考勤系统设计报告文档

****

**学 院\_ 智能与计算学部\_**

**专 业\_\_ 软件工程\_\_\_\_**

**组 员\_\_\_ 李济孚\_\_ \_**

**组 员\_\_\_ 李嘉浩\_\_ \_**

**组 员\_\_\_ 陈子贺\_\_ \_**

**组 员\_\_\_ 张赫 \_\_ \_**

**年 级\_\_\_\_ 2018\_\_\_\_\_\_**

**班 级\_\_\_\_\_\_ 5\_\_\_\_\_\_\_**

目录

[考勤系统设计报告文档 1](#_Toc41490802)

[一、 系统概要设计 3](#_Toc41490803)

[二、 数据库设计 4](#_Toc41490804)

[三、Controller层设计 9](#_Toc41490805)

[1. ApprovalController 9](#_Toc41490806)

[2. AuthorityController 11](#_Toc41490807)

[3. DeptController 12](#_Toc41490808)

[4. ManageController 13](#_Toc41490809)

[5. ViewController 14](#_Toc41490810)

[四、Service层设计 15](#_Toc41490811)

[1. ApprovalServiceImpl 16](#_Toc41490812)

[2. DeptServiceImpl 17](#_Toc41490813)

[3. ManagaServiceImpl 19](#_Toc41490814)

[4. AuthorityServiceImpl 21](#_Toc41490815)

[5.OtherServiceImpls 21](#_Toc41490816)

[五、Entity和Dao层设计 22](#_Toc41490817)

[1. Entity设计 22](#_Toc41490818)

[2. Dao层设计及其mapper映射xml文件 23](#_Toc41490819)

[1） ApprovalDao 24](#_Toc41490820)

[2） MangageDao 26](#_Toc41490821)

[3） RecordDao 28](#_Toc41490822)

[4）DeptDao 28](#_Toc41490823)

[5）UpLoadDao 28](#_Toc41490824)

[6）AuthorityDao 29](#_Toc41490825)

[7）DataProceDao 29](#_Toc41490826)

[8）AttendDao 29](#_Toc41490827)

[六、实现效果 30](#_Toc41490828)

[1. 鉴权 30](#_Toc41490829)

[2. 登录 30](#_Toc41490830)

[3. 员工管理 31](#_Toc41490831)

[4. 考勤记录（上班记录） 32](#_Toc41490832)

[5. 考勤管理（签到早退情况） 33](#_Toc41490833)

[6. 权限管理 33](#_Toc41490834)

[7. 审批管理 33](#_Toc41490835)

[8. 部门管理 35](#_Toc41490836)

[9. 数据导入 36](#_Toc41490837)

# 系统概要设计

本系统主要分为Controller层，Service层，Dao层三层设计，主要运用springboot及jdbc等技术，Dao层实现对数据库的增删查改操作，Service使用Dao层提供的接口并为Controller层提供接口，Controller层方法主要是调用Service层提供的接口。在调用时Service会进行相关的操作同时调用Dao层的接口实现数据库的操作，最后将结果传回给Controller层。

功能方面见于需求分析文档，主体功能实现代码在Service层，Controller层只是执行获取数据并将数据作为传参调用Service层接口的操作。Dao层也只是给Service层提供修改数据库的接口，本身并无功能代码。Dao层功能实现代码主要通过映射的方式，在resources文件夹下建立mapper文件夹，再在内编写相关xml文件映射定义Dao层对数据库的具体操作。整个系统分为不同层对应的package，其中只有Service的实现类（serviceImpl）有具体的操作代码，其他包内大部分类只提供接口和调用其他层方法。（utils内为启动类）

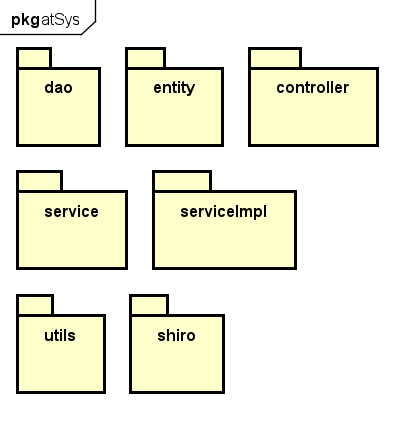


图1-1 考勤系统内部package

用户鉴权使用了Apache Shiro作为安全框架（**有参考源代码**），定义了shiro Config和Realm，用户的权限储存在本地数据库当中，Realm从本地数据库获取数据，在用户登录时将相关数据提供给Shiro，这样限制了用户在后续所能执行的操作，当权限不足时会跳转到设定好的页面提示权限不足。

定义html文件得到网页前端，定义相应的脚本获取后台数据库数据（**有参考源代码**），登录界面依靠web获取数据。在Controller层的方法内添加相应注释（@RequestParam）去获取http传入数据作为传参实现前后端交互。同时运用thymeleaf组件进行前端页面的跳转。

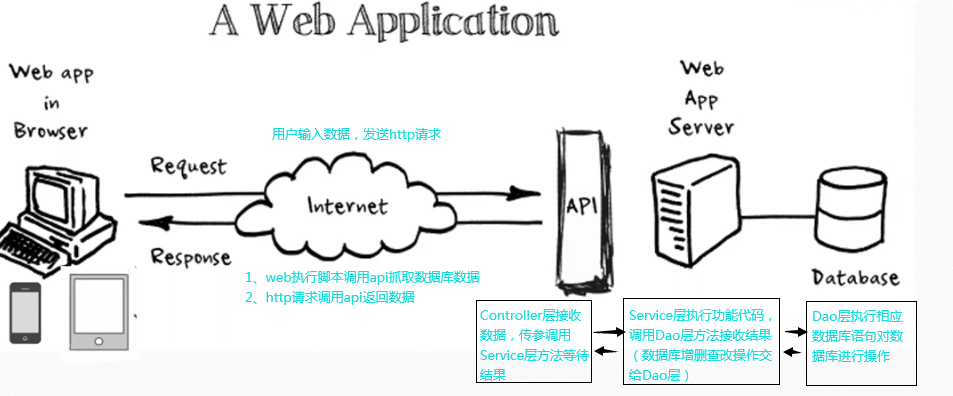


图1-2 系统前后端交互概述图

# 数据库设计

1.attendRecord表



该表中存储的数据为员工的打卡记录。

2.dept表



用于存储公司部门信息。

3.Detailed表



用于存储员工详细的工作时间。

4.dataProce\_t表



此表用于统计员工的考勤相关的信息，包含员工姓名、部门、工作时长、迟到情况、加班情况等信息。

5.Emptable表



用于存储员工信息的数据库表，存储有员工姓名、性别、部门等基础的员工信息。

6.leaveTable表



此表用于存储员工的请假申请，包括请假类别、请假起始时间、结束时间、审批人和审批状态等信息。

7.resource表



此表用于存储前段功能，及相关的URL信息。

8.user\_resource表



此表用于存储用户资源，包括用户id和用户的资源id。

9.workTimeTable表



此表用于存储员工的工作时间，包含当天工作的具体时间。

# 三、Controller层设计

在Controller层中，我们方法都是调用Service层提供的接口，在方法中设定传入参数的值和类型，采用@RequestParam去获取http请求传入的参数作为方法参数，并用@ResponseBody将返回的结果直接写入 HTTP 响应正文中，

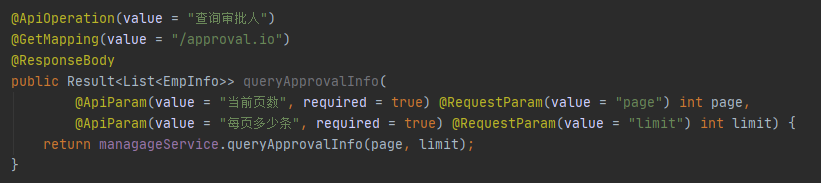
## ApprovalController

ApprovalController作为审批管理的类，调用了ManagageService，ApprovalService，AuthorityService接口作为私有变量。定义网页端访问链接为/approval，Result为网页端返回http请求情况。

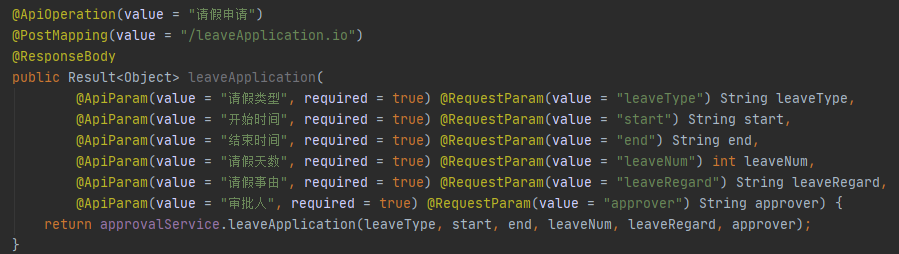


ApprovalController是员工查询自己请假申请及相关情况的类，分别定义了以下功能：

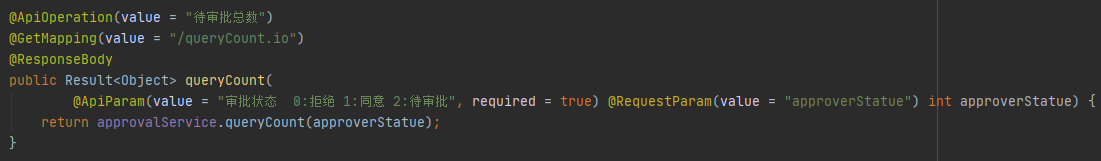
查询审批人功能：调用mangageService接口中的queryApprovalInfo方法进行查询。



