

# 人脸考勤管理系统

## 设计说明书

美亚柏科 2021 刘祥南创业基金夏令营

2021 年 7 月

---

# 目录

- 一、 需求分析 ..... 1
- 二、 总体架构方案 ..... 2
  - 2.1 物理架构设计 ..... 2
  - 2.2 系统架构设计 ..... 2
  - 2.3 前端架构设计 ..... 4
  - 2.4 后端架构设计 ..... 4
  - 2.5 前后端通信模式设计..... 5
  - 2.6 摄像头部署与交互设计 ..... 6
- 三、 功能模块与业务流程设计..... 7
  - 3.1 功能模块设计 ..... 7
  - 3.2 业务流程设计 ..... 8
- 四、 数据库设计 ..... 10
- 五、 核心接口设计 ..... 13
- 六、 创新模块设计 ..... 15
  - 6.1 打卡提醒小程序模块设计..... 15
  - 6.2 人脸与形象检测模块..... 16
- 七、 总结..... 19

## 一、需求分析

本次的项目为人脸考勤管理系统。需求是为公司提供一套完整的人脸打卡与考勤管理的流程。具体的需求分析划分如下：

1. 为员工个人或者人力资源部门提供员工的注册功能，注册信息包括但不限于姓名、工号、部门、邮箱以及正面清晰无遮挡的脸部照片。
2. 为人力资源部门提供员工注册信息的查询、更改和删除功能，方便其对整个公司的员工进行一个信息管理。
3. 为员工提供每天上下班的人脸识别打卡，记录打卡时间，打卡照片和打卡类型等信息。
4. 通过微信小程序向员工定时的发送打卡提醒，并可让员工随时查看自己的历史打卡信息。
5. 根据员工的打卡信息，统计相关的考勤信息，包括但不限于迟到、早退、缺勤、旷工、灵活打卡、加班情况等信息。可为人力资源部门对员工奖惩提供依据。
6. 记录员工的异常着装情况和个人形象情况，包括但不限于衣领、裤长、发色等信息。
7. 根据员工打卡时的表情推测员工的心理状态，供人力资源部门进行针对性的关怀和帮助。
8. 将考勤信息通过邮件发送给员工个人或者员工领导，以起到督促作用。

## 二、 总体架构方案

### 2.1 物理架构设计

人脸考勤管理系统的物理架构主要由服务端、PC 端、移动端与 IPC 端组成。首先服务端与 IPC 端通过以太网连接，实现人脸照片的捕捉与上传。服务端包含了本地服务器与阿里云服务器，二者以 Internet 连接，实现员工图片的上传与分析结果回传。服务端、PC 端与移动端通过一个无线网络连接，实现用户与服务端的数据交换。物理架构如图 1 所示。

