1、**结题报告书**

请实习学生在企业指导教师的指导下认真填写《计算机科学与技术系卓越工程师计划"工程实习"结题报告书》，由企业导师在基本信息表页中给出评语，评定分数。

2、**成果说明材料**

提交项目的成果说明材料电子版一份（包括调研报告、实验系统目标、实验系统设计、系统实现说明、系统性能评估、设计产品及成果实物的照片等；如有发表论文或申请专利请将论文、专利证书的扫描件附送，已投稿的论文和正在审理过程中的专利请将论文投稿刊物、专利申请进度等信息写出说明，并有导师签字。

3、**提交所有材料的电子版、基本信息表打印页**

文件名以“选课序号-学生姓名-项目名称”的格式命名。在《结题报告书》中列举的项目成果应与报送的项目成果材料附件严格一致，个别不能以附件形式报送的成果材料，请附导师签字的证明。

提交本报告第3页打印版并签字。

所有材料提交给计算机系工程实习课程任课教师。

**计算机科学与技术系**

**卓越工程师计划“工程实习”**

**结题报告书**

项目名称： 无感考勤

学生姓名： 岳宇轩

企业导师： 魏跃跃

二○二三 年 三 月 三十一 日

中国海洋大学信息科学与工程学部

填 表 说 明

一、 本结题报告适用于在企业进行实习的学生项目总结，并作为“工程实习”课程材料和成绩依据。

二、填写结题报告前，请先征求企业导师意见。报告书的各项内容要实事求是，逐条认真填写。表达要明确、严谨。

三 、在本报告书后，可附上与项目有关的研究资料、结果和证明材料（论文、实验报告、专利申请等）的电子版，或扫描件。

四、“一、基本信息表”须打印，本人签字，企业导师评分并签字。

五、提交本结题报告电子版和“基本信息表”打印页。

**一、基本信息表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1、学生情况** | | | | | | | |
| 姓 名 | 岳宇轩 | | 性 别 | 男 | | 出生年月 | 2001.6 |
| 学部、院(系) | 信息科学与工程学部计算机科学与技术学院 | | 专业年级 | 19计算机科学与技术 | | 学 号 | 19020011038 |
| E-mail | **yueyuxuan01@163.com** | | | | | 联系电话 | **13697680587** |
| **2、企业导师情况** | | | | | | | |
| 姓 名 | 魏跃跃 | 企业/部门 | | 青岛网信信息科技有限公司/研发部 | | | |
| 职称/职务 | 大数据高级工程师 | E-mail | | | weiyueyue@wxdata.cn | | |
| **3、项目基本信息** | | | | | | | |
| 项目名称 | **无感考勤** | | | | | | |
| 摘要（200字以内） | | | | | | | |
| 本项目主要基于企业考勤的基本原理，运用jeecg-boot框架搭建一个动态的企业考勤信息管理网站。它是在Windows系统环境下，以Mysql数据库开发平台，采用java技术开发的企业考勤信息管理网站。本系统分为前端与后端两部分组成。前端主要针对企业管理人员，前端包括了设备管理、订阅管理、人员管理、部门管理、班次管理、抓拍记录；后端主要用于接收摄像头的订阅上报信息，处理与数据库的交互。 | | | | | | | |
| **学生签名:岳宇轩**  **日期：2023.03.31** | | | | | | | |
| **4、企业导师评语：** | | | | | | | |
| 见实习证明（第23、24页）  **评分：97 企业导师签名：**魏跃跃  **日期：2023.03.30** | | | | | | | |
| **5、备注：** | | | | | | | |

