**考勤系统设计文档**

****

**第19组**

3018216116 尤嘉宁

3018216108 满显凡

3018216127鲍子龙

3018216081任林杰

目录

[0. 设计思路 3](#_Toc38314427)

[1. RESTful API 4](#_Toc38314428)

[1.1 登录页面 4](#_Toc38314429)

[1.3 申请页面 5](#_Toc38314430)

[1.4 个人页面 6](#_Toc38314431)

[1.5 其他员工页面 7](#_Toc38314432)

[1.6 行政页面 8](#_Toc38314433)

[1.7 财务页面 10](#_Toc38314434)

[1.8 消息页面 11](#_Toc38314435)

[1.9 审批检查页面 12](#_Toc38314436)

[1.10 全体员工一览页面 13](#_Toc38314437)

[2.Controller VO设计 14](#_Toc38314438)

[2.Service BO设计 16](#_Toc38314439)

[3. Repository DO设计 18](#_Toc38314440)

[4.Controller-Service-Repository接口设计 20](#_Toc38314441)

[5.数据库设计(E/R图) 24](#_Toc38314442)

## 0. 设计思路

如图所示, 由于本系统内部使用TCP/IP进行通信,以网站的形式供员工在内网访问.

当系统外部人员(或者设备)发来请求时, 首先应该访问暴露给外部的RESTful api, 按照API规定的动词和字段构造请求. 当请求到达服务器后, Controller负责提取请求中的字段和动词, 将字段中的数据封装入BO(business object)对象, 并根据url中的路径和动词路由至Service层中的相关业务方法.

随后, Service中的具体业务方法应该按照预先编写的流程和BO中设置的相关方法, 对数据进行操作, 例如更新相关变量的值, 发送邮件通知和(如果有必要)调用Repository层中对数据库的操作. 如果请求的动词是GET的话, 还需要构造response并以VO(ViewObject)的形式返回给Controller. Controller将其转为Json的形式后返回给客户端.

而Repository则负责具体的对数据库增删查改. DO(data object)的成员变量和数据库中的数据表字段相对应.

## 1. RESTful API

**1.1 登录页面**

**1.1.1 登录**

注: 如果只有一个json代码段, 说明此API不需要回复任何消息, 若为两个json代码段, 则第一个为请求, 第二个为回复. 下同.

// 登录接口, 发送id和password

// 权限: 匿名可访问

# POST /api/user

{

    "username":123456,

    "password":"654321"

}

{

"token":" ... "

}

**1.1.2 获取个人数据**

// 通过token获取个人数据

// 权限: 普通员工及以上

# GET /api/user

{

"userId":123,

"username":"yjn",

"title":"普通员工"

}

1.2 打卡页面

**1.2.2 加班登记**

// 在页面的某个部位有个加班登记的按钮, 需用户在正常上班时期外点击.

//登记加班

//id 用户id

//datetime 开始加班时间

// 权限: 普通员工及以上

# POST /api/record/overtime

{

    "id":123456,

    "datetime":"1590506262"

}

**1.3 申请页面**

**1.3.1 申请请假**

// 请假申请,

// id: 用户id

// startTime: 开始时间

// endTime: 终止时间

// type: 请假类型

// reason: 请假理由

// 普通员工及以上

# POST /api/application/leave

{

    "id":1,

    "startTime":1590506262,

    "endTime":1590851862,

  "type":"annualLeave",

    "reason":"I want to play a game"

}

**1.3.2 外出申请**

// 申请外出

// id: 用户id

// 权限: 普通员工及以上

# POST /api/application/out

{

    "id":123456,

    "startTime":1590506262,

    "endTime":1590851862,

    "reason":"I want to play a game"

}

**1.3.3 补签**

//补缺勤

//id: 用户id

// 权限: 普通员工及以上

# POST /api/application/remedy

{

    "id":123456,

    "date":1590851862,

    "reason":"I wanted to play a game"

}

**1.3.4 修改请假申请**

//leaveId 标识该请假申请的key, 注: 每个申请都有一个唯一的id(下同)

// 权限: 普通员工及以上

# PUT /api/application/mod/leave

{

"id":123456,

    "leaveId":1,

    "startTime":1590592662,

    "endTime":1590938262,

    "type":"annualLeave",

    "reason":"I still want to play a game"