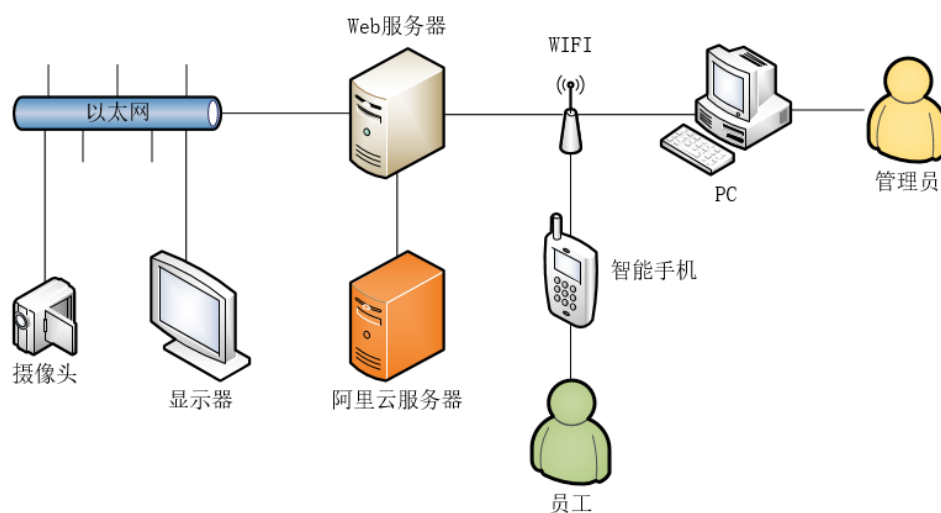


图1. 物理架构图

### 2.2 系统架构设计

本系统采用了前后端分离的架构模式，是对以往 MVC 架构的一种改进，SpringMVC 不再返回一个视图，而是直接返回 JSON 格式的交互数据。前后端部署于不同的服务中，构建工具也各不相同，基于 HTTP 协议进行交互。前端方面负责数据的收集和展示、监听处理用户事件、对 DOM 元素进行操作、向

后端发送请求并将响应数据渲染到页面中；后端则专注于对请求的处理、对业务逻辑的封装、与数据库的交互、提供良好的服务接口等工作。前后端通过 HTTP 协议和 JSON 格式的报文进行数据交互，各自分工明确，能够高效的协作，提升开发的专业性，降低维护成本，两方的架构也有了更好的伸缩性和扩展性，利于技术的不断深入。系统总体架构如图 2 所示。



图2. 系统总体架构图

## 2.3 前端架构设计

员工考勤管理系统的前端架构如图 3 所示，系统的资源文件均保存在 wamp 服务器上，当用户在自己的设备（手机、平板、PC）浏览器中输入员工管理系统的网址后，浏览器会自动解析网址定位网络资源并和服务器建立 HTTP 连接，连接建立之后服务器会把网络资源传输到用户浏览器，浏览器接收到网页资源之后将之构建为 DOM 树、CSSOM 树、渲染树，并根据渲染树计算布局，将所有网页元素渲染到屏幕上。

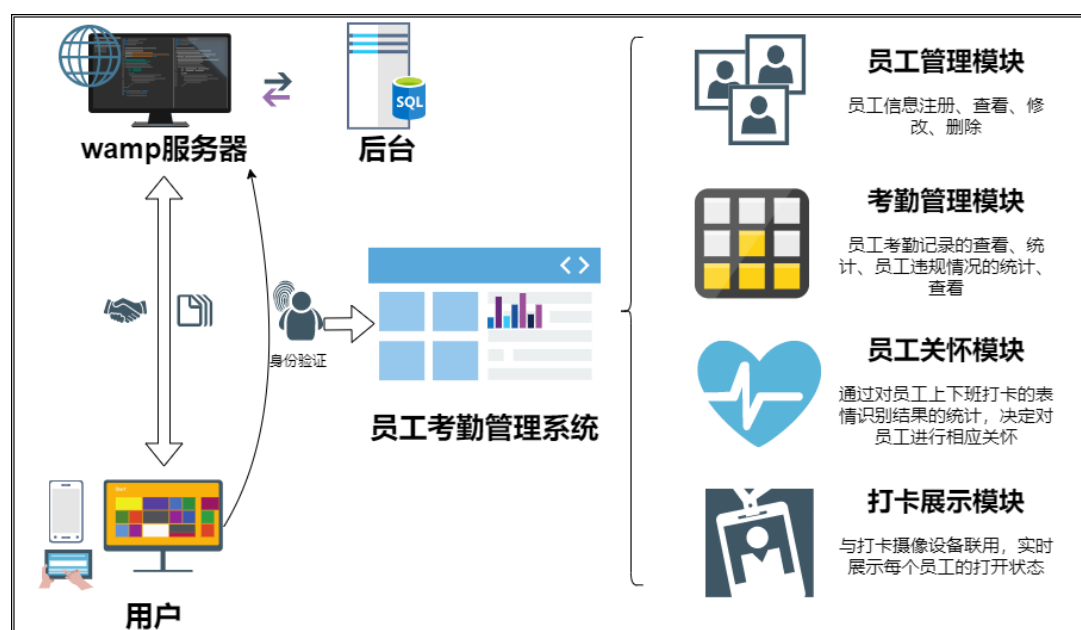


图3. 前端架构图

## 2.4 后端架构设计

系统后端架构如图 4 所示。系统采用 SpringMVC 中的分层控制结构，将后端分为控制层、业务层和持久层。Controller 层负责控制业务流程，需要基于 Service 层的接口，主要负责前端请求的接收和数据返回，是前端交互的中间控制层。Service 层主要对业务逻辑进行设计和处理，需要基于 Repository 层中定义的接口。Service 层对业务逻辑的封装可以较大程度增强通用业务逻辑的复用