请假申请功能：调用approvalService接口中的leaveApplication方法进行查询。



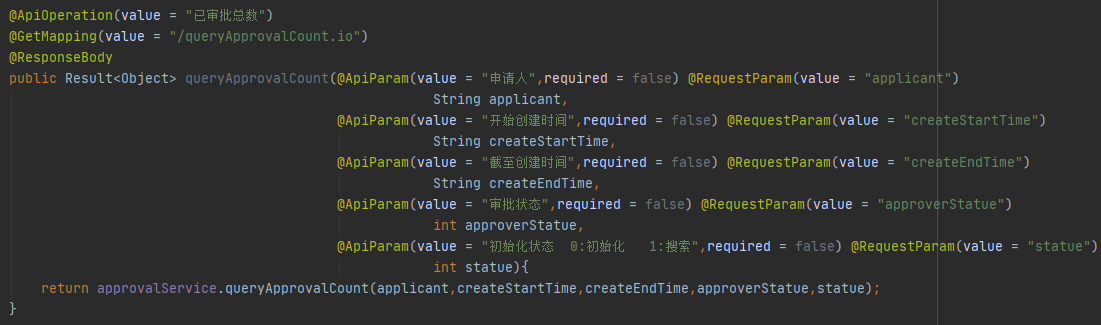
查询待审批总数功能：调用approvalService接口中的queryCount方法进行查询



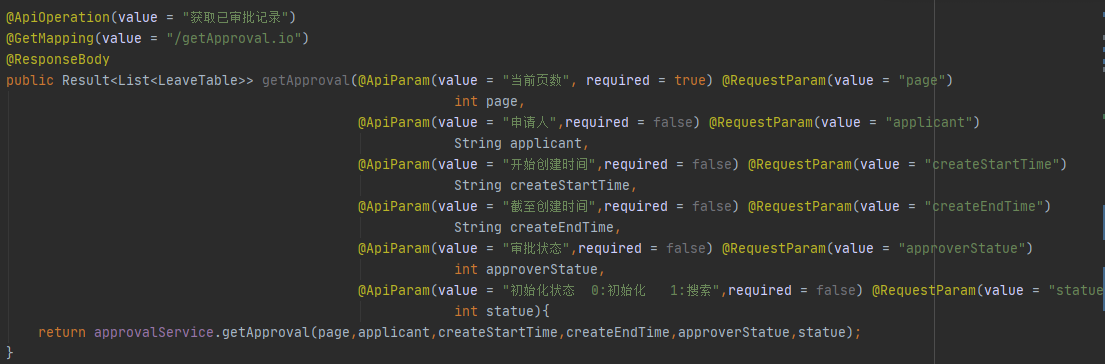
获取待审批记录功能：调用approvalService接口中的getPendingApproval方法进行查询



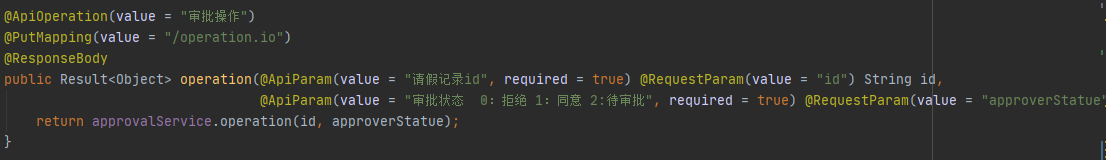
查询已审批总数功能：调用approvalService接口中的queryApprovalCount方法进行查询



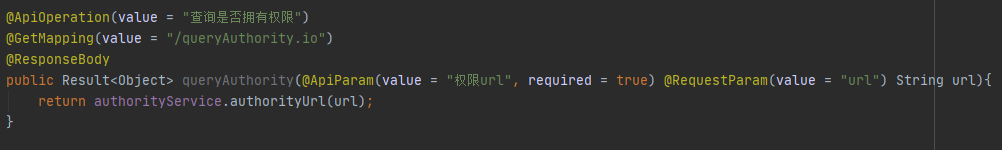
获取已审批记录功能：调用approvalService接口中的getApproval方法进行查询



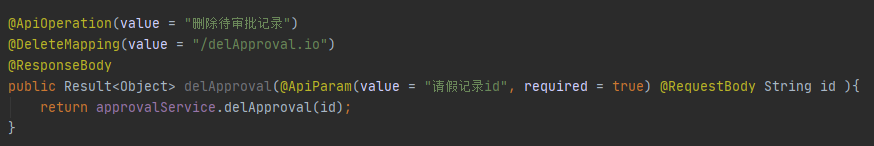
审批操作功能：调用approvalService接口中的operation方法进行查询



查询是否拥有权限功能：调用authorityService接口中的authorityUrl方法进行查询

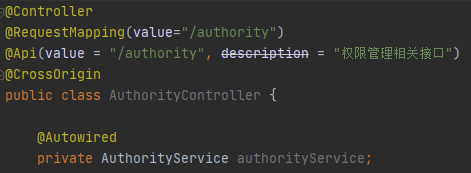


删除待审批记录功能：调用approvalService接口中的delApproval方法进行查询



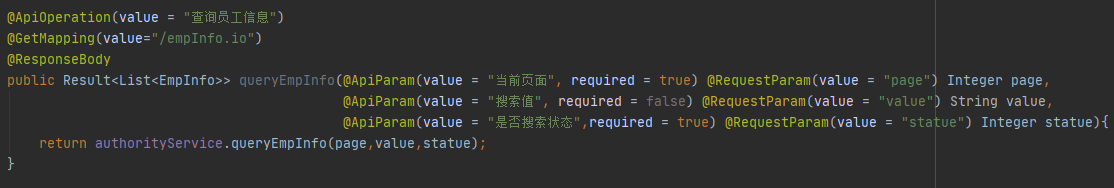
## AuthorityController

AuthorityController作为权限管理的类，调用了AuthorityService接口，定义网页端访问链接为/authority



AuthorityController是最高级别管理者（admin）所应用的类，定义了查询员工信息，查询员工总数，添加权限和查询权限的功能。

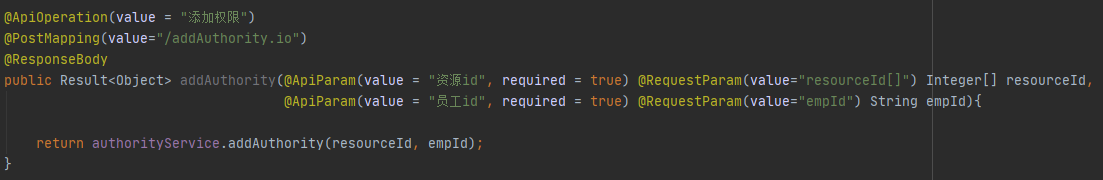
查询员工信息功能：调用AuthorityService接口的queryEmpInfo方法



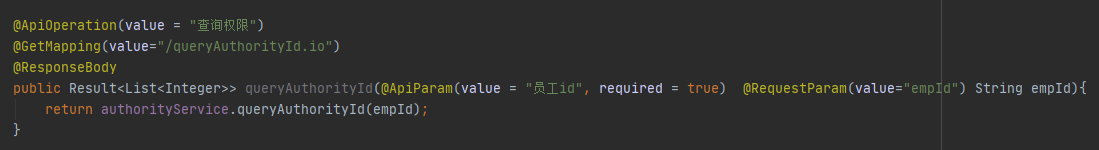
查询员工总数功能：调用AuthorityService接口的queryCount方法



添加权限功能：调用AuthorityService接口的addAuthority方法



查询权限功能：调用AuthorityService接口的queryAuthorityId方法



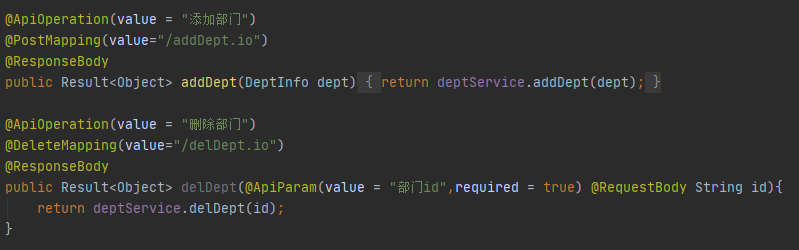
## DeptController

DeptController作为部门管理的相关类，调用了DeptService和AuthorityService接口，定义网页端访问链接为/dept

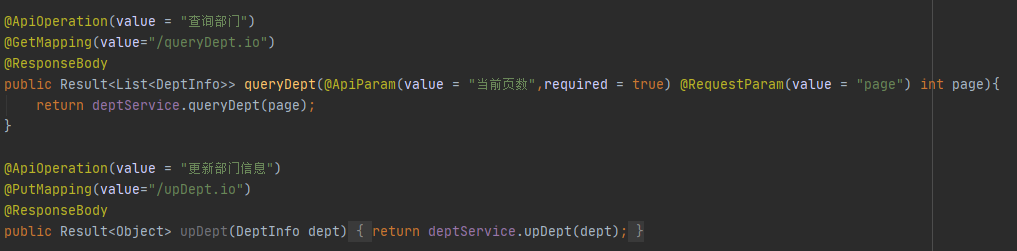


只有有部门管理权限的员工才可以进行相关的操作，定义了与部门操作相关的功能：

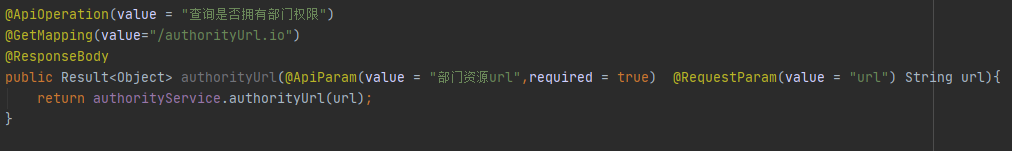
添加，删除部门：调用deptService接口的addDept和delDept方法



查询，更新部门信息：调用deptService接口的queryDept和upDept方法



查询是否拥有部门管理权限：调用authorityService接口的authorityUrl方法



查询部门总数，部门id，根据id查询部门信息：调用deptService接口的queryCount，queryDeptInfo和getDeptInfoById方法