**二、"工程实习"项目研究成果简介（要求撰写1500字以上）**

|  |
| --- |
| **请按照以下提纲进行撰写**   1. 项目背景   在现代企业中，考勤管理是一项非常重要的工作。通过对员工的考勤管理，企业可以更好地了解员工的工作情况，合理安排员工工作时间，并及时发现考勤异常情况，从而保证企业的正常运营。然而，传统的考勤管理方式存在着许多不足之处，如考勤效率低、人力成本高、考勤信息不及时等问题。因此，开发一种高效、智能化的考勤管理系统是非常有必要的。本项目基于 AI 智能人脸识别摄像头，搭建了一套智能化的企业无感考勤管理系统。  2. 项目研究目标  本系统基于B/S模式，用于满足企业管理人员考勤需求。从企业管理者的角度出发，以高效管理、准确管理需求为原则，要求本系统满足以下要求：   * 统一友好的操作界面，具有良好的用户体验； * 用户的登录功能； * 用户必须登录以后才能进行人员、班次、部门的管理； * 用户必须登录以后才能进行设备、订阅的管理和查看抓拍信息； * 后台需要及时、正确的接收摄像头上报的订阅信息； * 后台需要正确处理订阅上报的信息，前台要能准确展示考勤信息； * 后台能够管理各项基本数据，实现与数据库正确的交互； * 系统运行稳定，响应及时。   3. 项目研究的主要内容  根据无感考勤需求，系统分前台业务和后台业务。前台业务是针对企业管理者使用，主要要求对设备、订阅信息、人员信息、部门信息、班次信息进行在线管理，同时可以查看考勤结果。其功能模块如下图所示：  无感考勤前台  企业管理  考勤管理  用户登录  用户退出  人员管理  部门管理  抓拍记录  班次管理  设备管理  订阅管理  展示图表  后台业务是针对无感考勤系统管理员用户使用，管理员登录后台系统后可以对系统进行管理，包括：设备管理，订阅管理，人员管理，部门管理，班次管理。其功能模块如下图所示：  无感考勤后台  设备管理  人员管理  部门管理  增删改查设备  订阅管理  增删改查订阅  增删改查人员  增删改查部门  班次管理  增删改查班次  此外，后台还应该实现如下功能：   1. 接收A47 摄像头的上报订阅信息； 2. 正确处理接收到的订阅信息：将抓拍到的人员信息与人脸库中的数据进行匹配，并按照班次管理中的信息进行检查，记录相应的考勤状态； 3. 将抓拍信息储存至数据库中；   4. 项目研究的方法和步骤  项目采用了 Jeecg-boot 技术框架作为基础，使用 Java 语言进行开发，使用IntelliJ IDEA 2021.3 作为开发平台。后端整体采用 SSM 框架，并使用SpringMVC，Hibernate+Minidao 等技术；前端使用 HTML+Vue.js+jqgrid 进行页面展示，使用 axios 发送异步请求；数据库使用MySQL，并使用Navicat 在开发过程中进行可视化监控。  5. 项目成果（该项目研究已达到的目的、内容、成果及论文、专利等）  前端部分：  首先进入登录页面，输入用户名、密码和验证么（点击验证码会更换），点击登录    进入主页面后，点击左侧的设备管理，进入设备管理页面：    设备管理页面展示了所有已创建的设备的各项信息，如设备类型，设备IP，在线状态等。点击新增按钮可以添加设备，点击右侧操作按钮可以编辑信息，“更多”中可以进行查看和删除操作。  点击左侧导航栏中订阅管理进入订阅管理界面    订阅管理页面同样对订阅信息进行了展示，可执行的操作有对订阅的增、删、查、改。其中，订阅的IP信息和端口号信息，需要与后台进行对应，因为这是订阅上报的请求 url 组成部分。点击右侧“更多”按钮，可以选择“发送订阅”、“刷新订阅”和“删除订阅”。点击“发送订阅”、“刷新订阅”后，系统会自动根据请求的 response 解析出订阅的 ID ，开始时间和结束时间，并在前端界面予以展示。  点击左侧导航栏人员管理进入人员管理界面：    此处可以对企业的人员进行增删查改。点击右侧“更多”，可以选择将人员上传至人脸库中。  点击左侧导航栏部门管理可以对企业的部门信息进行管理：    点击左侧导航栏班次管理可以对企业员工的班次进行调整：    点击导航栏左侧的抓拍信息，可以看到摄像头的订阅上报内容：    此处的抓拍信息是我们将摄像头捕捉到的信息与人脸库中人员的身份证进行匹对后的结果，但是在“状态”一栏我们将人员与其班次信息进行了比对，显示为不同的状态。点击“查看统计信息图表”可以查看订阅商报的图表信息：        后端部分：  员工管理：    抓拍信息管理：    班次管理：    订阅管理：    设备管理：    部门管理：    6、项目特点与创新   1. 该项目是一套标准的企业考勤管理系统，具有完备的功能； 2. 基于JAVA语言设计，可以跨平台应用； 3. 项目整体采用Jeecg-boot 平台进行开发，集成度高，代码量低； 4. 注重与用户的交互，具有美观的UI界面； 5. 项目采用 A47 智能摄像头，搭配AI人脸识别功能； 6. 项目整体架构灵活，具有较高的可扩展性，方便后期的维护；   7、项目不足与设想   1. 对于登录后 token 的存取问题，我们没有实现很好的解决。