**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 七 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： 2017年6月3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 38 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**

现代社会，随着计算机技术和网络通信技术的不断发展和迅速崛起我们的生产生活方式也发生了巨大的变化。同时由计算机设计的一些软件和系统为我们的生活提供了许多方便，从而也代替了一些较复杂的手工程序。基于这种思想，本文设计实现了一个基于Java平台的学生签到系统。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，也提高了课堂质量，能够准确反映学生出勤情况。

本点名签到系统项目利用了Java、Spring等技术，采用IntelliJ IDEA 2017作为开发工具，运用了B/S与C/S结构。实现了课堂签到、签到记录查询、人脸信息添加等功能。本系统主要作用是对学生进行课堂签到、记录学生签到记录、统计学生签到记录。其根本目的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。

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

**ABSTRACT**

With continuous development and rapid enhancement of computer technologyand Network communications technology ,our life has changed tremendously in modernsociety.At the same time,some softwares and systems designed by the computer provide alot of convenience for us,thus it also replaced some of the complex manual

Procedures.Based on this idea,this paper designs and implements a student sign in systembased on Java platform.Compared to the traditional way of naming such as teacher's

Naming and sign in,it uses face recognition to solve students'signing problem which is bothconvenient and quick.Moreover,It enhanced classroom quality and can reflect rheattendance of the students accurately.

This name check-in system project uses Java,Spring and other technologies , with IntelliJ IDEA 2017 as a development tool . It still used B/S and C/S structure.The functions of classroom attendance,check in attendance record,and face information addition are realized. The main function of this system is to sign students in class , recoed students'attendance history and check students' attendence record.Its fundamental purpose is to help teachers to count students in their regular check-in,which can give a reasonable result for students' assessment.

**Keywords：**JAVA ; Spring; IntelliJ IDEA 2017; Name check-in system

目录

[第一章 绪 论](#_Toc21148_WPSOffice_Level1) [1](#_Toc21148_WPSOffice_Level1)

[1.1 研究背景](#_Toc2910_WPSOffice_Level2) [1](#_Toc2910_WPSOffice_Level2)

[1.2 研究内容](#_Toc2421_WPSOffice_Level2) [1](#_Toc2421_WPSOffice_Level2)

[1.3 运行环境](#_Toc16280_WPSOffice_Level2) [1](#_Toc16280_WPSOffice_Level2)

[1.4 系统相关技术介绍](#_Toc30633_WPSOffice_Level2) [1](#_Toc30633_WPSOffice_Level2)

[1.4.1 Spring Boot简介](#_Toc2910_WPSOffice_Level3) [2](#_Toc2910_WPSOffice_Level3)

[1.4.2 IDEA简介](#_Toc2421_WPSOffice_Level3) [2](#_Toc2421_WPSOffice_Level3)

[1.4.3 MyBatis简介](#_Toc16280_WPSOffice_Level3) [2](#_Toc16280_WPSOffice_Level3)

[1.4.3 Redis简介](#_Toc30633_WPSOffice_Level3) [3](#_Toc30633_WPSOffice_Level3)

[1.4.4 JavaCV简介](#_Toc15606_WPSOffice_Level3) [3](#_Toc15606_WPSOffice_Level3)

[1.4.5 Druid简介](#_Toc28095_WPSOffice_Level3) [3](#_Toc28095_WPSOffice_Level3)

[1.4.6 Vue.js简介](#_Toc15815_WPSOffice_Level3) [3](#_Toc15815_WPSOffice_Level3)

[1.4.7 前后端分离模式简介](#_Toc14640_WPSOffice_Level3) [4](#_Toc14640_WPSOffice_Level3)

[第二章 系统分析](#_Toc2910_WPSOffice_Level1) [6](#_Toc2910_WPSOffice_Level1)

[2.1 可行性分析](#_Toc15606_WPSOffice_Level2) [6](#_Toc15606_WPSOffice_Level2)

[2.1.1 技术可行性](#_Toc3170_WPSOffice_Level3) [6](#_Toc3170_WPSOffice_Level3)

[2.1.2 经济可行性](#_Toc27321_WPSOffice_Level3) [6](#_Toc27321_WPSOffice_Level3)

[2.1.3 操作可行性](#_Toc3939_WPSOffice_Level3) [7](#_Toc3939_WPSOffice_Level3)

[2.2 需求分析](#_Toc28095_WPSOffice_Level2) [7](#_Toc28095_WPSOffice_Level2)

[2.2.1 功能需求](#_Toc19465_WPSOffice_Level3) [7](#_Toc19465_WPSOffice_Level3)

[2.2.1 性能需求](#_Toc10606_WPSOffice_Level3) [7](#_Toc10606_WPSOffice_Level3)

[2.2.3 软件流程图](#_Toc7638_WPSOffice_Level3) [8](#_Toc7638_WPSOffice_Level3)

[第三章 总体设计](#_Toc2421_WPSOffice_Level1) [10](#_Toc2421_WPSOffice_Level1)