## ManageController

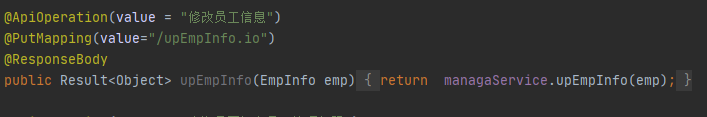
ManageController为员工管理类，调用了ManagaService和AuthorityService接口，定义网页端访问链接为/mangage



员工管理需要登录账户有员工管理权限才能进行相关的操作，定义了员工信息的相关操作。

增删查改员工信息：调用了managaService的queryDeptEmpInfo，delEmpInfo，addEmp和upEmpInfo方法





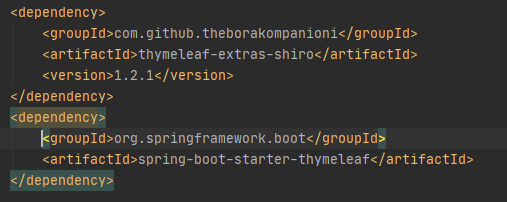
员工总数及修改员工密码：调用了managaService的queryDeptEmpInfo，delEmpInfo，addEmp和upEmpInfo方法



## ViewController

ViewController负责不同页面之间的跳转，根据我们使用的springboot，我们再运用了thymeleaf进行页面的跳转，html网页我们定义在templates文件夹中，在pom中添加了Thymeleaf组件，在ViewController设定好了各个网页的返回值以实现最终的动态页面跳转。

pom文件部分：



ViewController：



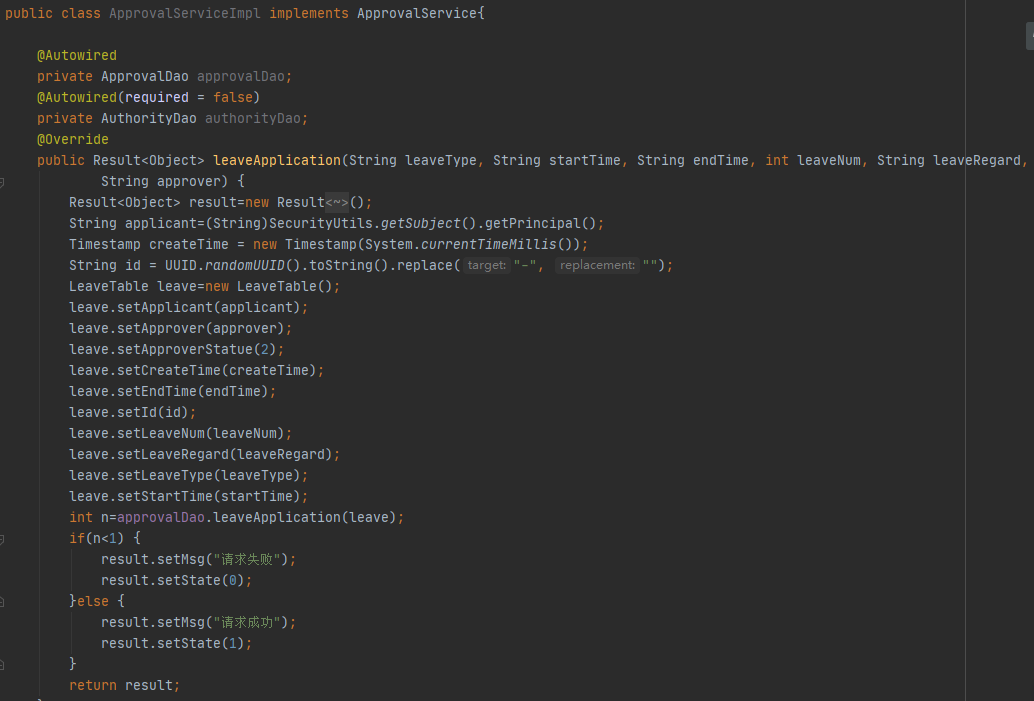
# 四、Service层设计

Service层作为中间层，既使用Dao层提供的接口，同时又向Controller层提供接口，用Entity作为不同层之间传递的参数进行相应的修改。我们在Service层定义了其实现类，在接收到Controller传来的数据后进行封装，让其像Dao层请求数据进行操作并返还结果给Controller层。这里**展示主要的Service接口的实现类，Service接口类和功能较边缘类忽略。**

## ApprovalServiceImpl

ApprovalServiceImpl调用了ApprovalDao和AuthorityDao接口进行请假审核的相应操作并鉴定权限，将权限applicant作为参数传给Dao层进行鉴权，功能如下：

1.请假申请，得到Controller层传入数据后即进行封装调用Dao接口进行申请（修改数据库），根据返回结果判断是否申请成功（**后述功能思想基本相同**）。



2.查询请假申请以及查询待定请假申请



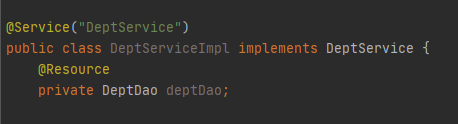
3.查询已批准请假申请



4.对请假申请进行审批操作或删除操作



## DeptServiceImpl

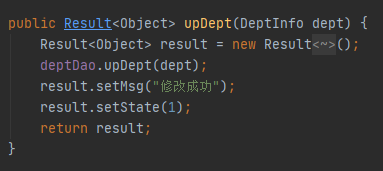


DeptServiceImpl为部门管理实现类，调用了DeptDao对部门表进行操作，功能如下：

1.对部门的增加删除，查询部门信息



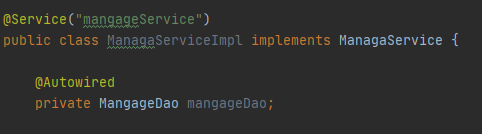
2.修改部门信息



3.获取所有部门信息，或根据id搜索对应部门信息



## ManagaServiceImpl

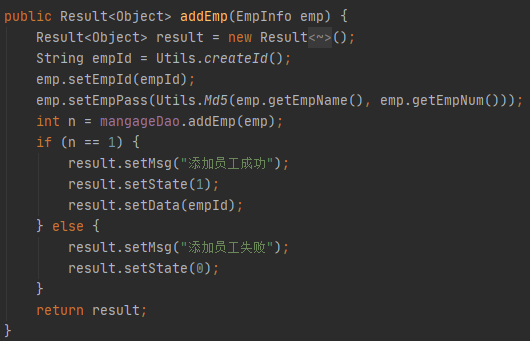


ManagaServiceImpl类为员工管理Service的实现类，调用了MangageDao接口，功能如下：

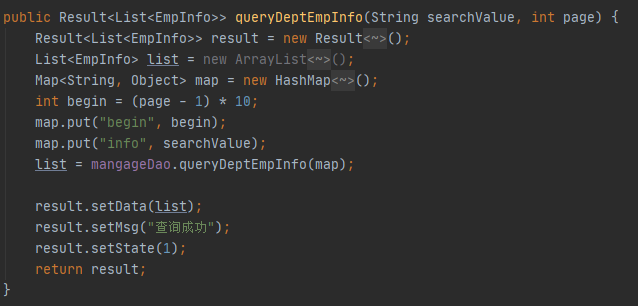
1.登录验证（初始页面登录功能）



2.添加员工信息，根据员工名字和number生成密码（后可能舍弃md5加密功能）



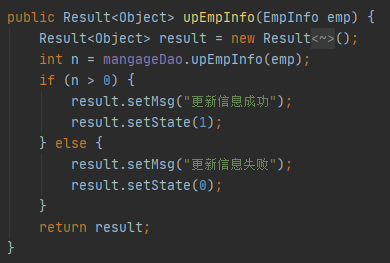
3.查询所有员工信息



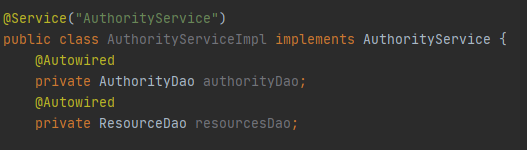
4.删除员工信息



5.更改员工信息

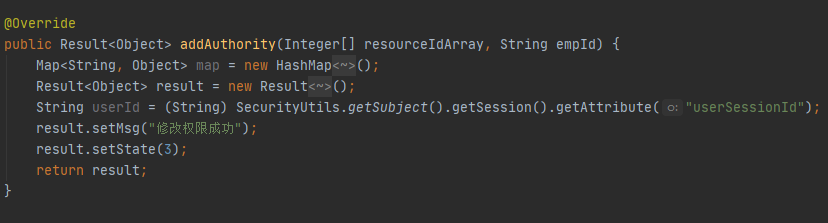


## 4. AuthorityServiceImpl

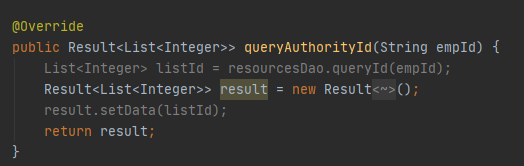


AuthorityServiceImpl作为实现鉴权Service类，调用了AuthorityDao和ResourceDao接口去获取Entity和数据库数据和相应权限判断。相应功能如下：

