

北方民族大学

本科毕业论文（设计）

题目: 基于web的人力资源管理系统的设计与实现----考勤管理模块的设计与实现

院(系)名 称: 计算机科学与工程学院

学 生 姓 名: 冯岩所旺

学 号: 20142301

专 业: 软件工程

指导教师姓名: 于千城

论文提交时间:

北方民族大学教务处制

基于web的人力资源管理系统的设计与实现----考勤管理模块的设计与实现

摘要

[第一章 概述 5](#_Toc510191592)

[1.1选题的背景与意义 5](#_Toc510191593)

[1.2国内外现状 6](#_Toc510191594)

[1.3选题特色 7](#_Toc510191595)

[1.3.1论文主要内容： 7](#_Toc510191596)

[1.3.2软件体系结构图： 8](#_Toc510191597)

[1.4 技术路线 9](#_Toc510191598)

[第二章 系统的需求分析 10](#_Toc510191599)

[2.1用户群定位 10](#_Toc510191600)

[2.2功能需求 11](#_Toc510191601)

[2.3系统非功能需求 16](#_Toc510191602)

[2.3.1 系统的操作需求 16](#_Toc510191603)

[2.3.2 性能需求 16](#_Toc510191604)

[2.3.3 安全性需求 17](#_Toc510191605)

[2.3.4 文化与政治需求 17](#_Toc510191606)

[第三章 人力资源管理系统概要设计 18](#_Toc510191607)

[3.1功能概要设计 18](#_Toc510191608)

[3.1.1 系统功能模块的划分 18](#_Toc510191609)

[3.1.2 系统业务流程 20](#_Toc510191610)

[3.2数据库概要设计 22](#_Toc510191611)

[3.2.1 系统概念模型 22](#_Toc510191612)

[3.2.2 系统的领域模型 23](#_Toc510191613)

[3.2.3 逻辑数据模型 24](#_Toc510191614)

[3.3用户界面概要设计 30](#_Toc510191615)

[第四章 详细设计与实现 32](#_Toc510191616)

[4.1 LZ-HR人力资源管理系统的总体设计 32](#_Toc510191617)

[4.2 考勤模块的总体设计思路 32](#_Toc510191618)

[4.3 考勤数据导入功能的实现 33](#_Toc510191619)

[4.4 考勤加班审核功能的实现 36](#_Toc510191620)

[4.5 考勤统计中月度汇总功能的实现 39](#_Toc510191621)

[第五章 遇的问题及解决办法 43](#_Toc510191622)

# 第一章 概述

本次毕业设计，我们的工作任务是设计并实现一个基于web的人力资源管理系统LZ-HR，支撑人力资源管理部门的日常管理工作和特殊处理工作的运行。LZ-HR系统由10个主要功能模块组成，分别是：用户管理模块、人员管理模块、机构管理模块、休假管理某块、考勤管理模块、招聘管理模块、出差和报销管理模块、固定资产管理模块、项目和资源池管理模块和个人查询中心模块。

## 1.1选题的背景与意义

进入21以来世纪，不管是经济，科技还是互联网等等都在飞速发展，其中互联网科技的飞速发展使我们的生活发生了天翻地覆的改变，人与人之间的通信不再局限于书信，购物不再局限于逛街、进实体店购买，学习知识也不再局限于学校学习。互联网让一个地区变成一个家园，让一个国家变成一个家园，让整个地球变成一个家园。在21世纪以来，处处充满了科技与互联网，互联网的发展使人与人之间的距离拉近，互联网信息发生爆炸式喷发，信息不再局限于少数地点和少数人的手里。

互联同时也让人力资源的管理变得更方便、便捷，从之前的文件与人工管理人力资源到现在使用系统软件时时管理人力资源，从繁琐与不及时、不准确到简洁与随时准确掌控的改变，使企业不再耗费大量人力物力去管理自己的人力资源，只需要开发或者购入一套人力资源管理系统，企业就能高效的管理和分配人力资源，还可以利用软件帮自己计算出人力资源在各个领域的分配比，使管理层人员能够把更多的时间放到产品创新和对未来公司的展望上，致使企业只需要花少量的人通过人力资源管理系统来管理人力资源。

现在的人力资源管理与过去的人力资源管理相比，不管是在思想上还是在管理方式上发生了翻天覆地的改变，首先，在其思维与管理的观念上的区别也很大，过去的传统人力资源管理主要是注重投入、使用和控制，它的中心点时放在“事”上，而现在的人力资源管理则主要是注重于“人”，主张的是以人为中心。其二，在其管理的内容和管理的中心上的不同，过去的传统人力资源管理是把人当成一种“机器”，它只管员工的“进、管、出”，也就是它把员工的招入、员工的离职和员工的管理放在第一位，其中，它把人员的调入和调出看成核心所在，而现在人力资源管理则是把人作为企业的重心，它的管理目的是“吸引、保留、激励和开发”公司所需要的人力资源，公司只需提供一定的指挥、控制和相应的管理，就能优化组合人力与物力之间的资源，从而提高员工的工作兴趣和积极性，以达到最大发挥人的潜在潜能，最终结果就是使公司的整体工作效率提高。

而我们实现LZ-HR人力资源管理系统，就是为了使HR的管理人员够摆脱繁琐的事务性处理工作，把精力都集中在具有战略性项目上，让公司把有限的人力资源合理分配到需要它的地方，以更好地吸引人才、留住人才、用好人才，从而最终实现提高企业核心竞争里的目标，同时也让公司高层能时时监控人力资源的流向，帮助公司做出更明确。更有前景的战略目标。

## 1.2国内外现状

HR人力资源管理系统发展至今，经历了从传统的与ERP集成在一起的重量级HR系统到目前比较流行的采用SaaS模式的Hr系统的转变，传统的HR人力资源管理系统中比较有名的有SAP公司开发的SAP BusinessOne和Oracle公司开发的Oracle ERP。而国内的钉钉人事、i人事、智思云、北森、薪人薪事等都是最新的采用SaaS模式开发的系统。SaaS(Software-as-a-Service，软件即是服务)，它是随着互联网发展和应用软件的成熟运用所诞生的一种全新的服务模式——基于互联网的全新的软件服务模式。SaaS提供商为企业搭建好所需的平台，还有网络基础设施、硬件等等，并且所有的前期实施和后期维护等一系列服务，企业不需购买软硬件、建设机房和招聘IT人员，就可以通过互联网来使用信息管理系统，这种模式的软件对于客户来说，就像买大米一样，直接就能煮，无需我们去管种植和加工等的问题，你只要会煮和吃就行，就如SaaS软件一样，你只要会使用就行，其他一切都不需要去管。SaaS的软件对于企业来说，企业无需在硬件和IT人员方面进行任何投资，就可以获得软件服务，这样既省心省力又省钱。软件的部署、升级和数据的维护上都不需企业自己来完成，这样不仅能够规避建设风险，还能够自动升级且持续获得价值服务，数据维护上又有免费的专业的强大的团队去维护，对于企业来说，在传统模式的管理系统和新模式SaaS的管理上的选择不言而喻。SaaS模式的软件将是以后软件发展的趋势，企业选择SaaS就好比是给企业上了保险，从保单生效开始直至终老，你都会受益，除非你自己退保。但保单的内容能否全部兑现，就取决于你是否选择了有实力的纯正SaaS软件公司为你来承保。Oracle ERP是由美国厂商制造，它适合于金融、教育、电信等大型的企业，产品相对来说灵活，对企业管理标准化的要求较低，但是产品和服务价格昂贵，并且缺乏足够的业务流程模板和软件功能的支持，无法满足企业管理上的要求；钉钉人事是SaaS中比较有代表性的软件，它是一个完全免费且基于阿里云的hr人力资源管理系统，可以帮助企业更高效的管理花名册，办理入职，离职，考勤等工作，使决策者能实时了解团队的人力情况，实现人力资源合理分配。SaaS模式的软件和传统模式软件相比较，传统的方式是叫软件许可，它的收费方式是用户一次性支付所有费用且需要培养一批专业的IT队伍，更新周期比较长，而SaaS模式则是一种按需定制，它的收费方式是用户按需定期支付费用且不需要配备IT队伍，SaaS软件供应商会为用户提供专业的运维，一出现bug会第一时间找出来，非常及时的更新和维护。SaaS模式的考勤系统与传统模式的ERP考勤系统相比，SaaS模式的考勤系统是智能化考勤，更灵活，更及时，更便捷，摆脱了传统的考勤机，同时员工还可以通过手机定位实现直接考勤，还有便捷式服务，如员工生日、入职日祝福，复岗、公司公告、通知送达及时、准确，更快更方便传递公司的沟通信息，让员工感受到公司的人文关怀。

## 1.3选题特色

### 1.3.1论文主要内容：

实现LZ-HR人力资源管理系统，用于支撑人力资源管理的工作和特殊处理工作的运行，使企业能高效的管理人力资源，同时便于管理人员能够合理的分配人力资源，以实现人力资源的价值，让公司分出更多的时间、人力与物力到部门上。本项目中，我实现的是考勤管理模块，主要任务有：

（1）模拟打卡机打卡：我实现后台的打卡信息录入功能，用导入表的方法模拟打卡机的信息，然后实现模拟打卡机打卡。

（2）考勤统计：这个功能还包括月度汇总，每月统计，打卡时间，原始数据4个子功能，主要用于查看考勤记录，对考勤数据的统计和修改等。

（3）审核：对加班和补打卡的审核。

（4）工位设置：包括工位管理和工位查询两个子模块，主要实现公司工位的设置和布置，并生成二维码提供扫码手机扫码签到等，还有查询工位信息的功能。

### 1.3.2软件体系结构图：

****

图1.1考勤模块

如上图所示，考勤模块主要由前端UI、展示层、业务层、持久层和数据库5个结构构成，它们组成了完整的考勤模块。其中前段UI主要React技术实现，而展示层则主要进行数据交互，业务层、持久层和数据库则运用经典的mvc模式来实现数据的处理、加载和存储。

系统的结构体系则主要也是前后端分开，前端主要由react、antdUI、HTML、CSS和图片来实现，而后端主要由spring boot+mybaties的技术实现的，并运用了mvc的模式来实现的，分为了controller层、service层、dao层和数据库层。系统结构体系如下图所示

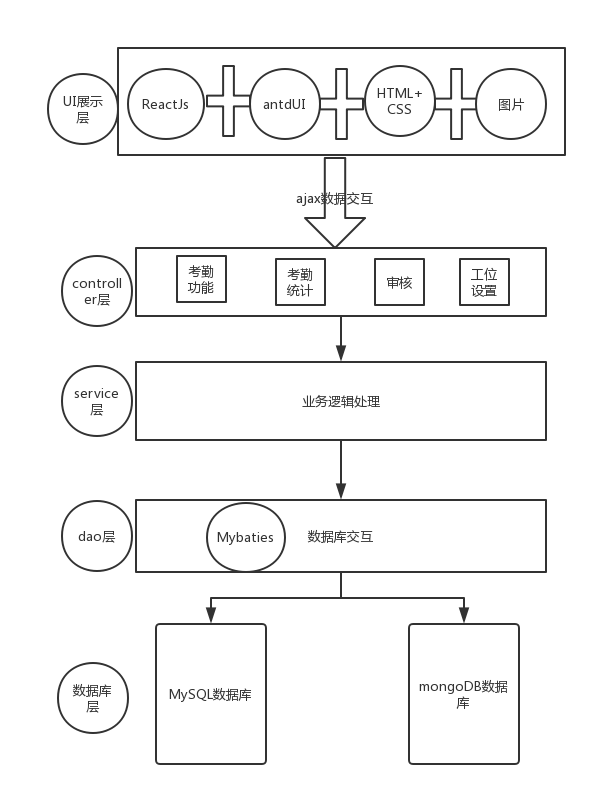
****

图1.2体系图

## 1.4 技术路线

后台主要采用java开发，运用mvc的思想模式，框架结构是spring boot+mybaties的框架结构，数据库主要采用mysql数据。

前台采用React+html+css+ajax的结构，ajax是用于数据前后台的数据异步交互，React是用于前台视图与前端逻辑的实现，其中React是Facebook开发的一款全新的JS库，它是通过操控虚拟DOM来实现真实DOM的渲染，利用组件状态的改变来进行组件之间信息传递，通过diff算法，把虚拟DOM与真实DOM相比较,更新差异的地方到真实DOM中，使真实DOM重新渲染，界面达到重新渲染。

开发工具：eclipse 软件,mysql数据库,mongDB数据库，tomcat服务(spring boot内部集成)，Microsoft VS Code软件，visio、StarUML软件。

# 第二章 系统的需求分析

## 2.1用户群定位

HR人力资源管理系统的用户群主要是HR人力资源管理系统的系统管理员、资源管理员，机构管理人员，使用该系统的hr人员和普通员工。

普通员工，可以通过登录进入系统，他可以查看看个人信息、查询出勤情况和工资单等信息，同时可以申请休假、补打卡和加班。

考勤管理员，主要管理考勤模块，他需要不定时的把打卡机中的信息导入数据，然后对考勤进行统计，员工的考勤信息都来自管理员的考勤信息录入，他还有对加班和补打卡的审核任务，最后他还需要对公司的工位进行分配和管理。