[3.1 功能模块划分](#_Toc14640_WPSOffice_Level2) [10](#_Toc14640_WPSOffice_Level2)

[3.1.1 系统功能结构图](#_Toc18374_WPSOffice_Level3) [10](#_Toc18374_WPSOffice_Level3)

[3.1.2 系统流程图](#_Toc23603_WPSOffice_Level3) [12](#_Toc23603_WPSOffice_Level3)

[3.2 数据库设计](#_Toc6980_WPSOffice_Level2) [13](#_Toc6980_WPSOffice_Level2)

[3.2.1 系统E-R图](#_Toc12579_WPSOffice_Level3) [13](#_Toc12579_WPSOffice_Level3)

[3.2.2 关系模型](#_Toc10606_WPSOffice_Level2) [14](#_Toc10606_WPSOffice_Level2)

[3.2.3 数据库表设计](#_Toc7638_WPSOffice_Level2) [14](#_Toc7638_WPSOffice_Level2)

[第四章 详细设计](#_Toc16280_WPSOffice_Level1) [19](#_Toc16280_WPSOffice_Level1)

[4.1 用户通用模块](#_Toc16882_WPSOffice_Level2) [19](#_Toc16882_WPSOffice_Level2)

[4.1.1 用户登录](#_Toc24796_WPSOffice_Level3) [19](#_Toc24796_WPSOffice_Level3)

[4.1.2 用户退出](#_Toc22148_WPSOffice_Level3) [20](#_Toc22148_WPSOffice_Level3)

[4.1.3 token验证](#_Toc30145_WPSOffice_Level3) [20](#_Toc30145_WPSOffice_Level3)

[4.2 学生模块](#_Toc31108_WPSOffice_Level2) [21](#_Toc31108_WPSOffice_Level2)

[4.2.1 查询签到记录](#_Toc11101_WPSOffice_Level3) [21](#_Toc11101_WPSOffice_Level3)

[4.2.2 统计签到记录](#_Toc21696_WPSOffice_Level3) [21](#_Toc21696_WPSOffice_Level3)

[4.2.3 课堂签到](#_Toc11619_WPSOffice_Level3) [22](#_Toc11619_WPSOffice_Level3)

[4.3 教师模块](#_Toc20198_WPSOffice_Level2) [23](#_Toc20198_WPSOffice_Level2)

[4.3.1 查询学生签到记录](#_Toc19138_WPSOffice_Level3) [23](#_Toc19138_WPSOffice_Level3)

[4.3.2 统计学生签到记录](#_Toc19393_WPSOffice_Level3) [24](#_Toc19393_WPSOffice_Level3)

[4.4 管理员模块](#_Toc3696_WPSOffice_Level2) [25](#_Toc3696_WPSOffice_Level2)

[4.4.1 添加学生人脸信息](#_Toc4258_WPSOffice_Level3) [25](#_Toc4258_WPSOffice_Level3)

[4.5 定时任务模块](#_Toc10706_WPSOffice_Level2) [26](#_Toc10706_WPSOffice_Level2)

[4.5.1 定时训练学生人脸模型](#_Toc12499_WPSOffice_Level3) [26](#_Toc12499_WPSOffice_Level3)

[第五章 系统测试](#_Toc30633_WPSOffice_Level1) [29](#_Toc30633_WPSOffice_Level1)

[5.1 测试的目标](#_Toc22364_WPSOffice_Level2) [29](#_Toc22364_WPSOffice_Level2)

[5.2 测试的原则](#_Toc25246_WPSOffice_Level2) [29](#_Toc25246_WPSOffice_Level2)

[5.3 测试方法](#_Toc28603_WPSOffice_Level2) [29](#_Toc28603_WPSOffice_Level2)

[5.4 测试内容](#_Toc23398_WPSOffice_Level2) [29](#_Toc23398_WPSOffice_Level2)

[5.4.1 用户通用模块测试](#_Toc24307_WPSOffice_Level3) [29](#_Toc24307_WPSOffice_Level3)

[5.4.2 学生模块测试](#_Toc20924_WPSOffice_Level3) [30](#_Toc20924_WPSOffice_Level3)

[第六章 总结](#_Toc15606_WPSOffice_Level1) [34](#_Toc15606_WPSOffice_Level1)

[参考文献](#_Toc28095_WPSOffice_Level1) [36](#_Toc28095_WPSOffice_Level1)

[致 谢](#_Toc15815_WPSOffice_Level1) [38](#_Toc15815_WPSOffice_Level1)

[附 录](#_Toc14640_WPSOffice_Level1) [39](#_Toc14640_WPSOffice_Level1)

[附录A：项目框架配置文件](#_Toc3170_WPSOffice_Level1) [39](#_Toc3170_WPSOffice_Level1)