1.根据员工id添加指定员工的权限



2.查询所有员工的权限情况



3.根据id查询某个员工的权限



## 5.OtherServiceImpls

剩下的非主要的类及其功能在此列出不做详细的代码截图：

AttendServiceImpl：记录员工打卡情况的Service实现类

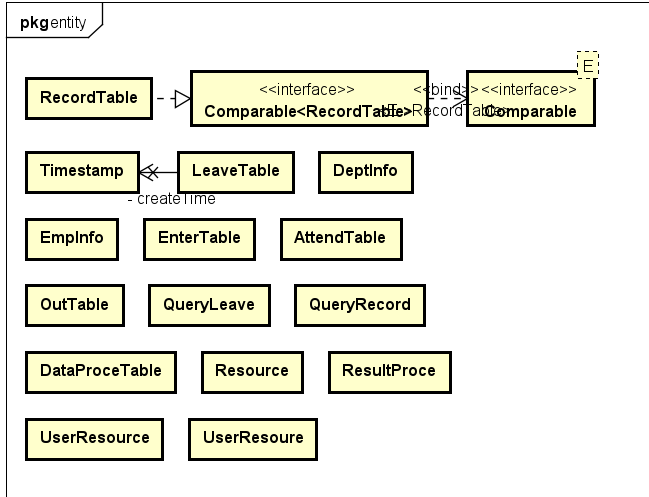
QueryRecordServiceImpl：查询员工个人或部门工时的Service实现类

UpLoadServiceImpl：记录员工考勤情况并插入到数据库中的Service实现类

# 五、Entity和Dao层设计

## Entity设计

Entity中定义了不同的表对应数据库包括员工信息，请教申请表，部门信息，考勤打卡表，出勤表，外出表，员工对应id，员工对应权限，有条件的检索表等。Dao层从数据库获取到数据传给Service层，Service将数据录入相应的Entity实体集表中作为返回结果参数。



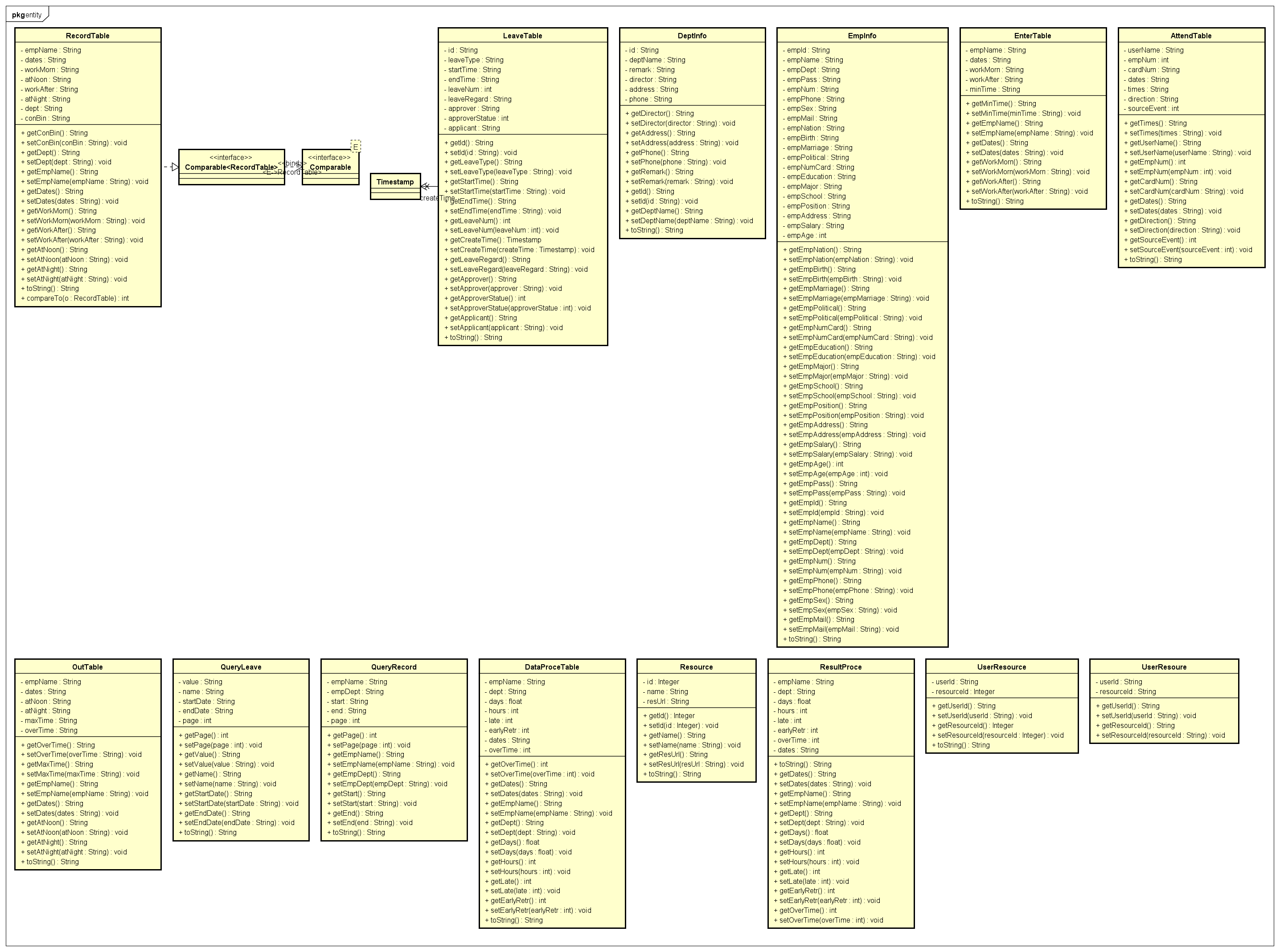
 图5-1 Entity表粗略情况图

图5-2 Entity表具体情况（请放大观看）

## Dao层设计及其mapper映射xml文件

Dao层主要是对数据库进行相应操作，本身作为接口提供给Service使用，具体的数据库语句通过resources文件夹下的mapper文件夹内写的xml文件映射相应的Dao层接口，在Service调用Dao层方法时会执行其对应映射文件下的sql语句对数据库进行相应操作。

Dao层主要根据需求功能进行相应的设计，提供的功能接口如下：

（**因Dao层包含较多接口，后续主要对部分Dao层主要功能接口进行详细叙述**）

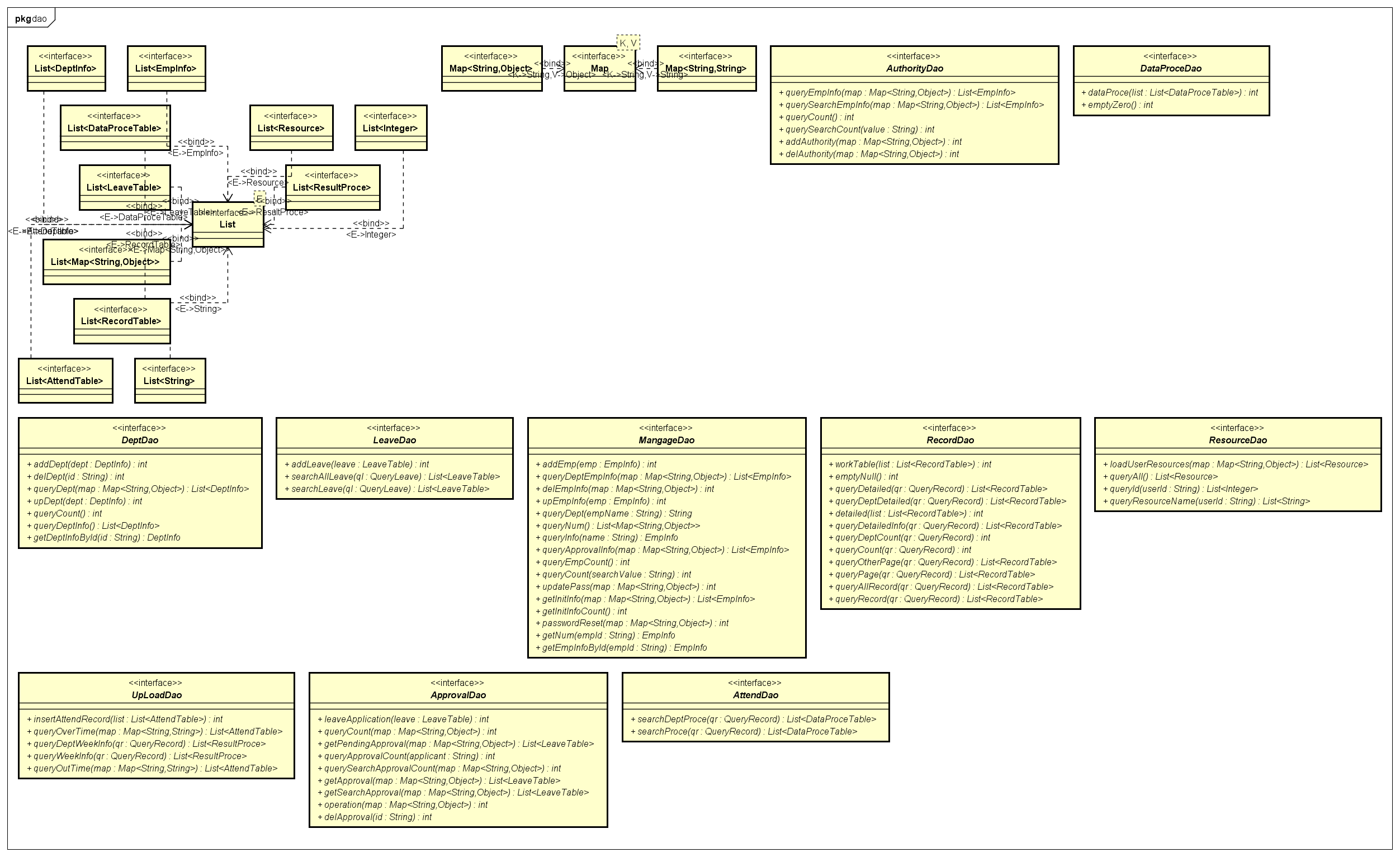
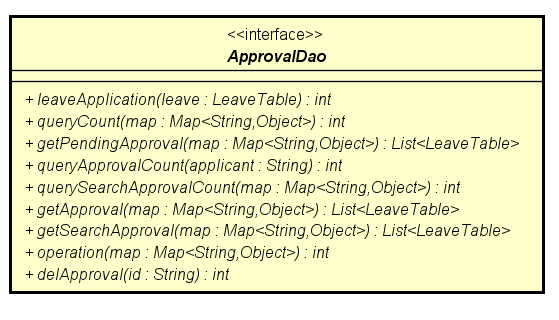