可以尝试使用redis缓存技术，将数据缓存到内存中。或者采用一些集成的框架，将 token 信息存储至运行缓存中。 2. 项目的抓拍信息展示不够灵活，只能以固定格式的图表进行订阅上报信息的展示。可以考虑采用动态的方式进行更加灵活的操作。 3. 参加工程实习的总结、体会、存在问题和建议   我在本次工程实习中负责全部与摄像头和人脸库有关的内容：   1. 设计数据结构 2. 测试A47相机说明文档中与本项目有关的接口 3. 设备管理、订阅管理、抓拍信息的前端页面设计和实现 4. 与企业指导老师交流问题 5. 人员信息上传至人脸库 6. 前端发送订阅、刷新订阅、删除订阅的功能实现 7. 后端处理摄像头的订阅上报请求，以及实现后端与MySQL之间的交互 8. 协助组员完成人员、统计信息部分的开发 9. 项目整合，测试和debug工作 10. 在项目汇报中展示项目内容 11. 上传项目至GitLab   下面是工作内容展示：   1. 设计数据机构：     2.测试A47相机说明文档中与本项目有关的接口    3.设备管理、订阅管理、抓拍信息的前端页面设计和实现  ①设备管理    ②订阅管理    ③抓拍信息    4.与企业指导老师交流问题      5.人员信息上传至人脸库  主要核心代码见后附代码1  6.前端发送订阅、刷新订阅、删除订阅的功能实现  主要核心代码见后附代码2  7.后端处理摄像头的订阅上报请求，以及实现后端与MySQL之间的交互  主要核心代码见后附代码3  本次实习极大的提高了我的工程实践能力。虽然之前在学校里的软件工程、软件系统项目与分析等课程上，均有过工程项目的开发经历，但是此次规格更高，要求也更多，也需要掌握新的知识与框架。下面简单概述一下此次实习经历的一些感悟：  刚接到项目的时候是在寒假期间，在听完企业指导老师介绍过项目需求之后，我和吕筱玮决定负责项目最关键的摄像头部分。在还没有上手的时候，我认为这是一项非常非常复杂的工作，于是我一直有点懈怠。光是 Jeecg-boot 基础框架项目的配置，都有很多篇幅的说明文件。但当我按照网站的说明文档一步一步配置直至成功后，我也逐渐增强了信心。在配置初始项目的过程中遇到了很多问题，但最终大多是因为没有认真阅读文档导致的。此后的学习过程中，在接触到一个新的，陌生的框架时，不要惧怕，先仔细按照说明文档进行配置和学习。  A47摄像头相关功能的开发是本次项目的核心内容。我在假期便与吕筱玮一同进行了一次线下的开发交流。我们首先尝试获取摄像头的IP地址，通过路由器的管理页面可以成功获得。随后，我们使用 Postman 工具按照 A47 相机接口文档中的说明，逐个调试项目需求中涉及到的接口，并理清了本次项目的相机工作流程，即：新增人员→订阅下达→摄像头检测→订阅上报。  我们首先测试了订阅的增删查改接口，均没有什么问题。但是在新增人员的时候，我们遇到了问题。我们按照接口文档中的说明，进行了多次排查，无果后寻求企业的指导教师魏工，魏工向我们说明了在新增人员的接口中，需要上传人脸图像的 base64 编码与编码长度需要按照字符数对应一致，且 base64 编码前不能有通用的前缀。我们按照魏工所述的方案修改后，成功测试了人脸库中人员的增删查改接口。  随后，我们在订阅上报接口的理解上出现了问题。魏工指导我们说，我们在订阅下达的时候用参数传递的 IP 和端口号，即是订阅上报的请求 url。 摄像头在识别到人脸后会自动向这个 url 发送数据。最终，我们在魏工的指引下成功测试了摄像头的人脸识别和上报订阅功能，并在后端完成了对订阅上报信息的初步解析。  返校后，我们对项目进行了进一步的开发。由于环境的变化（校园网环境），我们在宿舍中配置路由器、摄像头等设备时遇到了一系列的问题，比如路由器无法正常联网，摄像头看不到实况图像等。前者我们通过更改路由器配置方式解决，后者在请教魏工后，登录校园网账户并清除多余设备解决。此外，我们还遇到了跨域连接的问题，我们尝试了很久没有解决，最终通过 Chrome 的跨域配置插件 CORS 实现了在 Chrome 浏览器上的请求访问和response 接收。随后企业的指导老师在实习群中说明了配置跨域的方式，我们也按照说明进行了修改。有关摄像头的另一个主要部分是需要与数据库进行交互，因为我们需要去数据库中查询人员信息和班次信息，然后将抓拍到的信息与人员的信息和人员的班次信息进行匹配。  此外，我还帮助负责其它部分的同学进行开发，如帮助负责人员管理功能的同学开发了将人员信息上传至人脸库的功能，帮助统计信息展示的同学实现了部分抓拍信息的展示功能等。  在此次工程实习的项目中，我学会了对jeecg-boot 框架的基本使用操作，包括 Online 表单在线生成，在线图表的生成，导航栏设置等等；此外，我也巩固了使用 Js进行前端开发的一些知识，比如 Vue 的编写，axios 请求等等；同时，我也对 Java 编写后台程序有了初步的认知，包括如何构建、结构 json 格式文件，如何接收一个 post 请求，如何嵌入 sql 语句与数据库交互等等；我也对一些更为细节的操作，比如 utc 时间戳和本地时间格式的转换，base64和图像文件的转换等操作有了一定的掌握。 |
| 2-岳宇轩 |

