

# 啪啪打卡设计文档

## 1 引言

### 1.1 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。

### 1.2 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。

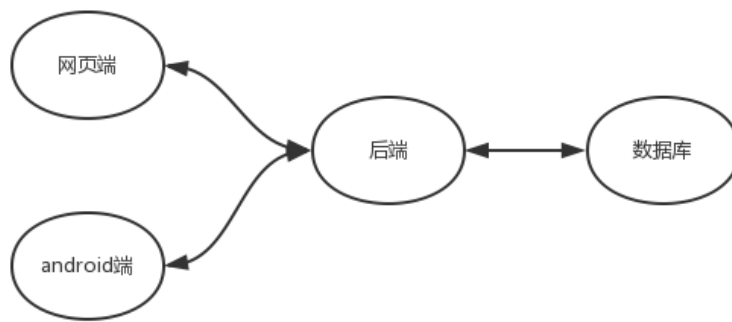
本文档旨在让读者更好了解啪啪打卡系统的设计结构，方便系统的二次开发和维护。使用者具体操作流程详见用户指南。

## 2 引用文件

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

## 3 系统级策略设计

- A. 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：



- B. 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。
- C. 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。
- D. 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。

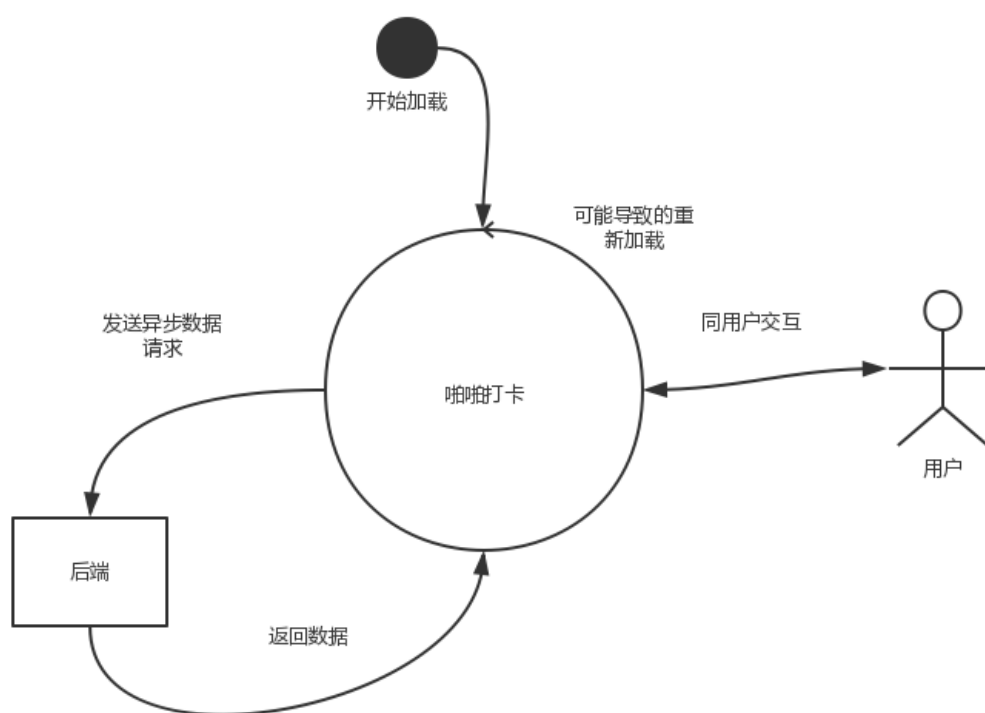
- **防止 GPS 漂移**  
由于硬件限制，一次 GPS 定位难免会出现位置不准确的情况。为避免之，我们要以某个时间间隔连续进行若干次定位，当前仅当这些位置相差不超过某个阈值时，我们才认可这一次的定位结果，并将其发送至服务器。