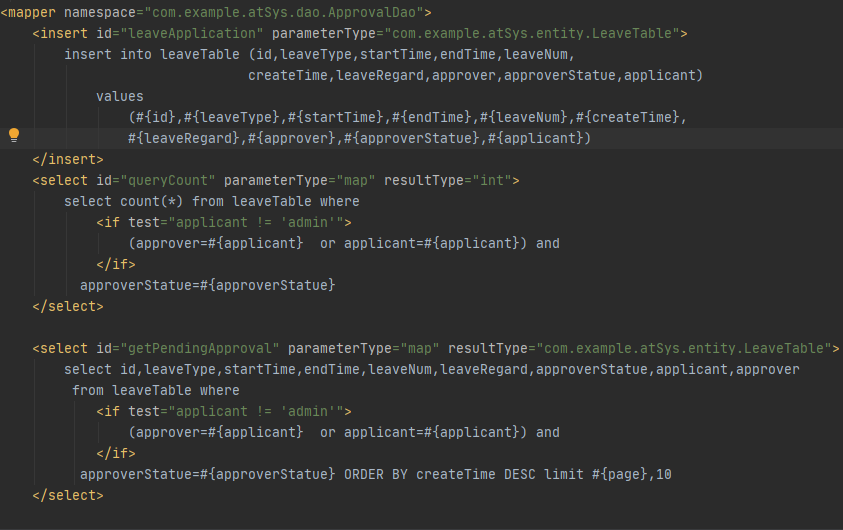
图5-3 Dao层及其对应功能（请放大观看）

### ApprovalDao

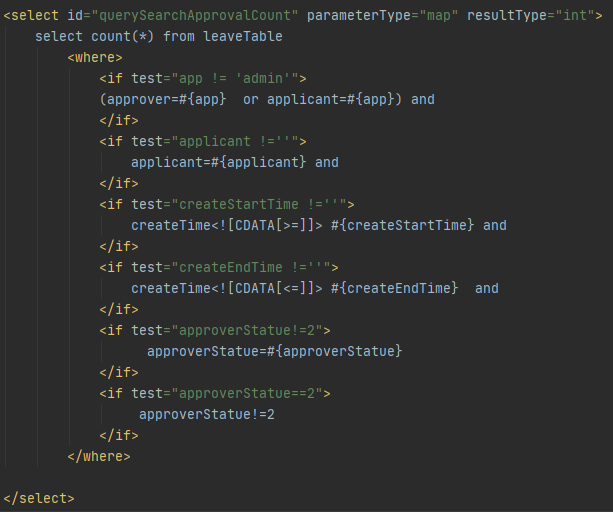


ApprovalDao负责请假申请的审核及申请，同时可查询申请情况狂，主要修改的是数据库的LeaveTable表即请假表，同时会对相关操作人员权限进行鉴定（applicant），映射对应xml文件如下所示：

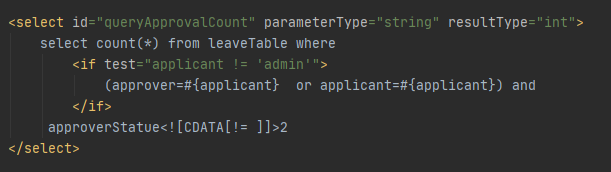
1.请假申请，查询待审批申请总数，获取待审批申请记录



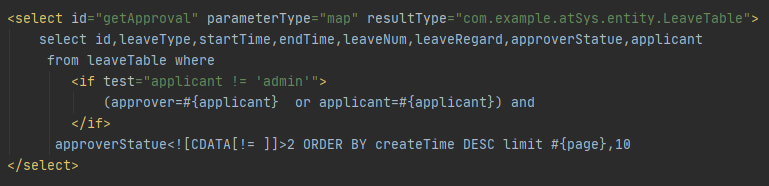
2.根据搜索条件查询已审批申请



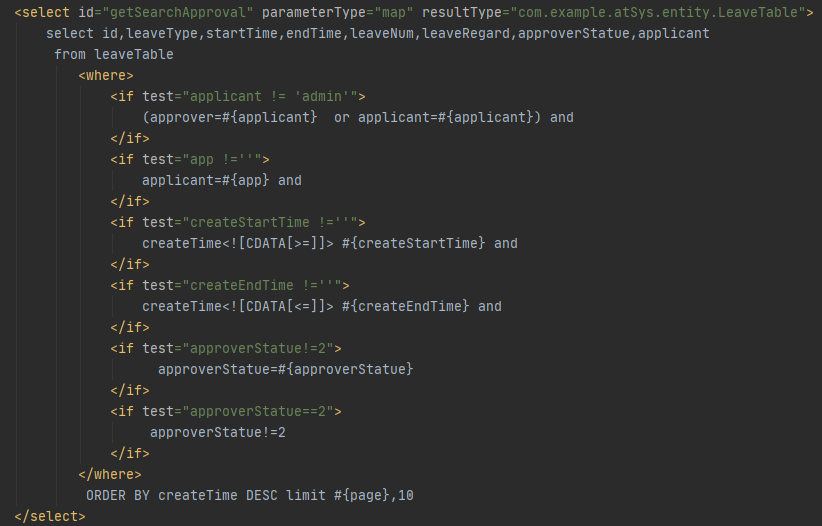
3.查询已审批总数



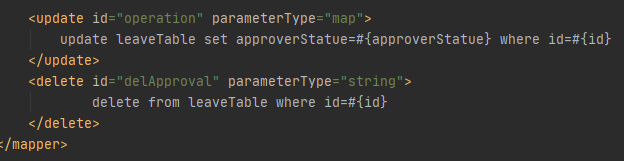
4.查询已审批记录



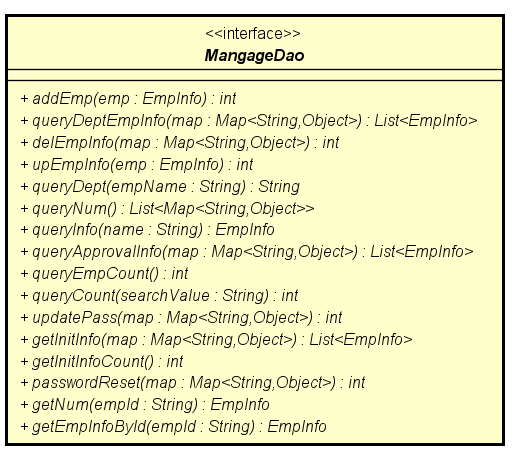
5.根据搜索条件查询已审批记录



6.进行请假申请审批，删除操作

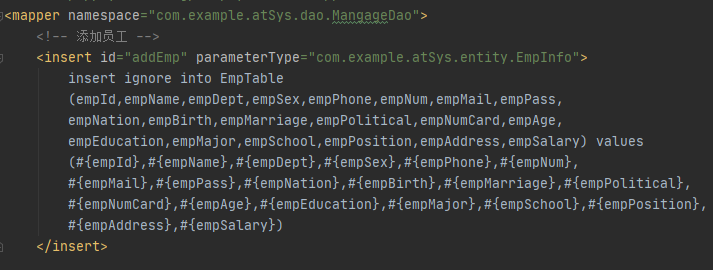


### MangageDao

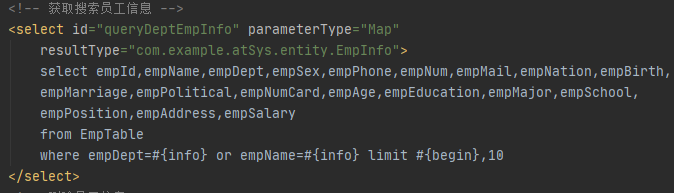


MangageDao负责员工的管理，基本只有管理员admin或者被赋予员工管理权限的用户才能执行相关的操作，主要修改的为数据库中的员工信息表，员工对应id表，会对相关操作人员权限进行鉴定（applicant），映射对应xml文件如下所示：