系统管理员，主管系统的权限分配问题，需要保证每个人员各司其职。

机构管理和休假管理都属于其他管理人员，他们主要是管理各个模块的功能模块，并完成各自的相关任务。

如图2.1上下文图。

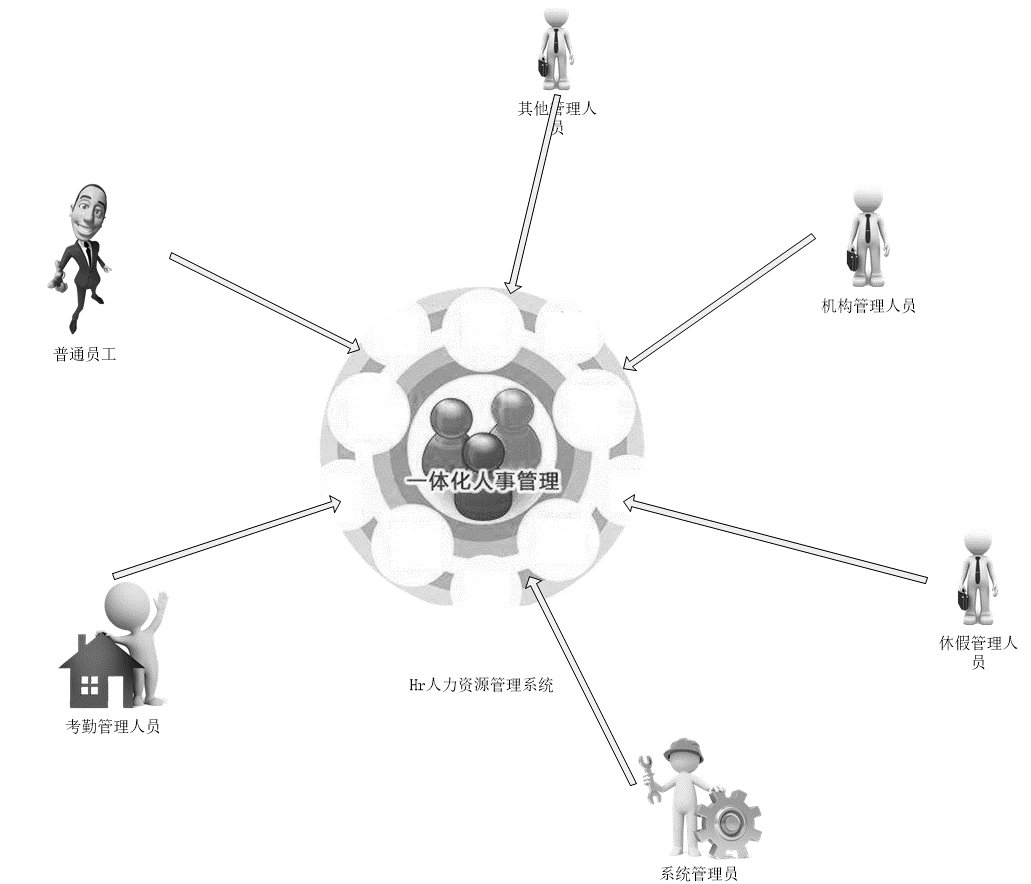


图2.1 上下文图

## 2.2功能需求

系统的操作者分为3类：普通员工，系统管理员，模块管理员，其中考勤管理员和其他管理员都属于模块管理员。

普通员工，通过账号密码登录进入系统，可以查询自己的出勤情况和自己的工资单，查询个人信息，普通员工还可以申请休假、报销、加班和补打卡，所有申请都需要相关管理员审批才能通过。

系统管理员主要是管理权限，给相应的人员分配合理的权限，使每个人员管理好自己管理领域内的事宜。

模块管理员，每个模块的管理员主要管理者自己模块的相应事宜，同时模块管理员还是普通同员工，有普通员工所有权限，可以进行普通员工的所有功能。但是每个管理员只能进入普通员工和自己管理模块的界面，除非被分配更高权限。

考勤模块是我实现的管理模块，所以单独提出来说明，考勤管理员的功能有：考勤打卡(打卡机打卡)，审核，考勤统计，工位管理。管理需要对数据进行录入，然后对数据进行统计并导成表交给老板看，还需要审批加班和补打卡的申请，

，如2.2图用例图。

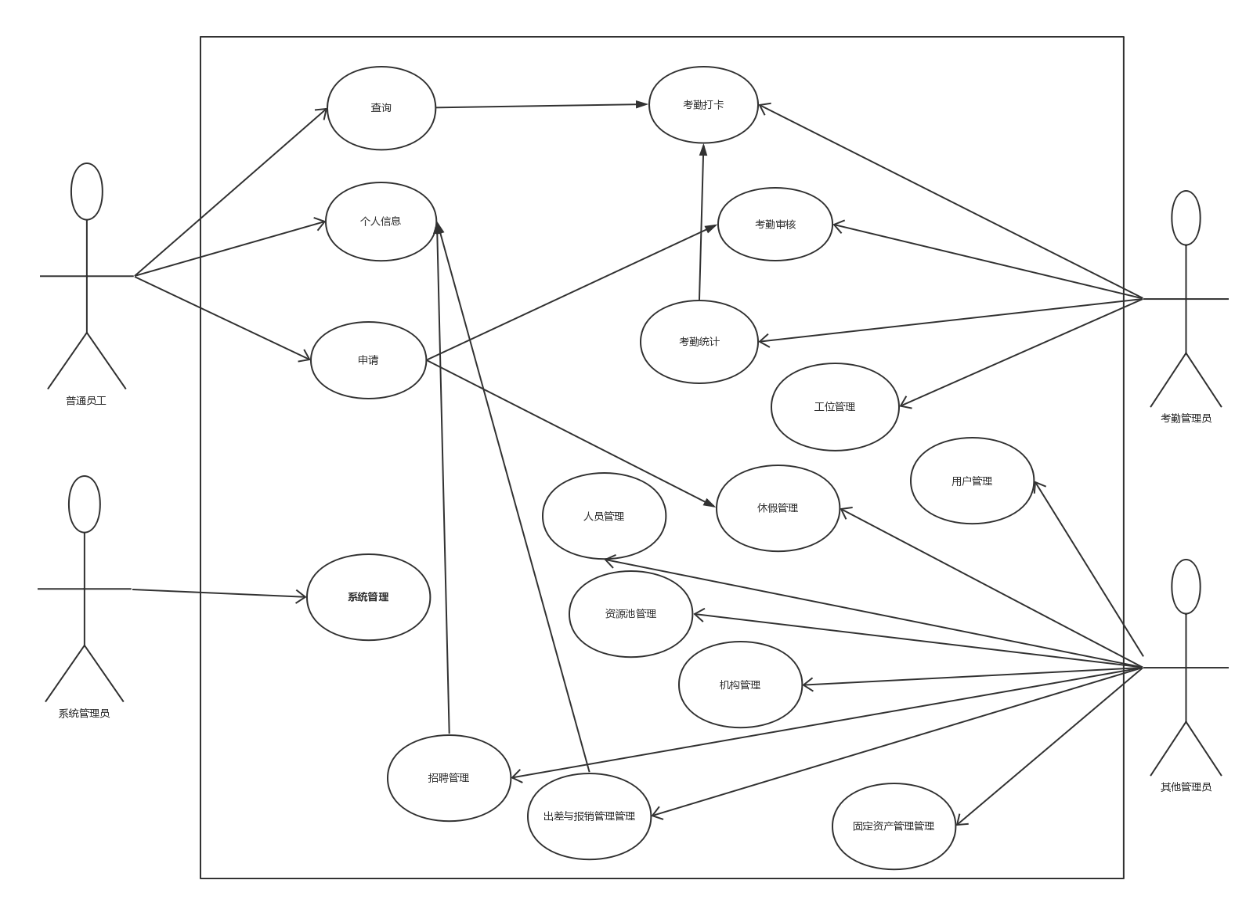


图2.2 系统用例图

|  |  |  |
| --- | --- | --- |
| 用例编号 | 001 | |
| 用例名称 | 申请加班 | |
| 用例概述 | 公司员工使用系统申请加班 | |
| 用例类型 | 业务用例 | |
| 参与者 | 员工，考勤管理员 | |
| 用例相关人 | 员工 | |
| 前置条件 | 员工登录系统，进入到申请页面进行加班申请 | |
| 后置条件 | 系统显示要填写的相关信息，员工按要求填写信息并申请，系统  提示申请成功 | |
| 基本事件流 | 步骤 | 活动(Action) |
| 1 | 员工进入申请界面 |
| 2 | 系统显示：休假申请、加班申请、报销申请 |
| 3 | 员工选择加班申请 |
| 4 | 系统显示功能：加班申请，加班记录，待批记录，驳回记录 |
| 5 | 员工选择加班申请 |
| 6 | 系统跳出一个框，并提示员工填写相关的加班信息，并提交 |
| 7 | 系统提示提交成功并等待审核 |
| 8 | 员工可以重新选择待批记录，然后撤销申请，然后终止下面的步骤 |
| 9 | 系统提示管理员有申请需要审批 |
| 10 | 管理员登录进入系统，对申请进行审核 |
| 11 | 员工可以随时查询审核结果 |
| 12 | 管理员不同意申请 |
| 13 | 员工可以查看驳回的原因，并根据情况可以  选择重新申请或者放弃申请 |
| 14 | 管理员同意 |
| 15 | 员工的加班申请成功 |
| 扩展事件流 | 1a | 系统检测出员工信息错误  1.系统提示，员工信息有误，请重新登录  2.系统转接员工到登录界面 |
| 4a | 系统查询不到数据   1. 系统显示无数据 2. 系统提示服务器崩溃 |
| 7a | 申请提交时失败   1. 系统提示信息填写有问题 2. 系统提示服务崩溃 |
| 8a | 撤销失败  1.系统提示申请已被管理员审核 |
| 9a | 系统查出不是管理员身份   1. 系统提示身份信息错误 2. 系统自动跳转到登录界面 |
| 12a | 审核驳回失败   1. 系统提示申请已被撤销 2. 系统提示系统出现问题 |
| 14a | 审核通过失败   1. 系统提示申请已被撤销 2. 系统提示系统出现问题 |
| 补充说明 | 1. 员工填写加班信息时需要按照规定格式 2. 员工的加班原因一定要写且必须真实，不然管理不给通过 | |

表2-3 员工加班申请用例描述

|  |  |  |
| --- | --- | --- |
| 用例编号 | 002 | |
| 用例名称 | 考勤管理员对工位的管理 | |
| 用例概述 | 考勤管理员使用系统对公司的工位进行分配、管理和记录 | |
| 用例类型 | 业务用例 | |
| 参与者 | 考勤管理员 | |
| 用例相关人 | 考勤管理员 | |
| 前置条件 | 管理员登录进入系统，选择工位管理 | |
| 后置条件 | 系统显示成楼层工位图 | |
| 基本事件流 | 步骤 | 活动(Action) |
| 1 | 管理员登录进入系统 |
| 2 | 进入考勤管理选择工位管理 |
| 3 | 系统显示出各个功能:工位管理，工位查询 |
| 4 | 管理员选择工位管理进入园区管理界面 |
| 5 | 若无园区信息则选择添加园区信息并形成卡片信息 |
| 6 | 管理员点击园区卡片信息进入建筑管理界面 |
| 7 | 若无建筑信息则选择添加建筑信息并形成卡片信息 |
| 8 | 管理员点击建筑卡片信息进入楼层管理界面 |
| 9 | 若无楼层信息则选择添加楼层信息(其中的工位的坐标要填写，若不填写则默认x:10,y:10)并形成卡片信息 |
| 10 | 管理员点击楼层信息卡片，进入到工位信息界面 |
| 11 | 系统显示功能：工位，WIFI，工位图，且系统会默认显示工位，工位显示的是根据工位坐标形成的x行y列的表格 |
| 12 | 管理员选择工位图，上传楼层信息图，与工位表格想映射，最后想成工位管理图 |
| 13 | 管理员选择工位，可以对表格的每一个填写相应的工位信息 |
| 14 | 管理员选择WiFi，进行WiFi信息填写 |
| 15 | 管理员把工位信息表导成一个个的工位二维码信息，提供给移动端扫码打卡使用 |
| 扩展事件流 | 1a | 系统检测出管理员登录信息错误  1.系统提示，管理员信息有误，请重新登录  2.系统转接管理员到登录界面 |
| 4a | 界面无信息显示  1.系统显示无数据，需要录入  2.系统提示服务器崩溃 |
| 6a | 点击卡片无法进入下个界面  1.系统提示信息错误并跳回园区界面  2.系统提示服务崩溃 |
| 9a | 信息添加失败   1. 系统提示信息填写格式错误 2. 系统服务器错误 |
| 12a | 工位图上传失败   1. 系统提示图片格式不对 2. 系统提示图片过大 |
| 15a | 二维码导出失败   1. 系统提示导出失败 2. 系统提示服务崩溃 |
| 补充说明 | 1.管理员填写相应信息时需要按照规定格式  2.工位图的大小和格式保证正确 | |

表 2-4 工位管理用例描述

考勤管理员可以对公司的工位进行管理，管理员需要把园区、建筑、楼层信息填写好，系统会根据信息形成工位表格，并与上传的图片形成工位管理图，可以更形象的显示公司的工位。

## 2.3系统非功能需求

### 2.3.1 系统的操作需求

技术环境：管理系统需要在windows7及以上的版本运行，其中数据库是采用mysql数据，数据库软件采用的是MySQL Server 5.5版本，其浏览器必须要IE10以上或者使用Google浏览器，不然会影响页面的正常展示。员工和管理员们只需通过网址进入登录系统，无需在自己的电脑上部署项目。

系统集成：工的基础信息是来自员工管理的管理员的录入，考勤数据是来自打卡机的记录，考勤管理员需要不定期把考勤数据从打卡机中导出并导入到系统的数据库中，系统会从数据库独处信息并对信息进行处理和统计，而其他数据也是来自相应模块管理员的录入。其中考勤模块的数据处理是交给专门用来处理数据的功能模块来进行相关处理，其他模块的数据处理也是通过各自模块的相应功能模块来进行处理。整个系统是有10大模块构建而成，每个模块相互关联又相互排斥。数据与功能构成了模块，模块与模块构成了整个系统。

可移植性：随着人力资源市场的变化、公司需求的变化和web开发技术的影响，系统都会随着变化，功能也会跟着不断的变得完善。

可维护性：预计每个7个月都会对系统进行维护，服务器的硬件设备也会不断的维修和更换。

### 性能需求

响应速度：1.功能的响应速度必须小于5秒。

2.数据库必须能灵活做相应变化，以达到随时做升级做准备。

系统容量：1.在高峰时期，系统最多能承受500～600人同时登陆在线。

2.员工的申请功能最大支持同时200个人进行申请。

3.系统对数据的处理速率必须快

可用性和可靠性：

1. 系统能24小时运行与使用。
2. 系统应该拥有99%的时间正常运行的能力。

### 2.3.3 安全性需求

系统的价值：系统必须要给公司带来高效的效益，同时尽可能的降低公司的成本，一切都以公司的利益为主要目的。

访问控制：

1. 系统管理员会设置权限，只有有相应的权限进入相应的地方，才能获取操作模块功能的权力。
2. 系统中的某些功能会拥有更高的权限设置，用户必须要有更高的权限才能操作该功能。