## 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。

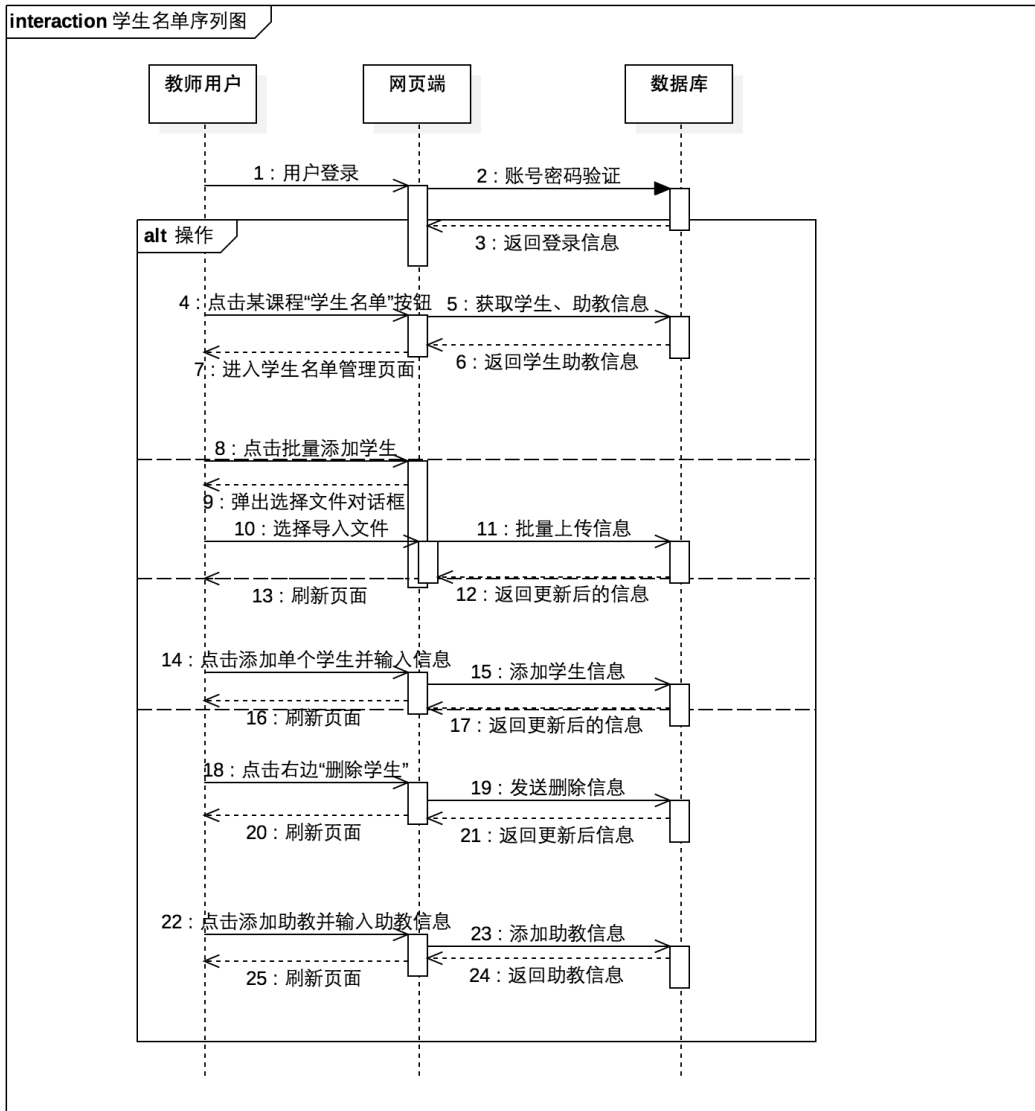
网页端的页面加载流程如下：



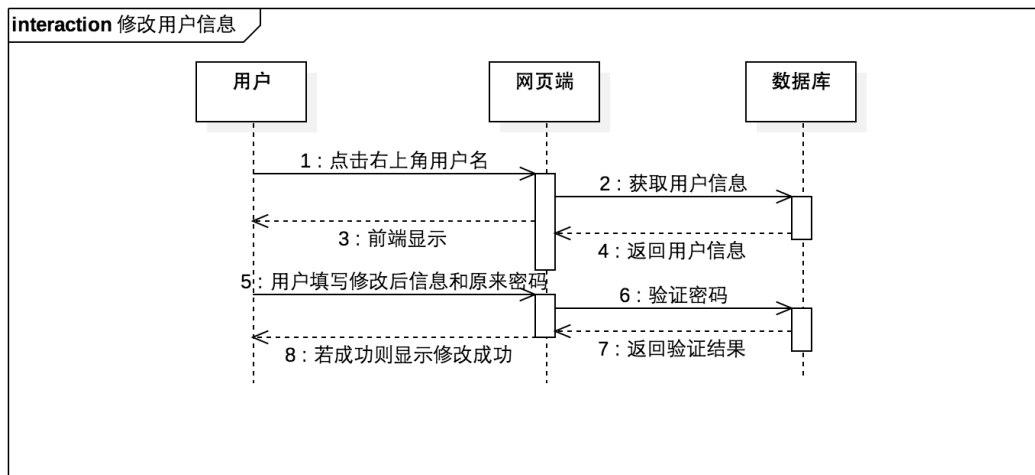
下面具体介绍一下各具体功能的运行序列：

- (1) 查看课程学生助教名单。

序列图如下：

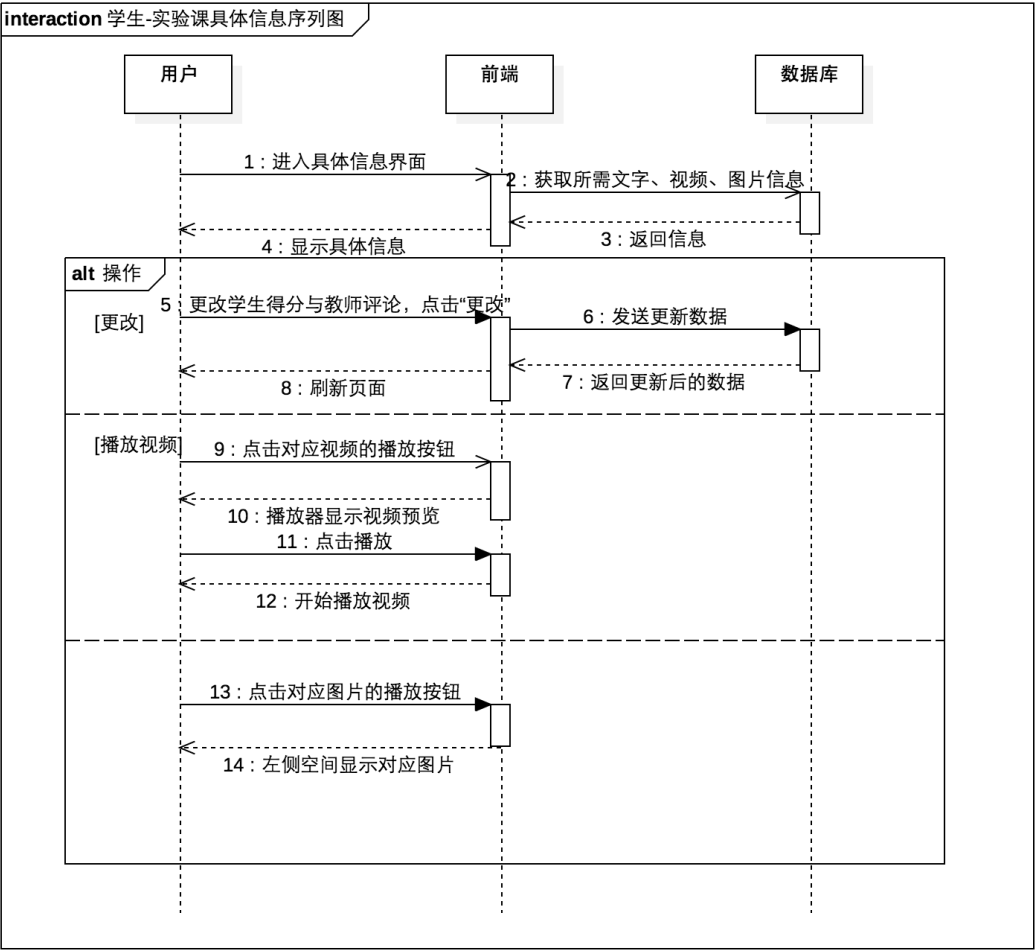


