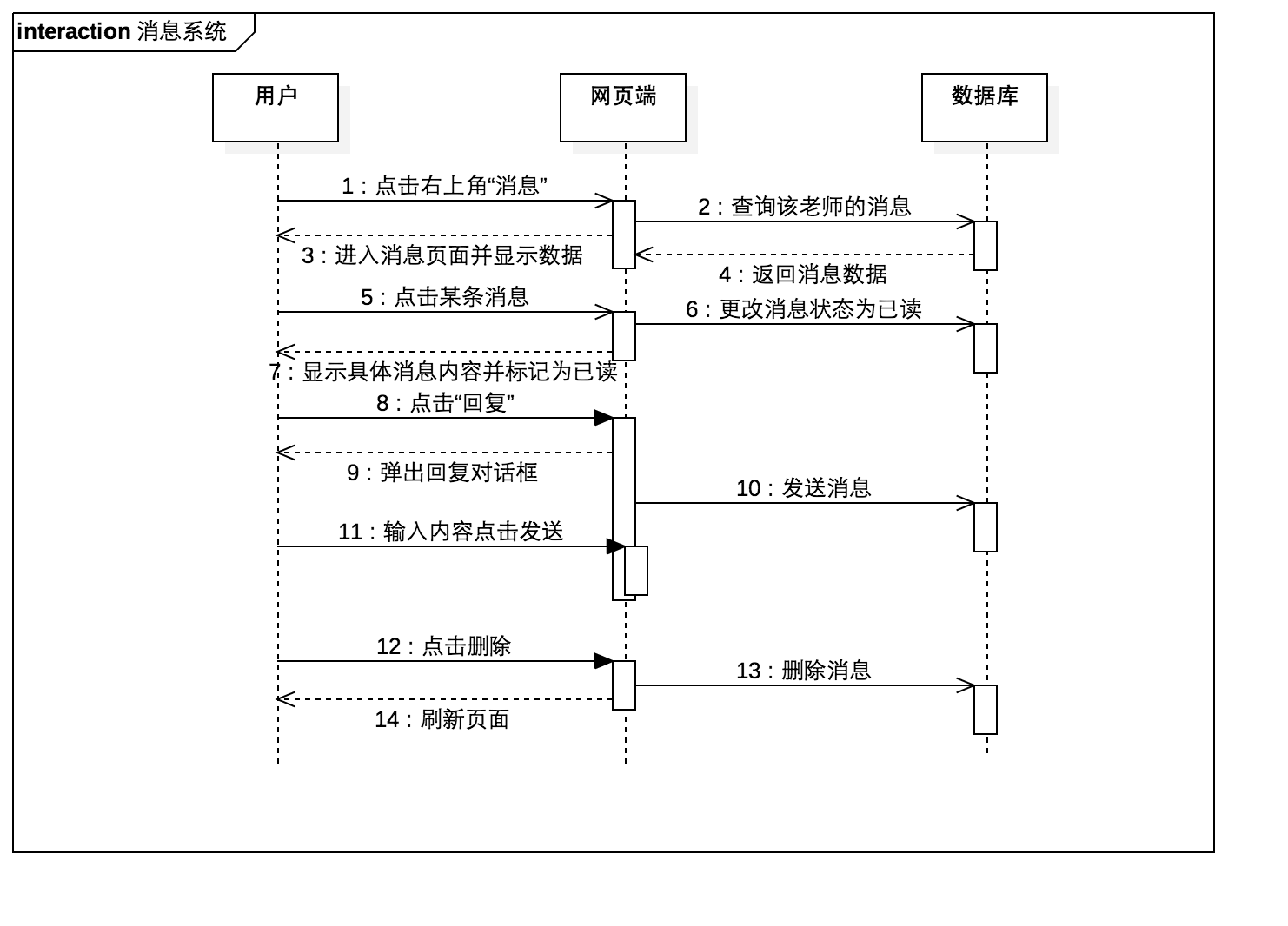
1. 查看具体学生-实验课信息。

序列图如下：

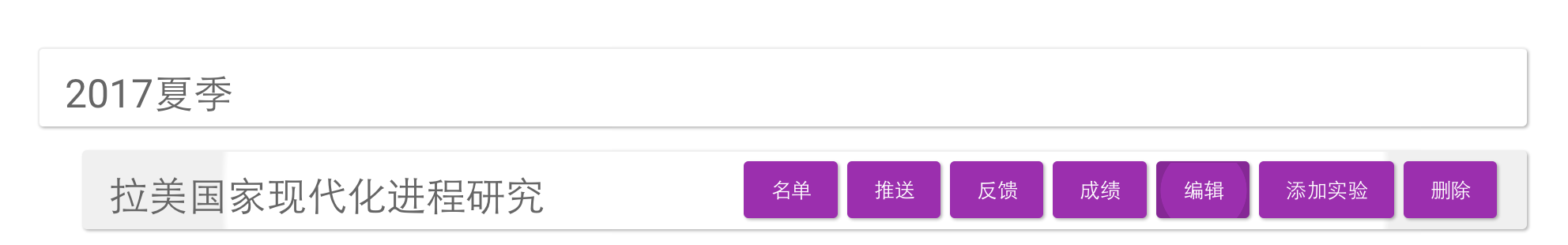


1. 消息系统

序列图如下：



1. 其他功能如导出成绩，修改课程信息等，只需要点击界面对应按钮即可。

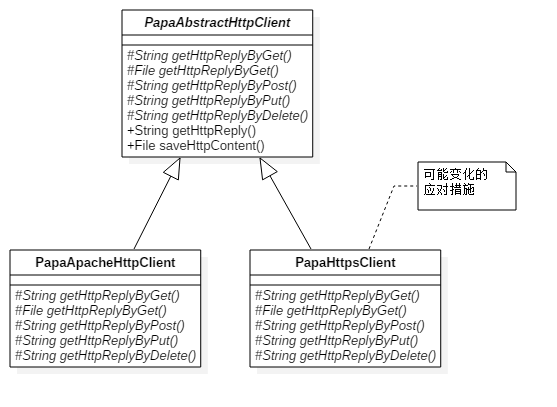


4.3.2 Android端

App 端是为了给学生和助教提供一个课程成绩管理、课程评价管理的平台而设计的。在实现过程中，我们充分利用了当今智能手机普遍具有的GPS、摄像、网络等功能，完成了一键式签到、精彩实验瞬间回放、与教师沟通互动、课程通知提醒等功能。

啪啪打卡 App 端在设计之初就充分考虑到了将来可能会应对到的变化。对于访问数据库的过程，我们将其划分为了几个不同类型的类之间的交互，以尽可能的实现将来的复用。

##### 4.3.2.1 底层HTTP(S) 协议接口



虽然目前本系统所有的信息交互都是基于 HTTP 协议，但众所周知，其安全性令人堪忧。因此，一个可能的变化就是需要使用 HTTPS协议。

在设计之初，我们就考虑设计了一个 HTTP 抽象类，其涵盖了基本的 HTTP和HTTPS的信息交互接口，全部封装成

[1](http://stash.secoder.net/projects/PAPA/repos/papa/browse/app/kernel/app/src/main/java/com/Back/NetworkAccess/papa/PapaAbstractHttpClient.java?at=dev#1) **public** **abstract** **class** PapaAbstractHttpClient;