加密/验证：系统所有用户的密码都进行加密处理，系统中都需要权限的验证，只有验证通过了才能进行相应的操作。

病毒控制：服务器会装上杀毒软件，并且每周都对系统进行扫描，确保系统的安全性。

系统故障修复：系统开发人员会对系统不定时进行更新和版本变更。

### 2.3.4 文化与政治需求

多语言设计：

系统预计只会支持简体中文，因为公司目前对外文和繁体中文没有相应的需求。如若之后公司有相应的需求，则系统也会支持多种语言，以此来保证公司的正常使用。

法律责任：

1. 系统绝不能泄露用户个人信息，如若为之则需要承担相应的法律责任。
2. 系统内不能传播违法犯罪的信息。

# 第三章 人力资源管理系统概要设计

## 3.1功能概要设计

### 3.1.1 系统功能模块的划分

LZ-HR人力资源管理系统主要由10个大模块构成的，其主要是用户管理模块、查询中心模块、休假管理模块、考勤管理模块、人员管理模块、机构管理模块、招聘管理模块、出差与报销管理模块、固定资产管理模块、资源池管理模块，其中用户管理主要是执行超级管理员的功能设置用户权限，查询中心主要是给所有用户提供查看信息和提交申请，休假管理则主管休假事物，考勤模块主管考勤，机构管理主管机构相关的事物，其他模块也同样主管各自模块的事物。如下图3.1系统功能模块图。

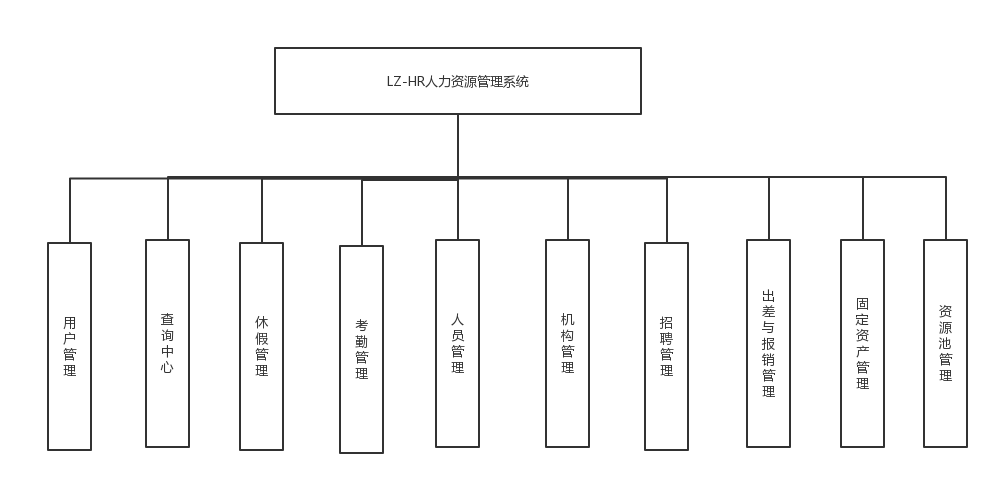


图3.1　系统功能模块图

考勤管理模块的功能主要有４个大功能组成，分别为考勤打卡，考勤审核、考勤统计、工位设置。其中考勤审核又分别由加班审核和补打卡审核组成，考勤统计由月度汇总、每日统计、打卡时间、原始数据４个功能组成，工位设置由工位管理和工位查询组成。如下图3.2考勤模块功能图。

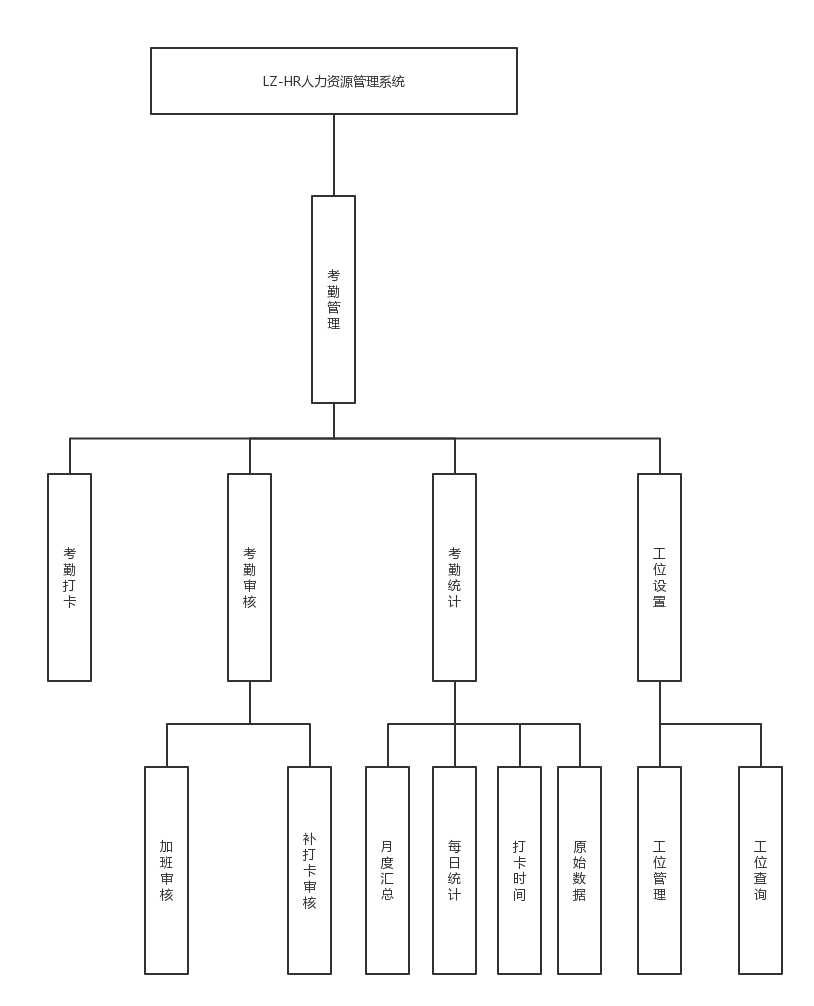


图3.2 考勤模块功能图

考勤打卡是整个系统考勤数据的来源，考勤管理员需要不定期的从打卡机中导出数据并导入到系统中，然后系统会自动对考勤数据进行处理、统计。而整个系统的权限管理则由用户管理模块管理，权限管理主要是保证系统正常运行而存在的，若无权限管理，则整个系统就会乱套，导致系统崩溃。

### 3.1.2 系统业务流程

员工根据登录信息登录验证进入到系统，员工可以看见个人查询中心这一过程，员工可以进入个人查询中心去查看基本信息、合同记录、工作变更、岗位调整、调薪记录、薪资福利和笔记本补贴，这些个人信息是用户管理过程提供的数据，员工可以进入查询界面，查询员工本人的出勤日志，出勤日志信息是来自于考勤管理过程，系统把考勤数据处理以后显示给员工，查询员工本人的工资单详情，该详情是来自于休假管理过程中工资计算，员工还以进入到申请界面，对系统提交加班申请、休假申请和报销申请，加班申请会被系统提交给考勤管理审核，休假申请会被系统提交给休假管理审核，报销申请则会提交给出差与保险管理审核。由于系统功能太复杂，为了简化，其数据流图大致如图3.3所示。

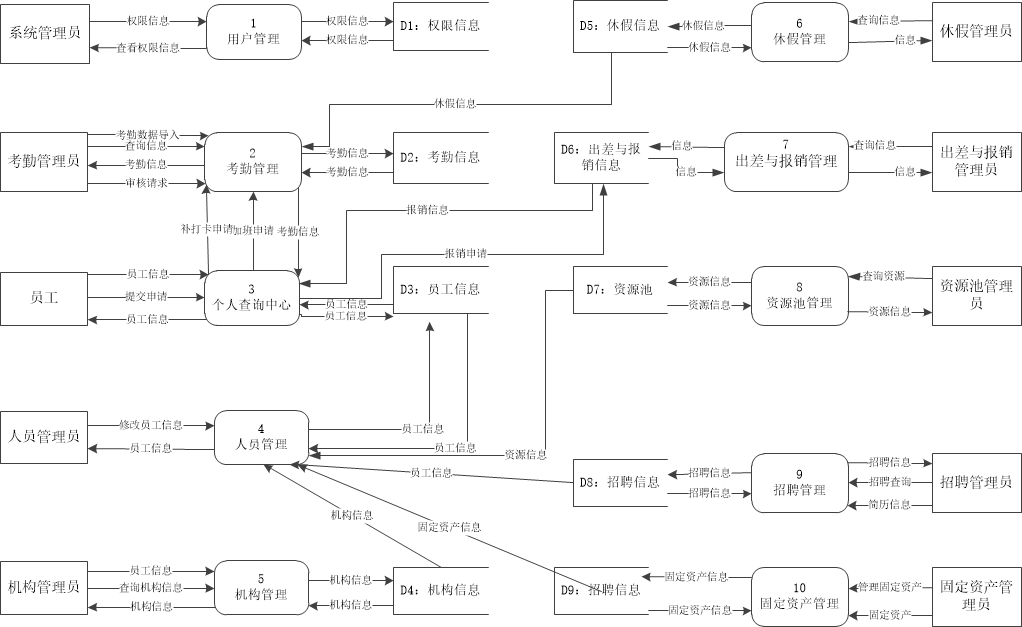


图3.3 系统0层图

注：系统的每个过程都要有相应的权限才能进行。

HR-LZ人力资源管理系统大致分解成10个高层过程，分别是：1.用户管理、2.考勤管理、3.个人查询中心、4.人员管理、5.机构管理、6.休假管理、7.出差与报销管理、8.资源池管理、9.招聘管理、10.固定资产管理。用户管理这一过程是系统管理员向系统提交权限信息，系统把权限信息存到数据库，系统管理员可以查看管理权限。考勤管理过程，考勤管理员会把打卡机上的数据导入到系统中，系统对数据进行处理后存到数据库中，考勤管理员可以根据月份查询每个月的考勤数据，也可以查询每个的月考勤统计，考勤管理员对系统发出审核的请求，系统会把所有的申请信息查询到界面以提供审核。个人查询中心是每个员工都进行的过程，每个员工提供员工信息给系统，系统会把员工的个人信息提和考勤等其他信息反馈给员工，员工还可以提交申请到系统让系统提交到其他过程去审核，审核通过就反馈给员工审核结果。人员管理过程是是对所有员工信息的处理，系统会给人员管理员提供所有员工的员工信息，人员管理员可以对员工的信息进行修改，而其中，员工的信息、资源信息和机构信息都是分别来自于招聘信息库、资源池信息库和机构信息库。机构管理员可以根据员工信息查询员工的机构信息。休假管理员可以通过系统查询所有员工的休假信息，也可以添加员工的休假信息，还可以对每个员工的信息进行管理。出差与报销管理过程，出差与报销管理员可以添加员工的出差信息和报销信息到数据库。资源池管理员可以查询资源池的信息，可以对资源池的资源进行操作和维护。招聘管理员把每个员工的简历提交到系统，系统会把简历信息进行存储，然后把信息提交给人员管理，招聘管理员根据管理员信息查看所有的招聘信息。固定资产管理员可以查看和管理员工的固定资产。系统过程2的数据流图如3.4所示。

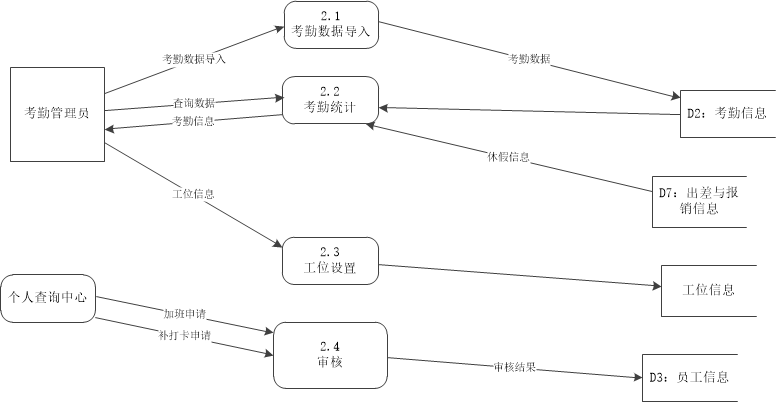


图3.4 系统过程1（考勤管理模块）1层DFD

系统过程1考勤管理又分解成4个子过程：2.1.考勤数据录入、2.2.考勤统计、2.3.工位设置、2.4.审核过程。考勤数据录入是管理员把打卡机中的数据导入到系统中，考勤统计是系统对考勤数据进行处理，工位设置是管理员对公司工位进行录入和管理分配，审核时管理员对加班和补打卡申请的审核。

## 3.2数据库概要设计

### 3.2.1 系统概念模型

根据系统的业务流程，系统由10大模块组成，分别是：1.用户管理、2.考勤管理、3.个人查询中心、4.人员管理、5.机构管理、6.休假管理、7.出差与报销管理、8.资源池管理、9.招聘管理、10.固定资产管理。由于系统太复杂，涉及到的实体实在是太多，画图地方有限，只画出了我负责的考勤模块的主要实体属性，各个实体之间的关系图：如图3.5 考勤E-R图。