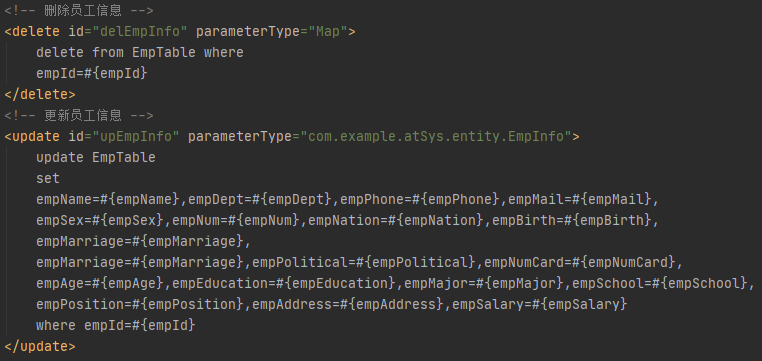
1.添加员工信息



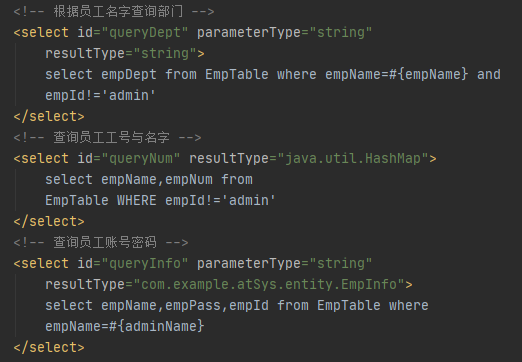
2.根据部门和名字搜索员工信息



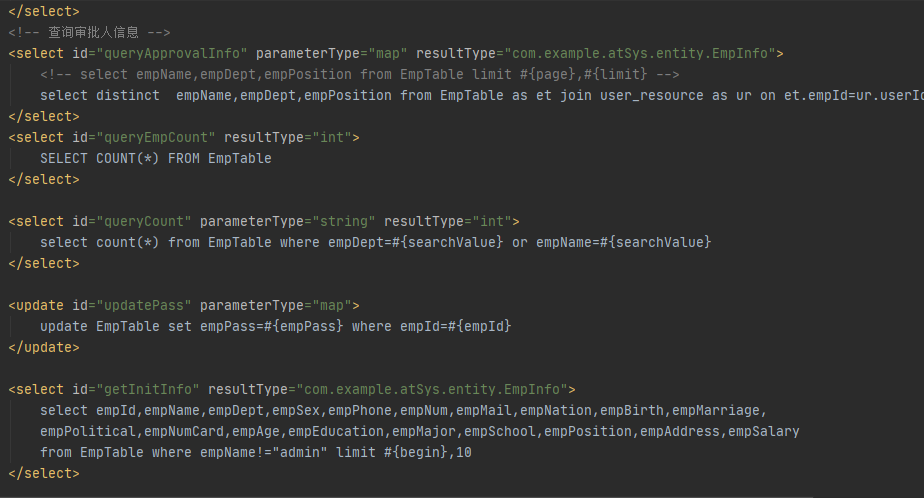
3.根据员工id删除或更新员工信息



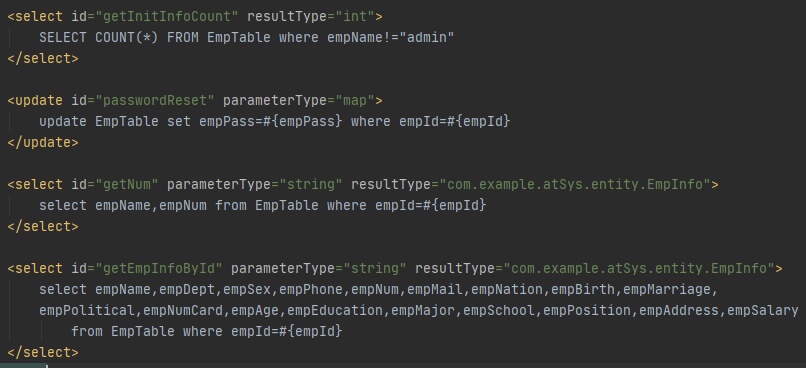
4.根据员工名字查询部门，管理员查询所有员工信息



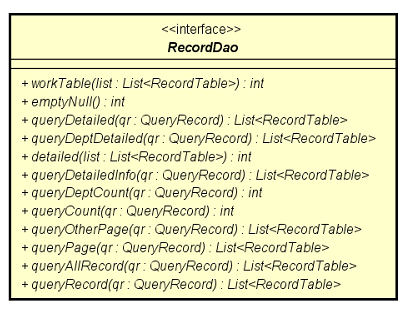
5.查询有权限的审批人信息，查询员工总数，更新员工密码，初始化员工信息



6.根据id查询员工的信息，得到被初始化的员工信息

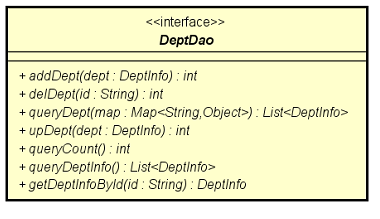


### RecordDao



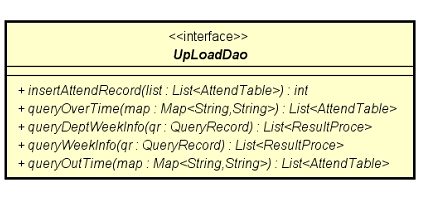
RecordDao主要记录员工的打卡信息，对数据库中的员工打卡表进行相应的修改，同时管理员或拥有相关权限的人员可以进行员工打卡情况的查询等操作。

### 4）DeptDao



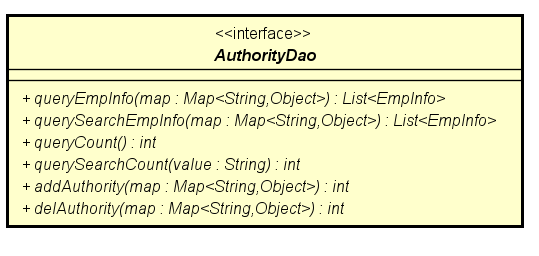
DeptDao对应数据库中的部门信息表，实现对数据库内部门表进行增删查改部门信息，支持通过id来查找相应部门。

### 5）UpLoadDao



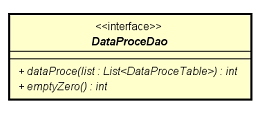
UpLoadDao对应数据库中的打卡表，实现将员工上下班打卡记录情况插入到数据库中（数据库初始化时已有预设值）

### 6）AuthorityDao



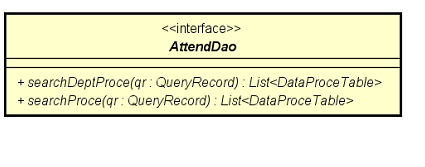
AuthorityDao对应数据库中的员工id信息表，员工权限表和权限对应id表，实现查询数据库内员工信息表得到员工信息及员工总数，可以对指定员工的权限进行修改（需要管理员admin权限）。

### 7）DataProceDao



DataProceDao将员工表工时数据插入到数据库中。

### 8）AttendDao



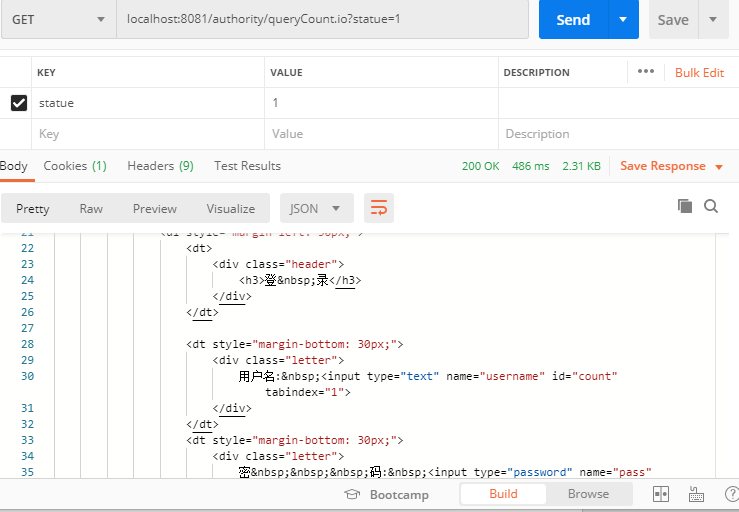
AttendDao实现从Entity中获取员工个人总工时统计以及部门的总工时统计

# 六、实现效果

## 鉴权

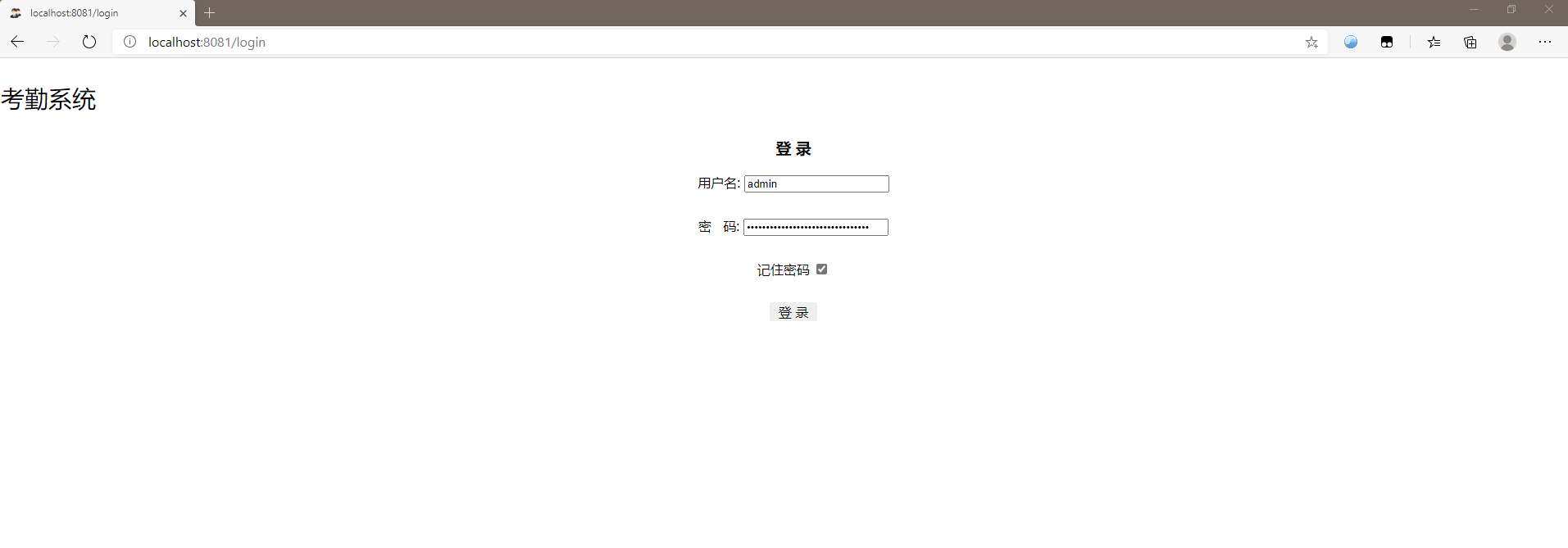
考虑到安全性，我们不在URL上进行账号密码的明文传输，我们使用shiro框架进行及安全设计，而相应的登录接口的调用（manage/login.io）则交给JS脚本（参考代码：<https://www.jb51.net/article/160473.htm>）