1. **绪 论**

## 1.1 研究背景

## 现代社会，随着计算机技术和网络通信技术的不断发展和迅速崛起我们的生产生活方式也发生了巨大的变化。同时由计算机设计的一些软件和系统为我们的生活提供了许多方便，从而也代替了一些较复杂的手工程序。基于这种思想，本文设计实现了一个基于Java平台的学生签到系统。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，也提高了课堂质量，能够准确反映学生出勤情况。

## 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/HTML；

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用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。

Spring Boot具有以下特征：

1. 遵循“习惯优于配置”原则，使用Spring Boot只需很少的配置，大部分时候可以使用默认配置。
2. 项目快速搭建，可以务配置整合第三方框架。
3. 可以完全不适用xml配置，只使用自动配置和Java Config。
4. 内嵌Servlet（如Tomcat）容器，应用可用jar包运行（java -jar）。
5. 运行中应用状态的监控。

### 1.4.2 IDEA简介

### IDEA 全称 IntelliJ IDEA，是java语言开发的集成环境，IntelliJ在业界被公认为最好的java开发工具之一，尤其在智能代码助手、代码自动提示、重构、J2EE支持、Ant、JUnit、CVS整合、代码审查、 创新的GUI设计等方面的功能可以说是超常的。

### 1.4.3 MyBatis简介

MyBbatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。MyBatis 避免了几乎所有的 JDBC代码和手动设置参数以及获取结果集。MyBatis 可以使用简单的XML或注解来配置和映射原生信息，将接口和Java的POJOs(Plain Old Java Objects，普通的Java对象)映射成数据库中的记录。

Mybatis的功能架构分为三层：

(1)API接口层：提供给外部使用的接口API，开发人员通过这些本地API来操纵数据库。接口层一接收到调用请求就会调用数据处理层来完成具体的数据处理。

(2)数据处理层：负责具体的SQL查找、SQL解析、SQL执行和执行结果映射处理等。它主要的目的是根据调用的请求完成一次数据库操作。

(3)基础支撑层：负责最基础的功能支撑，包括连接管理、事务管理、配置加载和缓存处理，这些都是共用的东西，将他们抽取出来作为最基础的组件。为上层的数据处理层提供最基础的支撑。

### 1.4.3 Redis简介

Redis是一个远程内存数据库，它不仅性能强劲，而且还具有复制特性以及为解决问题而生的独一无二的数据模型。Redis提供了5种不同类型的数据结构，各式各样的问题都可以很自然地映射到这些数据结构上：Redis的数据结构致力于帮助用户解决问题，而不会像其他数据库那样，要求用户扭曲问题来适应数据库。除此之外，通过复制、持久化（persistence）和客户端分片（client-side sharding）等特性，用户可以很方便地将Redis扩展成一个能够包含数百GB数据、每秒处理上百万次请求的系统。

### 1.4.4 JavaCV简介

JavaCV 提供了在计算机视觉领域的封装库，包括：OpenCV、ARToolKitPlus、libdc1394 2.x 、PGR FlyCapture和FFmpeg。此外，该工具可以很容易地使用Java平台的功能。

JavaCV 还带有硬件加速的全屏幕图像显示（CanvasFrame），易于在多个内核中执行并行代码（并行），用户友好的几何和色彩的相机和投影仪校准（GeometricCalibrator，ProCamGeometricCalibrator，ProCamColorCalibrator ），检测和特征点（ObjectFinder），一类是实现投影，摄像系统（直接图像对齐设置匹配主要GNImageAligner，ProjectiveTransformer，ProjectiveGainBiasTransformer，ProCamTransformer 和ReflectanceInitializer），以及在 JavaCV 类杂项功能。

### 1.4.5 Druid简介

Druid首先是一个数据库连接池。Druid是目前最好的数据库连接池，在功能、性能、扩展性方面，都超过其他数据库连接池，包括DBCP、C3P0、BoneCP、Proxool、JBoss DataSource。Druid已经在阿里巴巴部署了超过600个应用，经过一年多生产环境大规模部署的严苛考验。

同时Druid不仅仅是一个数据库连接池，它包括三个部分：

（1）基于Filter－Chain模式的插件体系。

（2）DruidDataSource 高效可管理的数据库连接池。

（3）SQLParser。

### 1.4.6 Vue.js简介

Vue.js 是一个JavaScriptMVVM库，是一套构建用户界面的渐进式框架。它是以数据驱动和组件化的思想构建的，采用自底向上增量开发的设计。相比于Angular.js，Vue.js提供了更加简洁、更易于理解的API，使得我们能够快速地上手并使用Vue.js；同时比起 React + Redux 相对复杂的架构，Vue.js 更加轻量级也更加容易上手，是初创项目的首选前端框架。Vue 的核心库只关注视图层，它不仅易于上手，还便于与第三方库或既有项目整合。并且作者是华人的关系，Vue拥有着对华人开发者最友好的api文档和官方教程。

### 1.4.7 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）：前端先写一个静态页面，写好后，让后端去套模板。静态页面可以本地开发，也无需考虑业务逻辑只需要实现View即可。不足是还需要后端套模板，这些前端代码后端需要浏览一遍，以免出错。