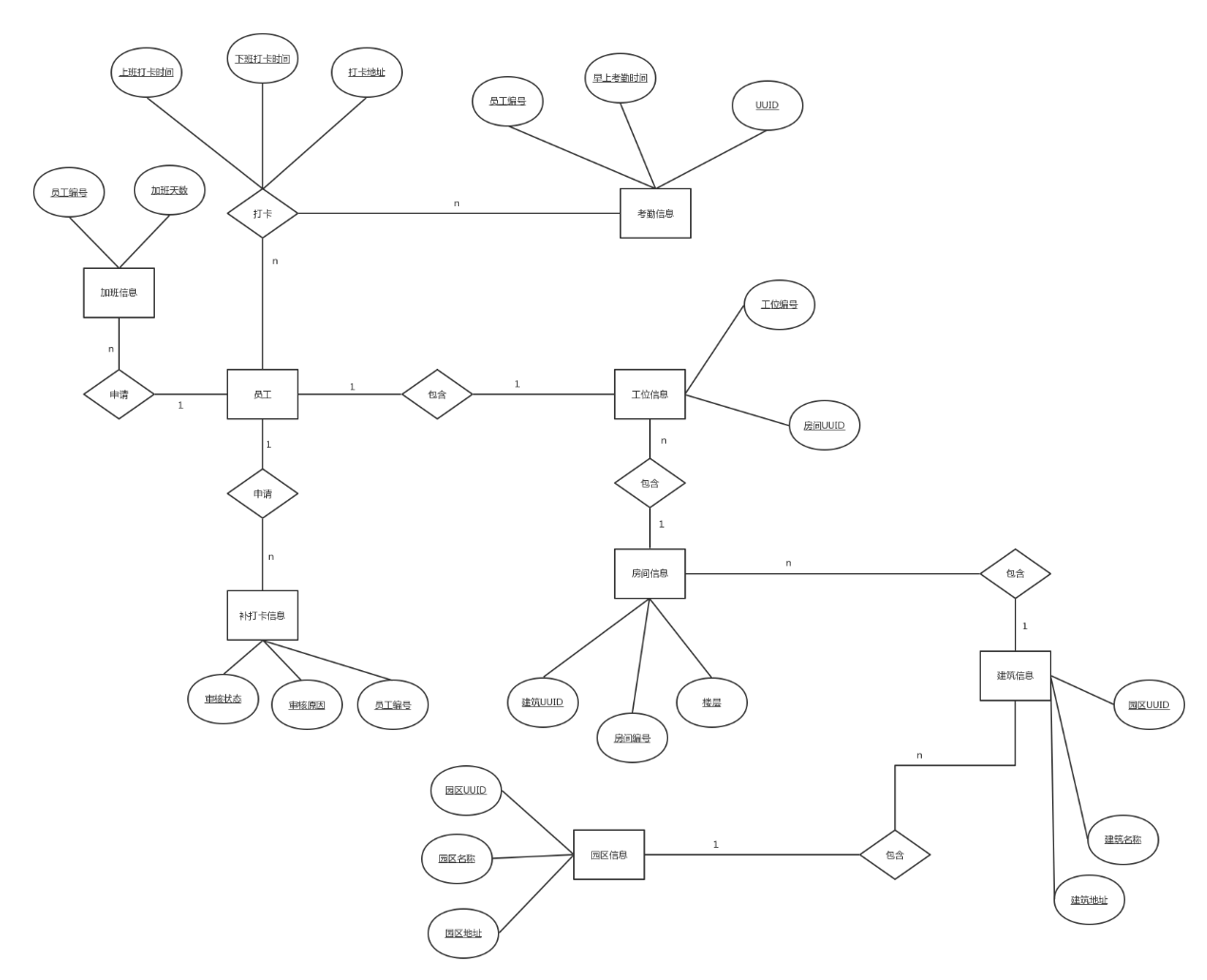


图3.5 考勤E-R图

其中打卡所派生出来的属性包括UUID、考勤规则名称、上班打卡时间、下班打卡时间、打卡wifi、打卡地址。下班补充派生实体属性：如图3.6 派生实体属性。

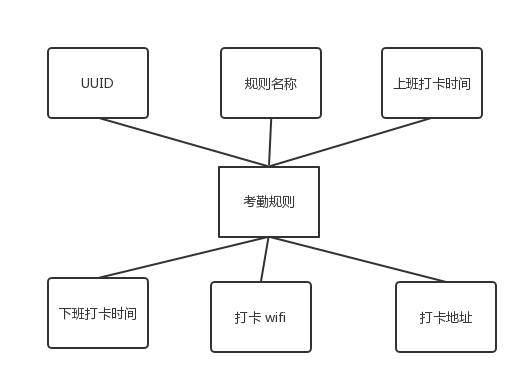
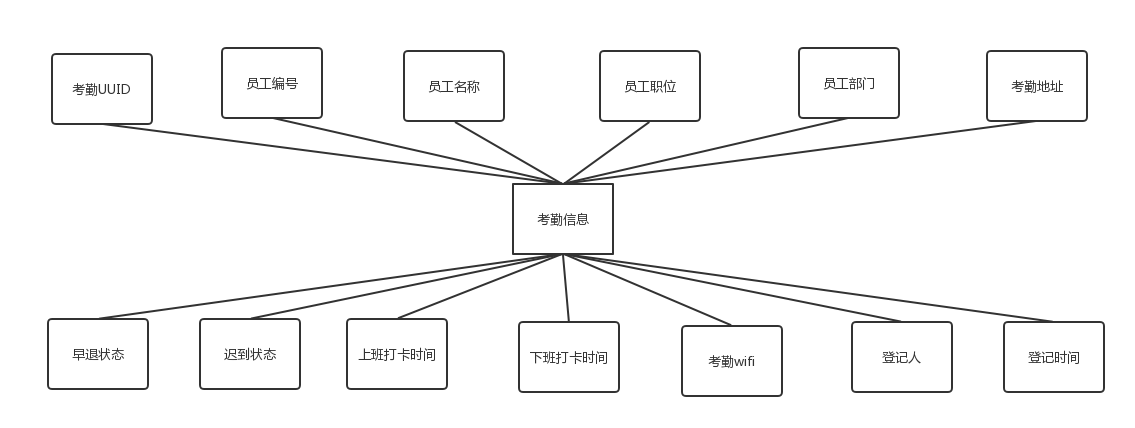


图3.6 派生实体属性

下面补充一下涉及到的实体图的完整属性。考勤信息包含的数据项有：考勤UUID、员工编号、员工名称、员工职位、员工部门、考勤地址、早退状态、迟到状态等内容。有考勤信息的数据项可以得到考勤信息实体属性图，如图3.7考勤信息实体图属性。



3.7考勤信息实体图属性

### 3.2.2 系统的领域模型

LZ-HR的系统包含的表实在是太多，关系也太复杂，所以这里只说明我负责的模块的领域模型。考勤模块一共包含了10张表，分别是员工表、考勤信息表、考勤规则表、休假信息表、加班信息表、补打卡信息表、园区信息表、建筑信息表、房间信息表、工位信息表。整个领域模型图之间的关系主要围绕着员工信息表而展开的，考勤信息表主要是存储员工打卡的信息，考勤规则表则是对员工考勤状态(早图、迟到)的判断，休假信息表、加班信息表和补打卡信息表都是员工提交的申请信息，而工位信息表、园区信息表、建筑信息表和房间信息表则是存储员工工位的完整信息表，这些信息是用来给移动端扫码打卡准备的。模块的领域模型如图3.8所示：

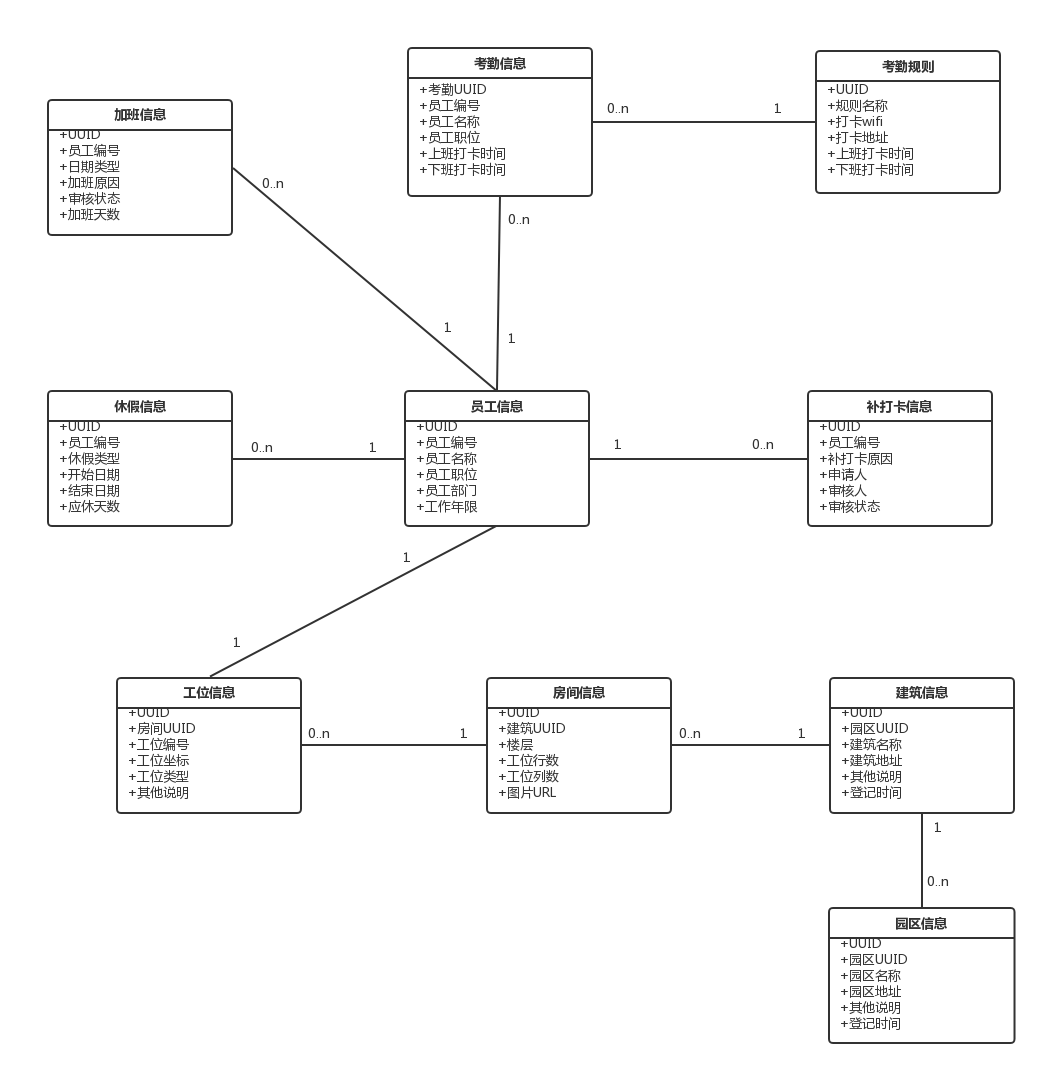


图3.8 考勤模块的领域模型图

### 3.2.3 逻辑数据模型

由于整个LZ-HR人力组员管理系统规模太大，涉及到的表过多，无法一一把它列出，所以还是展示我所负责的模块，考勤管理模块。首先在mysql数据库管理系统中建立一个名为hr的数据库，然后在数据库中创建考勤模块所需的数据表，系统一共包含10张表：员工信息表、考勤信息表、考勤规则表、加班信息表、休假信息表、补打卡信息表、园区信息表、建筑信息表、房间信息表、工位信息表。

1. 员工信息表（hr\_employee）

包含的所有字段名，长度和数据类型如表3.9所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 主键/外键 | 约束 | 描述 |
| user\_uuid | varchar | 32 | 主键 | 非空 | 员工uuid |
| group\_uuid | varchar | 32 |  |  | 公司uuid |
| branch\_uuid | varchar | 32 |  |  | 分公司uuid |
| dept\_uuid | varchar | 32 |  |  | 部门uuid |
| id\_type | varchar | 32 |  |  | 证件类型 |
| id\_code | varchar | 64 |  |  | 证件编号 |
| staff\_code | varchar | 24 |  |  | 员工编号 |
| per\_name | varchar | 32 |  |  | 员工名称 |
| per\_gender | varchar | 2 |  |  | 性别 |
| birth\_date | varchar | 24 |  |  | 出生日期 |
| job\_title | varchar | 64 |  |  | 职位 |
| phoneno | varchar | 32 |  |  | 电话 |
| email | varchar | 64 |  |  | 电子邮箱 |
| manager | varchar | 24 |  |  | 直接主管 |

表3.9 员工信息表

1. 考勤信息表（hr\_attendance）

包含的所有字段名，长度和数据类型如表3.10所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 主键/外键 | 约束 | 描述 |
| attendance\_uuid | varchar | 64 | 主键 | 非空 | 考勤uuid |
| staff\_code | varchar | 24 |  |  | 员工编号 |
| per\_name | varchar | 32 |  |  | 员工名称 |
| dept\_name | varchar | 128 |  |  | 部门名称 |
| job\_title | varchar | 64 |  |  | 员工职位 |
| attendance\_status\_leave | varchar | 32 |  |  | 早退状态 |
| attendance\_status\_late | varchar | 32 |  |  | 迟到状态 |
| attendance\_morning\_time | varchar | 128 |  |  | 上班打卡 |
| attendance\_afternoon\_time | varchar | 128 |  |  | 下班打卡 |
| attendance\_wifi | varchar | 128 |  |  | 打卡wifi |
| attendance\_address | varchar | 128 |  |  | 打卡地址 |
| reg\_name | varchar | 255 |  |  | 登记人 |
| reg\_time | varchar | 64 |  |  | 登记时间 |

表3.10 考勤表

1. 考勤规则表（hr\_attendance\_rules）

包含的所有字段名，长度和数据类型如表3.11所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 主键/外键 | 约束 | 描述 |
| attendance\_set\_uuid | varchar | 64 | 主键 | 非空 | 规则uuid |
| attendance\_set\_name | varchar | 24 |  |  | 规则名称 |
| morning\_set\_time | varchar | 64 |  |  | 上班时间 |
| afternoon\_set\_time | varchar | 64 |  |  | 下班时间 |
| attendance\_set\_wifi | varchar | 64 |  |  | 打卡wifi |
| attendance\_set\_address | varchar | 64 |  |  | 打卡地址 |

