**湖 南 科 技 大 学**

**毕 业 设 计（ 论 文 ）**

|  |  |
| --- | --- |
| **题目** | **基于JAVA的点名签到系统** |
| **作者** | **蔡鸿铭** |
| **学院** | **数学与计算科学学院** |
| **专业** | **信息与计算科学** |
| **学号** | **1407020401** |
| **指导教师** | **刘灯明** |

二〇一八 年 五 月 二十二 日

**湖 南 科 技 大 学**

**毕业设计（论文）任务书**

数学与计算科学 学院 信息与计算科学 系

系（教研室）主任:（签名） 年 月 日

**学生姓名: 蔡鸿铭 学号: 1407020401专业: 信息与计算科学**

1 设计（论文）题目及专题： 基于JAVA的点名签到系统

2 学生设计（论文）时间：自 2018年 1月 5日开始至 2018年 5 月 29日止

3 设计（论文）所用资源和参考资料：

[1] 计文柯.Spring技术内幕[M].北京:机械工业出版社,2011.

[2] 赵正文.现代数据库技术[M].成都:电子科技大学出版社,2013.

[3] 李珍.软件工程专业毕业设计(论文)指导[M].北京:清华大学出版社,2010.

[4] 文斌.软件工程与软件文档写作[M].北京:清华大学出版社,2005.

[5] 张海藩.软件工程导论[M].北京:清华大学出版社,2013.

[6] 刘邵华.J2EE设计模式[M].北京:中国电力出版社,2005.

[7] 王珊.数据库系统概论[M].北京:高等教育出版社,2014.

[8] 汪云飞.JavaEE的颠覆者 Spring Boot实战[M].北京:电子工业出版社,2016.

4 设计（论文）应完成的主要内容：

本系统将由学生用户模块、教师用户模块、管理员用户模块与定时器任务模块4个模块组成。使用权限控制为学生用户、教师用户和管理员用户提供了不同的信息的查看、录入和修改的功能。

5 提交设计（论文）形式（设计说明与图纸或论文等）及要求：

按照《湖南科技大学本科生毕业设计（论文）规范与撰写要求》的规定完成毕业设计（论文），图表搭配合理，格式符合规范，注重创新。按要求提交毕业设计（论文）打印稿1份，电子版及所有程序附件需要提交实验室存档。

6 发题时间： 2018 年 1 月 4 日

指导教师： （签名）

学 生： （签名）

**湖 南 科 技 大 学**

**毕业设计（论文）指导人评语**

[主要对学生毕业设计（论文）的工作态度，研究内容与方法，工作量，文献应用，创新性，实用性，科学性，文本（图纸）规范程度，存在的不足等进行综合评价]

**指导人：** （签名）

年 月 日

**指导人评定成绩：**

**湖 南 科 技 大 学**

**毕业设计（论文）评阅人评语**

[主要对学生毕业设计（论文）的文本格式、图纸规范程度，工作量，研究内容与方法，实用性与科学性，结论和存在的不足等进行综合评价]

**评阅人： （签名）**

**年 月 日**

**评阅人评定成绩：**

**湖 南 科 技 大 学**

**毕业设计（论文）答辩记录**

**日期： 2018年6月2日**

**学生： 蔡鸿铭 学号： 1407020401 班级： 信息与计算科学四班**

**题目： 基于JAVA的点名签到系统**

**提交毕业设计（论文）答辩委员会下列材料：**

**1 设计（论文）说明书 共 43 页**

**2 设计（论文）图 纸 共 0 页**

**3 指导人、评阅人评语 共 2 页**

**毕业设计（论文）答辩委员会评语：**

[主要对学生毕业设计（论文）的研究思路，设计（论文）质量，文本图纸规范程度和对设计（论文）的介绍，回答问题情况等进行综合评价]

**答辩委员会主任：** （签名）

**委员：** （签名）

（签名）

（签名）

（签名）

**答辩成绩：**

**总评成绩：**

**摘 要**

本文基于JAVA语言，使用IntelliJ IDEA 2017作为开发工具，设计了点名签到系统。本系统具有课堂签到，记录学生签到记录，统计学生签到记录和添加学生人脸信息等功能。本系统采用人脸识别技术，相比于传统的点名方式如教师点名签到方式，利用人脸识别解决了学生代签的问题，既方便又快捷，也提高了课堂质量，能够准确反映学生出勤情况。但该系统在设计时需要对接教务系统，所以在设计时简单模拟了一下教务系统的数据库，所以系统在发布到实际生产环境时，需要对数据库进行一些调整。

关键词**：**JAVA；IntelliJ IDEA 2017；点名签到系统

**ABSTRACT**

This paper,based on the JAVA language, uses IntelliJ IDEA 2017 as a development tool to design a roll-call check-in system. This system has classroom check-in record and student check-in record, which statistics student check-in record and adds student face information and other functions. Compared with the traditional methods of roll call, such as student attendance check, this system uses facial recognition technology, to solve the problems of the cheat in student attendance check. It improves the quality of the classroom and can accurately reflect students attendance, which is convenient and efficient. But the system need to dock the educational administration system in design, so that we need a simple simulation of the educational administration system of database in design, and the system in the release to the actual production environment, needs some adjustment on the database.

**Keywords：**JAVA; IntelliJ IDEA 2017; Roll-call check-in system

目 录

**[第一章 绪论](#_Toc5178_WPSOffice_Level1)**[1](#_Toc5178_WPSOffice_Level1)

[1.1 研究背景](#_Toc25634_WPSOffice_Level2) [1](#_Toc25634_WPSOffice_Level2)

[1.2 研究内容](#_Toc4894_WPSOffice_Level2) [1](#_Toc4894_WPSOffice_Level2)

[1.3 运行环境](#_Toc14005_WPSOffice_Level2) [1](#_Toc14005_WPSOffice_Level2)

[1.4 系统相关技术介绍](#_Toc9793_WPSOffice_Level2) [1](#_Toc9793_WPSOffice_Level2)

[1.4.1 Spring Boot简介](#_Toc25634_WPSOffice_Level3) [2](#_Toc25634_WPSOffice_Level3)

[1.4.2 Spring MVC简介](#_Toc4894_WPSOffice_Level3) [2](#_Toc4894_WPSOffice_Level3)

[1.4.3 前后端分离模式简介](#_Toc14005_WPSOffice_Level3) [2](#_Toc14005_WPSOffice_Level3)

**[第二章 系统分析](#_Toc25634_WPSOffice_Level1)**[3](#_Toc25634_WPSOffice_Level1)

[2.1 需求分析](#_Toc17624_WPSOffice_Level2) [3](#_Toc17624_WPSOffice_Level2)

[2.1.1 功能需求](#_Toc9793_WPSOffice_Level3) [3](#_Toc9793_WPSOffice_Level3)

[2.1.2 性能需求](#_Toc17624_WPSOffice_Level3) [3](#_Toc17624_WPSOffice_Level3)

[2.1.3 软件流程图](#_Toc27856_WPSOffice_Level3) [4](#_Toc27856_WPSOffice_Level3)

[2.2 可行性分析](#_Toc27856_WPSOffice_Level2) [4](#_Toc27856_WPSOffice_Level2)

[2.2.1 技术可行性 4](#_Toc26400_WPSOffice_Level3)

[2.2.2 经济可行性](#_Toc17587_WPSOffice_Level3) [5](#_Toc17587_WPSOffice_Level3)

[2.2.3 操作可行性](#_Toc21531_WPSOffice_Level3) [5](#_Toc21531_WPSOffice_Level3)

**[第三章 总体设计](#_Toc4894_WPSOffice_Level1)**[7](#_Toc4894_WPSOffice_Level1)

[3.1 功能模块划分](#_Toc18276_WPSOffice_Level2) [7](#_Toc18276_WPSOffice_Level2)

[3.1.1 系统功能结构图](#_Toc21037_WPSOffice_Level3) [7](#_Toc21037_WPSOffice_Level3)

[3.1.2 系统流程图](#_Toc21286_WPSOffice_Level3) [9](#_Toc21286_WPSOffice_Level3)

[3.2 数据库设计](#_Toc26400_WPSOffice_Level2) [13](#_Toc26400_WPSOffice_Level2)

[3.2.1 系统E-R图](#_Toc2826_WPSOffice_Level3) [13](#_Toc2826_WPSOffice_Level3)

[3.2.2 关系模型](#_Toc8654_WPSOffice_Level3) [14](#_Toc8654_WPSOffice_Level3)

[3.2.3 数据库表设计](#_Toc29428_WPSOffice_Level3) [14](#_Toc29428_WPSOffice_Level3)

**[第四章 详细设计](#_Toc14005_WPSOffice_Level1)**[19](#_Toc14005_WPSOffice_Level1)

[4.1 用户通用模块](#_Toc17587_WPSOffice_Level2) [19](#_Toc17587_WPSOffice_Level2)

[4.1.1 用户登录](#_Toc17104_WPSOffice_Level3) [19](#_Toc17104_WPSOffice_Level3)

[4.1.2 用户退出](#_Toc5150_WPSOffice_Level3) [20](#_Toc5150_WPSOffice_Level3)

[4.1.3 token验证](#_Toc17645_WPSOffice_Level3) [21](#_Toc17645_WPSOffice_Level3)

[4.2 学生模块](#_Toc21531_WPSOffice_Level2) [22](#_Toc21531_WPSOffice_Level2)

[4.2.1 查询签到记录](#_Toc22297_WPSOffice_Level3) [22](#_Toc22297_WPSOffice_Level3)

[4.2.2 统计签到记录](#_Toc9837_WPSOffice_Level3) [23](#_Toc9837_WPSOffice_Level3)

[4.2.3 课堂签到](#_Toc19614_WPSOffice_Level3) [24](#_Toc19614_WPSOffice_Level3)

[4.3 教师模块](#_Toc21037_WPSOffice_Level2) [26](#_Toc21037_WPSOffice_Level2)

[4.3.1 查询学生签到记录](#_Toc19770_WPSOffice_Level3) [26](#_Toc19770_WPSOffice_Level3)

[4.3.2 统计学生签到记录](#_Toc6928_WPSOffice_Level3) [28](#_Toc6928_WPSOffice_Level3)

[4.4 管理员模块](#_Toc9477_WPSOffice_Level2) [29](#_Toc9477_WPSOffice_Level2)