（2）：前端直接去写模板，这样做的问题在于，前端编写过程中很依赖与后端环境，如果当后端没写完的情况下，前端几乎没法干活。

显然这两种方式似乎都有很多问题，但至少这还是目前为止大部分公司所采用的模式。他们从物理层来区分前后端的开发，同时淡化了前端在逻辑上的色彩。由于前端所做的事情就是来实现一个页面的静态版本，所以，大多数公司又给前端工程师们找了点活干。你去看现在公司在招聘的时候前端工程师的要求，除了对页面的基本制作技能外还有额外的设计职责。

到这里原本我们以为已经将前后端分离开来了，但是在模版这个尴尬的问题上，前后端的工程师们绝对吃过不少苦头，因为在整体网站架构上，这并不是前后端的分离。

但是呢，后来开发的时候遇到很多问题，前端(html，css，js)工程师对php代码可能读不懂，php工程师也不对css那些内容非常熟悉，所以后来人们想出一个办法，把代码分离!这样就能让每个人在各自的领域更好的发挥，这样虽然消耗了一定的效率，但是对于开发人员从某一角度来说是友好的。

前后端分离并没有网上说的那么复杂。

（1）：首先要知道所有的程序都是一数据为基础的，没有数据的程序没有实际意义，程序的本质就是对程序的增删改查。

（2）：前后端分离就是把数据操作和显示分离出来。前端专注做数据显示，通过文字，图片或者图标等方式让数据形象直观的显示出来。后端专注做数据的操作。前端把数据发给后端，有后端对数据进行修改。

（3）：后端一般用java，c#等语言，现在的node属于JavaScript也能进行后端操作，此处不意义裂解语言。后端来进行数据库的链接，并对数据进行操作。

（4）：后端提供接口给前端调用，来触发后端对数据的。

1. **系统分析**

## 2.1 可行性分析

并非任何问题都有简单明显的解决方法，事实上，许多问题不可能在预定的系统规模或时间期限之内解决。如果问题没有可行的解，那么花费在这项工程上的任何时间、人力、软硬件资源和经费，都是无谓的浪费。

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。

### 2.1.1 技术可行性

本项目采用Java语言进行编写，Java语言有如下有优点：

（1）：Java为纯面向对象的语言。Java是一种everything is object的语言，能够直接反映现实生活中的对象。

（2）：平台无关性。Java语言可以一次编译到处运行，编译后的代码可以在多种平台上运行。由于Java是解释性语言，编译器会把Java代码编程中间代码，然后在JVM中运行，因此Java可以跨平台运行，具有良好的可移植性。

（3）：为完成特定任务内置的多种库。通过使用这些库，简化了开发者的开发难度，例如：Java对多线程的支持，网络通信的支持，以及Java中很重要的垃圾回收机制，异常处理等。

（4）：提供了对WEB应用开发的支持。例如：Applet，Server ，JSP等

（5）：具有良好的安全性和健壮性。网络是Java的一个重要使用领域，为了增强安全性，Java提供例如防止恶意代码攻击的安全机制(数组边界检测和ByteCode校验等)，Java的强类型机制，垃圾回收器，异常处理，安全检查机制等都使得Java编写的程序有着良好的健壮性。

（6）：去除了C++语言中难以理解，容易混淆的特性，指针，结构，单元，运算符重载，虚拟基础类，多重继承等等，使得程序更加严谨，简洁。

基于以上优点我选择Java作为本项目的开发语言。

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。2000年至今，人工智能的“数据挖掘”时代。随着各种机器学习算法的提出和应用，特别是深度学习技术的发展，人们希望机器能够通过大量数据分析，从而自动学习出知识并实现智能化水平。这一时期，随着计算机硬件水平的提升，大数据分析技术的发展，机器采集、存储、处理数据的水平有了大幅提高。特别是深度学习技术对知识的理解比之前浅层学习有了很大的进步，Alpha Go和中韩围棋高手过招大幅领先就是目前人工智能的高水平代表之一。

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。

### 2.1.2 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。

### 2.1.3 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。

操作可行性研究主要研究本系统的操作方式在这个用户组织内是否行得通。

本点名操作系统界面清晰直观、易于操作。操作本系统的用户只需懂得基本计算机操作即可操作本系统，所以，本系统是操作可行的。

## 2.2 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。

### 2.2.1 功能需求

从上述方式对本系统进行功能需求分析，本系统应满足功能如下：

1. 学生用户部分
2. 、用户登录
3. 、用户退出
4. 、查询签到记录
5. 、统计签到记录
6. 、课堂签到
7. 教师用户部分
8. 、用户登录
9. 、用户退出
10. 、查询学生签到记录
11. 、统计学生签到记录
12. 、学生请假状态记录
13. 、学生补签
14. 管理员部分