}

**1.3.5 修改外出申请**

//outId 标识该外出申请的key

// 权限: 普通员工及以上

# PUT /api/application/mod/out

{

    "id":123456,

    "outId":1,

    "startTime":1590592662,

    "endTime":1590938262,

    "reason":"I still want to play a game"

}

**1.3.6 修改补签申请**

//remedyId 标识该补签申请的key

// 权限: 普通员工及以上

# PUT /api/application/mod/remedy

{

    "id":123456,

    "remedyId":1,

    "date":"2020-4-10",

    "reason":"I still wanted to play a game"

}

**1.4 个人页面**

**1.4.1 查看考勤记录**

// 发出个人id

// 返回考勤数组 records[record]

// record { date, status }

// 权限: 普通员工及以上

# GET /api/record?id=123456

{

    "records":[

        {

            "date":"2020-4-09",

            "status":"leave"

        },

        {

            "date":"2020-4-10",

            "status":"present"

        },

        {

            "date":"2020-4-11",

            "status":"non-present"

        }

    ]

}

**1.4.2 查看当前状态**

// 返回该Id用户的当前请假或外出情况

// 权限: 普通员工及以上

# GET /api/user/status?id=123456

{

    "status":"present"

}

**1.4.3 查看剩余假期额度**

//返回该Id用户的假期剩余

// 权限: 普通员工及以上

# GET /api/user/Holidaybalance?id=123456

{

"annualLeave": 7

"maternityLeave":7

"homeLeave":7

}

**1.5 其他员工页面**

**1.5.1 查看员工列表**

// 权限: 普通员工及以上

# GET /api/user/employees // 这个是改之后的

{

    "members":[

        {

            "id":123456,

            "name":"尤嘉宁",

            "title":"总经理"

"status":""

        },

        {

            "id":456789,

            "name":"mv",

            "title":"副总经理"

"status":""

        }

    ]

}

**1.5.2 查看其它员工当前状态**

// 权限: 普通员工及以上

# GET /api/user/employees/654321 // 这个没变

{

    "id":654321,

    "name":"胡阿尤",

"title":"",

    "status":"present"

}

**1.6 行政页面**

**1.6.1 上传行政部处理后的数据**

// 权限: 行政部员工

# POST /api/administration-department/after-process/data/

excel 文件上传点

excel 字段

|  |  |
| --- | --- |
| id | result |
| 123 | signed |
| 124 | unsigned |
| 125 | signed |

**1.7 财务页面**

**1.7.1 获取系统处理后数据**

// // 权限: 财务部员工

# GET /api/administration-department/after-process/data?id=963852

// id: 用户id

// overTime 单位: 小时

// unsigned 单位: 天

{

"data":[

{

"userName":"mxf",

"leaves":[

{

"startTime":"2020-4-12",

"endTime":"2020-4-13",

"leaveType":"chanjia"

}

],

"outs":[

{

"startTime":"2020-4-12",

"endTime":"2020-4-13",

}

],

"unsigned":4,

"overTime":10,

}

]

}

**1.8 消息页面**

**1.8.1 获取消息列表(包括审批结果和申请信息)**

//id与返回的信息相对应

//返回的消息包括自己的申请的情况，以及需要自己审批的其他人的申请

// 例如如果是项目经理, 既cha需要处理普通员工的申请, 也可以查看自己的申请

// 权限: 普通员工及以上

# GET /api/user/messages/application?type=0&id=123456

// 返回 //此处请假申请和外出申请表示相同，外出申请type为out

{

        "applications":[

{

            "applicationId":1,

"memberId":123

"name":"jdy"

"startTime":"2020-4-11",

"endTime":"2020-4-12",

"type":"annualLeave",

"authority":0 //查看自己的申请功能下，该参数为0 无效

"reason":"I want to play a game"

            "LeadersOpinion": [{

                    "title":"总经理",

                    "name":"尤嘉宁",

"result":"agree",

                    "opinion":"I agree with you!"

                } ]

         }

        ]

}

# GET /api/user/messages/process-application?type=0&id=123456

// 权限: 项目经理及以上

// 返回 此处请假申请和外出申请表示相同，外出申请type为out

{

        "applications":[

{

            "applicationId":1,

"memberId":123

"name":"jdy"

"startTime":"2020-4-11",

"endTime":"2020-4-12",

"type":"annualLeave",

"authority":0

"reason":"I want to play a game"

            "LeadersOpinion": [{

                    "title":"总经理",

                    "name":"尤嘉宁",

"result":"agree",

                    "opinion":"I agree with you!"