性，提高开发效率。Repository 层使用了 Mybatis 来完成数据持久层的工作。这避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集。MyBatis 可以使用简单的注解来配置和映射原生信息，将接口和 Java 的对象映射成数据库中的记录。

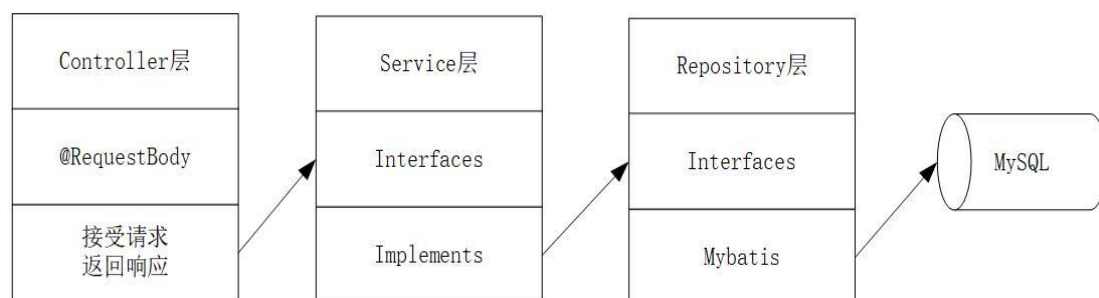


图4. 后端架构图

## 2.5 前后端通信模式设计

系统使用了 Layui 框架作为 View 层渲染数据，故较于传统的 SpringMVC 方式，系统将 View 层从后端中抽离，后端结合 SpringBoot 框架只完成控制层和 Model 层的业务逻辑，提供接口为前端返回数据。前后端基于 HTTP 协议进行数据通信，前端使用 Ajax 发送请求，后端控制层通过 @RequestMapping 注解配置请求映射处理类，并使用 @RequestBody 将请求中的 JSON 报文与 Java 对象进行转换，做相应的业务处理。前后端通信模式如图 5 所示。

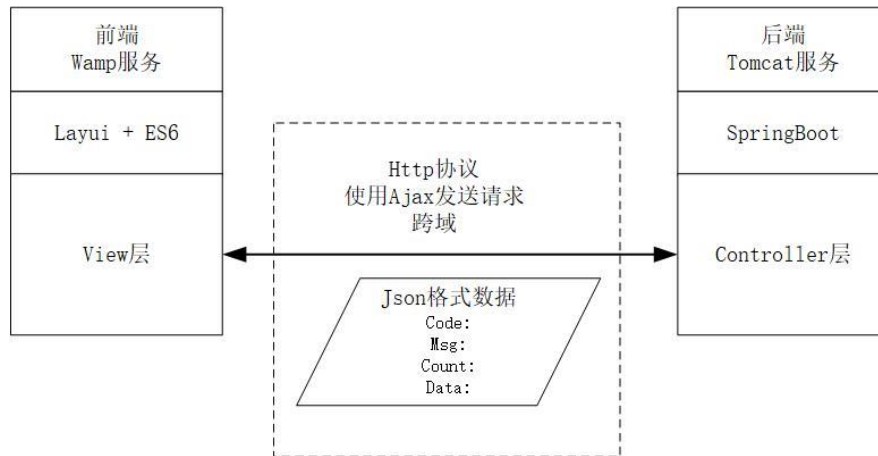


图5. 前后端通信模式

## 2.6 摄像头部署与交互设计

本系统相较于传统的前后端交互系统，特殊点在于与 IPC 的数据交互，交互设计如图 6 所示。IPC 捕捉人脸图像后，将图片文件通过 FTP 协议上传到后端的 FTP Server。由于 FTP Server 是集成到 SpringBoot 中的，因此当每一张图片传输完成后，后端通过回调接口获取图片的保存路径，并进行接下来的业务处理。