类。设计之初，我们是通过封装org.apache.http.client.HttpClient包实现的网络访问，故有一个子类

[1](http://stash.secoder.net/projects/PAPA/repos/papa/browse/app/kernel/app/src/main/java/com/Back/NetworkAccess/papa/PapaAbstractHttpClient.java?at=dev#1) **public** **class** PapaApacheHttpClient **extends** PapaAbstractHttpClient;

实现具体的网络访问过程。如果将来需求发生变化，要实现 HTTPS 协议，只需建一个新类

[1](http://stash.secoder.net/projects/PAPA/repos/papa/browse/app/kernel/app/src/main/java/com/Back/NetworkAccess/papa/PapaAbstractHttpClient.java?at=dev#1) **public** **class** PapaHttpsClient **extends** PapaAbstractHttpClient

2{

3 //...

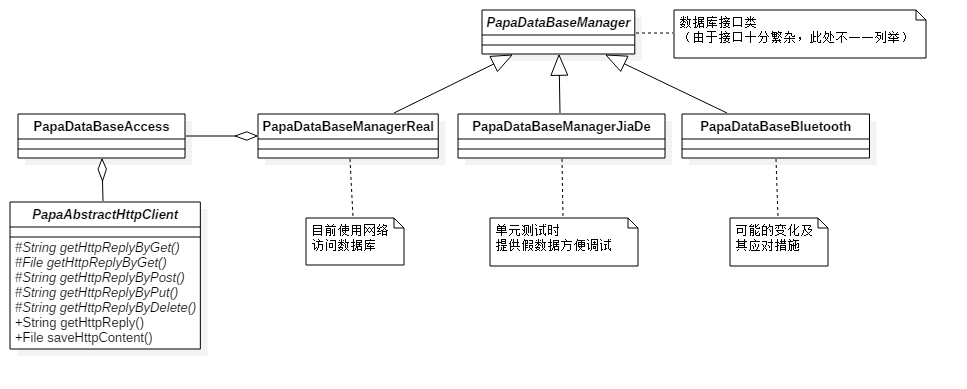
4}

并实现相关接口即可。

注意到无论是对于用户还是对于我们程序其他部分的逻辑，我们都不关心信息是采用何种协议传输的。因此，这样的设计是合理的。

##### 4.3.2.2 数据库访问接口

再退一步，目前本系统仍然是通过TCP/IP协议来与数据库交互的。可能将来我们会尝试走其他的协议（比如蓝牙协议等），来实现与数据库的交互。这时，整套HTTP(S) 协议的接口都可能无法继续使用。因此，我们设计了我们啪啪打卡标准数据库的相关接口。