(2) 修改用户信息。  
序列图如下：

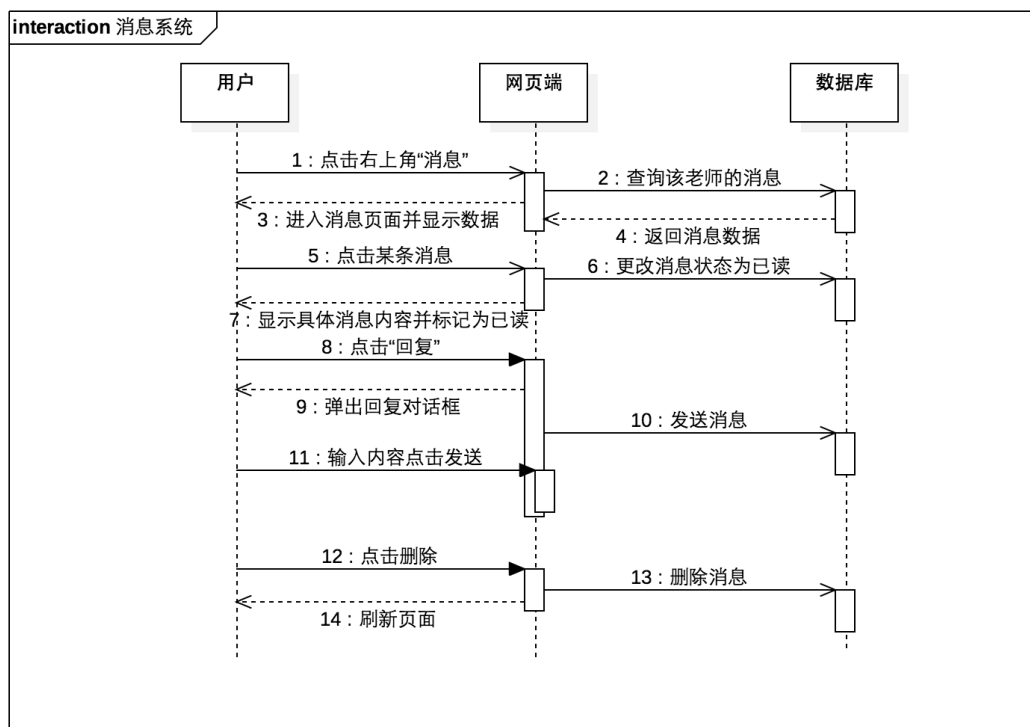


(3) 查看具体学生-实验课信息。

序列图如下：



(4) 消息系统  
序列图如下：



(5) 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。

2017夏季

拉美国国家现代化进程研究

名单

推送

反馈

成绩

编辑

添加实验

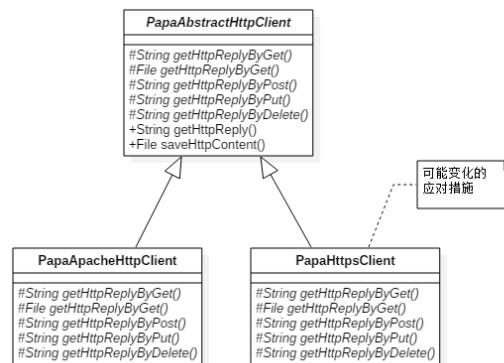
删除

### 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 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 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