为了保证人脸打卡时的实时视频显示，前端的 View 层需要解析 IPC 的实时 RTSP 视频流并播放。而打卡成功后前端通过 HTTP 协议和长轮询即时的获取打卡信息并进行显示。

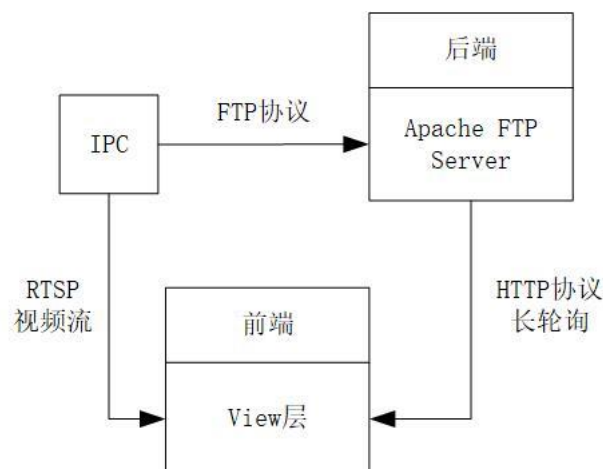


图6. 摄像头部署与交互设计图



### 三、 功能模块与业务流程设计

#### 3.1 功能模块设计



图7. 功能模块图

人脸考勤管理系统功能模块如图 7 所示。总共包含四个模块：员工管理模块、考勤管理模块、员工关怀模块以及打卡状态展示模块。其中员工管理模块包括员工信息注册、查看、搜索、修改、删除等功能；考勤管理模块包括 5 个功能面板，打卡记录面板可以搜索查看员工打卡信息，考勤统计面板可以统计员工考勤状态包括指定月/年正常上班总次数、缺勤次数、迟到次数、早退次数以及灵活打卡次数，加班榜面板可以统计指定时间段内的加班时长并将 top10 绘制天梯，考勤异常面板包括指定时间段内迟到和早退员工列表，着装异常面板可以统计仪容仪表违规的员工（包括发色不合格、衣领不合格、裤装不合格的员工信息）；员工关怀模块可以统计所有员工每天打卡时的表情状态分布，根据一段时间内员工的表情状态分布，管理人员可以有针对地发放福利、弹性工作时间，从而更好地挖掘员工潜能；打卡展示模块和摄像头设备匹配，一般和

考勤管理系统分离，面向的用户主要是员工，让员工通过摄像头打卡之后，可以实时看到自己的打卡状态（包括打卡时间、迟到早退信息、发色/衣领/裤装等仪容仪表检测信息等）。

## 3.2 业务流程设计

### A. 用户注册

1.用户通过微信小程序自行注册：用户打开微信小程序，输入姓名、工号、职位、邮箱信息并选择五官清晰无遮挡的脸部照片，信息填写完毕后提交，上传至服务器，服务器接收到数据后保存至数据库并学习人脸信息。

2.HR 通过员工考勤管理系统注册员工信息：HR 使用用户名和密码登录员工考勤管理系统，通过员工管理模块的员工注册功能填写员工信息并上传员工照片至数据库。

### B. 员工信息管理

1.HR 可以通过员工管理模块管理员工：HR 可以在员工考勤管理系统中查看（点击头像可显示大图、也可以使用关键字快速查找员工）并修改员工信息（包括修改、删除）。

### C. 员工打卡

1.小程序提醒功能：小程序在每天固定时间向员工推送信息，提醒员工按时打卡。

2.员工通过打卡摄像头进行上下班打卡：与摄像头配套的现实界面会实时反馈并展示打卡信息，包括迟到、早退的时长、个人信息、打卡成功的具体时间以及仪容仪表检测结果（包括发色、衣领、长裤、表情）。