表3.11 考勤规则表

1. 加班信息表（hr\_work\_overtime）

包含的所有字段名，长度和数据类型如表3.12所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 主键/外键 | 约束 | 描述 |
| uuid | varchar | 24 | 主键 | 非空 | uuid |
| staff\_code | varchar | 64 |  | 非空 | 员工编号 |
| proj\_name | varchar | 128 |  |  | 项目名称 |
| data\_type | varchar | 84 |  |  | 日期类型 |
| memo | varchar | 256 |  |  | 备注 |
| reason | varchar | 256 |  |  | 申请原因 |
| audit\_status | varchar | 10 |  |  | 申请状态 |
| approve | varchar | 24 |  |  | 审核人 |
| approve\_day | varchar | 24 |  |  | 审核时间 |
| begin\_day | varchar | 24 |  |  | 开始日期 |
| end\_day | varchar | 24 |  |  | 结束日期 |
| apply\_day | varchar | 24 |  |  | 申请时间 |
| work\_overtime\_days | varchar | 24 |  |  | 加班天数 |
| counterview | varchar | 256 |  |  | 审核原因 |

表3.12 加班信息表

1. 休假信息表（hr\_leave\_log）

包含的所有字段名，长度和数据类型如表3.13所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 主键/外键 | 约束 | 描述 |
| uuid | varchar | 24 | 主键 | 非空 | uuid |
| user\_uuid | varchar | 24 | 外键 | 非空 | 员工uuid |
| leave\_type | varchar | 128 |  |  | 休假类型 |
| begin\_date | varchar | 24 |  |  | 开始日期 |
| end\_date | varchar | 24 |  |  | 结束日期 |
| accrued | varchar | 16 |  |  | 应休天数 |
| spend | varchar | 16 |  |  | 判断应休天数 |
| proposer | varchar | 16 |  |  | 申请人 |
| apply\_day | varchar | 16 |  |  | 申请日期 |
| manager | varchar | 16 |  |  | 直接主管 |
| approve | varchar | 16 |  |  | 审批人 |
| approve\_day | varchar | 16 |  |  | 审批日期 |
| reason | varchar | 204 |  |  | 休假原因 |
| staff\_code | varchar | 64 |  |  | 员工编号 |

表3.13 加班信息表

1. 补打卡信息表（hr\_fill\_card）

包含的所有字段名，长度和数据类型如表3.14所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 主键/外键 | 约束 | 描述 |
| fill\_uuid | varchar | 32 | 主键 | 非空 | uuid |
| fill\_staff\_name | varchar | 32 |  |  | 员工名称 |
| fill\_staff\_code | varchar | 32 |  |  | 员工编号 |
| fill\_staff\_job | varchar | 32 |  |  | 员工职位 |
| fill\_reason | varchar | 255 |  |  | 申请原因 |
| fill\_morning\_time | varchar | 32 |  |  | 早上时间 |
| fill\_afternoon\_time | varchar | 32 |  |  | 下午时间 |
| fill\_applicant | varchar | 32 |  |  | 申请人 |
| fill\_auditor | varchar | 32 |  |  | 审核人 |
| fill\_audit\_time | varchar | 32 |  |  | 审核时间 |
| audit\_status | varchar | 32 |  |  | 审核状态 |
| reject\_reason | varchar | 255 |  |  | 审核原因 |

表3.14 加班信息表

1. 园区信息表（hr\_camp）

包含的所有字段名，长度和数据类型如表3.15所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 主键/外键 | 约束 | 描述 |
| uuid | varchar | 24 | 主键 | 非空 | uuid |
| camp\_uuid | varchar | 24 |  | 非空 | 园区uuid |
| camp\_name | varchar | 64 |  |  | 园区名称 |
| camp\_address | varchar | 256 |  |  | 园区地址 |
| camp\_desc | varchar | 256 |  |  | 园区描述 |
| reg\_name | varchar | 32 |  |  | 登记人 |
| reg\_time | varchar | 24 |  |  | 登记时间 |

表3.15 加班信息表

1. 建筑信息表（hr\_build）

包含的所有字段名，长度和数据类型如表3.16所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 主键/外键 | 约束 | 描述 |
| uuid | varchar | 24 | 主键 | 非空 | uuid |
| camp\_uuid | varchar | 24 | 外键 | 非空 | 园区uuid |
| city\_name | varchar | 64 |  |  | 城市名称 |
| build\_code | varchar | 128 |  | 非空 | 建筑名称 |
| address | varchar | 256 |  |  | 地址 |
| memo2 | varchar | 256 |  |  | 描述 |
| reg\_name | varchar | 32 |  |  | 登记人 |
| reg\_time | varchar | 24 |  |  | 登记时间 |

表3.16 加班信息表

1. 房间信息表（hr\_room）

包含的所有字段名，长度和数据类型如表3.17所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 主键/外键 | 约束 | 描述 |
| uuid | varchar | 24 | 主键 | 非空 | uuid |
| build\_uuid | varchar | 24 | 外键 | 非空 | 建筑uuid |
| corp\_uuid | varchar | 24 |  | 非空 | 公司uuid |
| room\_code | varchar | 64 |  | 非空 | 房间编号 |
| floor | varchar | 12 |  |  | 楼层 |
| manager | varchar | 24 |  |  | 管理员 |
| seat\_rows | varchar | 12 |  |  | 工位行数 |
| seat\_cols | varchar | 12 |  |  | 工位列数 |
| image | varchar | 128 |  |  | 图片URL |
| memo2 | varchar | 256 |  |  | 其他说明 |
| reg\_name | varchar | 32 |  |  | 登记人 |
| reg\_time | varchar | 24 |  |  | 登记时间 |

表3.17 加班信息表

1. 工位信息表（hr\_seat）

包含的所有字段名，长度和数据类型如表3.18所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 主键/外键 | 约束 | 描述 |
| uuid | varchar | 24 | 主键 | 非空 | uuid |
| room\_uuid | varchar | 24 | 外键 | 非空 | 房间uuid |
| seat\_code | varchar | 256 |  | 非空 | 工位编号 |
| seat\_desc | varchar | 256 |  |  | 工位描述 |
| seat\_pos | varchar | 256 |  |  | 工位坐标 |
| row\_index | varchar | 12 |  |  | 行编号 |
| col\_index | varchar | 12 |  |  | 列编号 |
| row\_span | varchar | 12 |  |  | 行数 |
| col\_span | varchar | 12 |  |  | 列数 |
| seat\_type | varchar | 32 |  |  | 工位类型 |
| memo2 | varchar | 256 |  |  | 其他说明 |

表3.18 加班信息表

## 3.3用户界面概要设计

由于系统太大，涉及到的用户太多，这里就大概的列出整个系统的界面设计，分别为：普通员工界面、考勤管理员界面、休假管理员界面、人员管理员界面、用户管理员界面、机构管理员界面、招聘管理员界面、出差与报销管理员界面、固定资产管理员界面、资源池管理员界面、个人信息界面、查询界面、申请界面、考勤打卡界面、考勤审核界面、考勤统计界面、工位设置界面。页面的调用关系如图3.19 系统界面。

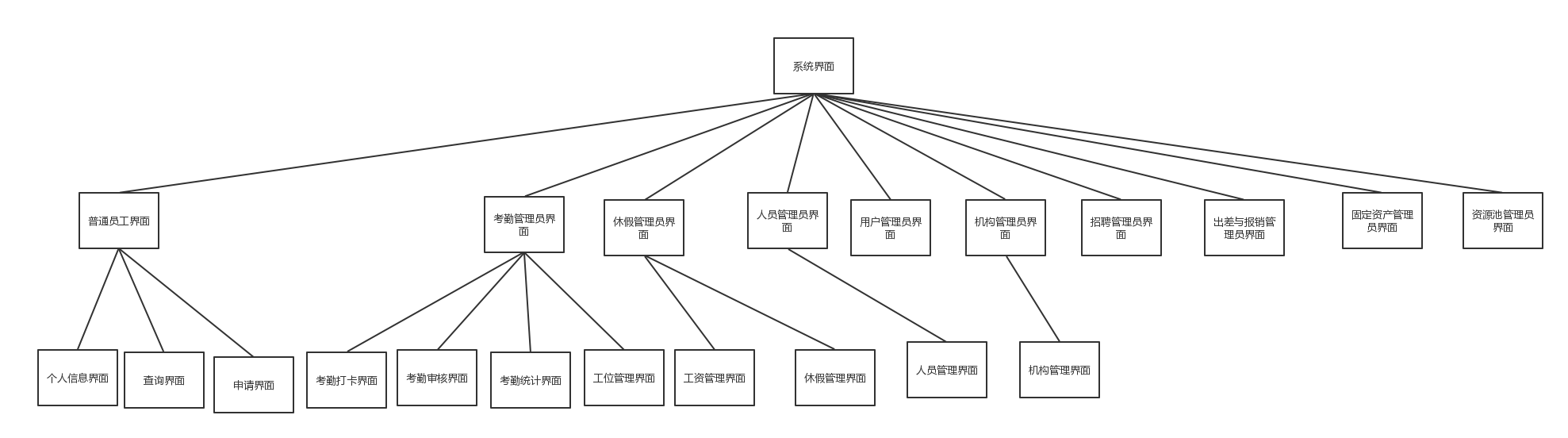


图3.19 系统界面

其中考勤管理员界面是我管理和设计的界面，分别由考勤打卡界面、考勤结果显示界面、导入界面、导出界面、考勤审核界面、加班审核界面、审核界面，待批记录界面、审核结果界面、驳回原因界面、考勤统计界面、月度汇总界面、详情界面、打卡状态修改界面、每日统计界面、打卡时间界面、原始数据界面、工位设置界面、工位管理界面、工位查询界面、建筑管理界面、房间管理界面、工位管理界面。如图3.20考勤管理员界面。

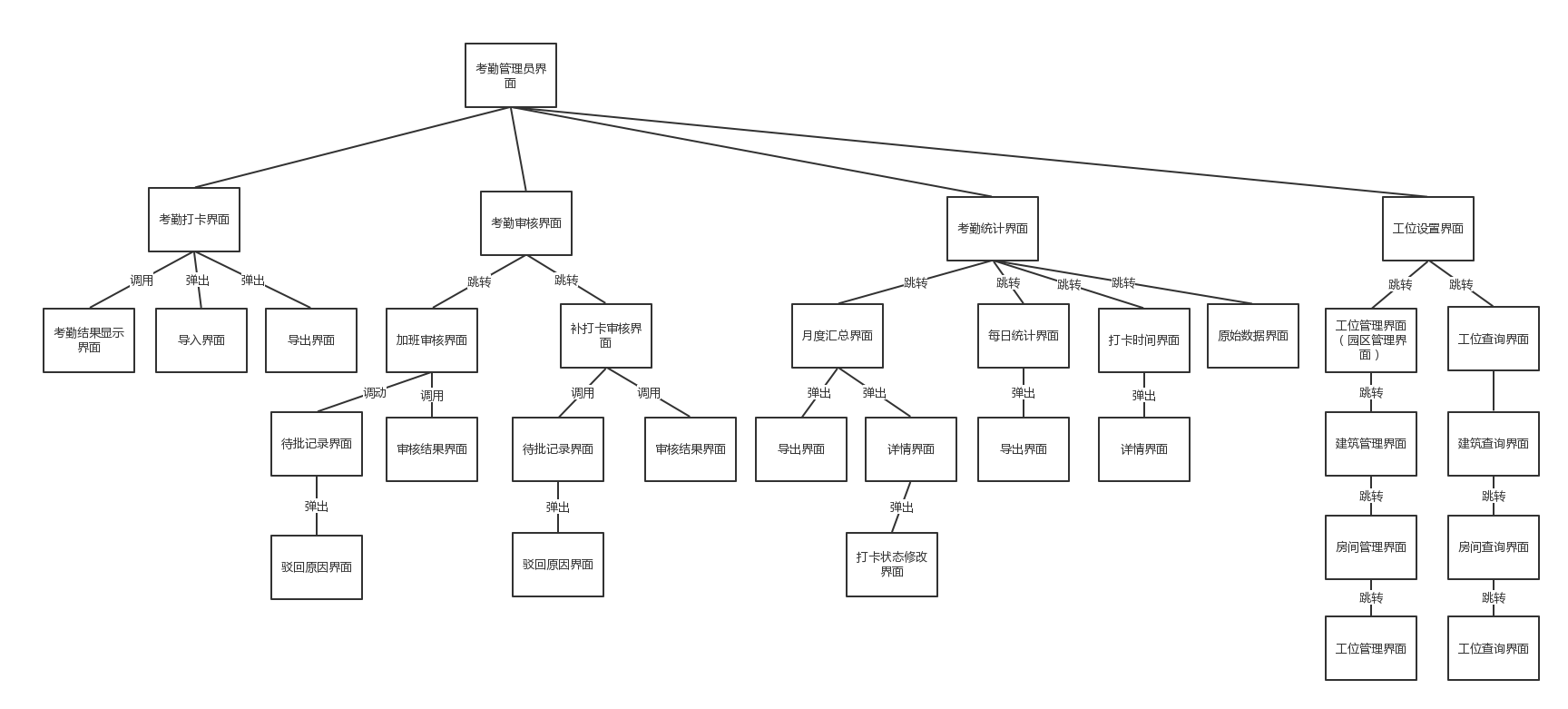


图3.20 考勤管理员界面

# 第四章 详细设计与实现

## 4.1 LZ-HR人力资源管理系统的总体设计

LZ-HR人力资源管理系统主要是按模块分配权限，用户有10类用户，分别是：普通员工、用户管理员、休假管理员、考勤管理员、人员管理员、机构管理员、招聘管理员、出差与报销管理员、固定资产管理员、资源池管理员，当然，所有管理员都是员工，所以都有普通员工的权限。系统主界面如图4.1系统主界面设计。