**代码1：**

addtofaceku(record,t2) {  
 ***console***.log(record.id);  
 const data = {  
 Num: 1,  
 PersonInfoList: [{  
 PersonID: 0,  
 LastChange: 1610760604087,  
 PersonName: record.personname,  
 Gender: 0,  
 Birthday: "2021-01-14",  
 Region: {  
 Nation: "1111",  
 Province: "2222",  
 City: "3333"  
 },  
 TimeTemplate: {  
 BeginTime: 0,  
 EndTime: 4294967295,  
 Index: 0  
 },  
 IdentificationNum: 1,  
 IdentificationList: [{  
 Type: 0,  
 Number: record.identificationnum  
 }],  
 ImageNum: 1,  
 ImageList: [{  
 FaceID: 0,  
 Name: "",  
 Size: t2.length,  
 Type: 1,  
 LastChange: 1610760604087,  
 Data: t2,  
 }]  
 }]  
 };  
 ***console***.log(data);  
  
 ***axios***.post('http://' + ***IP*** + '/LAPI/V1.0/PeopleLibraries/1/People', data)  
 .then(  
 response => {  
 ***console***.log(response);  
 })  
 .catch(error => {  
 ***console***.log(error);  
 });  
},

**代码2：**

updateDatabase(record, createID, CurrentTime, TerminationTime) {  
 ***axios***.put('http://localhost:8080/subscribe/subscribe/edit', {  
 id: record.id,  
 createid: createID,  
 currenttime: CurrentTime,  
 terminationtime: TerminationTime,  
 }, {headers: {"X-Access-Token": this.$ls.get(***ACCESS\_TOKEN***)}})  
 .then(response => {  
 ***console***.log(response.data);  
 })  
 .catch(error => {  
 ***console***.log(error);  
 });  
},  
  
RefreshSubscribe(record) {  
 const params = {  
 Duration: parseInt(record.duration),  
 };  
  
 ***axios***.put('http://' + ***IP*** + '/LAPI/V1.0/System/Event/Subscription/' + record.createid, params)  
 .then(  
 response => {  
 ***console***.log(response);  
 this.CurrentTime = response.data.Response.Data.CurrentTime;  
 this.TerminationTime = response.data.Response.Data.TerminationTime;  
  
  
 const CurrentTime = response.data.Response.Data.CurrentTime + 28800;  
 const TerminationTime = response.data.Response.Data.TerminationTime + 28800;  
  
 this.updateDatabase(record, record.createid, CurrentTime, TerminationTime);  
 ***console***.log(this.CurrentTime);  
 ***console***.log(this.TerminationTime);  
 })  
 .catch(error => {  
 ***console***.log(error);  
 });  
},  
DeleteSubscribe(record) {  
 ***axios***.delete('http://' + ***IP*** + '/LAPI/V1.0/System/Event/Subscription/' + record.createid)  
 .then(  
 response => {  
 ***console***.log(response);  
 })  
 .catch(error => {  
 ***console***.log(error);  
 });  
},  
  
subscribeEvent(record) {  
 ***console***.log(record.id);  
 const params = {  
 AddressType: 0,  
 IPAddress: record.ip,  
 Port: parseInt(record.port),  
 Duration: parseInt(record.duration),  
 Type: 16,  
 SubscribePersonCondition: {  
 LibIDNum: 65535,  
 LibIDList: [  
 {  
 LibID: 0  
 }  
 ]  
 }  
 };  
 ***console***.log(params);  
  