3.邮件反馈打卡结果：员工打卡成功之后，后台将打卡信息记录到数据库

并向用户发送邮件反馈打卡结果，邮件内容包括基本打卡信息以及一句根据员工打卡时的表情个性推荐的名言警句。

#### **D. 考勤管理**

1.HR 通过管理系统查看打卡记录：管理系统的考勤管理模块中的打卡记录页面会展示每一条打卡记录，HR 也可以通过时间或员工信息快速筛选打卡记录。

2.加班统计：后台根据每天上下班打卡时间推算出每个人的加班数据，管理系统中考勤管理模块的加班榜页面会将最近一个月的加班情况展示出来，并将加班时长top10 的员工通过天梯凸显。HR 也可以根据时间筛选加班数据。

3.着装异常统计：考勤管理模块的着装异常页面会统计并展示每个员工的着装违规行为，包括衣领 不合格、裤装不合格、发色不合格，并以饼图的形式展示不同违规行为的比例，HR 可以针对高比例 的违规行为发布公告，提醒员工注意仪容仪表。

4.考勤异常统计：考勤管理模块中的考勤异常页面会记录所有员工一个月内（可指定）的正常上班、迟到、早退、旷工以及灵活打卡次数（灵活打卡，是根据美亚柏科公司深度定制的功能。指员工若在 8 点 40至9 点之间上班打卡，且当日工作时长大于 9小时，则视为正常打卡。每月有 10 次机会 灵活打卡），HR 可以根据这些数据调整工资或警告懈怠的员工。

## 四、 数据库设计

本系统根据业务的持久化需求，设计了 5 张数据表，用于为员工管理，人脸打卡，考勤统计等一系列业务需求提供可靠的持久化支持。数据库设计 E-R 图如图 8 所示。其中管理员信息表用于管理员进入系统的登陆验证。员工信息表用于保存公司员工的所有信息。打卡信息表用于保存员工每一次打卡的记录。考勤统计表用于统计保存员工的考勤情况，该表以月为单位，每月初自动生成。情绪信息表用于保存对于不同情绪状态进行推送的名言警句等信息。