[4.4.1 添加学生人脸信息](#_Toc25382_WPSOffice_Level3) [29](#_Toc25382_WPSOffice_Level3)

[4.5 定时任务模块](#_Toc3973_WPSOffice_Level2) [32](#_Toc3973_WPSOffice_Level2)

[4.5.1 定时训练学生人脸模型](#_Toc25345_WPSOffice_Level3) [32](#_Toc25345_WPSOffice_Level3)

**[第五章 系统测试](#_Toc9793_WPSOffice_Level1)**[35](#_Toc9793_WPSOffice_Level1)

[5.1 测试的目标](#_Toc2137_WPSOffice_Level2) [35](#_Toc2137_WPSOffice_Level2)

[5.2 测试方法](#_Toc4244_WPSOffice_Level2) [35](#_Toc4244_WPSOffice_Level2)

[5.3 测试内容](#_Toc3248_WPSOffice_Level2) [35](#_Toc3248_WPSOffice_Level2)

[5.3.1 登录拦截测试](#_Toc1439_WPSOffice_Level3) [35](#_Toc1439_WPSOffice_Level3)

[5.3.2 跳转拦截测试](#_Toc1433_WPSOffice_Level3) [36](#_Toc1433_WPSOffice_Level3)

[5.3.3 学生模块测试](#_Toc31614_WPSOffice_Level3) [36](#_Toc31614_WPSOffice_Level3)

**[第六章 总结](#_Toc17624_WPSOffice_Level1)**[39](#_Toc17624_WPSOffice_Level1)

**[参考文献](#_Toc27856_WPSOffice_Level1)**[41](#_Toc27856_WPSOffice_Level1)

**[致谢](#_Toc18276_WPSOffice_Level1)**[43](#_Toc18276_WPSOffice_Level1)

**[附录](#_Toc26400_WPSOffice_Level1)**[45](#_Toc26400_WPSOffice_Level1)

1. **绪 论**

## 1.1 研究背景

课堂到课率是教学质量的基本保证，而传统的点名签到方式效率不高浪费教学时间而且容易让学生蒙混过去。随着现代信息技术不断发展，新的技术不断涌现，传统的点名签到方式已经落后。为减少学生迟到、旷课行为，提高到课率，本文设计了使用人脸识别技术进行签到的点名签到系统。相比于传统的点名方式如教师点名签到方式，利用人脸识别解决了学生代签的问题，既方便又快捷。

## 1.2 研究内容

本论文研究的是基于JAVA的点名签到系统，即一种代替传统点名方式转变为人脸识别方式的系统。本论文用于阐述点名签到系统的设计、实现一并验证、测试的一个过程。本点名签到系统的开发工具是IntelliJ IDEA 2017，MySQL为本系统的数据库，并且运用了B/S与C/S结构。本论文介绍系统的设计与搭建过程，及数据库的设计分析。通过用户应用场景代入，进行需求分析与需求评审，以达到需求分析的合理性与可行性。本系统的内容对于教师与学生来说都是非常重要的，所以点名签到系统应该能够为学生与教师提供可靠的签到方式与签到记录，以便教师以此为参考对学生的考核给出一个合理的成绩。

## 1.3 运行环境

开发环境如下：

开发环境： Windows8 Tomcat8.0 JDK1.8 maven3.5.2；

硬件实现： Inter-Core i5-5200U 8G内存 400G硬盘；

开发语言: JAVA；

数 据 库: MySQL；

开发工具: IntelliJ IDEA 2017。

## 1.4 系统相关技术介绍

点名签到系统是一种不同于传统点名方式的签到系统。该系统主要作用是可以让老师对学生平时签到进行统计，并以此为参考对学生的考核给出一个合理的成绩。一个软件系统是否可以开发，首先是要分析当前的技术是否能达到开发的要求，因此下面简单介绍一下开发本系统所涉及的相关技术。

### 1.4.1 Spring Boot简介

在当今JavaEE开发中，Spring框架是当之无愧的王者。而Spring Boot是Spring主推的基于“习惯优于配置”的原则，让我们能够快速搭建应用框架，从而使得JavaEE开发变得异常简单。

Spring Boot是为了简化Spring开发而生，从Spring 3.x开始，Spring社区的发展方向就是弱化xml配置文件而加大注解的戏份。最近召开的SpringOne2GX2015大会上显示：Spring Boot已经是Spring社区中增长最迅速的框架，前三名是：Spring Framework，Spring Boot和Spring Security，这个应该是未来的趋势。

### 1.4.2 Spring MVC简介

在了解Spring MVC之前，要先了解一下什么是MVC设计模式。MVC：Model+View+Controller，Model是模型；View是视图页面；控制器是Controller。

Spring MVC是Spring框架对MVC模式的实现，它可以解决状态管理、工作流以及验证等重要特性。它能够构建像Spring框架那样灵活和松耦合的Web应用程序。

在Spring MVC中，有一个专门的类叫Model，用来和View之间的之间的数据交互、传值；View的类型包含JSP、freeMarker、Velocity、Thymeleaf、Tile等；Controller是Spring MVC的注解@Controller的类。

### 1.4.3 前后端分离模式简介

在介绍前后端分离开发模式之前，我们先介绍一下传统开发的分离模式，一般传统上的开发协作模式有两种：

1. 前端先编写静态的HTML模版页面，开发完成后，交由后端去使用。这样做的好处是前端可以不用考虑业务逻辑，专心实现视图。坏处是后端使用模版时需要浏览前端的代码，免得出错，在时间上加大了后端的压力。
2. 前后直接将模版模版写好，这样做的好处是后端可以不用参与前端的代码。坏处是前端依赖后端接口的实现，在后端没有完成代码编写的情况下，前端的进度将被拖延。

这两种方式是业界使用最广开发模式，这两种开发方式的前后端分离程度都不是很高，为此，本次系统设计的开发模式将使用前后端分离模式开发。

本系统使用的前后端分离开发模式：

1. 后端编写数据请求接口文档，接口文档中包括请求地址、请求方式、请求案例与返回结果案例。
2. 前端根据接口编写一个伪请求地址，伪返回结果，以此编写前端界面代码。
3. 前后端联调，前端将请求地址修改为后端指定的请求地址，并进行接口调试，直至后端返回无误，前端展示无误。
4. **系统分析**

## 2.1 需求分析

本系统的使用对象主要为学生和教师，我在拥有学生身份的同时将自己代入系统应用场景。并结合平时上课实际情况对系统做出需求分析。

### 2.1.1 功能需求

**（1）学生用户部分** 学生用户应当实现用户登录、用户退出、查询签到记录、统计签到记录和课堂签到等功能。

**（2）教师用户部分** 教师用户应当实现用户登录、用户退出、查询学生签到记录、统计学生签到记录、学生请假状态记录、和学生补签等功能。

**（3）管理员部分** 管理员用户应当实现用户登录、用户退出和添加学生人脸信息等功能。

1. **定时任务训练人脸识别模型** 定时任务模块应当实现定时训练学生人脸模型功能。

### 2.1.2 性能需求

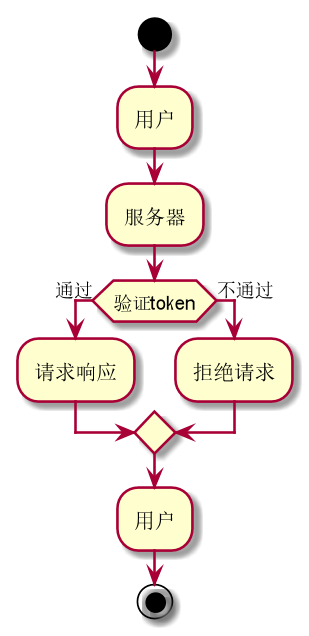
点名签到系统要满足用户需求，并且长期稳定运行，使用高效，操作方便，及时准确，那么就需要满足以下性能需求：

**（1）及时性和可靠性** 系统的响应速度应该在用户接受的范围内，但考虑签到的时候会出现多间教室同时签到的高并发情况，所以本系统推荐主机配置应该采用分布式主机。而且采用分布式主机部署的好处：如果一个机器崩溃,整个系统还可以运转，系统计算能力可以逐渐有所增加，微处理机提供了比大型主机更好的性能价格比。这样不仅提高了系统的响应速度，而且还提高了系统的稳定性、可靠性。

**（2）易用性** 为实现操作方便的性能需求，在界面设计时，应当力求简洁方便，设计出一个界面简洁美观而且操作方便能够让人机友好交互的界面。

**（3）安全性** 本系统采用不用与传统开发模式的前后端分离开发模式，所以在用户身份验证这一方面，采用了不用于传统的token验证方式。在用户登录的时候服务器将用户信息从数据库从取出，并且采用加密算法结合用户信息与随机数生成一个token。将此token作为name，用户信息作为value，存储于服务器本地的Redis数据库中，并且设置过期时间，然后将token返回给用户，用户每次发起请求的时候都会将token发送到服务器，服务器接收到token后会验证token的合法性，并且将该合法的token刷新过期时间，返回请求结果。

### 2.1.3 软件流程图



**图2.1 软件流程图**

## 2.2 可行性分析

可行性分析的目的，就是用最小的代价在尽可能短的时间内确定问题是否能够解决。

### 2.2.1 技术可行性

JAVA作为一门开源语言，具有许多第三方的开源框架（库），如Spring、Mybatis等，通过这些开源第三方开源框架（库）开源大大简化系统的开发难度。

近年来人工智能发展进入一个新的阶段，人工智能越来越得到重视。近年来，各类人工智能技术不断涌现，最具代表性的是阿尔法狗与人类顶级棋手对弈并将其战胜，这说明人工智能正在逐渐替换人类手工操作。

人脸识别作为人工智能发展的一个重要方向，在国内有许多公司已经有了成熟的解决方案，并且有了免费对外开放的SDK（如：百度，北京旷视），所以，我选择人脸识别作为本项目的签到方式。