                } ]

         }

        ]

}

**1.8.2 处理申请**

// 项目经理及以上

# POST /api/application/process-command/

{

    "id":123456,

    "applicationId":1

    "result":"agree",

    "opinion":"I agree with you!"

}

**1.9 审批检查页面**

**1.9.1 检查其他管理者的审批**

// 副总经理及以上

// 例如总经理在此处可以看到副总经理, 项目经理等领导做出的审批, 但是无法修改或删除.

# GET /api/manager/approval-results?id=123456

{

    "otherManagerApprovalResults":[

        {

            "id":123456,

            "startTime":"2020-4-11",

            "endTime":"2020-4-12",

            "type":"annualLeave",

            "reason":"I want to play a game",

            "LeadersOpinion": {

"title":"总经理",

"name":"尤嘉宁",

"result":"pass",

"opinion":"I agree with you!"

}

        }

    ]

}

**1.10 全体员工一览页面**

// 总经理, 副总经理等领导可以查看下面三个页面的数据

**1.10.1 查看全体员工的请假情况 查看所有休假员工**

// 副总经理及以上

# GET /api/manager/all/leaving-members

{

    "leaveMembers":[

        {

            "id":741852,

            "userName":"mv",

"title":"员工"

        },

        {

            "id":987654,

            "userName":"seeler",

"title":"员工"

        }

    ]

}

**1.10.2 查看全体员工的外出情况**

// 副总经理及以上

# GET /api/manager/all/outing-members

{

    "outMembers":[

        {

            "id":741852,

            "userName":"mv"

"title":"员工"

        },

        {

            "userId":987654,

            "userName":"seeler"

"title":"员工"

        }

    ]

}

**1.10.3 查看全体员工的加班情况**

// 副总经理及以上

# GET /api/manager/all/overtime-members

{

    "overtimeMembers":[

        {

            "id":741852,

            "userName":"mv",

"title":"员工"

        },

        {

            "id":987654,

            "userName":"seeler",

"title":"员工"

        }

    ]

}

## 2.Controller VO设计

ViewObject对象在本项目中, 应在Controller层被转为json格式, 返回给api调用者.

0. ViewObject

public interface ViewObject {

}

1. ApplicationForBossVO

public class ApplicationForBossVO {

    private List<ApplicationForBossBO> otherManagerApprovalResults;

}

2. currentStatusViewObject

某用户的当前状态.

public class CurrentStatusVO implements ViewObject {

    private String status;

}

3. HolidayBalanceVO

某员工的假期余额

public class HolidayBalanceVO implements ViewObject {

    private int annualLeave;

    private int maternityLeave;

    private int homeLeave;

}

4. LeaveMembersVO

所有休假的员工

public class LeaveMembersVO implements ViewObject {

    private List<MemberBO> leaveMembers;

}

5. LoginStatusVO

public class LoginStatusVO implements ViewObject{

    private int userId;

    private String username;

    private String title;

}

6. MessagesVO

所有可查看的消息

public class MessagesVO implements ViewObject {

    private List<ApplicationBO> applications;

}

7. OtherMemberStatusVO

其他某员工的当前状态

public class OtherMemberStatusVO implements ViewObject {

    private int id;

    private String name;

    private String title = null;

    private String status;

}

8. OutMembersVO

public class OutMembersVO implements ViewObject {

    private List<MemberBO> outMembers;

}

9. OverTimeMembersVO

public class OverTimeMembersVO implements ViewObject {

    private List<MemberBO> overTimeMembers;

}

10. PersonRecordViewObject

某用户的考勤记录的集合.

public class PersonRecordVO implements ViewObject {

    private List<RecordBO> records;

}

11. UserDataVO

public class UserDataVO implements ViewObject {

    private UserDataBO data;

}

12. UserListVO

全体员工的列表

public class UserListVO implements ViewObject {

    private List<MemberBO> members;

}

## 2.Service BO设计

0. VO

VO基类

public interface BusinessObject {

}

6. ApplicationBO

代表申请表的基类

public class ApplicationBO implements BusinessObject {

    protected int applicationId;

    protected int memberId;

    protected String name;

    protected String startTime;

    protected String endTime;

    protected String reason;

}

0. ApplicationForBossBO

public class ApplicationForBossBO implements BusinessObject {

    private int id;

    private String startTime;

    private String endTime;

    private String type;

    private String reason;

    private LeaderOpinionBO leadersOpinion;