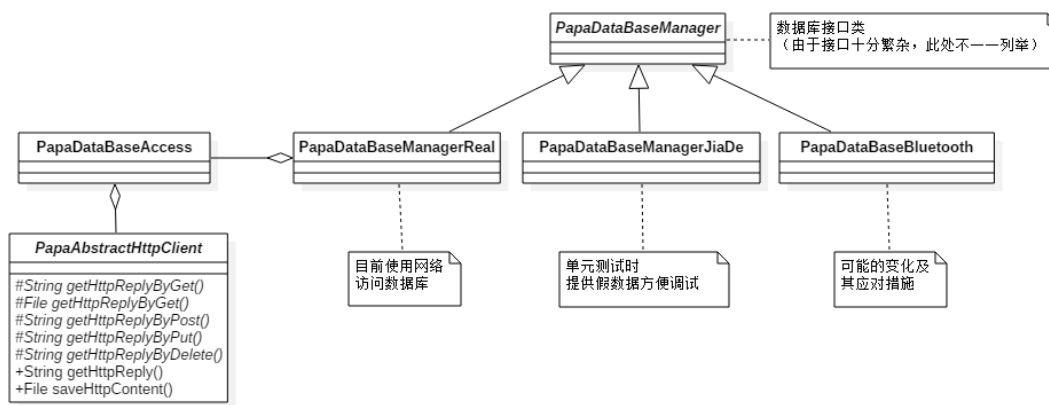
```
1 public class PapaHttpsClient extends PapaAbstractHttpClient
2 {
3     //...
4 }
```

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 public abstract class PapaDataBaseManager;
```

是总的数据库接口类。