### 2.2.2 经济可行性

点名签到系统与传统的点名方式相比，具有高效、快速、低成本、准确等优点。人工智能是今后科技发展的大趋势，随着时间推移与科技的发展人工智能会逐渐代替人力工作。

本点名签到系统使用百度对外提供的免费SDK，所以在开发成本上可以降低很多。

### 2.2.3 操作可行性

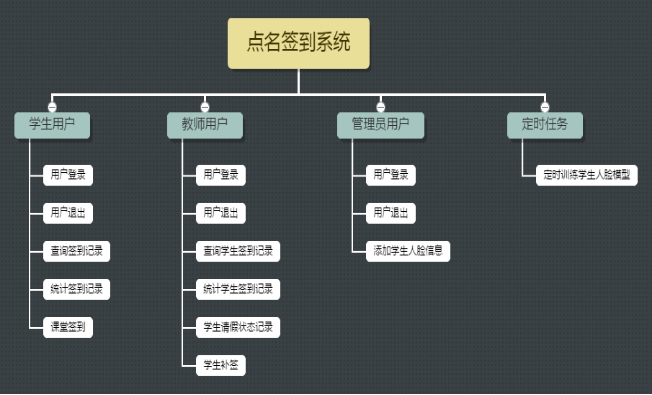
操作上是否可行主要是在了解使用本系统的用户的处理事务的原则和习惯之后，判断本系统的操作方式在这个用户组织中是否行的通。本系统采用使用计算机通过浏览器登录的方式进入系统，人机交互界面友好，如今计算机的普及程度也比较高，只要管理员和学生用户懂得基本的计算机操作就可以对本系统进行操作，所以，本系统在操作上可行。

# 第三章 总体设计

在经过上一个阶段的系统分析之后，已经基本确定了系统要实现的功能，在本章将决定这些功能如何实现。在本阶段将进行系统的功能模块划分和数据库设计。在功能模块划分部分将设计出系统功能结构图、各个模块中的功能实现大致思路以及各个模块的流程图。在数据库设计部分将设计本系统所用的系统E-R图、关系模型以及数据库表。

## **3.1** 功能模块划分

### 3.1.1 系统功能结构图



**图3.1 系统功能结构图**

**（1）学生用户模块** 用户登录，在学生输入学号和教务网密码并点击登录后，系统在后台取出相应学号对应的密码进行验证。验证通过后通过加密算法生成token，并将token作为name，用户信息作为value，将token和用户信息作为键值对存在服务器本地的Redis数据库中，然后将token返回给用户，并跳转相应的用户页面。否则，返回错误信息。

用户退出，用户将自己的token发送到后台，系统在后台取出token，并将该token在Redis中清除token与对应的用户信息，然后将用户界面跳转至登录界面。

查询签到记录，学生在完成登录进入用户界面后，可选择相应的课程（可不选，若没有则默认为全部课程），开始周数（可不选），结束周数（可不选），系统后台获取这些参数后，在数据库中查询出相应的记录，返回查询结果到前端界面，用户界面获取查询结果并展示。

统计签到记录，学生在完成登录进入用户界面后，使用查询签到记录功能时，会查询相应课程的统计记录，若不选择课程，则会默认统计全部课程的签到记录。

课堂签到，学生在教室内的签到摄像头前签到，摄像头在检测到有人脸信息的时候，会将画面截取下来，然后发送到服务器端。服务器端接收到图片后会调用百度提供的SDK，获取图片中的人脸信息是否为本次课堂中的某个学生，若是则根据当前时间将学生记录插入到签到记录表中。并且返回该名学生的信息到客户端，客户端得到签到学生的信息，将展示出来。否则，返回一个空的学生对象到客户端，表示没有当前签到的人不是本次课堂的学生。

**（2）教师用户模块** 用户登录，在教师输入工号和教务网密码并点击登录后，系统在后台取出相应工号对应的密码进行验证。验证通过后通过加密算法生成token，并将token作为name，用户信息作为value，将token和用户信息作为键值对存在服务器本地的Redis数据库中，然后将token返回给用户，并跳转相应的用户页面。否则，返回错误信息。

用户退出，用户将自己的token发送到后台，系统在后台取出token，并将该token在Redis中清除token与对应的用户信息，然后将用户界面跳转至登录界面。

查询学生签到记录，教师在完成登录进入用户界面后，可选择该教师所任课程列表，然后可以根据该课程列表查询所任课程的课堂记录，从而获取某一课堂记录中学生的签到记录。或者也可以查询某一课程的学生，获取该学生在本门课程的签到记录。系统后台获取到查询参数后，在数据库中查询出相应的记录，返回查询结果到前端界面，用户界面获取查询结果并展示。

统计学生签到记录，教师在完成登录进入用户界面后，使用查询学生签到记录功能时，会查询相应学生的统计记录。

**（3）管理员用户模块** 用户登录，在管理员输入账号和密码并点击登录后，系统在后台取出相应帐号对应的密码进行验证。验证通过后通过加密算法生成token，并将token作为name，用户信息作为value，将token和用户信息作为键值对存在服务器本地的Redis数据库中，然后将token返回给用户，并跳转相应的用户页面。否则，返回错误信息。

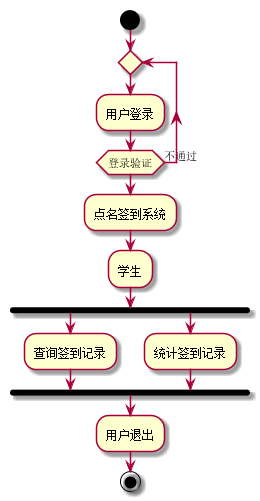
用户退出，用户将自己的token发送到后台，系统在后台取出token，并将该token在Redis中清除token与对应的用户信息，然后将用户界面跳转至登录界面。

添加学生人脸信息，管理员在完成登录进入用户界面后，可根据学院、专业、班级来筛选学生列表，然后选择某一个学生，然后在本地选择一张图片，将该学生学号和图片发送到后台，后台在获取图片后调用百度提供的SDK检测图片中是否有人脸。若存在人脸则将图片添加百度人脸库中，然后将该图片经过base64加密，存储在服务器本地数据库中。

**（4）定时任务模块** 每个学生的人脸特征会在不同的时间段有不同的改变，为避免因学生人脸特征变化过大而出现识别错误的情况。本系统采用的策略为：在每周，或者每个月的某个时间，将学生最新的十张人脸作为模型，替换到百度人脸库，以保证识别的准确率。

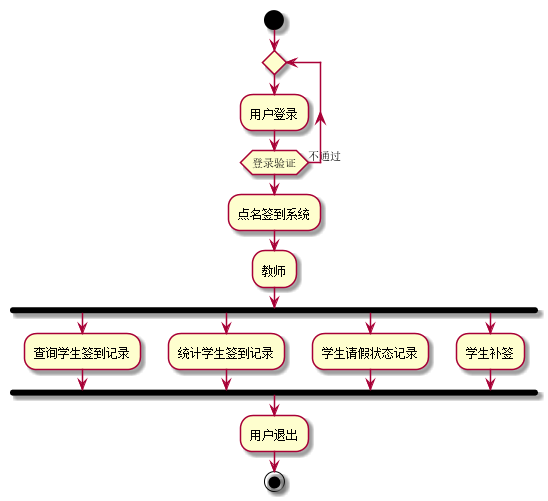
### 3.1.2 系统流程图

学生模块功能流程图如3.2所示。



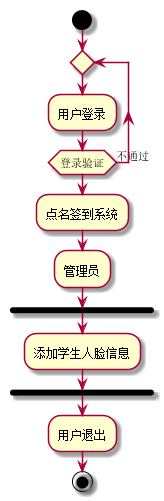
**图3.2 学生模块功能流程图**

教师模块功能流程图如3.3所示。

****

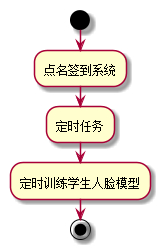
**图3.3 教师模块功能流程图**

管理员模块功能流程图如3.4所示。

****

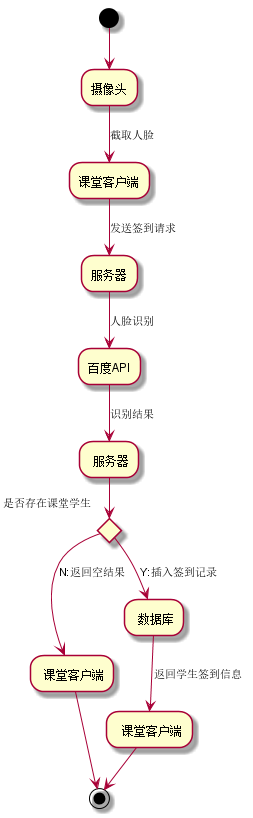
**图3.4 管理员模块功能流程图**

定时任务模块功能流程图如3.5所示。

****

**图3.5 定时任务模块功能流程图**

签到功能流程图如图3.6所示。

****

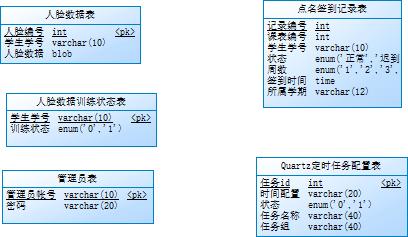
**图3.6 签到功能流程图**

## 3.2 数据库设计

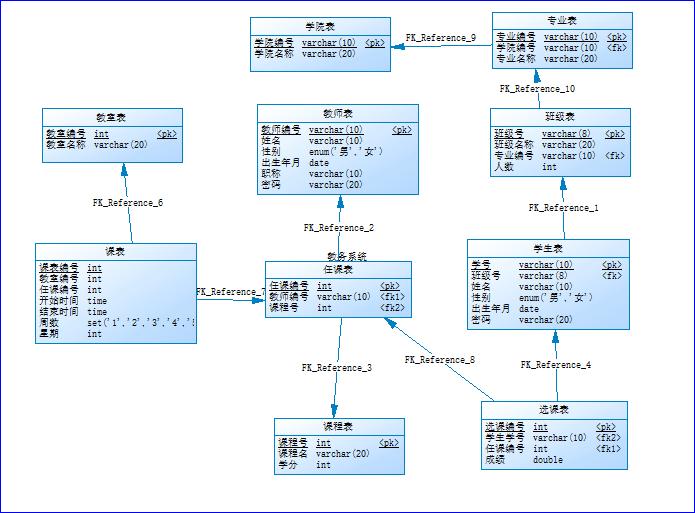
数据库本系统的信息储存源，储存着学生人脸信息，学生签到记录等信息。所以数据库的设计尤为重要，既要符合系统的高性能要求，又要有较高的范式，以便于维护升级和理解。