 ***axios***.post('http://' + ***IP*** + '/LAPI/V1.0/System/Event/Subscription', params)  
 .then(  
 response => {  
 const createID = response.data.Response.Data.***ID***;  
 const CurrentTime = response.data.Response.Data.CurrentTime + 28800;  
 const TerminationTime = response.data.Response.Data.TerminationTime + 28800;  
  
 this.updateDatabase(record, createID, CurrentTime, TerminationTime);  
 ***console***.log(response);  
 })  
 .catch(error => {  
 ***console***.log(error);  
 });  
},

**代码3：**

public static String doPostRequest(String url, HttpHeaders headers, String json) {  
  
 RestTemplate client = new RestTemplate();  
 headers.setContentType(MediaType.*APPLICATION\_JSON*);  
 HttpEntity<String> requestEntity = new HttpEntity<String>(json, headers);  
  
 ResponseEntity<String> response = client.postForEntity(url, requestEntity, String.class);  
 return response.getBody();  
 }  
  
 public static long localToUTC(String localTime) {  
 String dateString = localTime; // 日期字符串  
 DateTimeFormatter formatter = DateTimeFormatter.*ofPattern*("yyyy-MM-dd HH:mm:ss"); // 定义日期格式  
 LocalDateTime dateTime = LocalDateTime.*parse*(dateString, formatter); // 将字符串解析为LocalDateTime对象  
 Instant instant = dateTime.toInstant(ZoneOffset.*UTC*); // 转换为UTC时间  
 long utcSeconds = instant.getEpochSecond(); // 获取UTC时间的单位秒数  
  
 return utcSeconds;  
 }  
  
 public int dealSubscribe(String identificationnum, int passingtime) {  
 Connection conn = null;  
 Statement stmt = null;  
 ResultSet rs = null;  
  
 try {  
 String id = "";  
 String work\_date = "";  
 String work\_start\_time = "";  
 String work\_end\_time = "";  
 // 1. 加载数据库驱动程序  
 Class.*forName*("com.mysql.jdbc.Driver");  
  
 // 2. 建立数据库连接  
 String url = "jdbc:mysql://127.0.0.1:3306/jeecg-boot?characterEncoding=UTF-8&useUnicode=true&useSSL=false&tinyInt1isBit=false&allowPublicKeyRetrieval=true&serverTimezone=Asia/Shanghai";  
  
 String user = "root";  
 String password = "root";  
 conn = DriverManager.*getConnection*(url, user, password);  
  
 // 3. 创建Statement对象  
 stmt = conn.createStatement();  
  
 // 4. 执行SQL语句  
 String sql = "SELECT id FROM pm\_staff\_information WHERE identificationnum="+identificationnum;  
 rs = stmt.executeQuery(sql);  
  
 // 5. 处理结果集  
 while (rs.next()) {  
 id = rs.getString("id");  
 }  
  
 if(Objects.*equals*(id, "")) return -1;  
  
 // 查找班次表  
 // 4. 执行SQL语句  
 String sql2 = "SELECT work\_date,work\_start\_time,work\_end\_time FROM pm\_work\_information WHERE work\_person="+id;  
 rs = stmt.executeQuery(sql2);  
  
 // 5. 处理结果集  
 while (rs.next()) {  
 work\_date = rs.getString("work\_date");  
 work\_start\_time = rs.getString("work\_start\_time");  
 work\_end\_time = rs.getString("work\_end\_time");  
 }  
  
 if (Objects.*equals*(work\_date, "") || Objects.*equals*(work\_start\_time, "") || Objects.*equals*(work\_end\_time, ""))  
 return -1;  
  
 String lowerbound = work\_date.substring(0,11) + work\_start\_time;  
 String upperbound = work\_date.substring(0,11) + work\_end\_time;  
  
 long low = *localToUTC*(lowerbound);  
 long up = *localToUTC*(upperbound);  
  
 if(passingtime <= up && passingtime >= low)return 1;  
 else return 0;  
  