```
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 public class PapaDataBaseAccess;
```

记录了服务器的相关信息, 例如

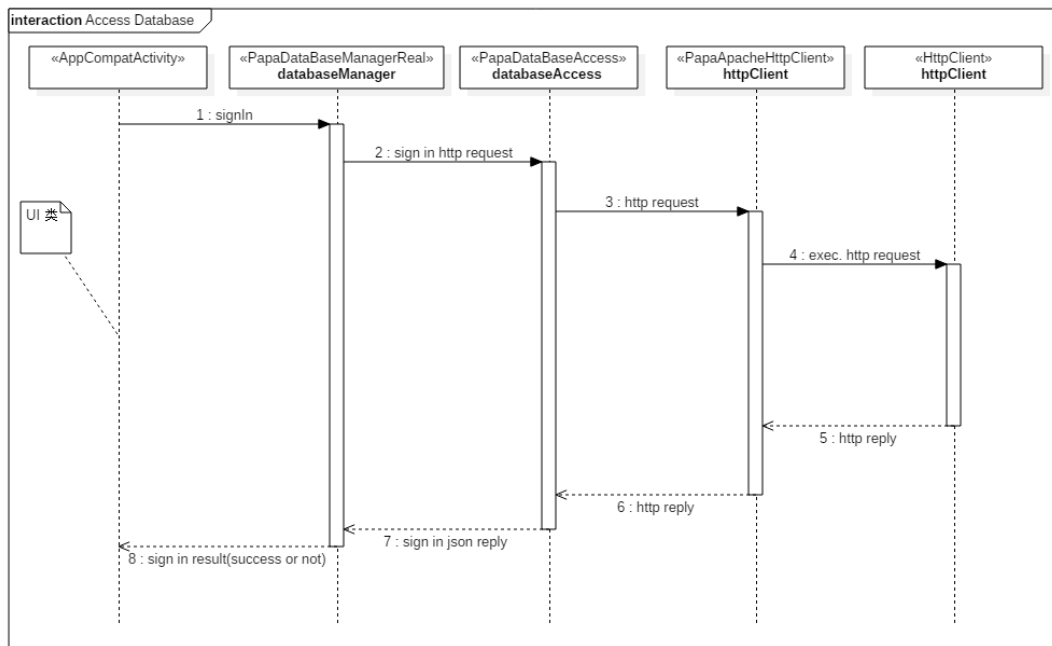
- IP 地址
- 端口号
- 等等

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



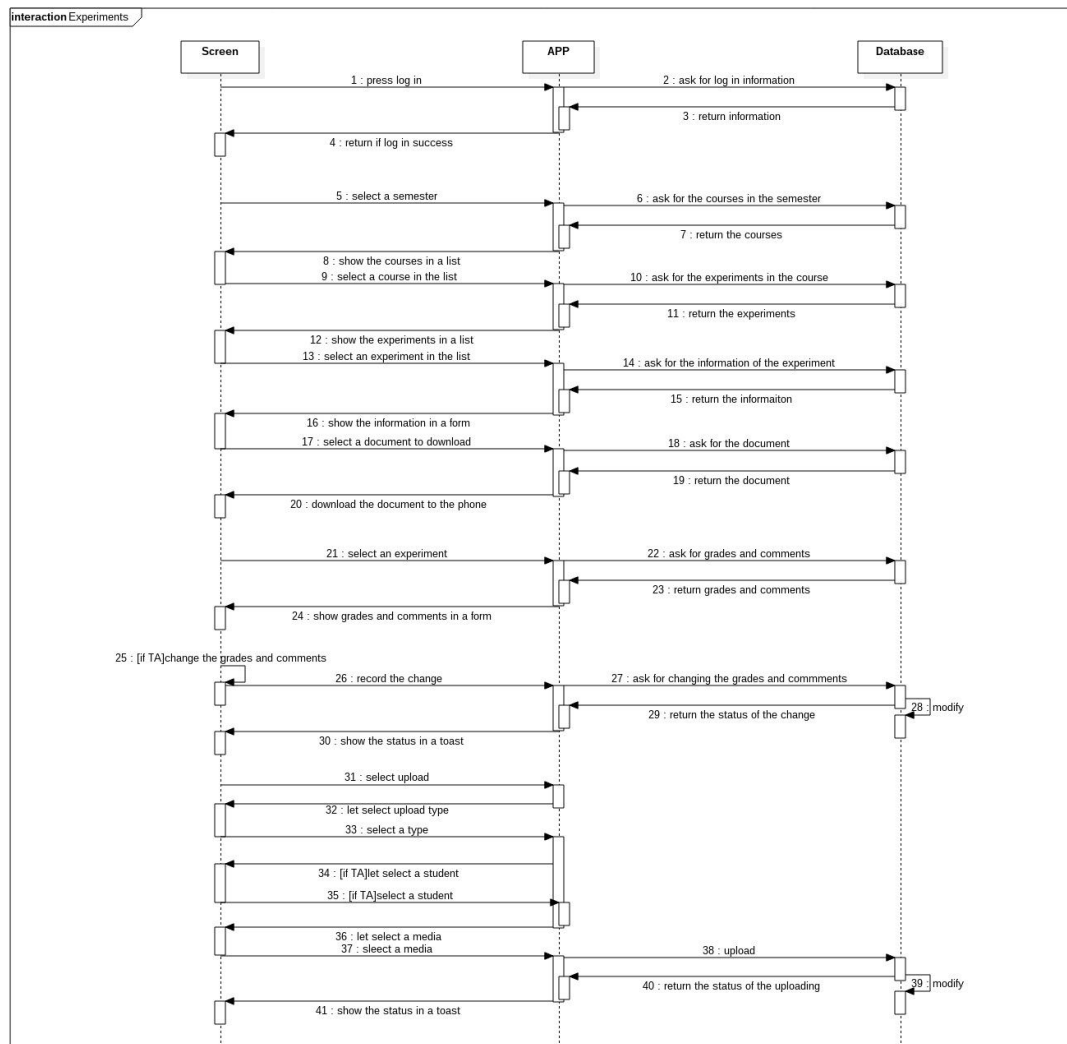


可以看到，用户的请求通过一层又一层的封装、嵌套，从最初的纯文本/文件信息，最后打包成数据包在网上传输。服务器端收到后，会根据协议解包，执行用户请求。

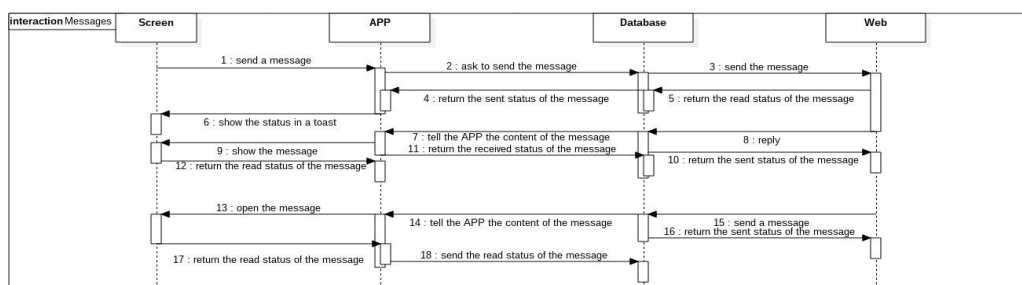
上图只是以登录为例举了个例子。得益于我们对数据库接口的定义，对于其他的用户操作，其序列图相同。

#### 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。

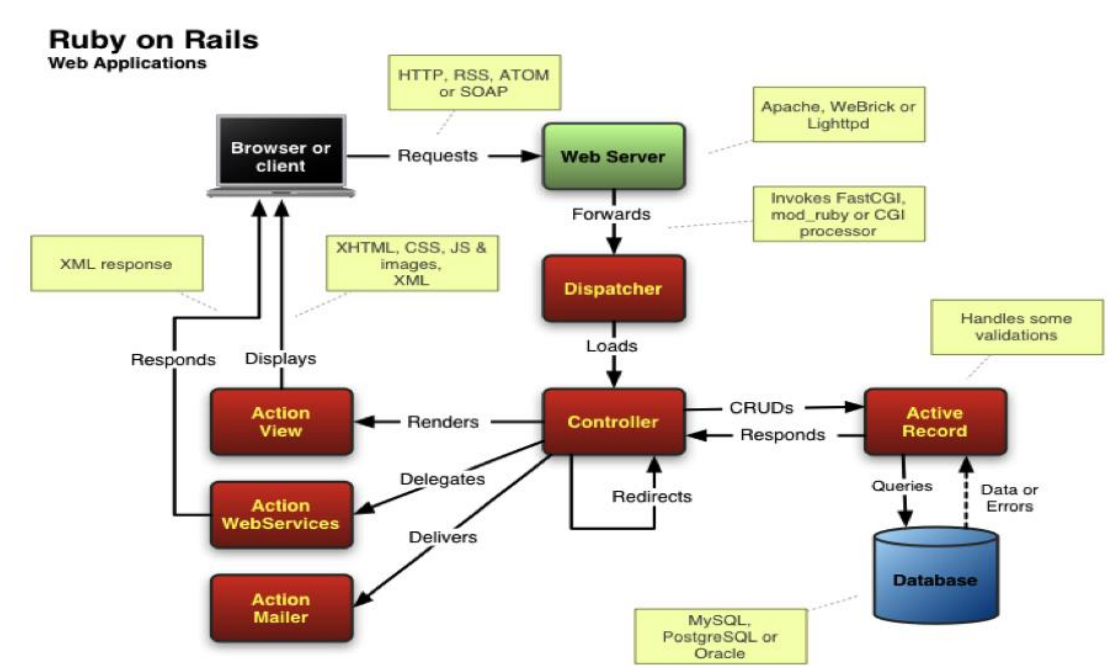
#### 4.3.2.5.6 本地化

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

#### 4.3.3.1 总体结构

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。



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

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

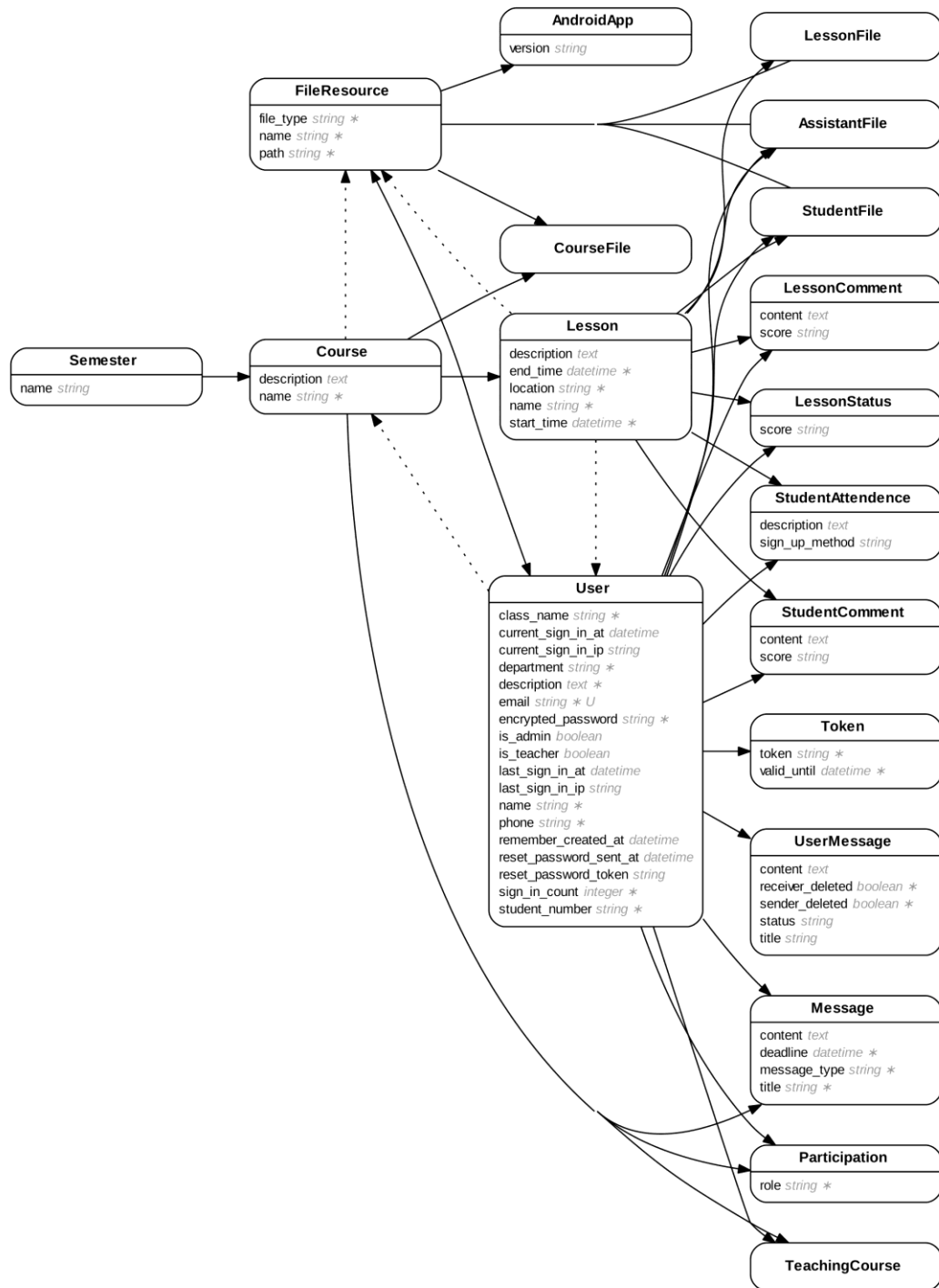
始文档。

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

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。