### 3.2.1 系统E-R图

通常，使用实体-联系图来建立数据模型。可以把实体联系图简称为E-R图。



**图3.7 点名签到系统E-R图**

****

**图3.8 教务系统E-R图**

### 3.2.2 关系模型

将上述E-R模型转换为下述关系模型（下划线表示主键）：

学院（学院编号，学院名称）

专业（专业编号，学院编号，专业名称）

班级（班级号，班级名称，专业编号，人数）

学生（学号，班级号，姓名，性别，出生年月，密码）

教师（教师编号，姓名，性别，出生年月，职称，密码）

课程（课程号，课程名，学分）

任课（任课编号，教师编号，课程号）

选课（选课编号，学生学号，任课编号，成绩）

教室（教室编号，教室名称）

课表（课表编号，教室编号，任课编号，开始时间，结束时间，周数，星期）

定时任务（任务id，时间配置，状态，任务名称，任务组）

管理员（管理员账号，密码）

人脸数据（人脸编号，学生编号，人脸数据）

人脸数据训练状态（学生学号，训练状态）

点名签到记录（记录编号，课表编号，学生学号，状态，周数，签到时间，所属学期）

### 3.2.3 数据库表设计

在MySQL中建立数据库rollcall，包含以下表：

### 表3.1 学院表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文字段名 | 英文字段名 | 字段类型 | 长度 | 主键/外键 | 字段值约束 |
| 学院编号 | AcaID | varchar | 10 | 主键 | NOT NULL |
| 学院名称 | AcaName | varchar | 20 | —— | NOT NULL |

### **表3.2** 专业**表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文字段名 | 英文字段名 | 字段类型 | 长度 | 主键/外键 | 字段值约束 |
| 专业编号 | ProID | varchar | 10 | 主键 | NOT NULL |
| 学院编号 | AcaID | varchar | 10 | —— | NOT NULL |
| 专业名称 | ProName | varchar | 20 | —— | NOT NULL |

### 

### **表3.3** 班级**表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文字段名 | 英文字段名 | 字段类型 | 长度 | 主键/外键 | 字段值约束 |
| 班级号 | ClassID | varchar | 8 | 主键 | NOT NULL |
| 班级名称 | ClassName | varchar | 20 | —— | NOT NULL |
| 专业编号 | ProID | varchar | 10 | —— | NOT NULL |
| 人数 | Count | int | —— | —— | —— |

### **表3.4** 学生**表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文字段名 | 英文字段名 | 字段类型 | 长度 | 主键/外键 | 字段值约束 |
| 学号 | StuID | varchar | 10 | 主键 | NOT NULL |
| 班级号 | ClassID | varchar | 8 | —— | NOT NULL |
| 姓名 | StuName | varchar | 10 | —— | NOT NULL |
| 性别 | StuSex | enum | —— | —— | NOT NULL |
| 出生年月 | Birthday | date | —— | —— | NOT NULL |
| 密码 | Password | varchar | 20 | —— | NOT NULL |

### **表3.5** 教师**表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文字段名 | 英文字段名 | 字段类型 | 长度 | 主键/外键 | 字段值约束 |
| 教师编号 | TeaID | varchar | 10 | 主键 | NOT NULL |
| 姓名 | TeaName | varchar | 10 | —— | NOT NULL |
| 性别 | TeaSex | enum | —— | —— | NOT NULL |
| 出生年月 | Birthday | date | —— | —— | NOT NULL |
| 职称 | Title | varchar | 10 | —— | NOT NULL |
| 密码 | Password | varchar | 20 | —— | NOT NULL |

### **表3.6** 课程**表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文字段名 | 英文字段名 | 字段类型 | 长度 | 主键/外键 | 字段值约束 |
| 课程号 | CouID | int | —— | 主键 | NOT NULL |
| 课程名 | CouName | varchar | 20 | —— | NOT NULL |
| 学分 | Credit | int | —— | —— | NOT NULL |

### 

### **表3.7** 任课**表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文字段名 | 英文字段名 | 字段类型 | 长度 | 主键/外键 | 字段值约束 |
| 任课编号 | TeachID | int | —— | 主键 | NOT NULL |
| 教师编号 | TeaID | varchar | 10 | —— | NOT NULL |
| 课程号 | CouID | int | —— | —— | NOT NULL |

### **表3.8** 选课**表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文字段名 | 英文字段名 | 字段类型 | 长度 | 主键/外键 | 字段值约束 |
| 选课编号 | SelectID | int | —— | 主键 | NOT NULL |
| 学生学号 | StuID | varchar | 10 | —— | NOT NULL |
| 任课编号 | TeachID | int | —— | —— | NOT NULL |
| 成绩 | Result | double | —— | —— | —— |

### **表3.9** 教室**表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文字段名 | 英文字段名 | 字段类型 | 长度 | 主键/外键 | 字段值约束 |
| 教室编号 | ID | int | —— | 主键 | NOT NULL |
| 教室名称 | Name | varchar | 20 | —— | NOT NULL |

### **表3.10** 课表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文字段名 | 英文字段名 | 字段类型 | 长度 | 主键/外键 | 字段值约束 |
| 课表编号 | SchID | int | —— | 主键 | NOT NULL |
| 教室编号 | RoomID | int | —— | —— | NOT NULL |
| 任课编号 | TeachID | int | —— | —— | NOT NULL |
| 开始时间 | StartTime | time | —— | —— | NOT NULL |
| 结束时间 | EndTime | time | —— | —— | NOT NULL |
| 周数 | WeekOfSemester | set | —— | —— | NOT NULL |
| 星期 | DayOfWeek | int | —— | —— | NOT NULL |

### **表3.11** 定时任务**表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文字段名 | 英文字段名 | 字段类型 | 长度 | 主键/外键 | 字段值约束 |
| 任务id | Id | int | —— | 主键 | NOT NULL |
| 时间配置 | Cron | varchar | 20 | —— | NOT NULL |
| 状态 | Status | enum | —— | —— | NOT NULL |
| 任务名称 | JobName | varchar | 40 | —— | NOT NULL |
| 任务组 | JobGroup | varchar | 40 | —— | NOT NULL |

### 

### **表3.12** 管理员**表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文字段名 | 英文字段名 | 字段类型 | 长度 | 主键/外键 | 字段值约束 |
| 管理员帐号 | Account | varchar | 10 | 主键 | NOT NULL |
| 密码 | Password | varchar | 20 | —— | NOT NULL |

### **表3.13 人脸数据表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文字段名 | 英文字段名 | 字段类型 | 长度 | 主键/外键 | 字段值约束 |
| 人脸编号 | FaceID | int | —— | 主键 | NOT NULL |
| 学生学号 | StuID | varchar | 10 | —— | NOT NULL |
| 人脸数据 | FaceDate | blob | —— | —— | NOT NULL |

### **表3.14 人脸数据训练状态表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文字段名 | 英文字段名 | 字段类型 | 长度 | 主键/外键 | 字段值约束 |
| 学生学号 | StuID | varchar | 10 | 主键 | NOT NULL |
| 训练状态 | Status | enum | —— | —— | NOT NULL |

### **表3.15 点名签到记录表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文字段名 | 英文字段名 | 字段类型 | 长度 | 主键/外键 | 字段值约束 |
| 记录编号 | RecID | int | —— | 主键 | NOT NULL |
| 课表编号 | SchID | int | —— | —— | NOT NULL |
| 学生学号 | StuID | varchar | 10 | —— | NOT NULL |
| 状态 | Status | enum | —— | —— | NOT NULL |
| 周数 | WeekOfSemester | time | —— | —— | NOT NULL |
| 签到时间 | SignedTime | time | —— | —— | NOT NULL |
| 所属学期 | Semester | varchar | 12 | —— | NOT NULL |

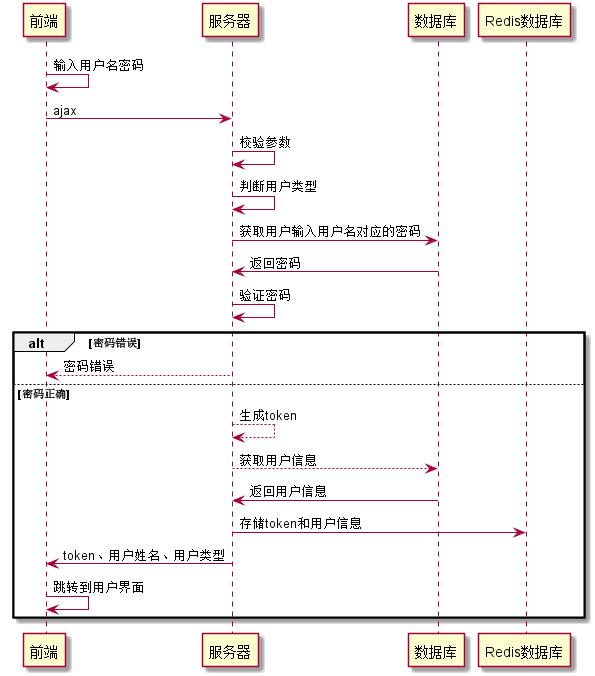
**第四章 详细设计**

在经过总体设计阶段，已经将本系统要实现的功能的大致思路设计出来了。而本阶段就要将这些思路具体化，通过本阶段的设计，要得出对本系统功能的精确描述，以便于在编码阶段可以直接精确编码。

## 4.1 用户通用模块

### 4.1.1 用户登录

用户进入用户登录界面，输入用户名和密码，点击登录。浏览器发起发起ajax异步请求，把用户名和密码组装成json格式，发送到服务器的登录Controller。Controller在接收请求后，校验参数是否齐全。验证登录身份与口令，若通过，则用过加密算法生成一个token，并从数据库中取出用户信息，将token作为name，用户信息作为value。将token和用户信息作为键值对存在服务器本地的Redis数据库中，并将生成的token，用户姓名，用户角色类型组装成json格式返回给用户。然后用户根据返回的用户角色类型跳转到相应的用户界面。