[1](http://stash.secoder.net/projects/PAPA/repos/papa/browse/app/kernel/app/src/main/java/com/Back/NetworkAccess/papa/PapaAbstractHttpClient.java?at=dev#1) **public** **abstract** **class** PapaDataBaseManager;

是总的数据库接口类。

[1](http://stash.secoder.net/projects/PAPA/repos/papa/browse/app/kernel/app/src/main/java/com/Back/NetworkAccess/papa/PapaAbstractHttpClient.java?at=dev#1) **public** **class** PapaDataBaseManagerReal **extends** PapaDataBaseManager;

2**public** **class** PapaDataBaseManagerJiaDe **extends** PapaDataBaseManager;

是通过网络真实访问数据库的类和提供假数据方便单元测试的类。核心代码通过调用接口类提供的接口，实现不同运行状态下的不同功能。

[1](http://stash.secoder.net/projects/PAPA/repos/papa/browse/app/kernel/app/src/main/java/com/Back/NetworkAccess/papa/PapaAbstractHttpClient.java?at=dev#1) **public** **class** PapaDataBaseAccess;

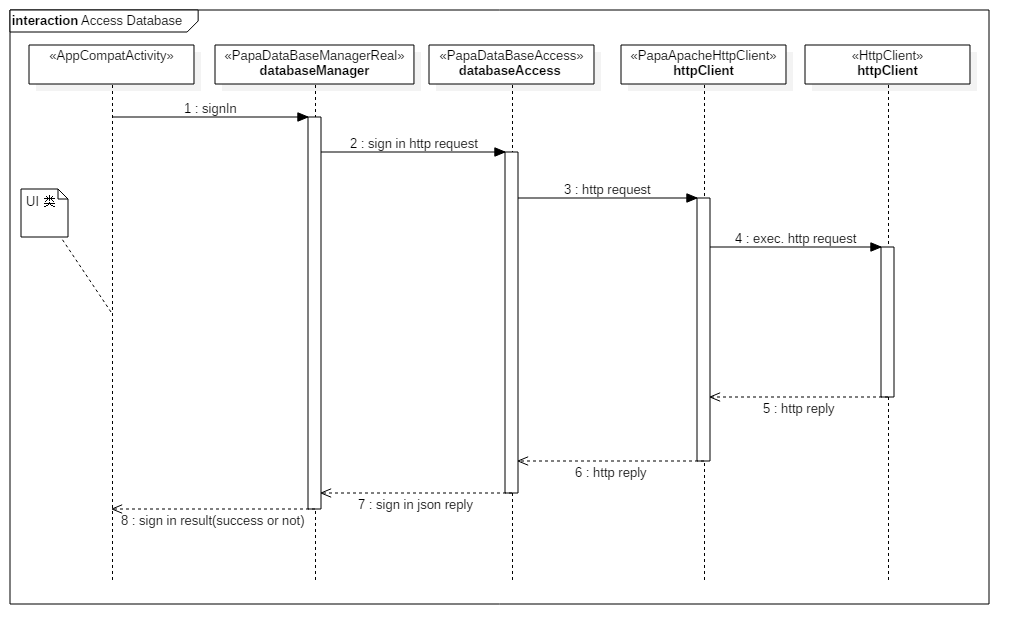
记录了服务器的相关信息，例如

* IP地址
* 端口号
* 等等

并对返回的文本内容进行JSON 解析，使之能够直接通过org.json.JSONObject 提供的相关接口进行进一步的操作。

##### 4.3.2.3 数据库访问过程

下面以登录为例，展示UI类访问数据库的过程



可以看到，用户的请求通过一层又一层的封装、嵌套，从最初的纯文本/文件信息，最后打包成数据包在网上传输。服务器端收到后，会根据协议解包，执行用户请求。

上图只是以登录为例举了个例子。得益于我们对数据库接口的定义，对于其他的用户操作，其序列图相同。

##### 4.3.2.4 整体时序图

###### 4.3.2.4.1 实验管理

###### 4.3.2.4.2 发信和收信

##### 4.3.2.5 UI设计

###### 4.3.2.5.1 Material Design

由于本APP是面向4.4以上的API，而且目前主流的安卓智能手机有很多是5.0以上的版本，因此编写的时候使用了一些5.0以上的特性，比如Floating Action Button, Recycler View, Coordinate layout 等。同时，在配色、布局等方面也参考了官方的设计文档。在低版本的手机上做了相应的处理，可以正常运行。

###### 4.3.2.5.2 BundleHelper

在Android程序中，如何在Activity之间传递数据是个重要的问题。在本APP中的解决方法是利用Intent的putExtras方法来传递。传递的内容是一个实现了Parcelable的类BundleHelper的对象，在不同的Activity之间这个对象只有一个，一个Activity被创建，除非是刚刚进入的EnterActivity，都会被传一个这个对象，从而保证了Activity之间信息的一致性。在这个类中，包装有信息username，student\_id等，需要的时候读取，获取了信息再写入即可，避免了传递信息的时候信息种类过多导致混乱的问题。