}

5. LeaderOpinionBO

代表某个领导意见

public class LeaderOpinionBO implements BusinessObject {

    private String title;

    private String name;

    private String result;

    private String opinion;

}

7. LeaveApplicationBO

代表一个休假申请

public class LeaveApplicationBO extends ApplicationBO {

    private String type;

}

2. MemberBO

代表某位员工

public class MemberBO implements BusinessObject {

    private int id;

    private String name;

    private String title;

    private String status = null;

}

3. MinApplicationBO.java

public class MinApplicationBO implements BusinessObject {

    protected String startTime;

    protected String endTime;

}

4. MinLeaveApplicationBO

public class MinLeaveApplicationBO extends MinApplicationBO {

    private String leaveType;

}

0. MLeaveApplicationBO

public class MLeaveApplicationBO extends ApplicationBO {

    private String type;

    private int authority = 0;

    private List<LeaderOpinionBO> leadersOpinion;

}

0. MOutApplicationBO

public class MOutApplicationBO extends ApplicationBO {

    private final String type = "out";

    private int authority = 0;

    private List<LeaderOpinionBO> leadersOpinion;

}

1. RecordBO

代表一条考勤记录

public class RecordBO implements BusinessObject {

    private String date;

    private String status;

}

0. UserDataBO

public class UserDataBO implements BusinessObject {

    private String userName;

    private List<MinLeaveApplicationBO> leaves;

    private List<MinOutApplicationBO> outs;

    private int unsigned;

    private int overTime;

    }

## 3. Repository DO(Entity)设计

0. Application

public abstract class Application {

    private int applicationId;

    private int userId;

    private String startTime;

    private String endTime;

    private String reason;

}

1. HolidayBalance

public class HolidayBalance {

    private int id;

    private int balance;

    private String type;

}

2. LeaderOpinion

public class LeaderOpinion {

    private int leaderId;

    private int applicationId;

    private String result;

    private String opinion;

}

3. LeaveApplication

public class LeaveApplication extends Application{

    private String type;

}

4. Member

public class Member {

    private int id;

    private String name;

    private String password;

    private String title;

    private String sex;

}

5. Message

public class Message {

    private int userId;

    private int applicationId;

}

6. OutApplication

public class OutApplication extends Application{

}

7. OverTime

public class OverTime {

    private int userId;

    private String date;

}

8. Record

public class Record {

    private int id;

    private String date;

    private String status;

}

## 4.Controller-Service-Repository接口设计

#### 4.1 Repository

0. ApplicationDao

public class ApplicationDao {

    JdbcTemplate jdbcTemplate;

    final private String applicationTable = "application";

    final private String holidayBalanceTable = "holiday\_balance";

    final private String leaderOpinionTable = "leader\_opinion";

    final private SimpleDateFormat simpleDateFormat = *new* SimpleDateFormat("yyyy/MM/dd");

*/\**

*\* 作用: 增加一条申请 输入: 要增加的申请, 例如如果增加一个外出申请, 就传来一个OutApplication的实例 返回: 无*

*\*/*

    public void insertApplication(Application application) {}

*/\**

*\* 作用: 根据 user id 和 application id 更新申请数据 输入: 需要更新的申请,*

*\* 注意必须保证传来的application中的userId和applicationId都是正确的, 否则无法更新 返回: 无*

*\*/*

    public void updateApplication(Application application) {}

*/\**

*\* 作用: 查询所有申请 输入: 无 返回: 所有的申请数据, 包括外出, 请假等*

*\*/*

    public List<Application> qureyAllApplication() {}

*/\**

*\* 作用: 通过 user id 查询申请记录 输入: 用户的id 返回: 某一位用户的所有的申请记录, 包括外出, 请假等*

*\*/*

    public List<Application> qureyApplicationById(int id) {}

*/\**

*\* 作用: 通过 application id 查询申请记录*

*\* 输入: 用户的id*

*\* 返回: 某一位用户的所有的申请记录, 包括外出, 请假等*

*\*/*

    public Application qureyApplicationByAppId(int applicationId) {}

*/\**

*\* 作用: 通过用户id和日期字符串来查询某条申请, 使得传入的 date 位于该 申请的 startDate 和 endDate 之间*

*\* 输入: 用户id和日期字符串, 注意 ! 日期字符串必须为 yyyy/MM/dd 格式, 例如 "2000/06/21" 而不能是 "2000/6/21"*