**图4.1 登录时序图**



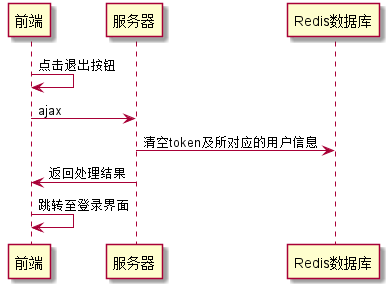
**图4.2 登录界面**



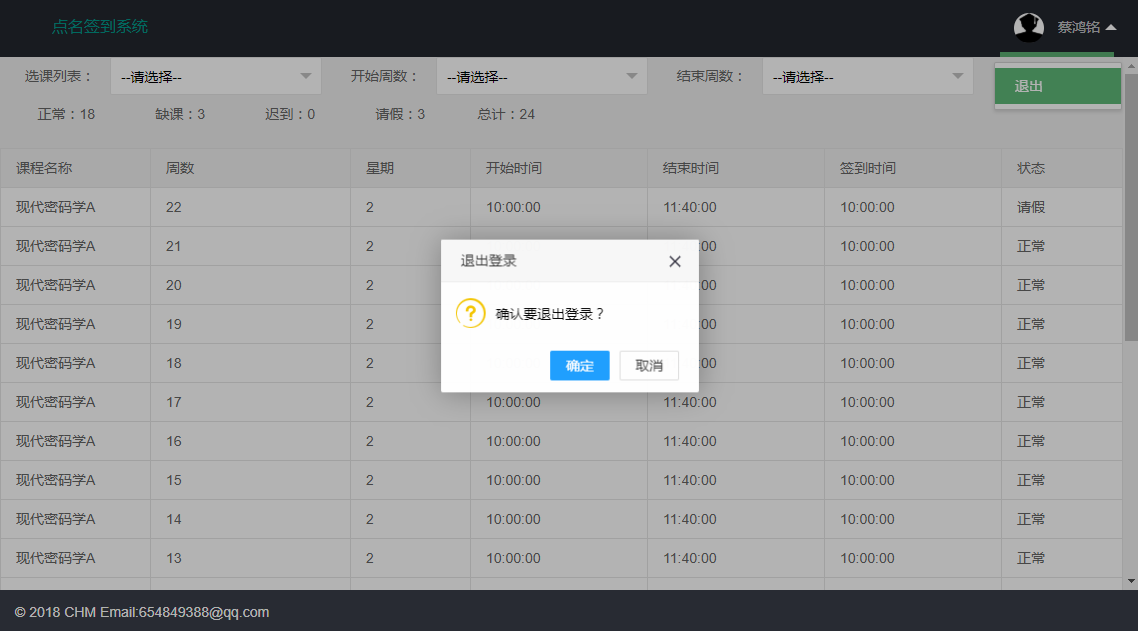
**图4.3 学生界面**

### 4.1.2 用户退出

用户在用户界面点击退出按钮，并点击确认按钮，浏览器发起发起ajax异步请求，把token发送到服务器的退出Controller。Controller在接收到请求后，将token及所对应的用户信息从服务器本地的Redis数据库中清除，然后前端界面跳转到登录界面。时序图如4.4所示，界面效果图如4.5所示。

****

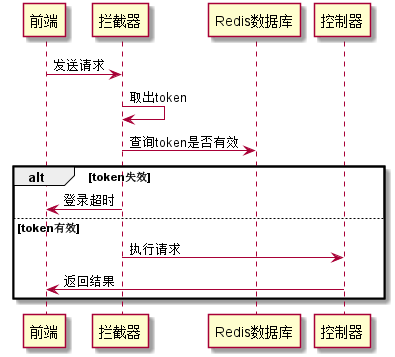
**图4.4 退出时序图**



**图4.5 退出登录确认界面**

### 4.1.3 token验证

在用户控制器中，我使用@Authorization自定义注解来注解提供服务的Controller，在前端请求这些被注解的Controller时，请求会被自定义拦截器拦截下来，并且验证用户的token是否有效。有效则通过并刷新token时效，无效则抛出登录失效异常。时序图如4.6所示。

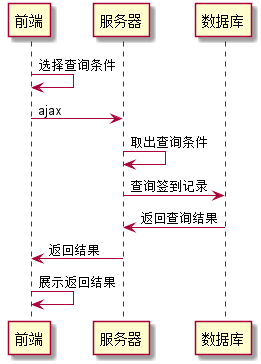
****

**图4.6 token验证时序图**

## 4.2 学生模块

### 4.2.1 查询签到记录

学生在完成登录进入用户界面后，可选择相应的课程（可不选，若没有则默认为全部课程），开始周数（可不选），结束周数（可不选），浏览器向服务器后端发起发起ajax异步请求，把选择参数发送到服务器后端，服务器后端获取这些参数后，在数据库中查询出相应的记录，返回查询结果到前端界面，用户界面获取查询结果并展示。时序图如4.7所示，界面效果图如4.8所示。



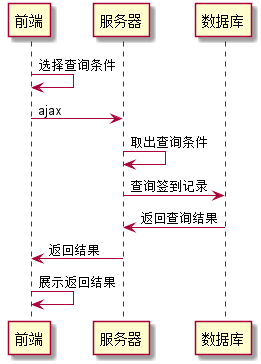
**图4.7 查询签到记录时序图**



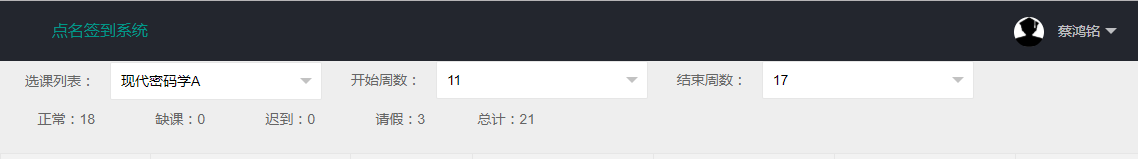
**图4.8 查询签到记录界面**

### 4.2.2 统计签到记录

学生在完成登录进入用户界面后，使用查询签到记录功能时，会查询相应课程的统计记录，若不选择课程，则会默认统计全部课程的签到记录。浏览器向服务器后端发起发起ajax异步请求，把选择参数发送到服务器后端，服务器后端获取这些参数后，在数据库中查询出相应的记录，返回查询结果到前端界面，用户界面获取查询结果并展示。时序图如4.9所示，界面效果图如4.10所示。



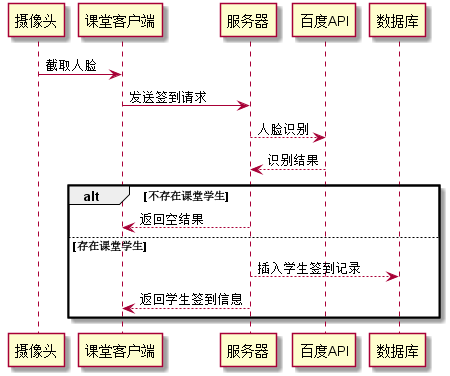
**图4.9 统计签到记录时序图**



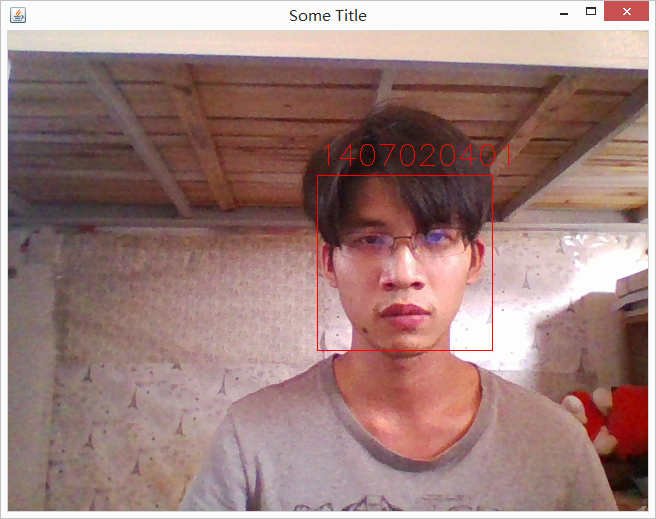
**图4.10 统计签到记录界面**

### 4.2.3 课堂签到

课堂签到，学生在教室内的签到摄像头前签到，摄像头在检测到有人脸信息的时候，会将画面截取下来，然后发送到服务器端。服务器端接收到图片后会调用百度提供的SDK，获取图片中的人脸信息是否为本次课堂中的某个学生，若是则根据当前时间将学生记录插入到签到记录表中。并且返回该名学生的信息到客户端，客户端得到签到学生的信息，将展示出来。否则，返回一个空的学生对象到客户端，表示没有当前签到的人不是本次课堂的学生。时序图如4.11所示，界面效果图如4.12所示。



**图4.11 课堂签到时序图**

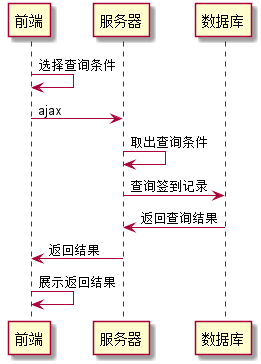


**图4.12 课堂签到效果图**

## 4.3 教师模块

### 4.3.1 查询学生签到记录

查询学生签到记录，教师在完成登录进入用户界面后，可选择该教师所任课程列表，然后可以根据该课程列表查询所任课程的课堂记录，从而获取某一课堂记录中学生的签到记录。或者也可以查询某一课程的学生，获取该学生在本门课程的签到记录。浏览器发起发起ajax异步请求，系统后台获取到查询参数后，在数据库中查询出相应的记录，返回查询结果到前端界面，用户界面获取查询结果并展示。时序图4.13所示，界面效果图如4.14至4.17所示。



**图4.13 查询学生签到记录时序图**



**图4.14 查询学生签到界面**



**图4.15 查询学生签到界面**



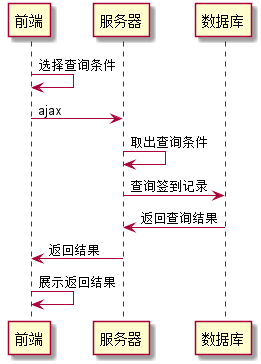
**图4.16 查询学生签到界面图**



**图4.17 查询学生签到界面图**

### 4.3.2 统计学生签到记录

教师在完成登录进入用户界面后，使用查询学生签到记录功能时，会查询相应课程的统计记录。浏览器向服务器后端发起发起ajax异步请求，把选择参数发送到服务器后端，服务器后端获取这些参数后，在数据库中查询出相应的记录，返回查询结果到前端界面，用户界面获取查询结果并展示。时序图如4.18所示，界面效果图如4.19所示。