（1）、添加学生人脸信息

1. 定时任务训练人脸识别模型
2. 、定时训练学生人脸模型

### 2.2.1 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。

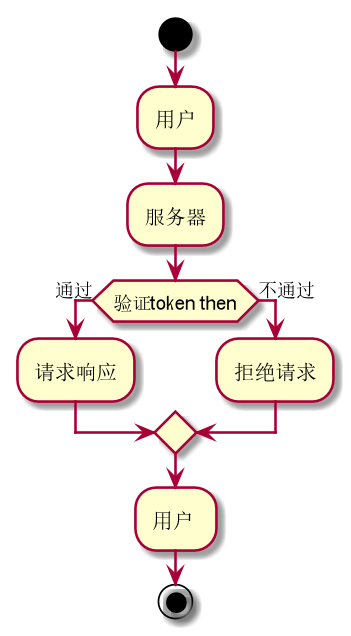
1. 易用性

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。

1. 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session存储用户信息的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。

### 2.2.3 软件流程图



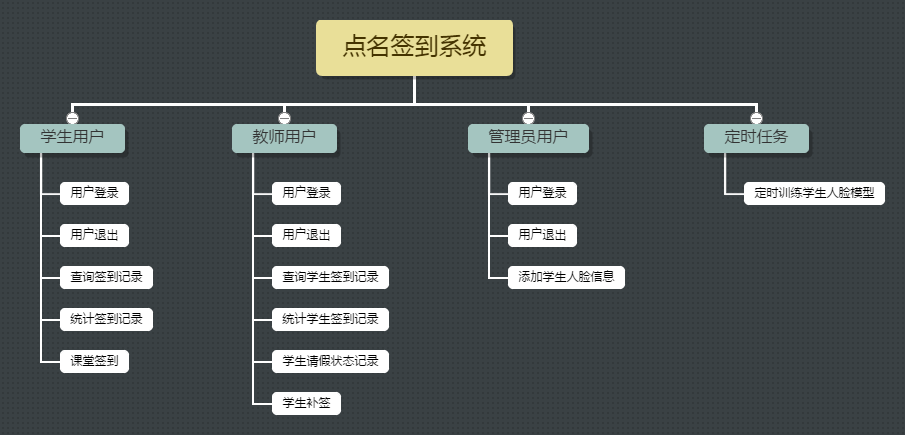
**图2.1 软件流程图**

# 第三章 总体设计

经过上一个阶段的分析，系统要“做什么”已经清楚了，现在是决定系统“怎么做”的时候了。在本阶段，主要是设计出软件的总体结构和划分出组成系统的物理元素——程序、文件、数据库、人工过程和文档等，但是写个元素都是处于黑盒子级的，这些元素的具体内容会将在以后仔细设计。

## **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。

1. 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，在使用查询学生签到记录查询学生签到记录时，会查询相应学生的统计记录。

1. 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。

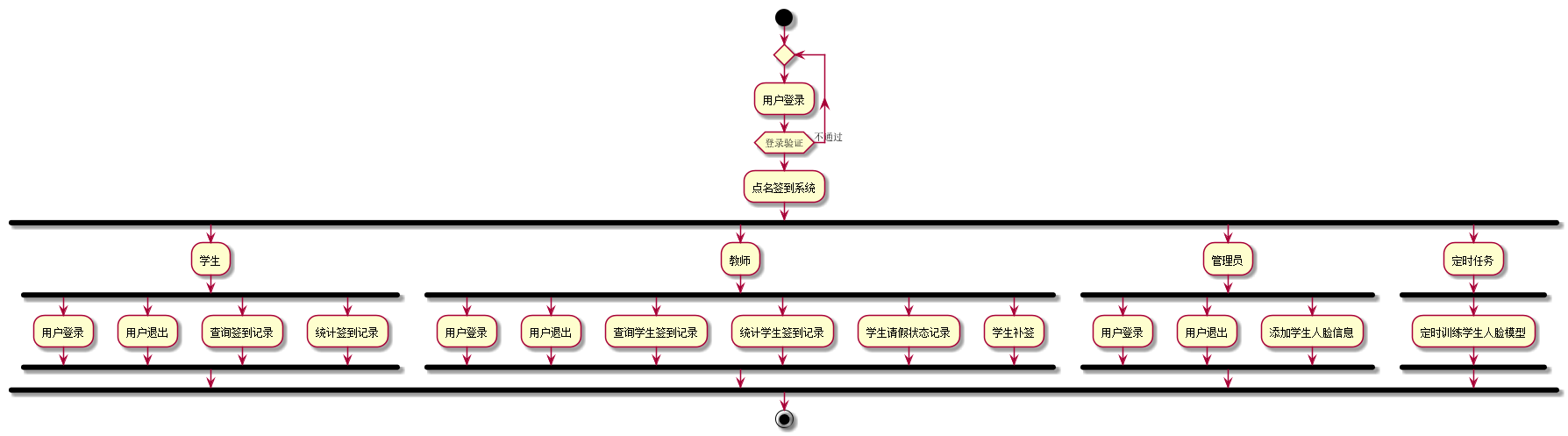
用户退出，用户将自己的token发送到后台，系统在后台取出token，并将该token在Redis中清除token与对应的用户信息，然后将用户界面跳转至登录界面。