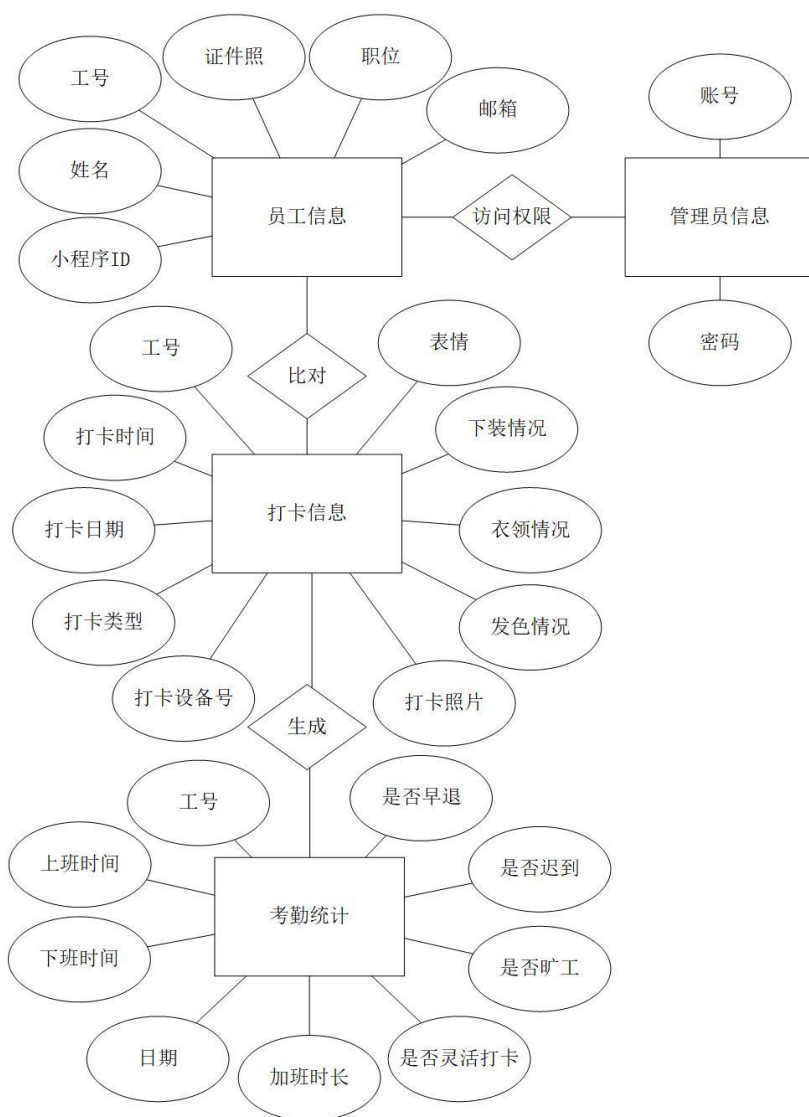


图8. 数据库设计 E-R 图

详细的数据表设计如表 1-表 5 所示：

表1.管理员信息表

Name	Type	Length	Nulls	Desc
username	varchar	30	YES	管理员用户名
password	varchar	30	YES	管理员密码

表2.员工信息表

Name	Type	Length	Nulls	Desc
usr_name	varchar	20	NO	姓名
usr_ID	varchar	20	NO	工号
usr_pic	varchar	255	NO	照片
usr_job	varchar	30	YES	职位
email_address	varchar	30	YES	邮箱
usr_openid	varchar	255	YES	小程序 ID

表3.打卡信息表

Name	Type	Length	Nulls	Desc
usr_ID	varchar	30	NO	工号
check_time	time		YES	打卡时间
check_date	date		YES	打卡日期
check_type	int	11	YES	打卡类型
device_num	int	11	YES	设备号
check_pic	varchar	255	YES	打卡照片
hair_color	varchar	30	YES	发色
low_cloth	varchar	30	YES	下装
expression	varchar	30	YES	表情
collar_round_neck	int	1	YES	衣领

表4.考勤统计表

Name	Type	Length	Nulls	Desc
usr_ID	varchar	32	NO	工号
check_in_time	time		YES	上班时间
check_out_time	time		YES	下班时间
check_date	date		YES	打卡日期
isBeLate	int	1	YES	迟到
isLeaveEarly	int	1	YES	早退
isFlexible	int	1	YES	灵活打卡
isNormal	int	1	YES	正常上班
isOut	int	1	YES	旷工
workHours	double	(10,2)	YES	工作时长
workOverHours	double	(10,2)	YES	加班时长
expression	int	2	YES	表情

表5.情绪信息表

Name	Type	Length	Nulls	Desc
expression	varchar	30	YES	表情
statement	varchar	255	YES	名言警句

## 五、 核心接口设计

本系统采用前后端分离的架构，后端采用 SpringBoot 框架，由 Controller 层负责前端请求的接收和数据返回。因此接口设计对于前后端的交互与业务的实现至关重要。本系统的核心接口设计如表 6-表 9 所示。

表6.员工注册接口

FUN	addUser
anno	员工注册
Parameter	usr_name (String, 必须) usr_ID (String, 必须) usr_pic (String, 必须) usr_job (String, 不必须) email_address (String, 不必须)
Return	成功: 1, 失败: 0
RequestMethod	POST

表7.打卡记录查询接口

FUN	selectChecker
anno	打卡记录查询
Parameter	select_str (String, 不必须) startDate (String, 不必须) startDate (String, 不必须) page (Integer, 必须) limit (Integer, 必须)
Return	根据所给参数进行搜索排序分页后的打卡记录
RequestMethod	GET

表8.加班排行查询接口

FUN	select_WorkHours
anno	加班排行榜
Parameter	select_month (String, 必须)
Return	所选月所有人的加班情况
RequestMethod	GET

表9.形象异常查询接口

FUN	select_TotalAbnormal
anno	形象异常记录查询
Parameter	select_month (String, 必须) abtype (String, 必须)
Return	所选月所有打卡异常的记录
RequestMethod	GET