图4.1 系统界面设计

## 4.2 考勤模块的总体设计思路

由于整个系统的设计过于复杂，功能太多，主要是整个系统是有一个10的团队合作实现的，每个人负责实现一个模块，而我是负责考勤模块的实现，所以我们就详细讲解本模块的设计思路。考勤模块主要由四大功能模块组成，分别为考勤打卡、考勤审核、考勤统计、工位设置，考勤审核则由加班审核和补打卡审核两个功能模块组成，分别对加班申请和补打卡申请做审核的功能，考勤统计则由月度汇总、每日统计、打卡时间和原始数据4个功能模块组成，根据不同需求对考勤数据做不同的处理，工位设置由工位管理和工位查询两个功能模块组成，工位管理主要是对公司工位的录入、设置和管理。如图4.2考勤主界面设计。



图4.2 考勤主界面设计

## 4.3 考勤数据导入功能的实现

考勤管理员登录进入到系统以后可以看到两大模块：个人查询中（普通员工模块）和考勤管理模块，当点击进入考勤模块后，管理员需要把本月打卡机中的数据导出形成Excel后再导入到该模块系统中，让整个模块系统来对数据进行相应的处理。考勤数据导入的活动图4.3所示：

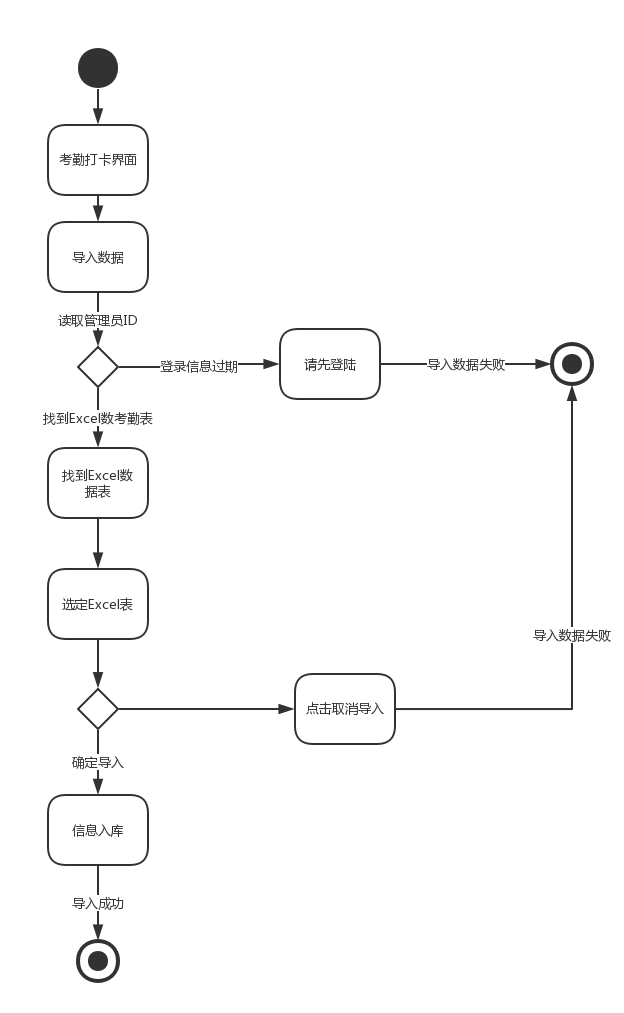


图4.3 考勤导入活动图

考勤数据导入时，先在考勤打卡界面找到导入按钮并点击，系统会先读取管理员ID，若登录信息过期则跳转至登录界面并提示用户重新登录，若登录信息有效弹出选择文件的界面，选择好文件若点击确定则完成导入，若点击取消这导入失败。考勤导入的时序图如图4.4所示：

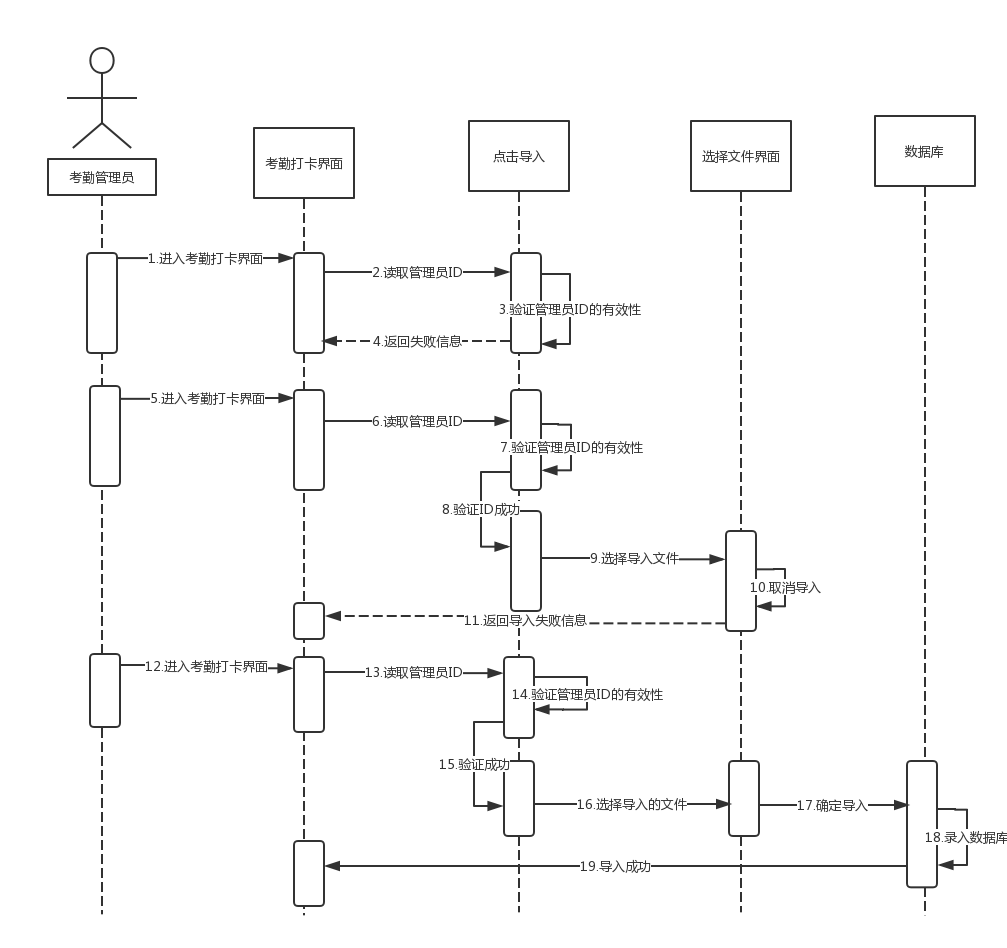


图4.4 考勤导入时序图

考勤打卡的界面如图4.5所示，实现导入功能，前端只是起到触发方法的作用，关键性的实现代码都在后台，所以只要点击考勤打卡界面中左上角的导入图标，就能完成触发导入的功能。



图4.5 考勤打卡界面

触发导入功能后，界面会弹出选择导入文件的界面，如图4.6所示，先找到文件所在地，然后选中文件然后确定，文件就会导入系统。

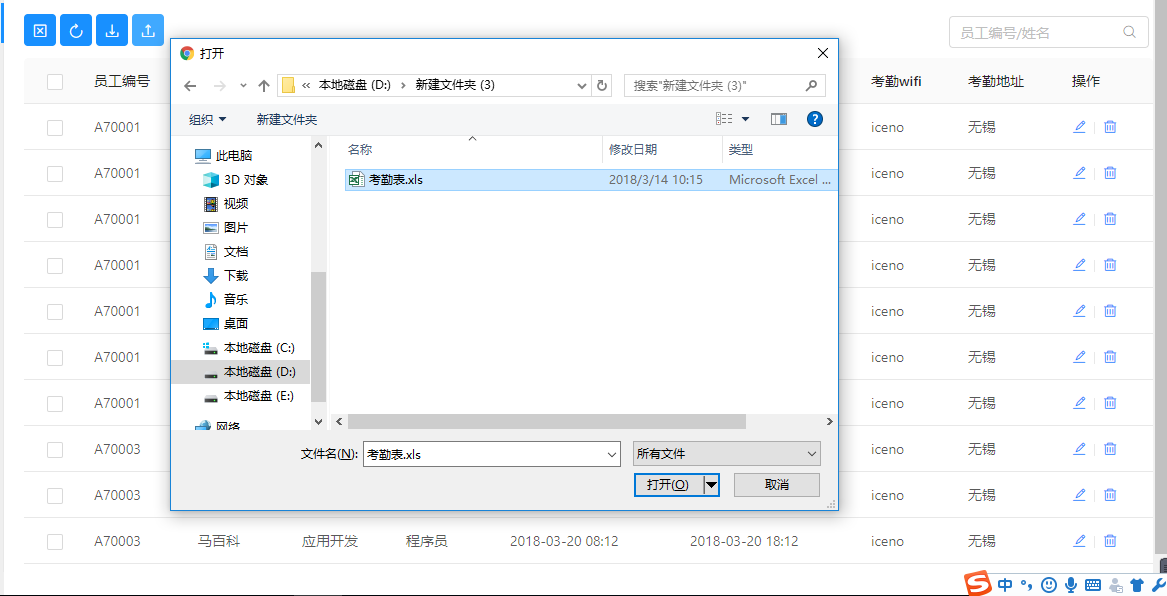


图4.6 选择文件界面

导入功能主要是后台实现，该功能涉及到了文件读写的io流和easyio导入功能，文件的读写是为了实现任何地方的文件都能导入系统的功能，代码如图4.7所示，而easyio则是实现把文件解析成对应实体的功能，其内部也使用到了io流，代码如图4.8所示。

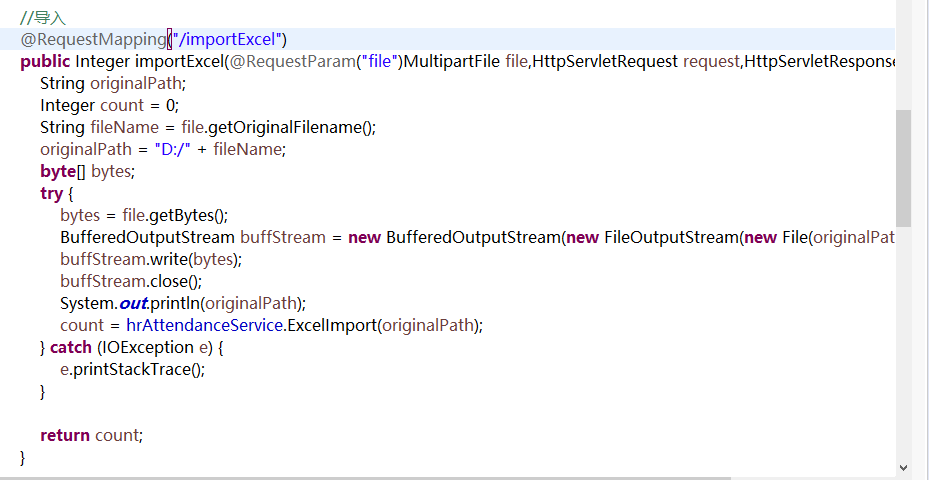


图4.7 io流

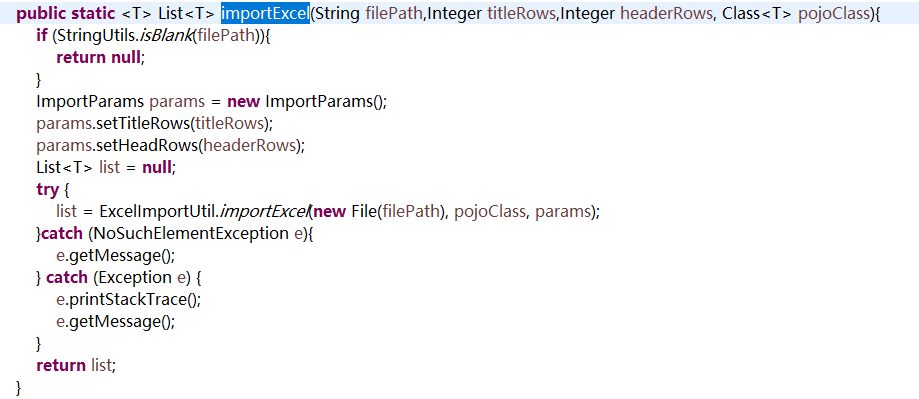


图4.8 easyio代码

## 4.4 考勤加班审核功能的实现

管理员打开加班审核的界面，系统会先读取管理员ID，若登录信息失效则跳转登录见面并提示重新登录。若登录信息有效，则管理员就可正常审核，若同意申请则直接审核成功并修改数据库中申请的状态，若驳回则需要填写原因，确定审核后审核成功并修改数据库中申请的状态。活动图如图4.9所示：

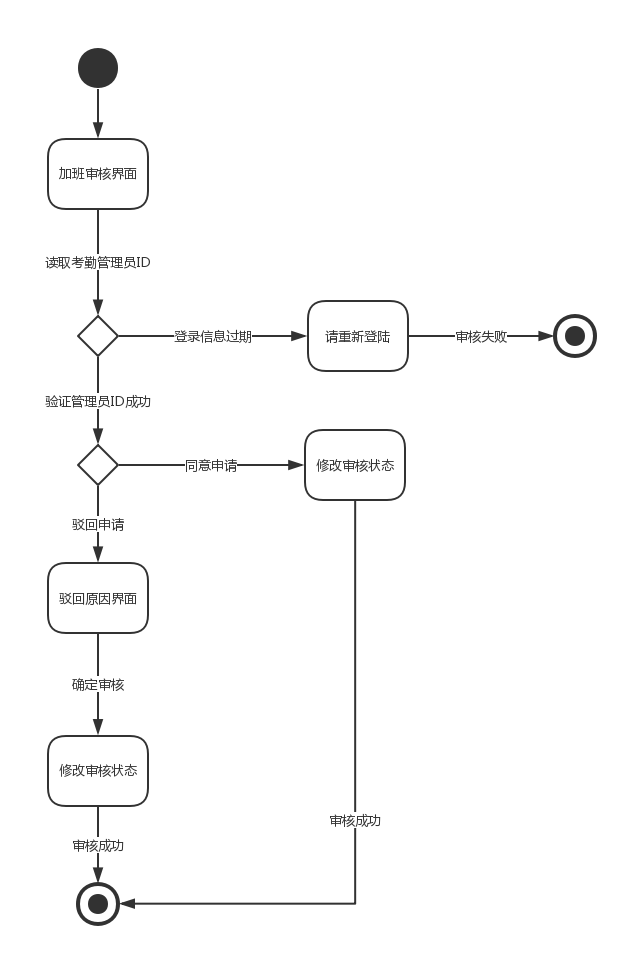


图4.9 加班审核活动图

加班审核功能是管理员向考勤系统提交审核要求，系统先检测管理员ID的有效性，若验证通过则才能进行审核的功能，为了能够更深动简洁的显示出审核功能，其时序图如图4.10所示：