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。

1. 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



**图3.2 系统流程图**

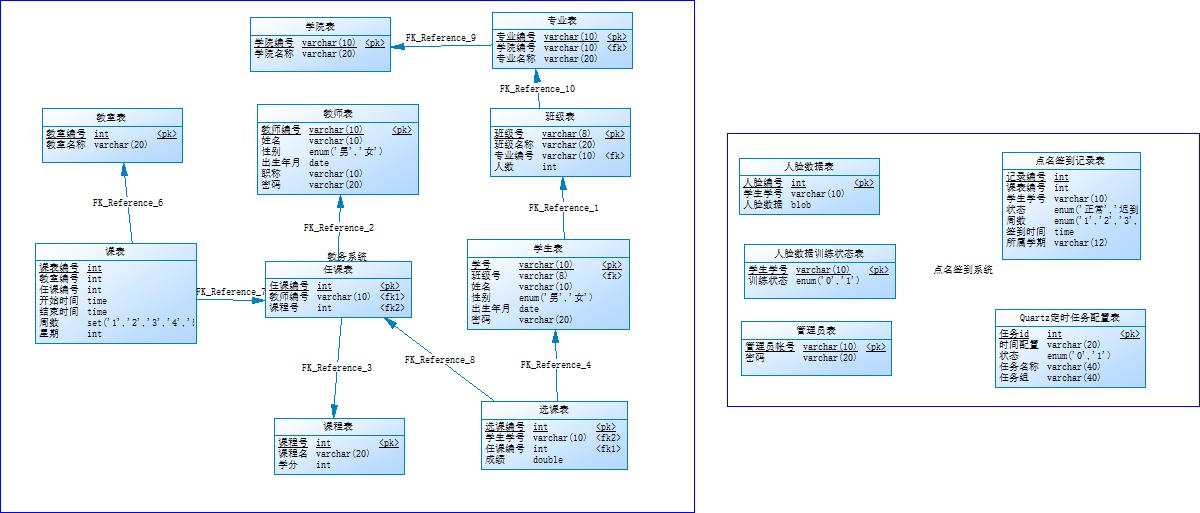
## 3.2 数据库设计

数据库设计的好坏是一个关键。如果把企业的数据比做生命所必需的血液，那么数据库的设计就是应用中最重要的一部分，是一个系统的根基。用于开发人员进行项目设计，以此作为编码的依据，同时也为后续的数据库维护工作提供了良好的使用说明，也可以作为未来版本升级时的重要参考资料。数据库设计的目标是建立一个合适的数据模型。这个数据模型应当是满足用户要求，既能合理地组织用户需要的所有数据，又能支持用户对数据的的所有处理功能。并且要具有较高的范式，数据完整性好，效益高，便于理解和维护，没有数据冲突。

### 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。

系统E-R图如图3.3



**图3.3 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3.1** 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3.1** 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3.1** 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3.1** 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3.1** 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3.1** 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3.1** 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3.1** 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3.1** 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3.1** 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3.1 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3.1 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3.1 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 |

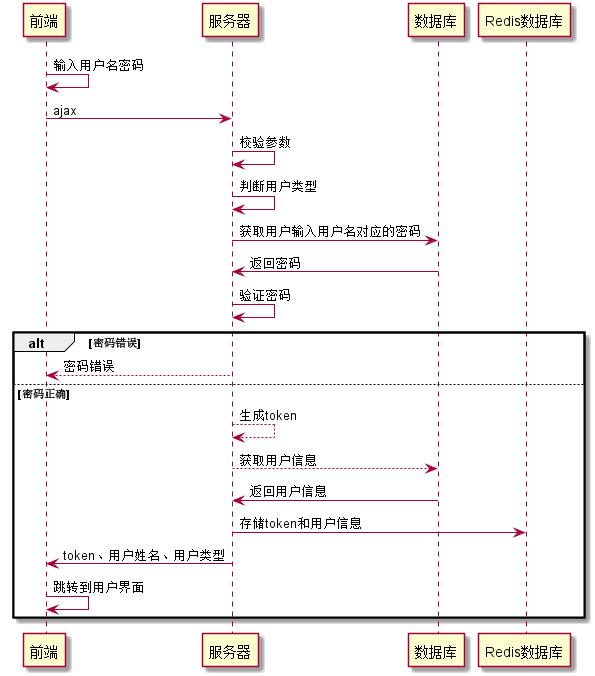
**第四章 详细设计**

详细设计阶段的根本目标是确定应该怎样具体地实现所要求的系统，也就是说，经过这个阶段的设计工作应该得出对目标系统的精确描述，从而在编码阶段可以把这个描述直接翻译成某种程序设计语言书写的程序。