*\* 返回: Application对象, 如果没有查询到, 则会返回 null*

*\*/*

    public Application qureyApplicationByDate(int id, String currentDate){ }

*/\**

*\*   作用:   查询某员工的假期余额*

*\*   输入:   某位员工的id*

*\*   返回:   该员工的所有假期的余额的数据*

*\*/*

    public List<HolidayBalance> qureyHolidayBalance(int id){ }

*/\**

*\* 作用:   增加一个假期余额*

*\* 输入:   假期余额实例*

*\* 返回:   无*

*\*/*

    public void insertHolidayBalance(HolidayBalance hb){ }

*/\**

*作用:   修改假期余额*

*输入:   假期余额实例, 需确保实例中的user id 和 type和现有数据对应*

*返回:   无*

*\*/*

    public void updateHolidayBalance(HolidayBalance hb){ }

*/\**

*作用:   查询现有的某申请的领导审批结果*

*输入:   某条申请的id*

*输出:   若干LeaderOrder对象*

*\*/*

    public List<LeaderOpinion> qureyLeaderOpinionByAppId(int applicationId){ }

*/\**

*作用:   为某一请求增加领导(审批)意见, 注意, 是增加, 而不是修改现有的*

*(因为一个申请往往需要多个领导的审批)*

*输入:   领导意见实例对象*

*返回:   无*

*\*/*

    public void updateApplicationResult(LeaderOpinion opinion){ }

*/\**

*内部使用*

*\*/*

    private Application transMapToApplication(Map<String, Object> result){ }

*/\**

*内部使用*

*\*/*

    private List<Application> transMapToApplication(List<Map<String, Object>> maps){ }

}

1. MemberDao

public class MemberDao {

    public JdbcTemplate jdbcTemplate;

    private String memberTable = "member";

*/\**

*作用:   根据 id 返回一条 member 记录, 并返回 Member对象*

*输入:   用户id*

*返回:   该用户所有个人数据, 包括密码等*

*\*/*

    public Member qureyUser(int id){ }

*/\**

*作用:   根据 id 返回 该id对应的 username.*

*输入:   用户id*

*返回:   该用户所有个人数据, 包括密码等*

*\*/*

    public String qureyUsernameById(int id){ }

*/\**

*作用:   根据 id 返回对应的 password 数据*

*输入:   用户 id*

*返回:   该用户的密码*

*\*/*

    public String qureyUserPassword(int id){ }

*/\**

*作用:   查询所有员工的所有信息*

*输入:   无*

*返回:   所有用户的所有数据*

*\*/*

    public List<Member> qureyMembersList(){}

*/\**

*作用:   增加一名员工*

*输入:   员工实例对象*

*返回:   新增的员工的id*

*\*/*

    public int insertMember(Member m){ }

}

2. MessageDao

public class MessageDao {

    JdbcTemplate jdbcTemplate;

    private String MessageTable = "message";

*/\**

*作用:   查询一个员工的个人消息(但是该对象中只有*

*用户id和申请id, 消息的具体内容需要你自己再去查)*

*输入:   个人id*

*返回:   消息对象列表*

*\*/*

    public List<Message> qureyPersonMessage(int id){    }

*/\**

*作用:   增加一个员工消息*

*输入:   消息的实例对象*

*返回:   无*

*\*/*

    public void insertMessage(Message m){ }

}

3. OverTimeDao

public class OverTimeDao {

    JdbcTemplate jdbcTemplate;

    final private String overTimeTable = "overtime";

    final private SimpleDateFormat simpleDateFormat = *new* SimpleDateFormat("yyyy/MM/dd");

*/\**

*作用:   增加一个加班记录*

*输入:   加班实例*

*返回:   无*

*\*/*

    public void insertOvertimeRecord(OverTime overTime){ }

*/\**

*作用:   查询某用户所有的加班记录*

*输入:   某用户的id*

*返回:   该用户所有加班记录*

*\*/*

    public List<OverTime> queryOvertimeById(int id){ }

*/\**

*作用:   通过 date 查询加班记录*

*输入:   date字符串*

*返回:   加班记录列表*

*\*/*

    public List<OverTime> queryOvertimeByDate(String date){    }

}

4. RecordDao

public class RecordDao {

    JdbcTemplate jdbcTemplate;