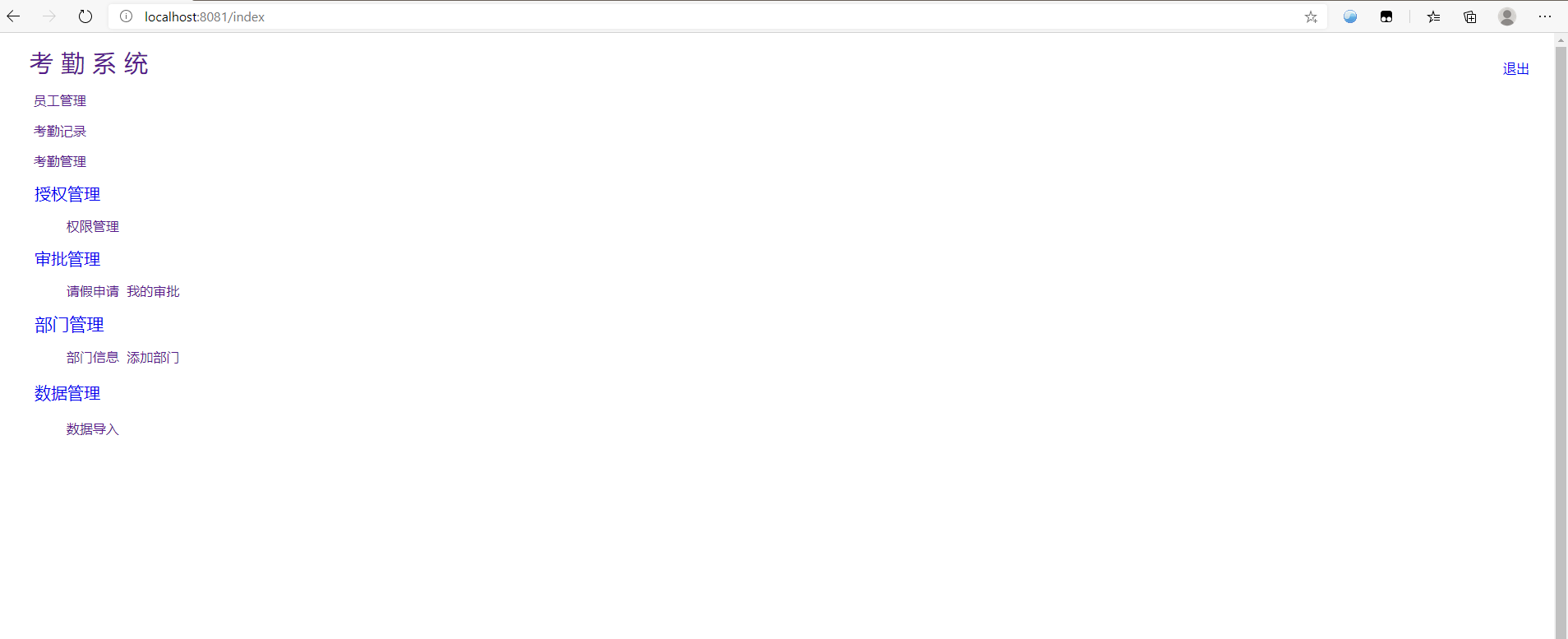
首先我们先用postman来测试鉴权效果，如果不登陆直接访问查询员工总数的接口，则会直接跳转至登陆界面如下图：