**图4.18 统计学生签到记录时序图**

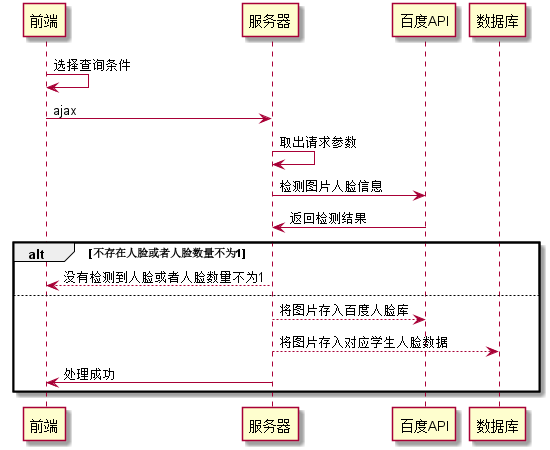


**图4.19 统计学生签到界面图**

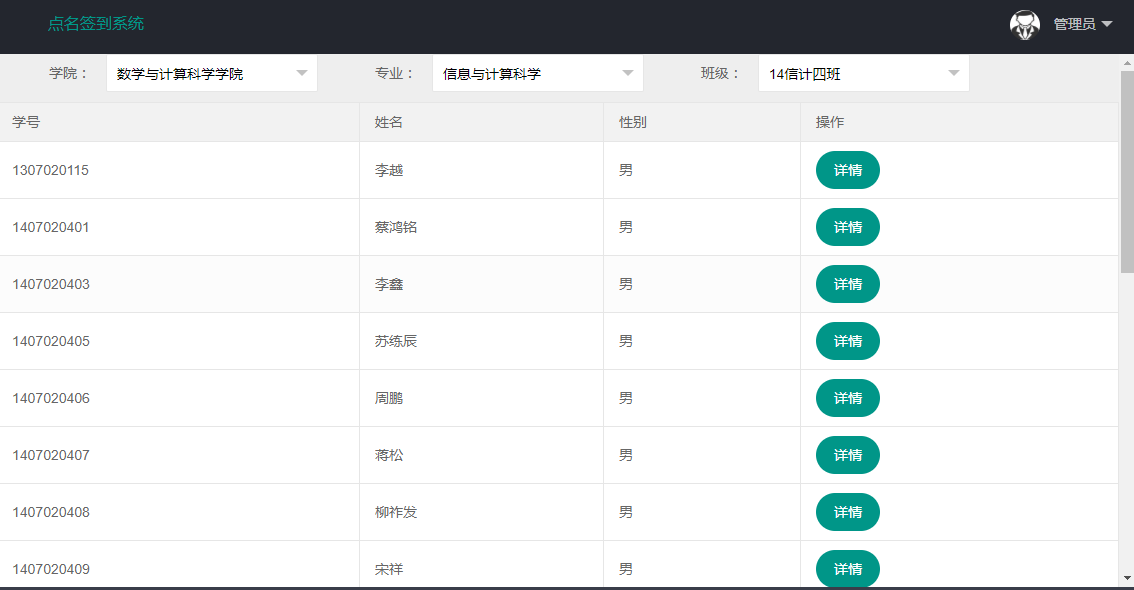
## 4.4 管理员模块

### 4.4.1 添加学生人脸信息

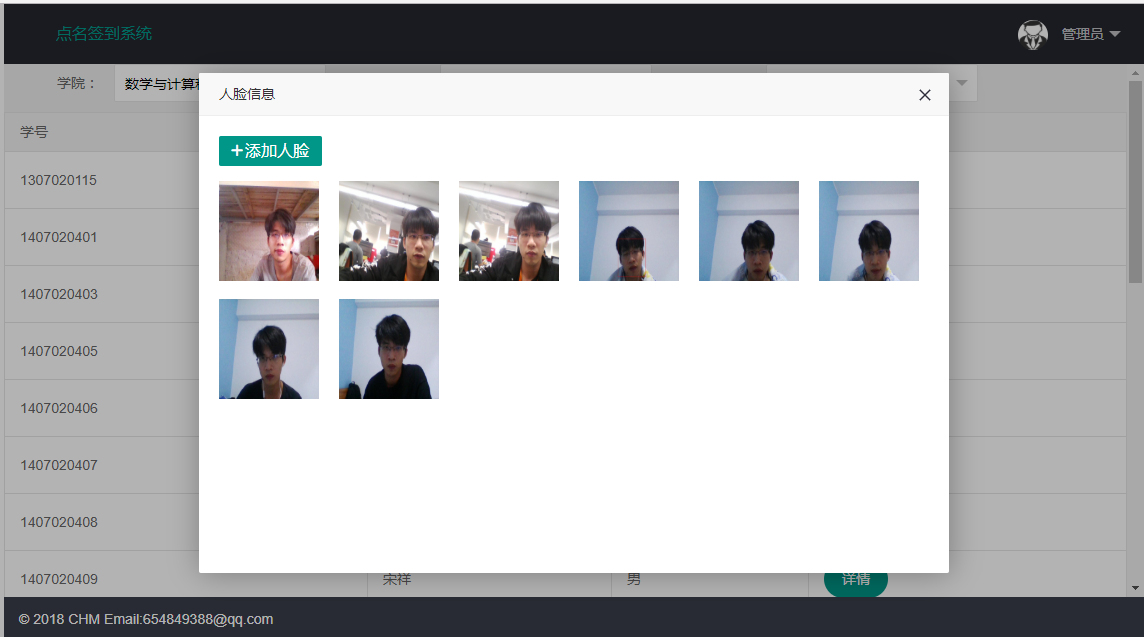
教师在完成登录进入用户界面后，可根据学院、专业、班级来筛选学生列表，然后选择某一个学生，然后在本地选择一张图片，浏览器发起发起ajax异步请求，将该学生学号和图片发送到后台，后台在获取图片后调用百度提供的SDK检测图片中是否有人脸。若存在人脸则将图片添加百度人脸库中，然后将该图片经过base64加密，存储在服务器本地数据库中。时序图如4.20所示，界面效果图如4.21至4.25所示。