###### 4.3.2.5.3 TabLayout 和 Fragments

为了减少用户的操作数量，本APP对于一类操作的处理是把它们放在同一个Activity中，创建多个Tab，通过Tab的点击和滑动，切换到不同的Fragment，在这些Fragment中实现有类似但是不一定相同的功能，比如当用户选择了某一个实验进入时，会在同一个Activity中显示实验详情、实验成绩和实验结果，也可以点击进入实验评价，这样把多个功能集中在一起方便用户使用。同时，避免了在Activity生命周期中不可预测的泄露问题。

###### 4.3.2.5.4 提示信息的显示

为了提高工作的效率也为了减少网络请求的次数，本APP中很多的列表项中都含有该项的信息预览，这样用户不用点进去就能看见内容，提高了效率。在用户发信的时候，也有收件人名字的自动补全，避免打错以致无法正常发送。同时，在使用过程中经常会有Toast和SnackBar出现提示用户，提高使用过程中的体验。

###### 4.3.2.5.5 多媒体压缩

在用户上传实验结果的时候，会调用本机的API来照相、录像，得到的结果（照片、视频）大小不一，格式不一。为了统一格式和减少网络的压力，在上传之前，会对图片进行压缩，支持jpeg和png格式。但是目前视频的支持不是很好，无法进行恰当的转码使上传后任何浏览器都能正常播放。

###### 4.3.2.5.6 本地化

目前本APP仅支持中文和英文，默认是中文，当用户把系统语言调成英文之后，APP内的语言会自动转成英文，拓宽使用人群。

4.3.3 后端及服务器

以下内容摘自后端概述文档 ../db/doc/overview.md。读者可查看原始文档。

以下内容摘自后端设计文档 ../db/doc/database.md。读者可查看原始文档。

##### 4.3.3.1. 总体结构

--- 使用了Ruby On Rails框架，该框架采用MVC架构 如图

网页端和App端共用一套JSON API，所有数据库操作均由JSON API进行，网页端不直接操作数据库。 支持使用cookie和token的认证方式，网页端用cookie认证，App端用token认证。

##### 4.3.3.2. 目录结构

app/ 应用程序主要代码

app/assets/ 动态资源文件（动态js，css）

app/controllers/ MVC的控制器部分

app/models/ MVC的模型部分

app/views/ MVC的视图部分

bin/ rails的可执行文件

config/ 配置文件

config/routes.rb 路由配置（URL和controller方法的对应关系）

config/initializers/ 初始化配置和常量

db/ 数据库

db/migrate 数据库结构文件

doc/ 文档

log/ 日志

public/ 静态资源文件的目录

tmp/ 运行时临时文件

libs/ vendor/ 第三方库

##### 4.3.3.3. 数据库设计

###### 4.3.3.3.1 UML

1. 考虑到某门课程的助教可能是另一门课程的学生， User表储存的是所有用户的信息，包括老师，助教，学生， 用Participation表代表某个用户参与某门课这个关系，Participation表有一个role字段， 该字段表示用户是以什么身份参加的该门课程。取值为assistant和student
2. User Model(app/models/user.rb)的login方法允许用用户用不同方法登陆，比如要增加用户名登陆，只需 在其中加入一个user.username即可
3. Course Model(app/models/course.rb)的students和assistants方法可以简单的获取某门课程的所有学生和助教
4. 其余的Model都是rails标准的使用方法没有多余的方法，很好理解。

5 接口设计

以下内容摘要自db/doc/api.md，读者可查看其原始内容。

### 5.1 表示法介绍

下表为所有API的使用方法，分别有ID， HTTP方法，URL，参数，返回值，错误，权限和注释

参数：标准的http参数

返回值：每个API返回均为JSON，包含一个status属性， 如果成功为success，失败为fail，如果发生错误 status属性会是错误信息，在“错误”中介绍。