  
 } catch (ClassNotFoundException e) {  
 e.printStackTrace();  
 } catch (SQLException e) {  
 e.printStackTrace();  
 } finally {  
 // 6. 关闭连接和释放资源  
 try {  
 if (rs != null) {  
 rs.close();  
 }  
 if (stmt != null) {  
 stmt.close();  
 }  
 if (conn != null) {  
 conn.close();  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
 return -1;  
 }  
  
 public static int boolToInt(boolean b) {  
 return b ? 1 : 0;  
 }  
  
 @PostMapping(value = "/PersonInfo")  
 public void PersonInfo(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
  
  
 System.*out*.println("run here");  
 BufferedReader br = null;  
 try {  
 br = new BufferedReader(new InputStreamReader(req.getInputStream(), "UTF-8"));  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 String line = null;  
 StringBuilder sb = new StringBuilder();  
 try {  
 while ((line = br.readLine()) != null) {  
 sb.append(line);  
 }  
 br.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
// String dir = "E:\\a.txt";  
// File file = new File(dir);  
//  
// if (!file.exists())  
// file.createNewFile();  
//  
// FileWriter writer = new FileWriter(file);  
// writer.write(sb.toString());  
// writer.flush();  
// writer.close();  
  
 String token = new String(Files.*readAllBytes*(Paths.*get*("E:\\token.txt")));  
  
 String content = sb.toString();  
  
 System.*out*.println(content);  
  
  
 // 解析字符串成 JSON 对象  
 JSONObject jsonObject = JSONObject.*parseObject*(content);  
  
 // 获取 JSON 对象中的 "foo" 属性，并将其转换成 JSONObject  
 JSONObject PersonEventInfo = jsonObject.getJSONObject("PersonEventInfo");  
  
 JSONArray FaceInfoList = PersonEventInfo.getJSONArray("FaceInfoList");  
  
 for (int i = 0; i < FaceInfoList.size(); i++) {  
 JSONObject jsonObj = FaceInfoList.getJSONObject(i);  
  
 // here , int or long or string  
 int PassingTime = jsonObj.getInteger("PassingTime") + 28800;  
 int Type = jsonObj.getInteger("Type");  
 if (Type == 0) continue;  
  
  
 String ChannelName = jsonObj.getString("ChannelName");  
  
 JSONObject CompareInfo = jsonObj.getJSONObject("CompareInfo");  
 JSONObject SnapshotImage = CompareInfo.getJSONObject("SnapshotImage");  
  
 JSONObject SmallImage = SnapshotImage.getJSONObject("SmallImage");  
  
 String snapshotimage\_name = SmallImage.getString("Name");  
 int snapshotimage\_size = SmallImage.getInteger("Size");  
 String snapshotimage\_data = SmallImage.getString("Data");  
  
  
 JSONObject PersonInfo = CompareInfo.getJSONObject("PersonInfo");  
 int PersonID = PersonInfo.getInteger("PersonID");  
 String PersonName = PersonInfo.getString("PersonName");  
 int gender = PersonInfo.getInteger("Gender");  
 String birthday = PersonInfo.getString("Birthday");  
  
 JSONArray IdentificationList = PersonInfo.getJSONArray("IdentificationList");  
 String Identification\_Number = IdentificationList.getJSONObject(0).getString("Number");  
  
  
 JSONObject json = new JSONObject();  
 //向json中添加数据  
 json.put("passingtime", PassingTime);  
 json.put("snapshotimageName", snapshotimage\_name);  
 json.put("snapshotimageSize", snapshotimage\_size);  
 json.put("snapshotimageData", snapshotimage\_data);  
 json.put("channelname", ChannelName);  
 json.put("personid", PersonID);  
  
 json.put("personname", PersonName);  
 json.put("gender", gender);  
 json.put("birthday", birthday);  
  
  
 String jsonStr = json.toString();  
 HttpHeaders headers = new HttpHeaders();  
 headers.set("X-Access-Token", token);  
  
 *doPostRequest*("http://localhost:8080/subscribe\_post/subscribePost/add",headers,jsonStr);  
  
 // true 为按时，否则为缺席  
 int status = dealSubscribe(Identification\_Number,PassingTime);  
  
 if(status==0||status==1)  
 {  
 JSONObject json2 = new JSONObject();  
 //向json中添加数据  
 json2.put("passingtime", PassingTime);  
 json2.put("personname", PersonName);  
 json2.put("identification", Identification\_Number);  
 json2.put("status",status);  
  
 String jsonStr2 = json2.toString();  
 *doPostRequest*("http://localhost:8080/status/status/add",headers,jsonStr2);  
 }  
  
  
 }  
  
  
  
 }