## 六、 创新模块设计

### 6.1 打卡提醒小程序模块设计

#### A. 注册功能：

用户在小程序填写需要的注册信息以及需要上传的照片统一发送给后端服务器的 user/add 接口，同时将当前用户在小程序的唯一 openid 上传。后端服务器将用户注册信息，照片，openid 一起储存在 sql 数据库中。注册成功返回 1，在小程序中弹窗提示用户注册成功，跳转至 index 界面，若数据库中已存在用户的 openid，则会返回 -1，在小程序中提示用户已存在。若数据请求出错则会在控制台打印错误信息，并在小程序界面中提示注册失败。

#### B. 查看考勤历史功能：

点击“考勤历史”按钮小程序将当前用户的 openid 发送至后端服务器 select\_str 接口，后端服务器根据 openid 去数据库中当前 openid 下用户得日常考勤记录，以 json 数组得形式返回给小程序。小程序解析完成之后对得到得 json 数组进行遍历在界面中按照时间先后的顺序显示。

#### C. 日常打卡提醒功能：

日常打卡提醒功能依托腾讯云后台开发，通过定时器触发，在每天上班和下班前 5 分钟对用户进行订阅消息推送，提醒用户打卡。以上班时间为例，每天 8:35 分云后台得 timer 函数通过定时器触发，通过 request-promise 方式去请求后端服务器 selectAll 接口，后端服务器接收到请求后以 json 数组得形式返回数据库中所有注册用户的信息。云后台解析接收到的信息将用户信息中的 openid 作为参数传递给同样依托在云后台的 sendMsg 函数，该函数调用 subscribeMessage.send 接口将定制的订阅消息发送到用户的微信服务通知窗口中，用户就能接收到打卡提醒。

#### D. 异常打卡提醒:

异常打卡提醒与日常打卡提醒有相似之处：都依托微信云后台开发，都通过定时器触发，都需要在开始阶段通过 request-promise 方式访问外部后端服务器的接口以获取订阅用户的信息。二者不同的是异常打卡提醒在获取了每一位用户的 openid 后需要再一次访问外部的后端服务器，查询当前 openid 下的用户是否已经打卡，若已经打卡则不作提醒，若还未打卡则会再次提醒用户进行打卡。还是以上班时间为例，在 8:39 分时通过定时器触发 errorRemind 函数，在接收到后端服务器返回的用户信息后，函数将每一个用户的 openid 上传到后端服务器的 isCheckIn 接口，若该用户已打卡，则服务器返回数字 1，云后台函数不作提示，若未打卡，则服务器返回数字-1，云后台将该用户的 openid 作为参数传递给 sendMsg 函数，通过 sendMsg 函数推送订阅消息提醒用户打卡。在 41 分时会再次触发 errorRemind 函数，此时若用户还未打卡，则会推送“您已迟到，请及时打卡”提醒用户。

## 6.2 人脸与形象检测模块

该模块任务是通过阿里云的视觉智能开放平台，以及自己训练的神经网络模型来识别图片中的人脸、发色、表情、衣领、下装。主要设计流程为当摄像头完成拍摄全身照并上传图片到 ftp 服务器上时，通过阿里云人脸识别功能将该图片云人脸数据库进行比对，从而识别人脸。通过阿里云的头发分割功能获得全身照的头发抠图，再通过阿里云的颜色识别功能判断头发的颜色种类。通过阿里云的表情识别功能全身照中的表情种类。通过阿里云的人体结构化属性识别全身照中的下装种类。最后用自建神经网络模型识别全身照中的衣领。

#### A. 人脸识别:

首先需要在人脸数据库中添加数据，即样本 ID 和人脸图片。当摄像头拍摄完全身图片后，我们调用阿里云人脸搜索 1:N 功能。用图片生成 imgUrl 并通过 http 通信与数据库中的人脸图片进行比较。比较完成后返回相似度排名前三