权限：分为student, assistant, teacher, admin，权限依次增大， “权限”一列表示的是所需的最低权限，如果权限不够会返回错误。

有些接口没有包含参数的注释，这是因为这是一个创建对象或者修改对象的方法，参数列表在后面“对象的属性”一章介绍。

有些接口没有包含返回值，那么默认就是返回status是success或者fail

有些接口的返回值包含一个对象，对象的属性在后面“对象的属性”一章介绍

### 5.2 API列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | HTTP方法 | 地址 | 注释 | 权限 |
| 用户相关 | | | | |
| 1 | GET | /users/1.json | 得到用户的信息 | "user": user |
| 2 | POST | /users/sign\_in.json | 登陆 |  |
|  | 参数 | user[login]=xxx&user[password]=123&user[remember\_me]=0 | | |
|  | 返回 | token: '123456777777777', id: 123 | | |
| 3 | DELETE | /users/sign\_out.json | 注销 | User |
|  | 错误 | 暂时不检查错误 | | |
| 4 | POST | /users.json | 注册普通用户, 非(老师或管理员) | Everyone |
|  | not\_implemented | | |  |
| 5 | PUT | /users/1.json | 修改用户信息 | User |
|  | not\_implemented | | |  |
| 8 | GET | /users/current.json | 获取当前用户id | Current User |
|  | 返回值 | "id": "123" | | |
|  | 错误 | permission\_denied: 未登录 | | |
| 学期相关 | | | | |
| 10 | GET | /semesters.json | 获得所有学年 | Student |
|  | 返回值 | "semesters": [semester, ...] | | |
|  | 错误 | always successful | | |
| 11 | POST | /semesters.json | 添加一个学年 | Admin |
|  | 错误 | permission\_denied: 不是管理员 | | |
|  |  | internal\_error: 其他内部错误, 如果遇到请联系开发者 | | |
| 12 | PUT | /semesters/1.json | 修改学年信息 | Admin |
|  | 错误 | permission\_denied: 不是管理员 | | |
|  |  | resource\_not\_found: semester\_id不存在 | | |
|  |  | internal\_error: 其他内部错误, 如果遇到请联系我 | | |
| 13 | DELETE | /semesters/1.json | 删除某个学年 | Admin |
|  | 错误 | permission\_denied: 不是管理员 | | |
|  |  | resource\_not\_found: semester\_id不存在 | | |
|  |  | internal\_error: 其他内部错误, 如果遇到请联系我 | | |
| 14 | GET | /semesters/1/courses.json | 获得某个学年的所有课程 | Student |
|  | 参数 | "courses": [course, ...] | | |
|  | 错误 | resource\_not\_found: semester\_id不存在 | | |
|  |  | internal\_error: 其他内部错误, 如果遇到请联系我 | | |
| 15 | GET | /semesters/default.json | 获得默认学期 | Student |
|  | 参数 | "semester": semester | | |
|  | 错误 | invalid\_fields: 总学年数为0 | | |
| 课程相关 | | | | |
| 20 | PUT | /courses/1.json | 修改课程 | Teacher |
|  | 错误 | permission\_denied: 当前用户不是该课程的老师 |  |  |
|  |  | internal\_error: 其他内部错误, 如果遇到请联系我 |  |  |
| 21 | GET | /courses/1.json | 获得ID为1的课程 | Student |
|  | 返回值 | Course对象 |  |  |
|  | 错误值 | resource\_not\_found: invalid course\_id |  |  |
| 22 | DELETE | /courses/1.json | 删除课程 | Teacher |
|  | 错误 | permission\_denied: 当前用户不是该课程的老师 | | |
|  |  | internal\_error: 其他内部错误, 如果遇到请联系我 | | |
| 课程相关 | | | | |
| 23 | GET | /courses/1/teachers.json | 获取该门课所有老师 | Student |
|  | 返回值 | Teacher数组 |  |  |
|  | 错误 | invalid\_operation: course\_id未指定 | | |
|  |  | resource\_not\_found: course\_id不存在 | | |
| 24 | POST | /courses/1/teachers/1.json | 把老师添加到课程 | Admin |
|  | 错误 | permission\_denied: 不是管理员 |  |  |
|  |  | resource\_not\_found: course\_id或teacher\_id不存在 |  |  |
| 25 | DELETE | /courses/1/teachers/1.json | 从课程中删除老师 | Admin |
|  | 错误 | permission\_denied: 不是管理员 |  |  |
|  |  | resource\_not\_found: course\_id或teacher\_id不存在 |  |  |
| 26 | GET | /courses/1/students.json | 获得id=1课的所有学生 | Teacher |
|  | 返回值 | "students": [student, ...] |  |  |