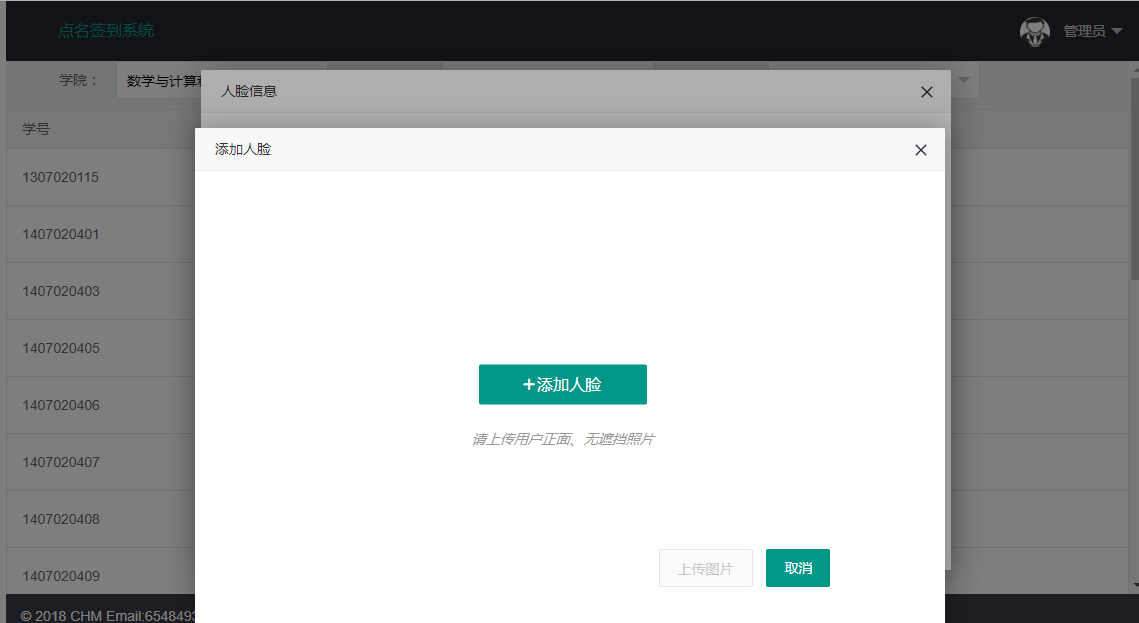
**图4.20 添加学生人脸信息时序图**



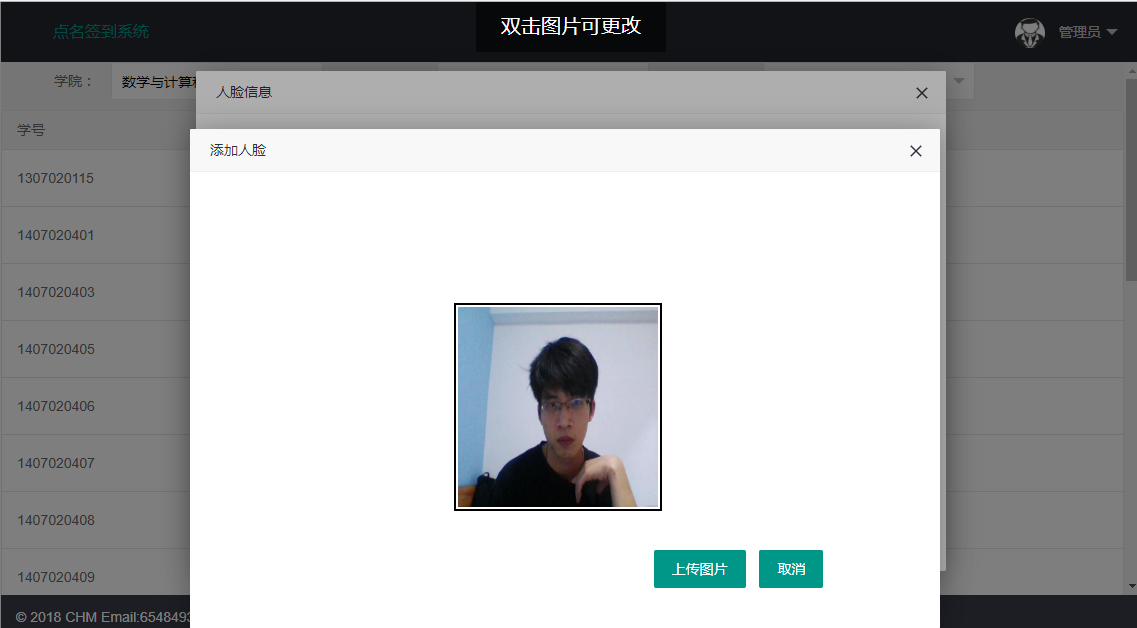
**图4.21 添加学生人脸信息界面**



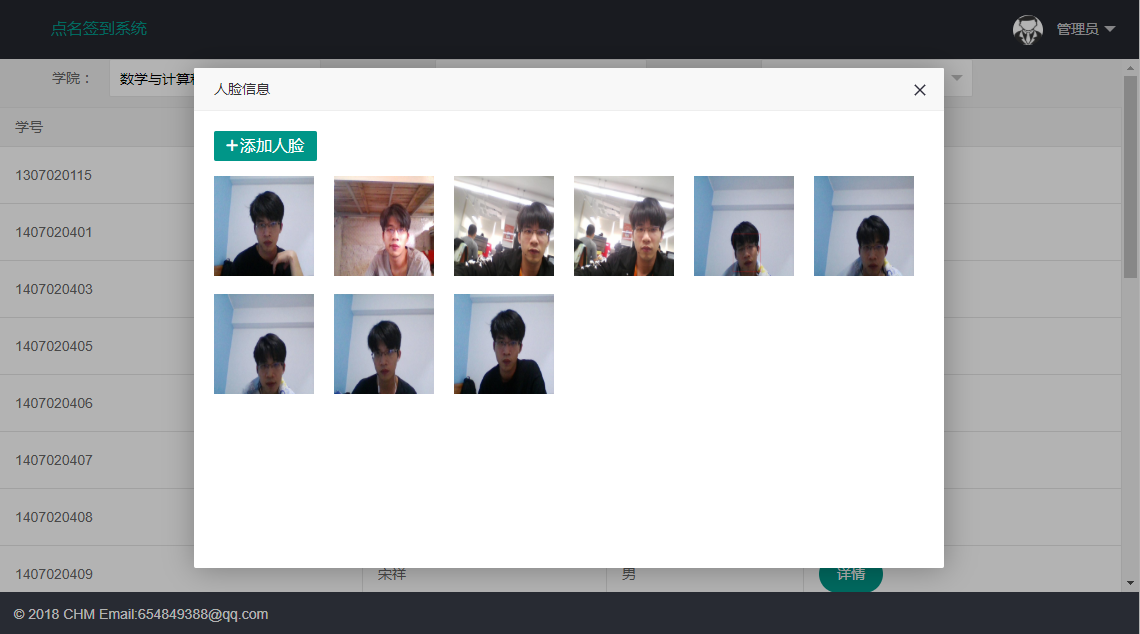
**图4.22 添加学生人脸信息界面**



**图4.23 添加学生人脸信息界面**



**图4.24 添加学生人脸信息界面**

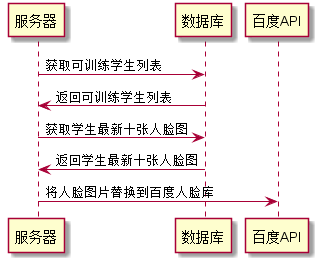


**图4.25 添加学生人脸信息界面**

## 4.5 定时任务模块

### 4.5.1 定时训练学生人脸模型

当定时任务启动时，查询数据库获取可训练学生列表。遍历列表，获取这些学生最新的十张人脸图片。使用百度API将人脸图片替换到百度人脸库。时序图如4.26所示。

****

**图4.26 定时训练学生人脸模型时序图**

# 第五章 系统测试

## 5.1 测试的目标

为了给系统的正常运行提供一个科学性的保证，将程序中存在的错误和漏洞在投入生产之前暴露出来并加以改正。

## 5.2 测试方法

软件测试方法是指测试软件的方法。主要的测试方法有两种：黑盒测试，白盒测试。在本次测试中，我将选择黑色测试为测试方法。

在本文中我将使用谷歌浏览器作为测试工具，操作系统为Windows 8.1的环境下进行功能测试。

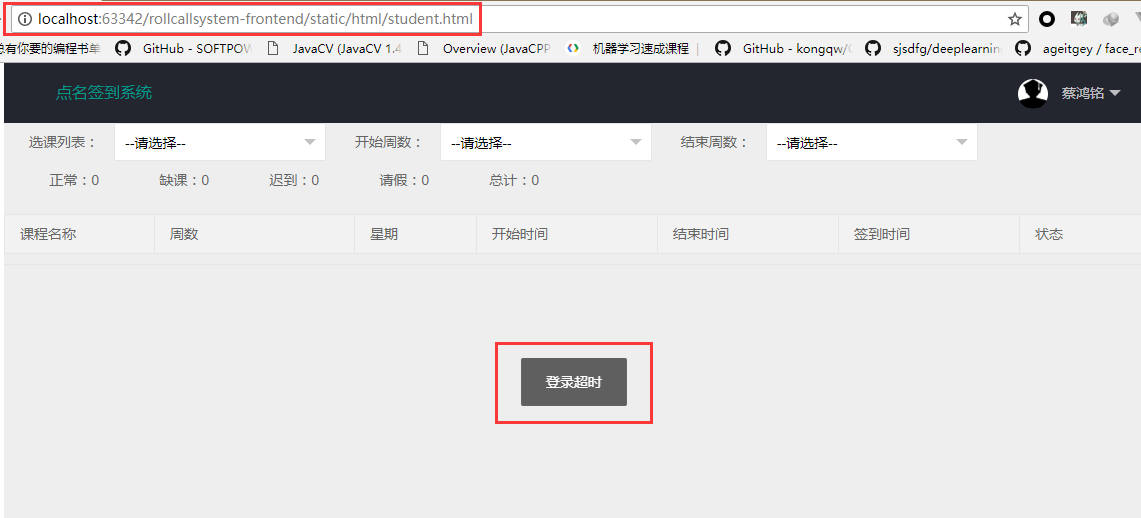
## 5.3 测试内容

本节将以系统部分功能测试过程为例，阐述软件测试方法。

### 5.3.1 登录拦截测试

测试用例：不输入账号密码，直接在浏览器输入学生用户界面的地址，查看系统有进行拦截。

测试结果：系统返回登录超时，并且将界面跳转至登录界面，如图5.1所示。



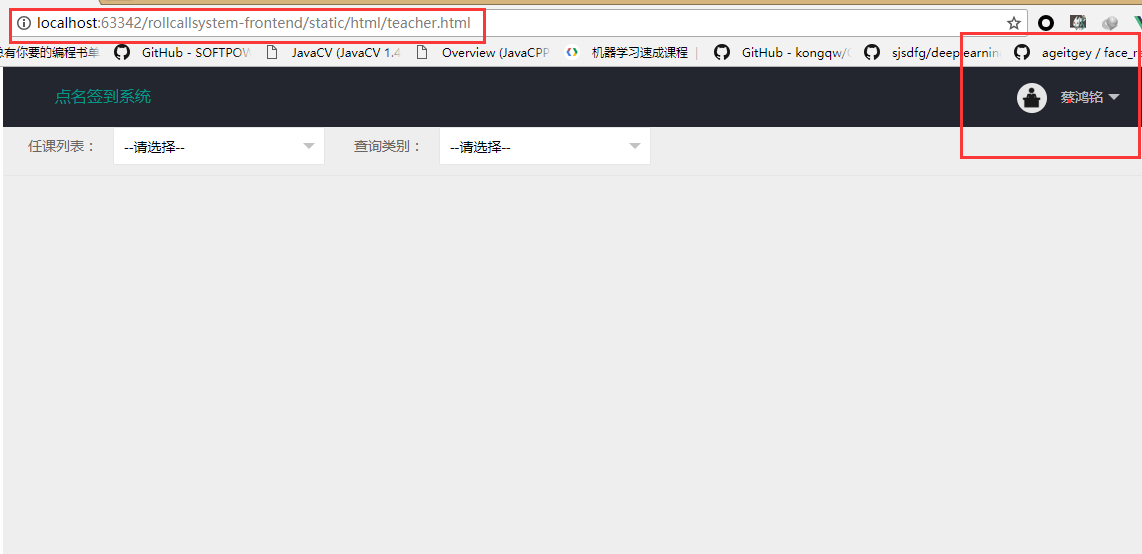
**图5.1 登录拦截测试结果**

### 5.3.2 跳转拦截测试

测试用例：在学生用户登录成功进入学生界面后，在浏览器的地址栏输入教师用户界面的地址，看系统是否阻止浏览器跳转到教师用户界面。

测试结果：浏览器成功跳转到教师用户界面，结果如图5.2所示。

改进方法：在浏览器对页面进行载入后对用户身份进行判断，若用户身份不符合当前页面所对应身份，则跳转到相应页面。



**图5.2 跳转拦截测试**

### 5.3.3 学生模块测试

测试目的：检测学生模块是否可以正常使用。

（1）使用学生身份登录系统，不做任何操作，看系统是否会返回全部签到记录。

测试结果：系统返回全部签到记录，结果如图5.3所示。



**图5.3 学生模块测试**

（2）使用学生身份登录系统，将选课列表选择为计算方法，看系统是否会返回计算方法课程的签到记录。

测试结果：系统返回计算方法课程的签到记录，结果如图5.4所示。



**图5.4 学生模块测试**

（3）使用学生身份登录系统，将选课列表选择为现代密码学A，开始周数为16，看系统是否会返回相应签到记录。

测试结果：系统返回相应签到记录，结果如图5.5所示。



**图5.5 学生模块测试**

（4）使用学生身份登录系统，将选课列表选择为现代密码学A，开始周数为16，结束周数为20，看系统是否会返回相应签到记录。

测试结果：系统返回全部签到记录，结果如图5.6。



**图5.6 学生模块测试**

# 总结

这次的毕业设计，是我大学学习阶段的一个总结以及检验。通过了几个月的时间的努力，我完成了我的毕业设计：基于JAVA的点名签到系统。我根据题目，按照软件工程介绍的方法进行研究，现在已经全部完成，并达到了预期的效果。

本项目使用的核心框架是时下最流行的微服务框架Spring Boot，Spring Boot集成了许多的第三方框架与组件，所以在使用MyBatis、Redis等这些框架与组件的时候，可以节省去许多框架间的整合所花费的时间。

在项目编码时，我严格按照各个类与各个接口的作用，严格分包，使项目结构变得清晰，各个模块之间的交互也是使用接口进行。在对功能模块进行维护升级时只需对接口实现做维护升级即可。

本项目还使用了与传统B/S模式不同的前后端开发模式，前后端的交互数据全部由json格式来进行。后端只需要向前端提供接口文档与请求实例，前端就可以自己进行开发，不用后端来参与前端的调试。从而实现真正意义上的前后端分离，便于维护。

本系统虽然可以成功运行，但是仍然有很多不足之处，比如教师的调课方式。今后还需结合学校的教务系统，进一步设计和完善改系统。

经过这次毕业设计，让我认识到学习基础知识的重要性，更让我体会到只有不断在实践中学习，才能真正掌握知识，才能学以致用。

# 参 考 文 献

[1] 计文柯.Spring技术内幕[M].北京:机械工业出版社,2011.