的 Json 字符串，如果相似度最高的 Score 值大于 0.4，那么我们认为人脸比对成功并得到样本 ID。

### **B. 发色识别：**

当摄像头拍摄完全身图片后，我们调用阿里云的头发分割功能。用图片生成 imgUrl 并上传到阿里云中。解析返回 Json 中的 imgUrl 路径，并将头发抠图保存到本地。接着调用阿里云的颜色识别功能，同样的将头发抠图生成 imgUrl 并上传到阿里云中，解析返回的 Json 字符串中的 color 键从而判断发色。颜色的分类共 11 种：black（黑）、white（白）、gray（灰）、red（红）、orange（橘）、yellow（黄）、green（绿）、cyan（青）、blue（蓝）、purple（紫）、magenta（紫红）。当发色不是黑或灰色时，认为该图片中的发色是异常的。

### **C. 表情识别：**

当摄像头拍摄完全身图片后，我们调用阿里云的表情识别功能。用图片生成 imgUrl 并上传到阿里云中。解析返回的 Json 字符串中的 expression 键从而判断发色。表情的分类共 7 种：neutral（中性）、happiness（开心）、surprise（惊喜）、sadness（悲伤）、anger（愤怒）、disgust（厌恶）、fear（害怕）。

### **D. 下装识别：**

当摄像头拍摄完全身图片后，我们调用阿里云的人体结构化数据功能。用图片生成 imgUrl 并上传到阿里云中。解析返回的 Json 字符串中的 lowerWear 键从而判断发色。下装的种类共三种：Trousers（长裤）；Shorts（短裤）；Skirt&Dress（裙子）。当下装为短裤时，我们认为该图片的下装是异常的。

### **E. 衣领检测：**

自建神经网络模型是用 python 语言的 keras 框架搭建，采用的是 2015 年有何凯明提出的 ResNet 网络结构。该神经网络由 18 层卷积层构成，并采用了短路机制和残差块，从而有效地解决了退化问题。我们从网上爬取了 772 张立领图片，以及 809 张无领图片，将它们处理为（32，32，1）形式后作为训练集，从而做二分类问题。以 64 为 batch，训练了 25 个 epoch 后，验证集的准确率大约在 80%。为了避免每次预测衣领时都加载模型，还利用 flask 框架搭建了服务端和客户端。对模型初始化并加载在服务端，在客户端中将拍取到的

全身照片根据图片比例截取衣领部分图片后，压缩成字节流发送到服务端进行判断。判断完成后会在客户端的控制台中输出 0 或 1。最后在 SpringBoot 框架中，当需要进行衣领识别时运行客户端脚本，并在 SpringBoot 获取客户端控制台输出。

## 七、 总结

本系统的设计开发旨在为公司员工提供一整套的人脸识别打卡管理系统。我们最初拿到的需求是实现员工管理，人脸打卡，记录查询。随着开发过程的推进，我们意识到简单的人脸打卡是缺乏创新性的，只有着单一的打卡功能，但实际上一个公司对于员工工作所需的信息却远远不止于此。因此，我们进行了第一步创新，考勤统计分析，其中最有点的是加班排行和灵活打卡机制。前者有助于对努力工作的员工给予奖励同时提高员工的工作积极性，后者体现了公司对员工的包容，人性化的打卡机制反而会提高员工的工作积极性。

在这些做完以后，我们又发现其实人脸打卡捕捉到的照片远不止人脸信息，我们还可以从中提取到更多的信息，比如发色信息，衣领信息和下装信息，这可以帮助公司提高员工的着装规范。最后我们也可以提取员工的表情信息，针对这一点受限于研发时间，我们并没有深入研究。但事实上员工的心理状况对于公司管理是至关重要的信息，在员工心理出现异常时，即时发现并提供帮助是十分有意义的。我们的系统捕捉员工每天上下班的情绪状态，可以提供给公司进行大量的定制化分析服务。

一个月开发时间很短暂，但我们团队精诚合作，快速的学习并进入了开发状态，及时完成了预定目标并在一些点上进行了创新，让打卡不仅仅是打卡。十分感谢美亚柏科公司给予我们机会参与到暑期夏令营活动中，十分感谢严建亮老师给与我们技术上的指导与建议，十分感谢公司的 HR 吴锰龙先生，吴程璐女士对我们在公司生活上的帮助。最后，希望我们团队的每个成员能够继续努力，在即将到来的秋招中斩获佳绩。