|  | 错误 | resource\_not\_found: course\_id不存在 |  |  |
|  |  | permission\_denied: 不是该课程的老师 |  |  |
| 27 | GET | /courses/1/assistants.json | 类似上一个 | Student |
|  | 返回值 | "assistants": [assistant, ...] |  |  |
|  | 错误 | resource\_not\_found: course\_id不存在 |  |  |
|  |  | permission\_denied: 不是该课程的老师 |  |  |
| 28 | POST | /courses/1/students/1.json | 添加学生到指定课程 | Teacher |
|  | 错误 | resource\_not\_found: course\_id不存在 |  |  |
|  |  | permission\_denied: 不是该课程的老师 |  |  |
|  |  | internal\_error: 其他内部错误, 如果遇到请联系我 |  |  |
| 29 | POST | /courses/1/students.json |  |  |
|  | 参数 | User对象， 必填：username，email，phone，student\_no | Teacher |  |
|  | 错误 | resource\_not\_found: course\_id不存在 |  |  |
|  |  | token\_invalid: 未登录 |  |  |
|  |  | permission\_denied: 当前用户不是该门课程的老师 |  |  |
|  |  | format\_error: json格式错误 |  |  |
|  |  | invalid\_fields: student\_number未指定 |  |  |
|  | \*\*注释 | 部分学生添加失败 invalid\_fields: ["123", "111"...] 数字为student\_number |  |  |
|  |  | 批量添加学生, 默认密码是学号 |  |  |
| 助教相关 | | | | |
| 30 | POST | /courses/1/assistants/1.json | 添加助教到指定课程 | Teacher |
|  | 错误 | resource\_not\_found: course\_id不存在 |  |  |
|  |  | permission\_denied: 不是该课程的老师 |  |  |
|  |  | internal\_error: 其他内部错误, 如果遇到请联系我 |  |  |
| 31 | POST | /courses/1/assistants.json | 批量添加助教 | Teacher |
|  | 参数 | User对象， 必填：username，email，phone |  |  |
|  | 错误 | resource\_not\_found: course\_id不存在 |  |  |
|  |  | token\_invalid: 未登录 |  |  |
|  |  | permission\_denied: 当前用户不是该门课程的老师 |  |  |
|  |  | format\_error: json格式错误 |  |  |
|  |  | invalid\_fields: student\_number未指定 |  |  |
|  |  | 批量添加助教, 默认密码是email |  |  |
| 32 | GET | /courses/1/lessons.json | 获得id=1课所有实验课 | Student |
|  | 返回值 | Lesson数组 |  |  |
|  | 错误 | resource\_not\_found: course\_id不存在 |  |  |
| 33 | POST | /courses/1/lessons.json | 向课程中添加实验课 | Teacher |
|  | 错误 | permission\_denied: 不是该课程的老师 |  |  |
|  |  | invalid\_fields: lesson参数无效 |  |  |
| 课程相关 | | | | |
| 40 | GET | /lessons/1.json | 获得某门实验课的信息 | Student |
|  | 错误 | resource\_not\_found: lesson\_id不存在 |  |  |
| 41 | DELETE | /lessons/1.json | 删掉实验课 | Teacher |
|  | 错误 | permission\_denied: 不是该课程的老师 |  |  |
|  |  | resource\_not\_found: lesson\_id不存在 |  |  |
|  |  | internal\_error: 其他内部错误, 如果遇到请联系我 |  |  |
| 42 | GET | /lessons/1/comments.json 获得某门课程的评价 | Teacher |  |
| 43 | POST | /lessons/1/start\_sign\_up.json | 发起签到 | Teacher |
|  | 错误 | not\_in\_lesson\_time: 当前不是实验课时间 |  |  |
| 对学生的评价相关 | | | | |
| 44 | GET | /lessons/1/students/1/comment.json | 助教对学生的评价 | Student |
| 45 | POST | /lessons/1/students/1/comments.json | 添加助教对学生的评价 | Assistant |
| 46 | GET | /lessons/1/students.json | 某门实验课的到课学生列表 | Teacher |
| 47 | POST | /lessons/1/attendance.json | 学生签到 | Student |
|  | 错误 | resource\_not\_found: lesson\_id或者student\_id找不到 |  |  |
|  |  | permission\_denied: 当前学生id不匹配 |  |  |
| 48 | DELETE | /lessons/1/attendance.json | 登出 |  |
| 文件相关 | | | | |
| 50 | GET | /lessons/1/files.json | 获得该门实验课的简介文件 | Student |
|  | 错误 | resource\_not\_found: lesson\_id不存在 |  |  |
|  |  | invalid\_fields: file参数或type参数未指定或者格式不正确 |  |  |