    String recordTable = "record";

*/\**

*作用:   插入一条签到记录*

*输入:   签到记录实例*

*返回:   无*

*\*/*

    public void insertSignInRecord(Record r){ }

*/\**

*作用:   根据用户id查询打卡记录*

*输入:   某用户的id*

*返回:   该用户所有的打卡记录*

*\*/*

    public List<Record> qureyPersonRecords(int id){}

*/\**

*作用:   查询所有人的所有打卡记录*

*输入:   无*

*返回:   所有员工所有的打卡记录*

*\*/*

    public List<Record> qureyAllRecord(){}

*/\**

*作用:   查询某日某用户的签到记录*

*输入:   用户 id 和 日期字符串*

*输出:   一个 Record 对象*

*\*/*

    public Record qureyRecordByDate(int id, String date){    }

}

#### 4.2 Service

0. LeaveSevice

public class LeaveService {

    private ApplicationDao applicationDao;

    private MessageDao messageDao;

    private UserService userService;

*/\**

*\* 添加请假*

*\*/*

    public void addLeaveApplication(LeaveInfo leaveInfo) {

    }

*/\**

*\* 修改请假申请*

*\*/*

    public void modifyLeaveApplication(ModifyLeaveInfo modifyLeaveInfo) {

    }

*/\**

*\* 获取年假剩余天数*

*\*/*

    public int[] getHolidayBalanceById(int id) {

    }

*/\**

*\* 获取迷你版请假申请*

*\*  "startTime":"2020-4-12",*

*"endTime":"2020-4-13",*

*"leaveType":"chanjia"*

*\*/*

    public List<MinLeaveApplicationBO> getMinLeaveApplicationById(int id) {

    }

*/\**

*\* 获取重量级请假申请*

*\* MLeaveApplicationBO*

*\*/*

    public List<ApplicationBO> getApplicationResultById(int id) {

    }

*/\**

*\* 获取重量级需要被该id处理的申请*

*\*  MLeaveApplicationBO*

*\*  doing*

*\*/*

    public List<ApplicationBO> getNeedToProcessById(int id) {

    }

*/\**

*\* 为请假申请添加处理意见*

*\*/*

    public boolean addOpinion(ProcessInfo processInfo) {

    }

*/\**

*\* 查看其他管理者的审批*

*\*/*

    public List<ApplicationForBossBO> getProcessApplicationResultsById(int id) {

    }

*/\**

*\* 获取全体请假中员工信息*

*\*/*

    public List<MemberBO> getLeaveMembers() {

    }

    private void getLeaderOpinion(MLeaveApplicationBO application) {

    }

    private void getLeaderOpinion(ApplicationForBossBO applicationFB, int appId, int leaderId) {

    }

    private int getBalance(int id, String type) {

    }

    private void addMessageToDepartmentManager(int applicationId) {

    }

    private void addMessageToGeneralManager(int applicationId) {

    }

    private void addMessageToDeputyGeneralManager(int applicationId) {

    }

    private void addMessage(List<Integer> ids, int applicationId) {}

}

1. OutService

public class OutService {

    private ApplicationDao applicationDao;

    private MessageDao messageDao;

    private UserService userService;

*/\**

*\* 添加外出申请*

*\*/*

    public void addOutApplication(OutInfo outInfo) {

    }

*/\**

*\* 修改外出申请*

*\* 加判断*

*\*/*

    public void modifyOutApplication(ModifyOutInfo modifyOutInfo) {

    }

*/\**

*\* 获得该id的迷你外出申请*

*\*/*

    public List<MinOutApplicationBO> getMinOutApplicationById(int id) {

    }

*/\**

*\* 获得该id的重量级外出申请*

*\*/*

    public List<ApplicationBO> getApplicationResultById(int id) {

    }

*/\**

*\* 获得该id的重量级需要处理申请*

*\*/*

    public List<ApplicationBO> getNeedToProcessById(int id) {

    }

*/\**

*\* 为外出申请添加处理意见*

*\*/*

    public void addOpinion(ProcessInfo processInfo) {

    }

*/\**

*\* 获取全部正在外出的员工信息*

*\*/*

    public List<MemberBO> getOutMembers() {

    }

    private void getLeaderOpinion(MOutApplicationBO application) {

    }

    private void addMessageToDepartmentManager(int applicationId) {

    }

    private void addMessageToGeneralManager(int applicationId) {

    }

    private void addMessageToDeputyGeneralManager(int applicationId) {

    }

    private void addMessage(List<Integer> ids, int applicationId) {

    }

}

2. OvertimeService

public class OvertimeService {

    private OverTimeDao overTimeDao;

    private MemberDao memberDao;