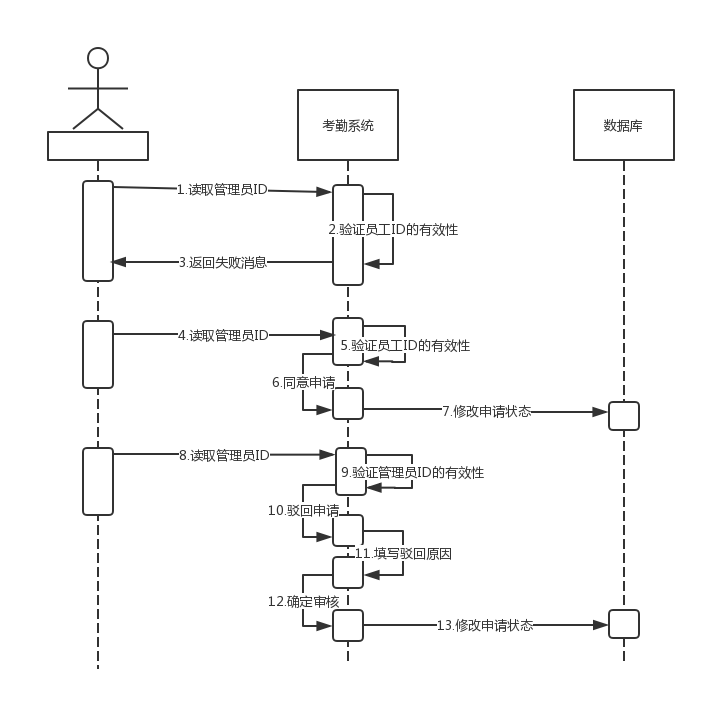


图4.10 审核功能时序图

审核功能的界面如图4.11所示，管理员进入界面，可以选择同意和驳回，驳回需要填写驳回原因。

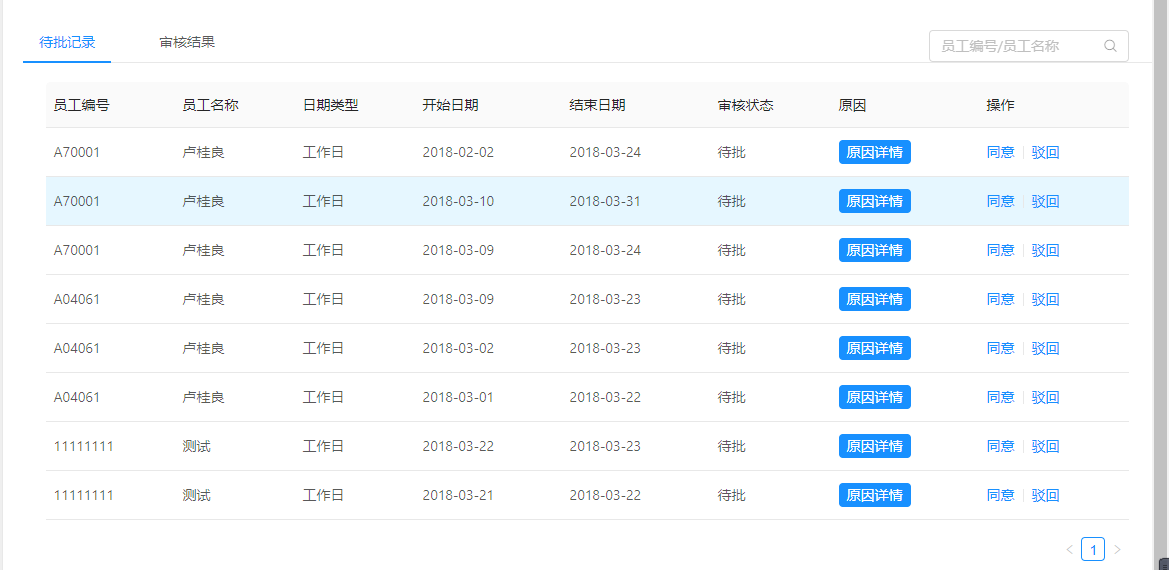


图4.11 审核界面

审核功能代码主要是分前后端，但由于代码太多，其中前端审核功能的代码是用fetch异步请求数据的方法，如图3.12所示，fetch异步请求数据是在不影响界面正常运行的情况下向后台请求数据。后台的代码是对前端的请求做相应的处理，如图3.13所示，若前端发出的是同意请求，则把申请状态改为通过，若前端发出的是驳回请求，则把审核状态改为驳回并添加审核原因进入数据库。



图3.12 前端代码

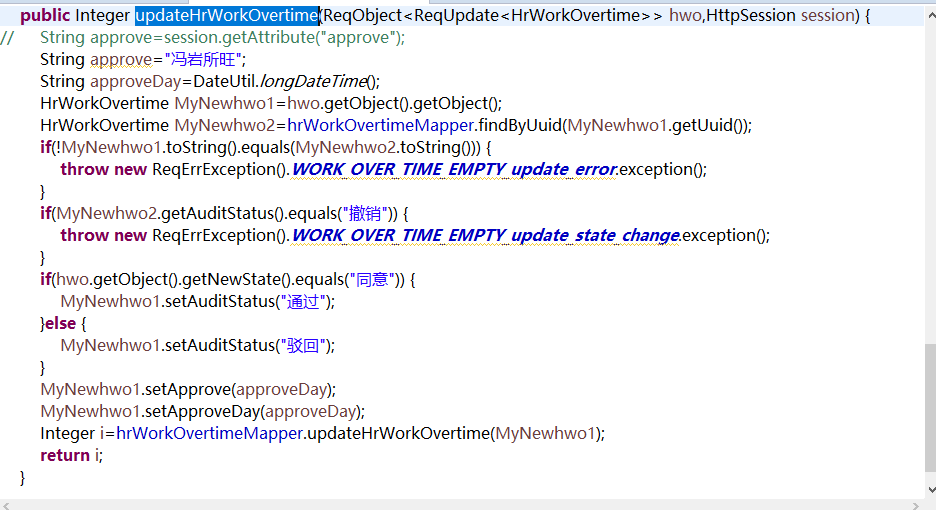


图3.13 后端代码

## 4.5 考勤统计中月度汇总功能的实现

月度汇总是系统对考勤数据进行查询和统计，然后显示到界面以给管理员查看本月的出勤情况，其活动图如图3.14所示。

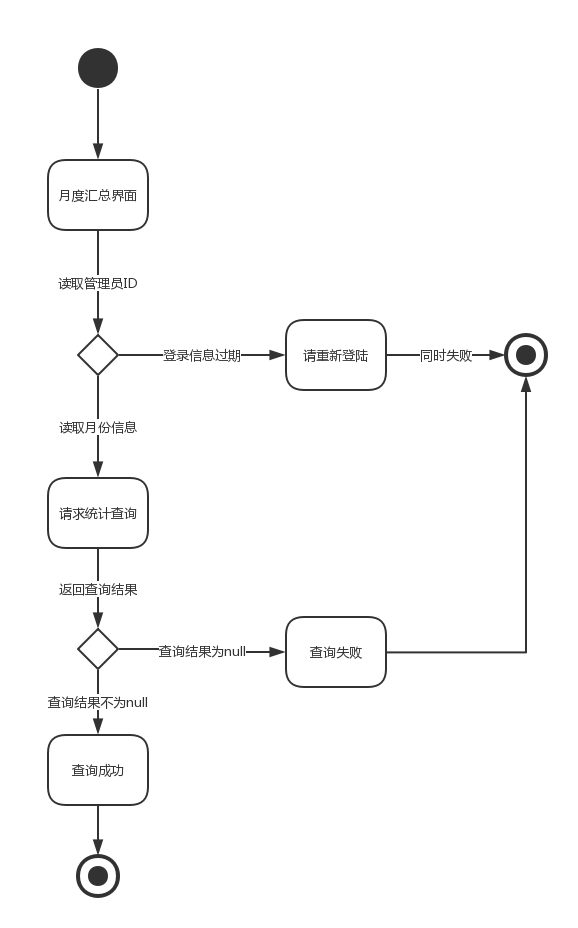


图3.14 月度汇总活动图

管理员先进入界面，系统会读取其ID，若已过期则跳转登录界面并提示重新登录。若验证通过则会向系统发送统计查询的请求，系统会读取月份参数并查询相应的考勤信息并做统计，之后给管理员反馈到统计界面。如图3.15所示。

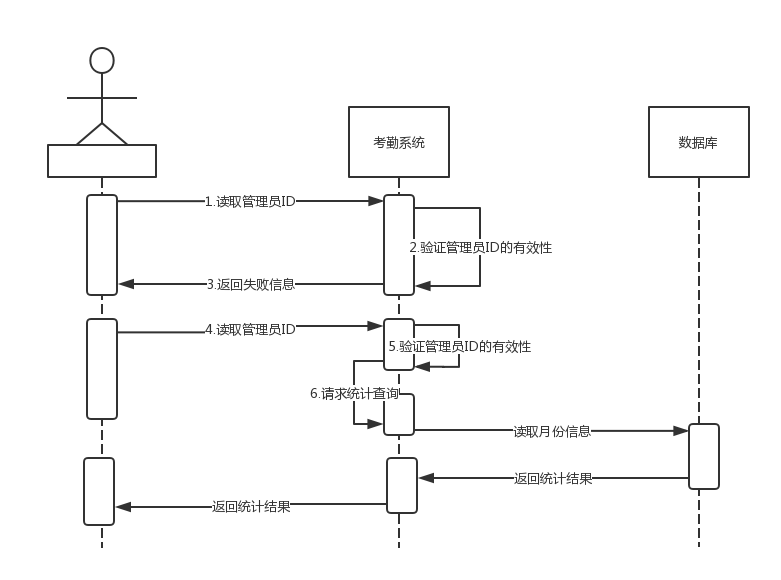


图3.15 月度汇总时序图

月度汇总的界面如图3.16所示，系统会把统计后的结果返回到前端界面，以表的形式展示给管理员，管理员还可以把表下载下来。



图3.16 月度汇总界面

月度汇总的前端代码如图3.17所示，主要涉及日期格式的转换和数据的异步请求，由于后端需要时间格式问题，需要在前端写一个日期转换的方法，而fetch异步请求则是对数据的更新。



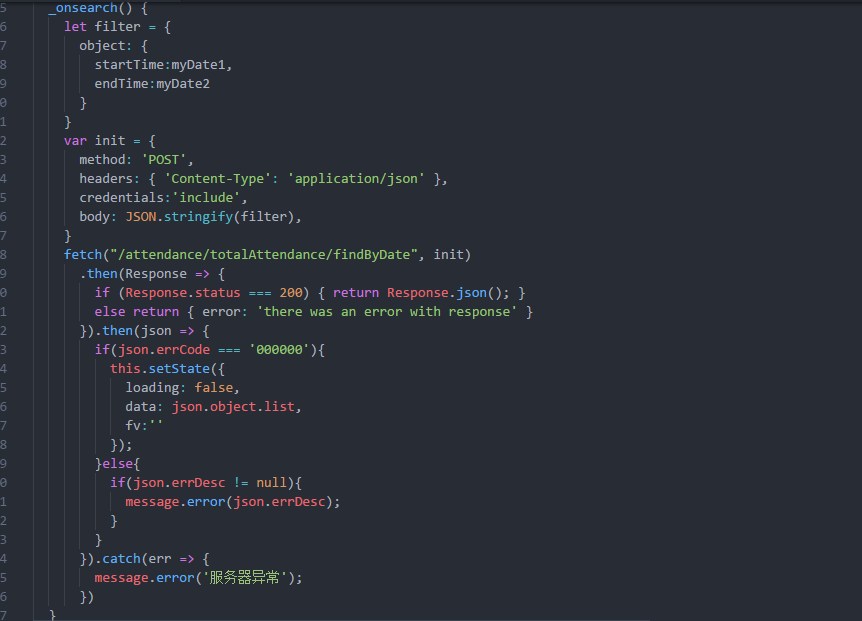
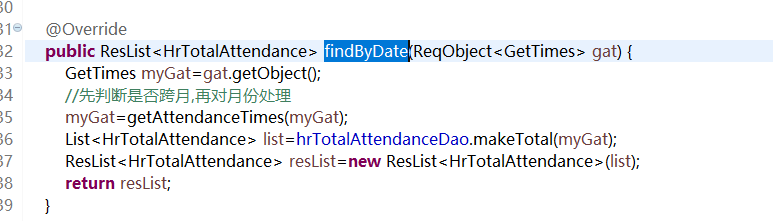