|  |  | file\_too\_big: 文件大小超过指定大小(100M, 该数值请查看config/initializers/constants.rb) |  |  |
| 51 | POST | /lessons/1/files.json | 向课程中添加文件 | Teacher |
|  | 错误 | resource\_not\_found: lesson\_id不存在 |  |  |
|  | \*\*注意 | 如果lesson\_id不正确或者超出范围, 不会返回错误, 只会查不到数据 |  |  |
| 学生相关 | | | | |
| 60 | GET | /students/1.json | 获得id=1学生的信息 | Student |
|  | 错误 | permission\_denied: 未登录 |  |  |
|  |  | resource\_not\_found: student\_id不存在 |  |  |
| 61 | GET | /students/1/files.json | 获得学生所有文件列表 | Student |
|  | 返回值 | FileResource数组 |  |  |
|  | 错误 | permission\_denied: 当前用户不是student\_id |  |  |
|  | \*\*注意 | 如果lesson\_id不正确或者超出范围, 不会返回错误, 只会查不到数据 |  |  |
| 62 | POST | /students/1/files.json | 学生添加文件 | Student |
|  | 错误 | resource\_not\_found: student\_id不存在 |  |  |
|  |  | permission\_denied: student\_id和当前用户不匹配, 或者lesson\_id不属于student\_id |  |  |
|  |  | invalid\_fields: file参数或type参数未指定或者格式不正确 |  |  |
|  |  | file\_too\_big: 文件大小超过指定大小(100M, 该数值请查看config/initializers/constants.rb) |  |  |
| 63 | DELETE | /students/1/files/1.json | 删除文件 | Creator |
|  | 错误 | resource\_not\_found: file\_id不存在 |  |  |
|  |  | permission\_denied: 当前用户不是文件创建者 |  |  |
|  | \*\*注意 | 这里忽略student\_id |  |  |
| 64 | GET | /students/1/lessons/1/comment.json 某位学生对课程的评价 |  |  |
|  | 错误 | resource\_not\_found: student\_id或者lesson\_id不存在或者该学生不在该课程中 |  |  |
| 65 | POST | /students/1/lessons/1/comments.json |  |  |
|  | 错误 | resource\_not\_found: student\_id或者lesson\_id不存在或者该学生不在该课程中 |  |  |
| 课程/实验课相关 | | | | |
| 70 | GET | /students/1/courses.json | 获得所有课程 | Student |
|  | 错误 | permission\_denied: student\_id和当前登陆用户不匹配 |  |  |
|  |  | resource\_not\_found: student\_id不存在 |  |  |
|  |  | internal\_error: 其他内部错误, 如果遇到请联系我 |  |  |
| 71 | POST | /students/1/courses/1.json | 给学生添加课程 | Student |
|  | 错误 | permission\_denied: student\_id和当前登陆用户不匹配 |  |  |
|  |  | resource\_not\_found: student\_id不存在 |  |  |
|  |  | internal\_error: 其他内部错误, 如果遇到请联系我 |  |  |
| 72 | DELETE | /students/1/courses/1.json | 把学生从课程中移除 | Teacher |
|  | 错误 | permission\_denied: 当前用户不是该课程的老师 |  |  |
|  |  | resource\_not\_found: student\_id不存在或者course\_id不存在或者student\_id和不在course\_id的学生列表中 |  |  |
|  |  | internal\_error: 其他内部错误, 如果遇到请联系我 |  |  |
| 73 | GET | /students/1/lessons/1/files.json | 获得某门实验课某个学生的所有文件 | Student |
|  | 错误 | permission\_denied: student\_id和当前登陆用户不匹配 |  |  |
|  | \*\*注意 | 如果lesson\_id不正确或者超出范围, 不会返回错误, 只会查不到数据 |  |  |
| 74 | POST | /students/1/lessons/1/files.json | 在某门实验课上添加视频图片 | Student |
|  | 错误 | permission\_denied: 一下三条至少有一条不满足: |  |  |
|  |  | | 1. 用户已经登陆或当前token有效 |  |
|  |  | | 2. student\_id包含在lesson\_id对应的课程的学生名单中 \*\*\*(或者当前用户是课程的老师) |  |
|  |  | | 3. 当前用户是student\_id或当前用户在lesson\_id的助教名单中 |  |
|  |  | invalid\_fields: file参数或type参数未指定或者格式不正确 |  |  |
|  |  | file\_too\_big: 文件大小超过指定大小(100M, 该数值请查看config/initializers/constants.rb) |  |  |
| 75 | DELETE | /students/1/lessons/1/files/1.json | 在某门实验课上删除视频照片 | Creator |
|  | 错误 | resource\_not\_found: file\_id不存在 |  |  |
|  |  | permission\_denied: 当前用户不是文件创建者 |  |  |
|  | \*\*注意 | 这里忽略student\_id和lesson\_id |  |  |