## 4.1 用户通用模块

### 4.1.1 用户登录

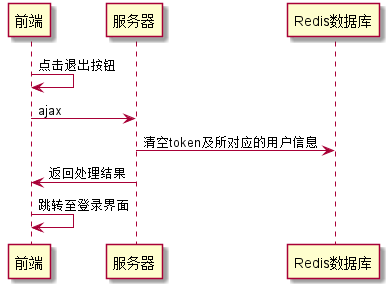
用户进入用户登录界面，输入用户名和密码，点击登录。浏览器发起发起ajax异步请求，把用户名和密码组装成json格式，发送到服务器的登录Controller。Controller在接收请求后，校验参数是否齐全。然后根据用户名判断用户的角色类型，从而从数据库相应的表中取出用户名对应的密码，然后将用户的输入的密码与数据库中取出的密码进行判断看是否一致。若一致，则用过加密算法生成一个token，并从数据库中取出用户信息，将token作为name，用户信息作为value。将token和用户信息作为键值对存在服务器本地的Redis数据库中，并将生成的token，用户姓名，用户角色类型组装成json格式返回给用户。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。时序图如图4.1



**图4.1 登录时序图**

### 4.1.2 用户退出

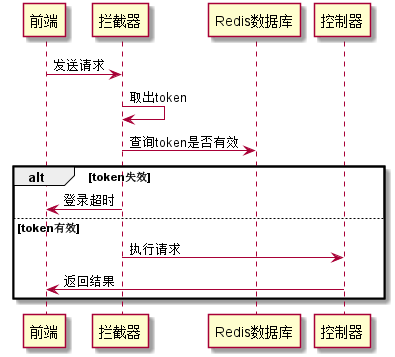
用户在用户界面点击退出按钮，浏览器发起发起ajax异步请求，把token发送到服务器的退出Controller。Controller在接收到请求后，将token及所对应的用户信息从服务器本地的Redis数据库中清除。然后前端界面跳转到登录界面。时序图如图4.2

****

**图4.2 退出时序图**

### 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。时序图如图4.3

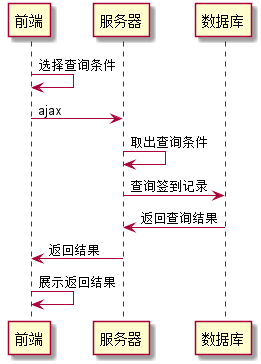
****

**图4.3 token验证时序图**

## 4.2 学生模块

### 4.2.1 查询签到记录

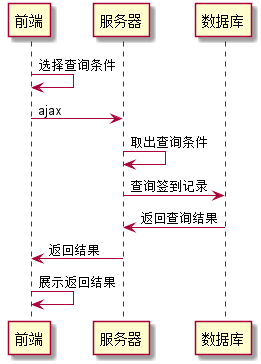
学生在完成登录进入用户界面后，可选择相应的课程（可不选，若没有则默认为全部课程），开始周数（可不选），结束周数（可不选），浏览器发起发起ajax异步请求，把选择参数发送到服务器，系统后台获取这些参数后，在数据库中查询出相应的记录，返回查询结果到前端界面，用户界面获取查询结果并展示。时序图如图4.4



**图4.4 查询签到记录时序图**

### 4.2.2 统计签到记录

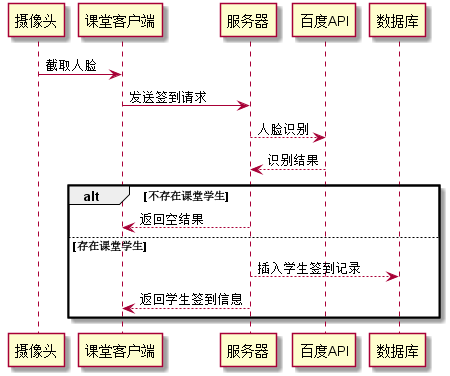
学生在完成登录进入用户界面后，使用查询签到记录功能时，会查询相应课程的统计记录，若不选择课程，则会默认统计全部课程的签到记录。浏览器发起发起ajax异步请求，把选择参数发送到服务器，系统后台获取这些参数后，在数据库中查询出相应的记录，返回查询结果到前端界面，用户界面获取查询结果并展示。时序图如图4.5



**图4.5 统计签到记录时序图**

### 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。时序图如图4.6。

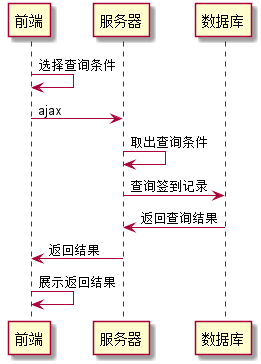


**图4.6 课堂签到时序图**

## 4.3 教师模块

### 4.3.1 查询学生签到记录

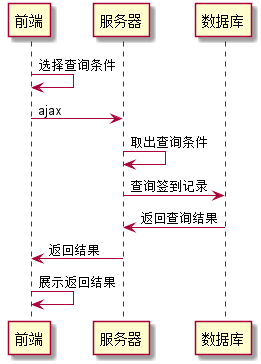
查询学生签到记录，教师在完成登录进入用户界面后，可选择该教师所任课程列表，然后可以根据该课程列表查询所任课程的课堂记录，从而获取某一课堂记录中学生的签到记录。或者也可以查询某一课程的学生，获取该学生在本门课程的签到记录。浏览器发起发起ajax异步请求，系统后台获取到查询参数后，在数据库中查询出相应的记录，返回查询结果到前端界面，用户界面获取查询结果并展示。时序图如图4.7。