图3.17 前端代码

月度汇总的代码如图3.18所示，其中最复杂的是在数据库查询语句，由于代码过长无法全部显示，只能展示部分代码。



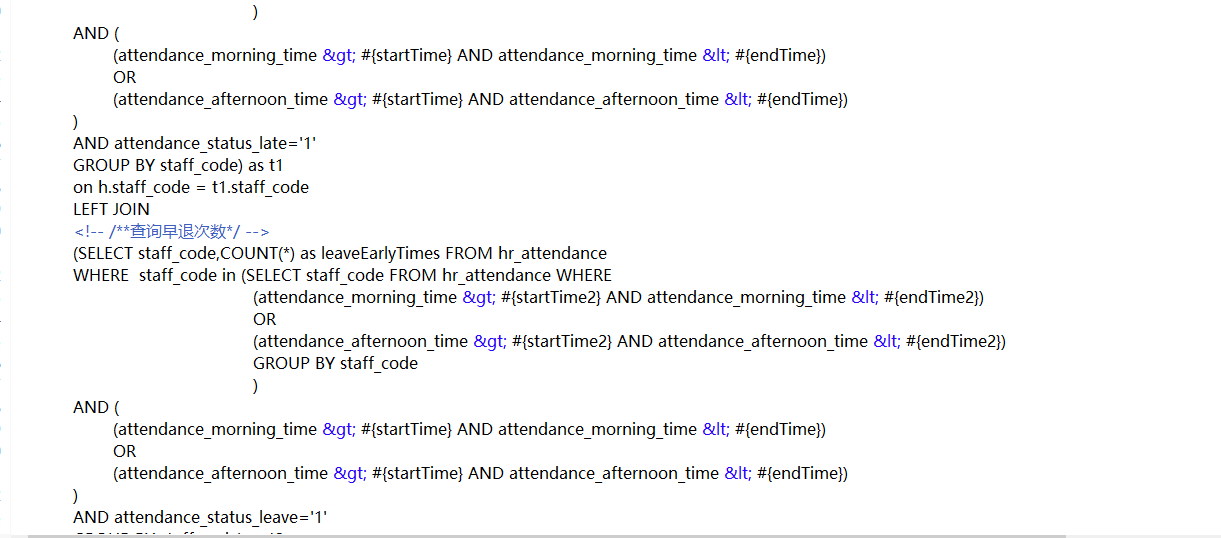
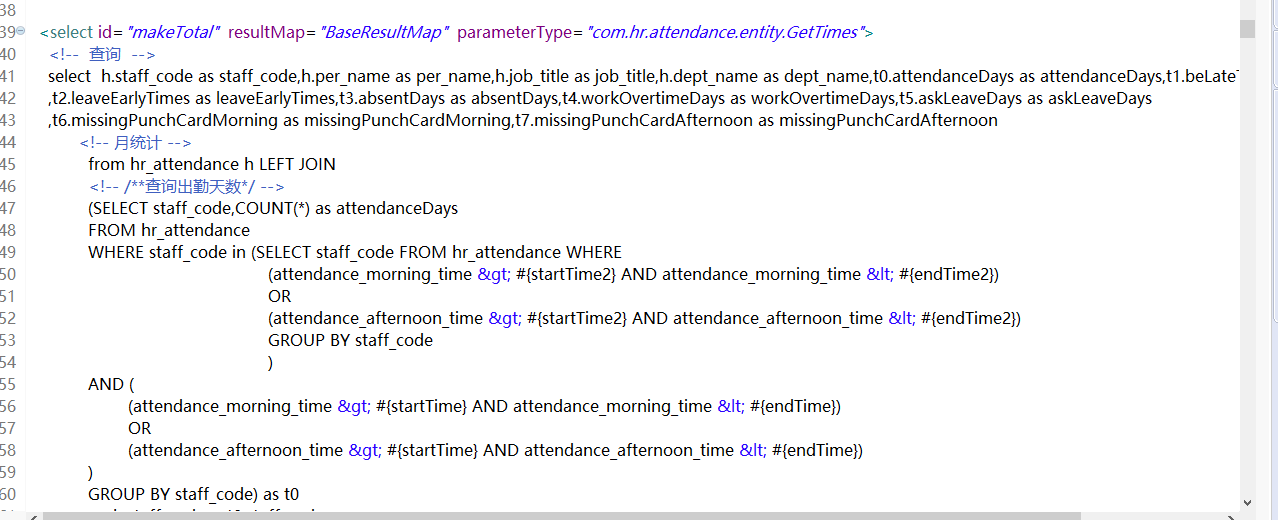


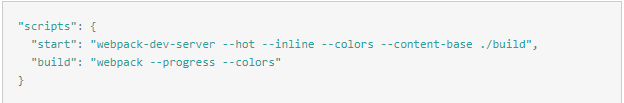
图3.18 后台代码

# 第五章 遇的问题及解决办法

问题一：如何搭建一个react项目？搭建项目之前需要安装什么软件？搭建成功以后如何与后台接口连接？连接过程中如何解决跨域问题？

首先在搭建react项目之前，需要安装node.js，它是一个js（javaScript）的运行环境，而react是一个js库，是对js语言的一种封装。首先我先来说明一下为何要使用react来写前端代码,react是目前前端最好的javaScript mvc框架之一，它的性能优越，更重要的是它的代码逻辑简单，同时也是为了响应当前兴起的前后端分离的思想，我们就采用react语言来完成前端代码的书写。

那如何搭建一个react的项目呢？这就得做很多的准备了，首先我们需要新建一个文件夹，在cmd命令窗口中进入到该文件，然后第一步就是输入命令行“nupm init”来初始化生成一个名为“package.json”的文件，该文件主要的作用是用来装各种依赖包；第二步是安装各种依赖，比如react、webpack、babel-core等等一堆依赖，而这些依赖是reac项目不可或缺的东西；第三步就是配置，首先我们需要在package.js中添加几行代码：



再在命令行中输入 “npm build”将会进行生产环境打包。

然后我们还需配置入口文件和出口文件，也就是启动webpack，当然由于需求不同配置的方法也不同，这里就使用默认的配置方法。我们需要到文件夹里新建一个webpack.config.js的文件，然后再文件中我们需要写很多的配置，首先要有入口，代码为“entry: './src/app.js'”，还要有出口，代码为：

“output: {

path: \_\_dirname + '/build',

filename: "bundle.js"

}”

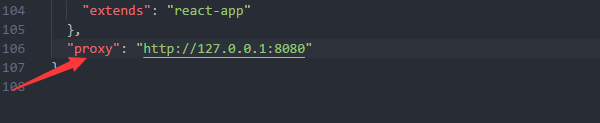
同时还需要一些其他配置，总的配置文件如下图所示：



配置完这些以后，我们的react项目基本上已经搭建成功了，之后就是填代码文件。

当然，有于搭建react项目的步骤太繁琐，自己搭建的项目问题太多，无法满足我们正常的开发需求，所有我们使用了react的脚手架，该脚手架已经把完整的react项目搭好，我们只需要改一些入口和出口的路径问题和导入yixie 我们需要的依赖，就可以直接投入到项目的开发过程中了。

项目搭建好了，代码也书写好项目代码以后，我们就需要向后台发送请求了，在这一过程中我们就会涉及到两个域，前端的一个域和后端的一个域，这时就会出现跨域的问题，由于node.js这个引擎自带了Nginx反向代理服务器，所以我们只需要在package.json文件中加入一句话：

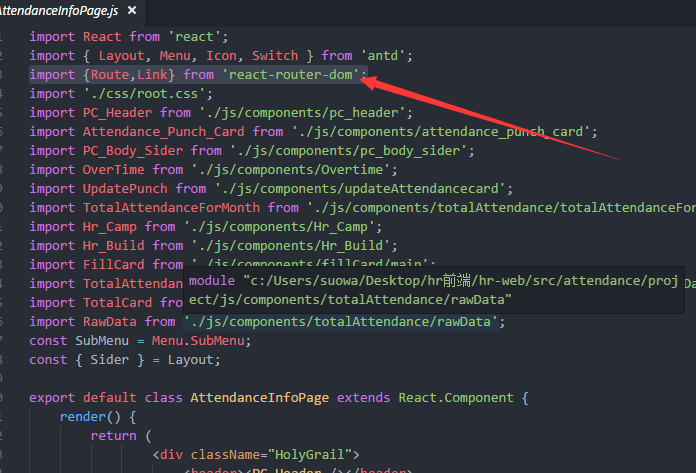


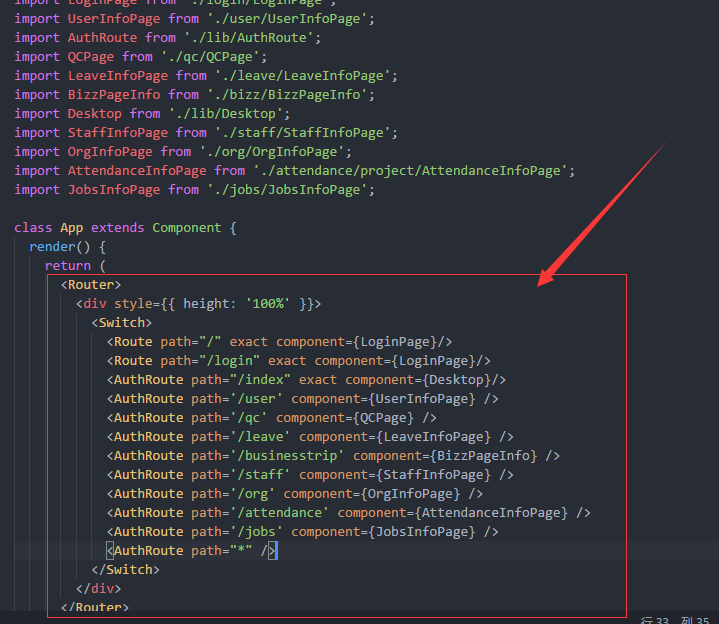
就能很好的解决了该问题，当然，由于我们的项目不是太复杂，反向代理的配置简单，当涉及复杂的跨域问题时，反向代理的配置就会变得复杂。

而跨域问题一解决，那么前后端与后台的连接就变得很简单了，前端只要异步向后台发出请求，然后后台响应并把数据交给前台。

问题二：前端react路由的如何配置和使用？

首先在react中，页面是以一个个组件的形式存在的，而页面之间的跳转是用路由来实现的。想要实现路由，我们就需要往项目中引入路由的依赖react-router-dom,引入的方法也简单，只需要在命令窗口运行npm install react-router-dom就能把依赖引入进来，而要实现路由的功能则需要写相应的代码：



首先，要先往组件里面引入路由：import {Route,Link} from 'react-router-dom'，然后用<Router>把所有组件都包起来，用<Route>来指定每个组件的跳转路径。

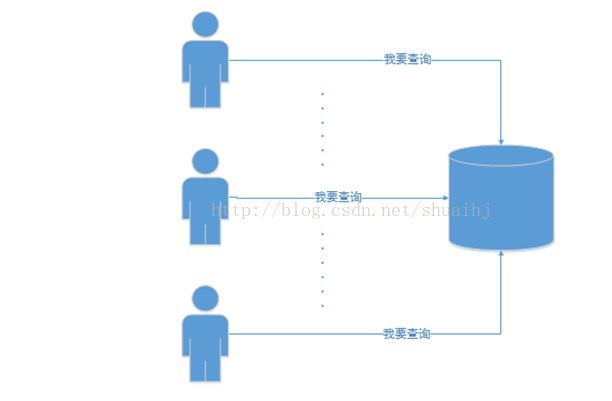


然后用<Link>来指定怎么触发跳转。

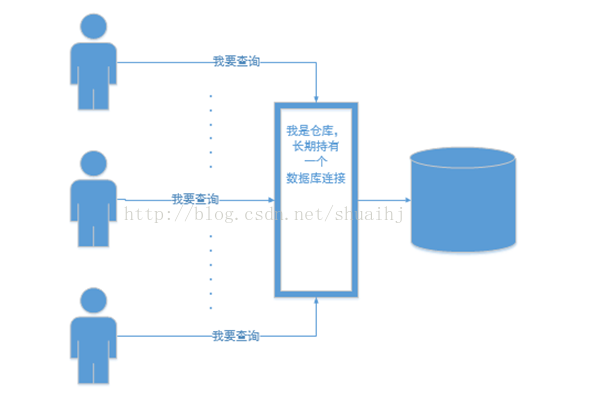
问题三：如何使用c3p0来配置连接池？

由于项目是有多个用户来访问的，数据库的连接也就会变得很频繁，如果使用spring自带的连接池的话会无法满足需求，所以需要配置c3p0连接池来管理数据库的连接，首先我们先讲一讲c3p0连接池的原理：

在正常情况下，我们每次请求会与数据库建立连接，每次用完又得断开连接，要是500个人同时用系统并每个人都发出多个请求，那么你可以想到后果，那就是数据库内存泄漏，导致数据库服务崩溃，需要重启数据库又才能重新运行项目，如下图所示就是正常情况下的连接。



而c3p0数据库连接池则是解决该问题的方法之一它的连接，它的连接原理图如下图所示：

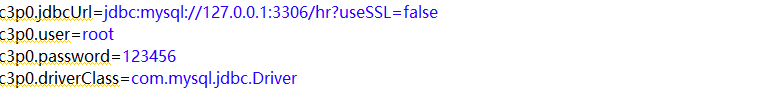


我们建立一个数据连接池，不断开与数据库的连接并设置他长期拥有几个数据库连接，若无人请求的情况下数据库连接就处于缓存的状态留在数据库连接，它保持着与数据库的连接不断开，当有人发出请求时则会自动分配数据库连接给他，用完以后又返回到数据库连接池中，这样我们只要设置数据库的最大连接池和最小连接池等变量，就能控制好数据库的连接问题。

C3p0数据库连接池的使用需要先在pom.xml中引入依赖并导入包，其依赖为：



然后在application.properties文件中配置参数，由于需求不同，需要的参数不同，我就只写了自己需要的依赖：



然后就可以正常使用了。

**致谢**

通过了一个多月的学习和论文的书写，在这一过程中我遇到了很多的问题和无数的困难，是身边的同学、实习过程中的同事、学长和老师们，让我能够非常好的解决问题，帮助我突破重重困难，突破我一次又一次的极限，这一个月不只是完成了论文的书写还是知识海洋的中的一次旅行。

在这里我想要由衷的感谢于千城老师，感谢于老师在工作繁忙的情况下对我论文的指导，感谢于老师不分昼夜的给我指导修正论文的错误，