GET /assistants.json 获得所有助教信息 "assistant": [assistant, ...] Student

not\_implemented(该API是否有用?)

GET /assistants/1.json 获得某个助教信息 "assistant": assistant Assistant

resource\_not\_found: 助教不存在

permission\_denied: 当前用户不是assistant\_id

GET /assistants/1/courses.json 获得某个助教的所有课程 "courses": [course, ...] Assistant

permission\_denied: assistant\_id和当前用户不匹配

resource\_not\_found: assistant\_id不存在

internal\_error: 其他内部错误, 如果遇到请联系我

POST /assistants/1/courses/1.json 给某个助教添加某门课 Assistant

permission\_denied: assistant\_id和当前用户不匹配

resource\_not\_found: assistant\_id不存在

internal\_error: 其他内部错误, 如果遇到请联系我

GET /assistants/1/files.json 助教上传的所有文件 "files": [file, ...] Assistant

permission\_denied: 当前用户和assistant\_id不匹配

resource\_not\_found: assistant\_id不存在

# 老师相关

GET /teachers/1.json 获得某个老师的信息 "teacher": teacher Student

resource\_not\_found: teacher\_id不存在

POST /teachers.json 添加老师 teacher parameters Admin

not\_implemented

DELETE /teachers/1.json 删除老师 id Admin

resource\_not\_found: teacher\_id不存在

internal\_error: 其他内部错误, 如果遇到请联系我

UPDATE /teachers/1.json 修改老师信息 teacher parameters The Teacher

not\_implemented

GET /teachers/1/courses.json 获得老师的所有课程 "courses": [course, ...] Student

permission\_denied: teacher\_id和当前登陆用户不匹配

resource\_not\_found: teacher\_id不存在

internal\_error: 其他内部错误, 如果遇到请联系我

POST /teachers/1/courses.json 给老师添加课程 course parameters Teacher

permission\_denied: teacher\_id和当前登陆用户不匹配

resource\_not\_found: teacher\_id不存在

internal\_error: 其他内部错误, 如果遇到请联系我

GET /teachers/1/lessons.json 获得一个老师的所有实验课

resource\_not\_found: teacher\_id不存在

# 消息推送

GET /students/1/messages.json 查询学生的所有消息 "messages": [message ...] Student

resource\_not\_found: student\_id不存在

permission\_denied: student\_id和当前用户不匹配

POST /courses/1/messages.json 发送消息到某门课的所有学生 "message": message Assistant

resource\_not\_found: course\_id不存在

permission\_denied: 当前用户不是course\_id的老师且不是course\_id的助教

invalid\_field: message创建参数取值范围或类型错误

# 文件相关

GET /files/1.json 得到该文件的信息 "file": file

resource\_not\_found: file\_id不存在

POST /files.json 上传文件 file parameters

permission\_denied: 未登录

file\_too\_big: 文件大小超过指定大小(100M, 该数值请查看config/initializers/constants.rb)

DELETE /files/1.json 删除自己上传的文件 Creator

resource\_not\_found: file\_id不存在

permission\_denied: 当前用户不是文件创建者

# 学生老师发消息

POST /users/1/messages.json 当前用户给该用户发送消息

resource\_not\_found: user\_id 不存在

too\_often: 发送消息太频繁

GET /messages.json 获取当前用户的所有消息

permission\_denied: 未登录

POST /messages/1/read.json

resource\_not\_found

GET /messages/new\_messages\_count.json

permission\_denied: 未登录

DELETE /messages/1.json

not\_implemented

# App更新

GET /android/current\_version.json 得到当前最新版本号 {"version": "xx", "apk\_path": "xx"} Student

always successful

POST /android/current\_version.json 上传apk version=xx apk Admin

permission\_denied: 当前用户不是管理员

# 快捷查询

GET /courses/1.json?full=true

{ status: successful,

course: { xx, 和从前一样, ..., students: [{ id: 1, name: '123', email: , phone: ,

comment: { score: , content: , creator\_name } } ]}

GET /lessons/1.json?full=true

### 5.3 对象属性

semester id=int

name=string,

has\_many courses

course id=int,

name=string,

description=string

,

belongs\_to semester,

has\_many lessons,

has\_many participations,

has\_many students, through: participation, as: :user

has\_many teachers, through: teaching\_course

has\_many messages

participation id=int,

role=string

,

belongs\_to course,

belongs\_to user

lesson id=int,

name=string,

description=string,

start\_time=datetime,

end\_time=datetime,

location=string,

,

belongs\_to course

has\_many attached\_files, as: :files

has\_many lesson\_comments,

has\_many student\_comments

lesson\_files id=int

belongs\_to lesson

belongs\_to file\_resource

student\_files id=int

belongs\_to student

belongs\_to file\_resource

belongs\_to lesson

assistant\_files id=int

belongs\_to assistant

belongs\_to file\_resource

belongs\_to lesson

user id=int,

name=string,

student\_number=int,

phone=string,

email=string,

class=string,

department=string

,

has\_many lesson\_comments

belongs\_to avator, as: :file

has\_many teaching\_courses

has\_many assisting\_courses

has\_many learning\_courses

has\_many lesson\_statuses

has\_many assistant\_to\_student\_comments, as: student\_comments, foreign\_key: :creator\_id

has\_many from\_assistant\_comments, as: student\_comments, foreign\_key: :student\_id

has\_many posted\_messages, as: :message, foreign\_key: :creator\_id

teaching\_course id=int

belongs\_to user

belongs\_to course

lesson\_status id=int

score=string

belongs\_to user

belongs\_to lesson

timestamps

lesson\_comment id=int, // 学生对实验课的评价

content=string,

score=int

timestamps

,

belongs\_to creator, as: :user,

belongs\_to lesson

student\_comment id=int, // 助教对学生的评价

content=string,

score=int(0-10)

timestamps

,

belongs\_to lesson

belongs\_to creator, as: :user,

belongs\_to student, as: :user,

message id=int,

type=string(homework|notification),

title=string,

deadline=datetime,

content=string

timestamps

,

belongs\_to creator, as: :user

belongs\_to course

file id=int,

type=string,

name=string,

path=string,

timestamps

,

belongs\_to creator, as: :user

android\_app id=int,

version=string,

timestamps

,

belongs\_to file

6 出错处理及维护设计

6.1出错处理：

* 如果网络不畅，在设定的时间内数据库没有回复，则会终止网络访问，并向用户提示。

6.2维护设计：

* Android端提供检查更新和错误反馈，在一定程度上实现后期的改进工作。

7 尚待解决的问题

1. 网页端视频播放器对部分格式的视频和部分浏览器支持不好。
2. 不能根据Android系统版本进行不同的适配，兼容性不是很好。
3. android视频使用标准的系统 API 录制。由于当今android 版本不一，因此录制出的视频的编码也不一。在网页端，可能因为出现不常见的编码而导致无法播放视频的现象。