**图4.7 查询学生签到记录时序图**

### 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。浏览器发起发起ajax异步请求，把选择参数发送到服务器，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。时序图如图4.8。

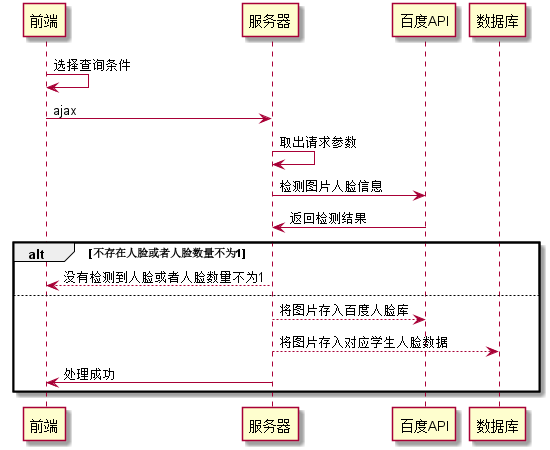


**图4.8 统计学生签到记录时序图**

## 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。时序图如图4.9。

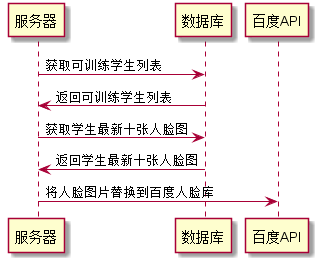


**图4.9 添加学生人脸信息时序图**

## 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。时序图如图4.10。

****

**图4.10 定时训练学生人脸模型**

# 第五章 系统测试

## 5.1 测试的目标

（1）软件测试是为了发现错误而执行程序的过程。

（2）测试是为了证明程序有错，而不是证明程序无错。

（3）一个好的测试用例在于它发现至今未发现的错误。

（4）一个成功的测试是发现了至今未发现的错误的测试。

## 5.2 测试的原则

（1）应当把“尽早地不断地进行软件测试“作为软件开发者的座右铭。

（2）测试用例应由测试数据和与之对应的预期输出结果这两部分组成。

（3）程序员应避免检查自己的程序。

（4）在设计测试用例时，应当包括合理的输入条件和不合理的输入条件。

（5）充分注意测试中的群集现象。

（6）严格执行测试计划，排除测试的随意性。

（7）应当对每一个测试结果做全面的检查。

（8）妥善保存测试计划、测试用例、出错统计和最终分析报告，为维护提供方便。

## 5.3 测试方法

软件测试方法是指测试软件的方法。随着软件测试技术的不断发展，测试方法也越来越多样化，针对性更强；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。

对于软件测试而言，黑盒测试法把程序看作一个黑盒子，完全不考虑程序的内部结构和处理过程。也就是说，黑盒测试是在程序接口进行的测试，它只检查程序功能是否能按照规格说明书的规定正常使用，程序是否能适当地接收输人数据并产生正确的输出信息，程序运行过程中能否保持外部信息(例如数据库或文件)的完整性。黑盒测试又称为功能测试。

## 5.4 测试内容

本节将以用户通用模块与学生模块测试过程为例，阐述软件测试方法。

### 5.4.1 用户通用模块测试

（1）用户登录测试：

测试目的：检测用户登录功能是否可以正常使用。

|  |  |  |
| --- | --- | --- |
| 测试用例 | 测试过程 | 测试结果 |
| 用户名1407020401，密码123456 | 用户名密码正确，页面直接跳转学生界面 | 测试成功 |
| 用户名1407020401，密码123453 | 登录失败，返回信息：账号或密码错误 | 测试成功 |

（1）用户退出测试：

测试目的：检测用户退出功能是否可以正常使用。

|  |  |  |
| --- | --- | --- |
| 测试用例 | 测试过程 | 测试结果 |
| 点击退出按钮 | 页面直接跳转至登录界面 | 测试成功 |

### 5.4.2 学生模块测试

测试目的：检测学生模块是否可以正常使用

测试用例及结果如图5.1、图5.2、图5.3、图5.4



**图5.1**



**图5.2**



**图5.3**



**图5.3**

通过本次对系统的测试，发现了一些错误，踩了很多坑，进一步完善了系统，也学到了许多系统所使用框架的技巧。

# 总结

这次的毕业设计，是我大学学习阶段的一个总结已经检验。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，从而增大了模块内部的内聚，减少了模块间的耦合。

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。在我遇到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");

}