[2] 赵正文.现代数据库技术[M].成都:电子科技大学出版社,2013.

[3] 李珍.软件工程专业毕业设计(论文)指导[M].北京:清华大学出版社,2010.

[4] 文斌.软件工程与软件文档写作[M].北京:清华大学出版社,2005.

[5] 张海藩.软件工程导论[M].北京:清华大学出版社,2013.

[6] 刘邵华.J2EE设计模式[M].北京:中国电力出版社,2005.

[7] 王珊.数据库系统概论[M].北京:高等教育出版社,2014.

[8] 汪云飞.JavaEE的颠覆者 Spring Boot实战[M].北京:电子工业出版社,2016.

# 致 谢

本次毕业设计能够顺利完成，首先要感谢我的指导老师，刘灯明老师。在我毕业设计遇到困难的时候，刘灯明老师给予了我很大的帮助。在此向刘灯明老师表示衷心的谢意。

我在公司实习期间，我的同事们给我的毕业设计提出了技术性的建议和指导。在此向我的同事们表示衷心的感谢。

另外也要感谢我的家人们，他们的理解和支持使我能专心完成我的学业。

# 

# 附 录

**附录A：项目框架配置文件**

**application.yml**

projectName: RollCallSystem

server:

port: 8080

tomcat:

uri-encoding: UTF-8

max-threads: 1000

servlet:

context-path: /${projectName}

context-path: /${projectName}

spring:

datasource:

url: jdbc:mysql://127.0.0.1:3306/rollcall

username: RollCall

password: 123456

driver-class-name: com.mysql.jdbc.Driver

# 使用druid数据源

type: com.alibaba.druid.pool.DruidDataSource

filters: stat

maxActive: 20

initialSize: 1

maxWait: 60000

minIdle: 1

timeBetweenEvictionRunsMillis: 60000

minEvictableIdleTimeMillis: 300000

validationQuery: select 'x'

testWhileIdle: true

testOnBorrow: false

testOnReturn: false

poolPreparedStatements: true

maxOpenPreparedStatements: 20

devtools:

restart:

exclude: static/\*\*,public/\*\*

mvc:

view:

suffix: .html

#redis配置

redis:

database: 0

host: localhost

port: 6379

# 连接超时时间 单位 ms（毫秒）

timeout: 3000ms

pool:

# 如果赋值为-1，则表示不限制；如果pool已经分配了maxActive个jedis实例，则此时pool的状态为exhausted(耗尽)。

max-active: 8

# 等待可用连接的最大时间，单位毫秒，默认值为-1，表示永不超时。如果超过等待时间，则直接抛出JedisConnectionException

max-wait: -1

mybatis:

mapper-locations: classpath:mapper/\*.xml

type-aliases-package: com.chm.domain

#配置分页插件pagehelper

pagehelper:

helperDialect: mysql

reasonable: true

supportMethodsArguments: true

params: count=countSql

#跨域

axios:

defaults:

withCredentials: true

#Logback config

logging:

config: classpath:logback-spring.xml

#日志配置

logging:

file: MyLog.log

#签到时间配置

NORMALSTART: 20 #正常签到开始时间(上课前20分钟)

NORMALEND: 5 #正常签到结束时间(上课后5分钟)

LATE: 20 #迟到(上课后5分钟至上课后20分钟)

#学期开始时间配置

STARTWEEK: 10 #今年的第几周

#默生人阈值

THRESHOLD: 75.0

#当前学期

SEMESTER: 2016-2017-1

**附录B：token拦截**

@Override

public boolean preHandle(HttpServletRequest request,

HttpServletResponse response, Object handler) throws Exception {

//如果不是映射到方法直接通过

if (!(handler instanceof HandlerMethod)) {

return true;

}

HandlerMethod handlerMethod = (HandlerMethod) handler;

Method method = handlerMethod.getMethod();

//该方法没有注明Authorization，跳过验证token

if (method.getAnnotation(Authorization.class) == null) {

return true;

}

String userToken = request.getHeader("token");

if (redisRepository == null) {

//解决service为null无法注入问题

BeanFactory factory = WebApplicationContextUtils.getRequiredWebApplicationContext(request.getServletContext());

redisRepository = (RedisRepository) factory.getBean("redisRepository");

}

//查询token是否存在

if (userToken != null && redisRepository.get(userToken) != null) {

redisRepository.update(userToken);

return true;

}

//验证token失败，并且方法注明了Authorization，返回401错误

throw new ParamExecption("登录超时");

}

**附录C：学生签到**

/\*\*

\* 学生签到方法

\*

\* @param image 学生图像

\* @param schid 课表id

\* @param signedTime 签到时间

\* @return 签到学生的实例

\*/

@Override

public synchronized Student signed(String image, Integer schid, LocalTime signedTime) {

//获取课表实例

Schedule schedule = scheduleMapper.selectByPrimaryKey(schid);

if (schedule.getStarttime().toSecondOfDay() - signedTime.toSecondOfDay() > NORMALSTART) {

//没到签到时间

return null;

}

//根据任课实例获取课堂所有学生相应的标签

List<String> labels = classMapper.getClasses(schid);

//识别，获取识别结果 label格式:groupId/userId

String label = faceRecognition.recogntion(image, labels);

//计算本周为第几周

String weekofsemester = String.valueOf(LocalDate.now().get(ChronoField.ALIGNED\_WEEK\_OF\_YEAR) - STARTWEEK + 1);

if (label != null) {

//获取学生ID

String stuId = label.split("/")[1];

//插入签到结果

//查询是否已签到

if (recordMapper.selectStatusByStuidAndSchidAndWeekofsemester(stuId, schid, weekofsemester, SEMESTER) == null) {

//进行签到

//比较签到时间

long time = schedule.getStarttime().toSecondOfDay() - signedTime.toSecondOfDay();

//新建签到实例

Record record = new Record();

//学生学号

record.setStuid(stuId);

//课表编号

record.setSchid(schid);

//周数

record.setWeekofsemester(weekofsemester);

if (time > NORMALEND && time < NORMALSTART) {

//正常签到

record.setStatus("正常");

} else if (time > LATE) {

//迟到

record.setStatus("迟到");

} else if (time < LATE && schedule.getEndtime().getSecond() > signedTime.getSecond()) {

//缺课

record.setStatus("缺课");

}

//签到时间

record.setSignedtime(LocalTime.now());

//当前学期

record.setSemester(SEMESTER);

//插入签到记录

recordMapper.insert(record);

//构造人脸数据训练状态对象

FaceDataTrainStatus faceDataTrainStatus = new FaceDataTrainStatus();

faceDataTrainStatus.setStuid(stuId);

faceDataTrainStatus.setStatus("0");

//插入人脸数据训练状态记录

faceDataTrainStatusMapper.insertOrUpdate(faceDataTrainStatus);

//插入人脸数据

FaceData faceData = new FaceData();

faceData.setFacedate(image);

faceData.setStuid(stuId);

faceDataMapper.insertSelective(faceData);

}

Student student = studentMapper.selectByPrimaryKey(stuId);

//返回学生学号

return student;

} else {

return null;

}

}

**附录D：用户登录**

/\*\*

\* 用户登录

\*

\* @param params

\* @return

\*/

@Override

public Result login(Map params) {

Validate.login(params);

String userId = params.get("userId").toString();

String password = params.get("password").toString();

String token = null;

String name = null;

String userType = null;

//判别登陆类型

if (RoleTypeUtils.discriminant(userId).equals(RoleTypeUtils.STUDENT)) {

if (password.equals(studentMapper.getPasswordByStuid(userId))) {

token = redisRepository.add(userId);

name = ((Student) redisRepository.get(token)).getStuname();

userType = "student";

}

} else if (RoleTypeUtils.discriminant(userId).equals(RoleTypeUtils.TEACHER)) {

if (password.equals(teacherMapper.getPasswordByTeaid(userId))) {

token = redisRepository.add(userId);

name = ((Teacher) redisRepository.get(token)).getTeaname();

userType = "teacher";

}

} else if (RoleTypeUtils.discriminant(userId).equals(RoleTypeUtils.MANAGER)) {

if (password.equals(managerMapper.getPasswordByAccount(userId))) {

token = redisRepository.add(userId);

name = "管理员";

userType = "manager";

}

}

//登陆成功

Map map = Maps.newHashMap();

if (token != null) {

map.put("token", token);

map.put("name", name);

map.put("userType", userType);

//返回token

return ResultUtils.success(map);

}

return ResultUtils.error("", "账号或密码错误");

}

**附录E：用户退出**

/\*\*

\* 用户退出

\*

\* @param request

\* @return

\*/

@Override

public Result logout(HttpServletRequest request) {

String token = request.getHeader("token");

redisRepository.delete(token);

return ResultUtils.success();

}