## 登录

使用浏览器输入localhost:8081/login来进入登陆界面，并输入账号密码来进行登录，这里我们使用管理员进行登录，方便演示所有功能：





## 员工管理

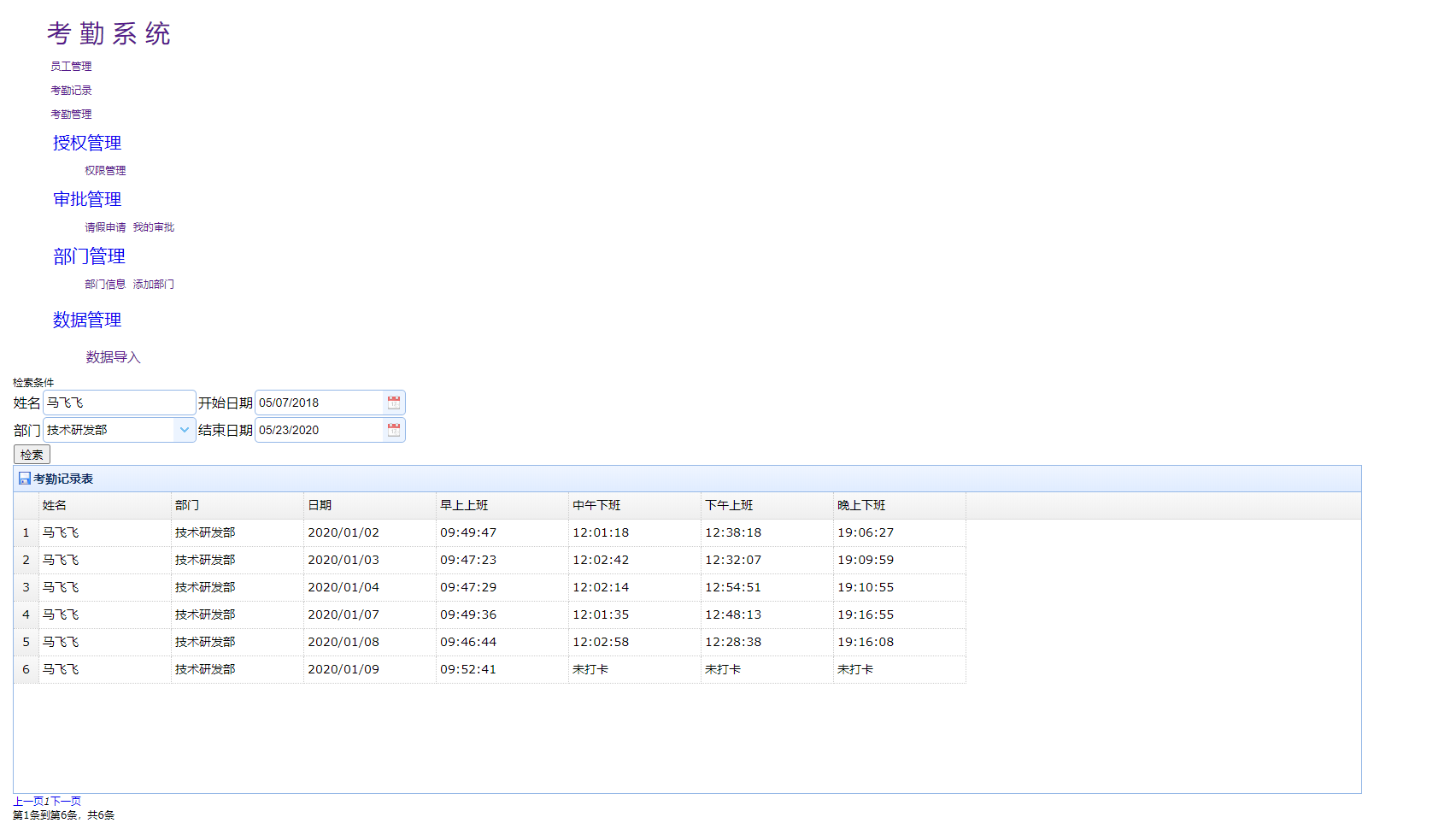


员工管理可以查看所有的员工情况。当然，首先会进行一个鉴权行为判断当前登录者是不是有权限管理员工。这里有一个添加员工按钮，有权限的人可以添加员工，如下图。



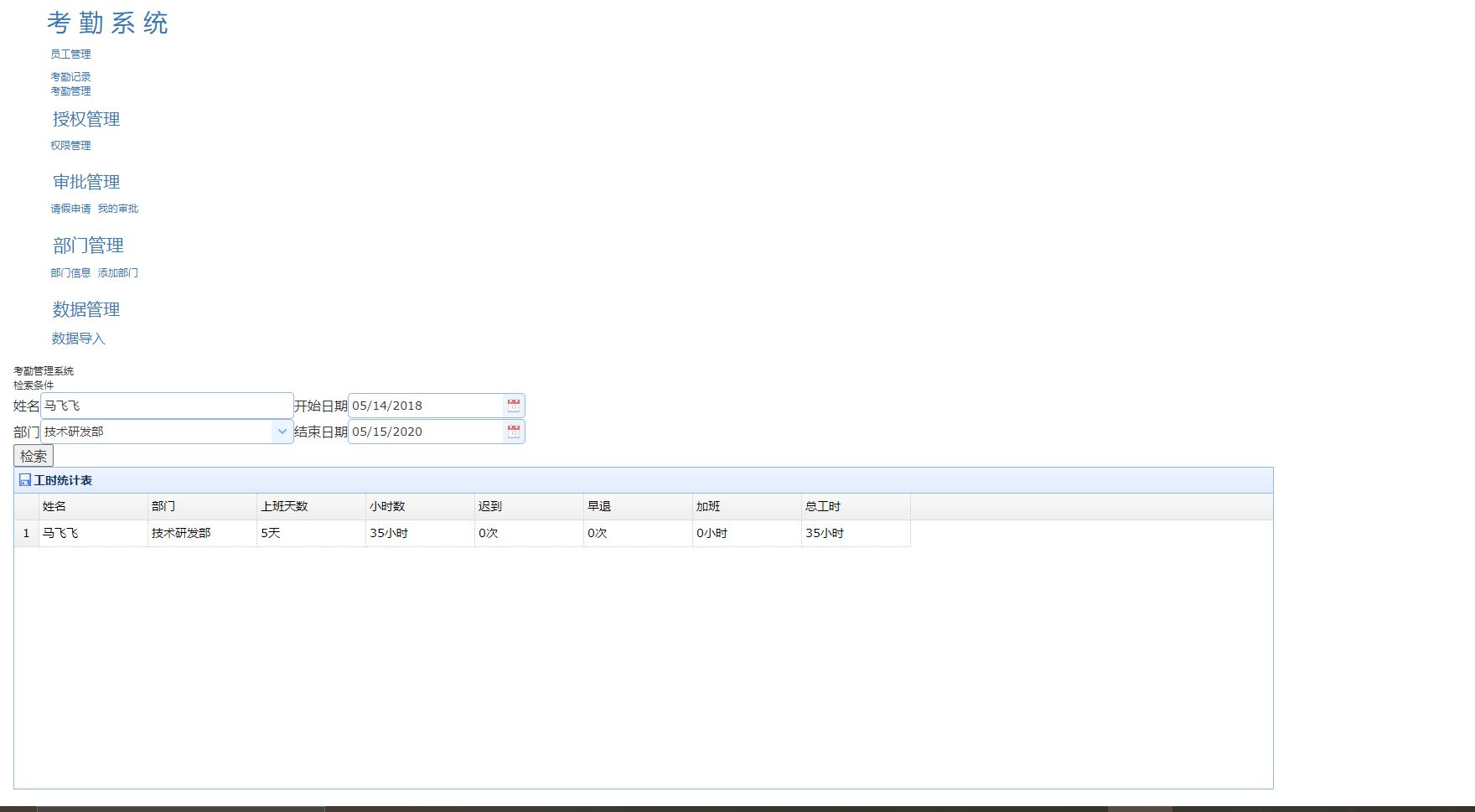
## 考勤记录（上班记录）

可以查看某个人的上班记录，如下图：



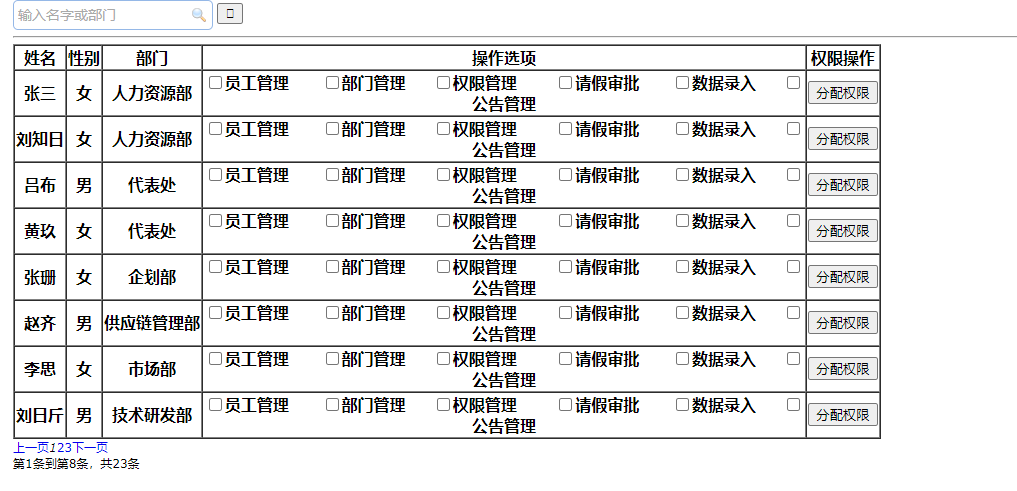
## 考勤管理（签到早退情况）

这个网页，管理员可以查看所有成员都签到打卡情况，如下图：



## 权限管理

管理员可以分配权限：

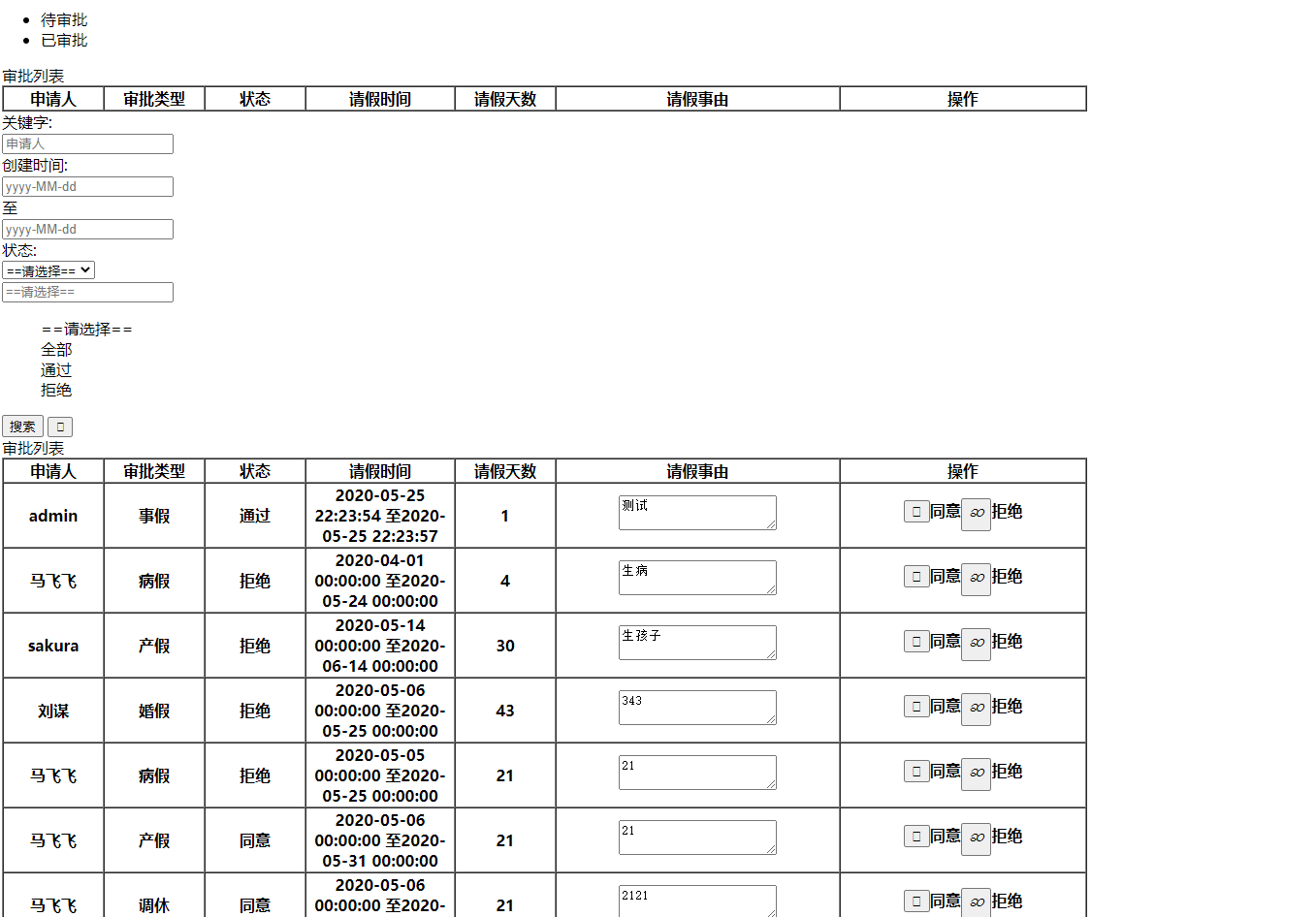


## 审批管理

在这里有两个功能：请假和审批，首先是请假功能：



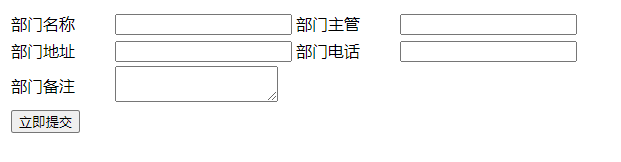
其次是审批：

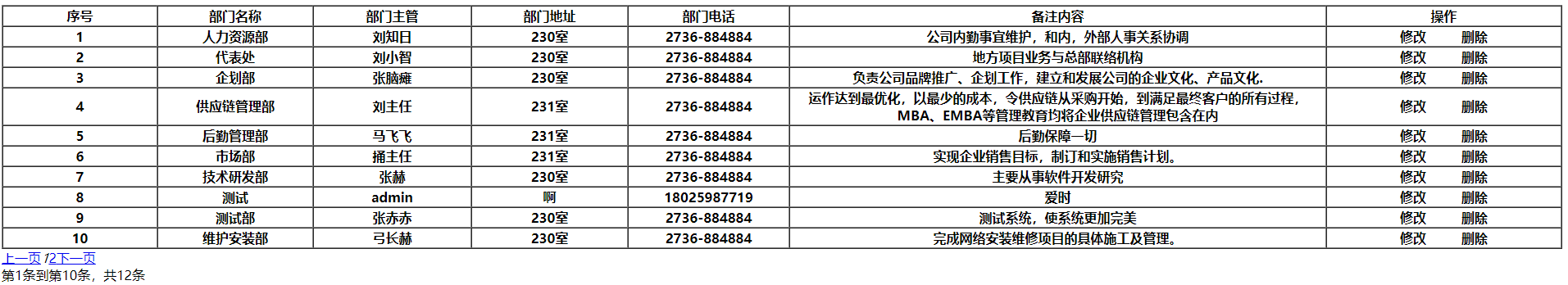


这里第一个是待审批的记录，第二个是已审批的记录，作为管理员都可以更改。

## 部门管理

第一个是添加部门，第二个是部门列表。（由于前端html和JS不知道怎么写，参考了老师给的和网上别人的代码还是存在bug，此功能使用postman测试是通过的）





## 数据导入

这个功能主要是实现上班记录的批量导入。服务器接收客户端传送的表格文件并进行数据处理，比如判断是否迟到等等。

（参考代码：<https://blog.csdn.net/qq_42382972/article/details/85764703>）