Papadb domain model



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** 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**

1. **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
2. **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

3. 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。

#### 4.3.3.4 类设计

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

1. Controllers（位于 `app/controllers`）

a. `SemestersController` 响应和学期有关的页面，负责处理 Semester model  
b. `CoursesController` 课程（每学期的课程）  
c. `LessonsController` 实验课（对应每一节课）  
e. `LessonCommentsController` 学生对课程的评价  
f. `StudentCommentsController` 助教/老师对学生的评价  
g. `MessagesController` 老师向同学们推送消息  
h. `UserMessagesController` 同学和老师之间互相发送消息，包括请假，批假  
i. `TeachersController` 老师  
j. `AssistantsController` 助教  
k. `StudentsController` 学生  
l. `User` 登录用户  
m. `AttendanceController` 管理学生上课的信息

2. Models（位于 `app/models`）

每个 Model 对应数据库的一张表。

3. Views（位于 `app/views`）

由于该应用程序的主要功能是提供 JSON API，所以 View 主要功能是根据 Controller 的处理结果，构造出一个 JSON 文件。

4. Routes（位于 `config/routes.rb`）

管理 URL 和 Controller 方法之间的对应关系。

5. 杂项

所有常量均位于 `config/initializers/constants.rb`

## 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		
<b>助教相关</b>				
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		

		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		
<b>课程相关</b>				
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		
<b>对学生的评价相关</b>				
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	
<b>文件相关</b>				
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		
<b>学生相关</b>				
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		
<b>课程/实验课相关</b>				
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

- A. 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。
- B. 不能根据 Android 系统版本进行不同的适配，兼容性不是很好。
- C. 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。