*/\**

*\* 加班登记*

*\*/*

    public void addOvertimeRecord(OvertimeInfo overtimeInfo) {

    }

*/\**

*\* 返回该id的加班次数*

*\*/*

    public int getOvertimeById(int id) {    }

*/\**

*\* 返回正在加班的人*

*\*/*

    public List<MemberBO> getOvertimeMembers() {    }

}

3. RecordService

public class RecordService {

    private RecordDao recordDao;

    private ApplicationDao applicationDao;

*/\**

*\* 返回对应id的全部打卡信息*

*\*/*

    public List<RecordBO> getRecordsById(int id) {

}

*/\**

*\* 返回该id没有签到次数*

*\*/*

    public int getUnsignedById(int id) {

}

*/\**

*\* 处理excel 保存记录*

*\*/*

    public void saveRecords(MultipartFile file) {

}

4. UserService

public class UserService {

    @Autowired

    private MemberDao memberDao;

    @Autowired

    private RecordDao recordDao;

*/\**

*\* 判断用户用户密码是否正确*

*\*/*

    public String logIn(LogInfo logInfo) {

    }

*/\**

*\* 登陆后返回某用户的个人信息*

*\* 6.5 add by yjn*

*\*/*

    public Member getLoginSatus(int userId){

    }

*/\**

*\* 返回某用户的状态*

*\* signed unsigned leave out*

*\*/*

    public String getStatusById(int id) {

}

    public Member getMembersById(int id) {

}

*/\**

*\* 返回全体成员信息*

*\*/*

    public List<MemberBO> getMembers() {

}

*/\**

*\* 返回对应id的名字*

*\*/*

    public String getNameById(int id) {

}

*/\**

*\* 返回对应id的职务*

*\*/*

    public String getTitleById(int id) {

}

*/\**

*\* 返回总经理id*

*\*/*

    public List<Integer> getGeneralManagerId() {

}

*/\**

*\* 返回副总经理id*

*\*/*

    public List<Integer> getDeputyGeneralManagerId() {

}

*/\**

*\* 返回部门经理id*

*\*/*

    public List<Integer> getDepartmentManagerId() {

}

    private List<Integer> getTitleId(String title) {

}

}

#### 4.3 Controller

public class AttendanceJsonController {

    private UserService userService;

    private OvertimeService overtimeService;

    private RecordService recordService;

    private LeaveService leaveService;

    private OutService outService;

    private RemedyService remedyService;

*/\**

*/\**

*\* 通过token获取用户的 id 和 username 等数据*

*\**

*\*/*

    @PreAuthorize("hasRole('普通员工')")

    @GetMapping("/api/user")

    LoginStatusVO loginStatus(@RequestHeader HttpHeaders headers){

    }

*/\**

*\* 行政部上传Excel文件，登记打卡记录*

*\*/*

    @PreAuthorize("hasRole('行政部员工')")

    @PostMapping("/api/administration-department/after-process/data")

    void signIn(@RequestParam("file") MultipartFile file) {

    }

*/\**

*\* 加班登记*

*\*/*

    @PreAuthorize("hasRole('普通员工')")

    @PostMapping("/api/record/overtime")

    void overtimeRecord(@RequestBody OvertimeInfo overtimeInfo) {

    }

*/\**

*\* 请假申请*

*\*/*

    @PreAuthorize("hasRole('普通员工')")

    @PostMapping("/api/application/leave")

    void leaveApplication(@RequestBody LeaveInfo leaveInfo) {

    }

*/\**

*\* 外出申请*

*\*/*

    @PreAuthorize("hasRole('普通员工')")

    @PostMapping("/api/application/out")

    void outApplication(@RequestBody OutInfo outInfo) {

    }

*/\**

*\* 补签申请*

*\*/*

    @PreAuthorize("hasRole('普通员工')")

    @PostMapping("/api/application/remedy")

    void remedyApplication(@RequestBody RemedyInfo remedyInfo) {

    }

*/\**

*\* 修改请假申请*

*\*/*

    @PreAuthorize("hasRole('普通员工')")

    @PostMapping("/api/application/mod/leave")

    void updateLeaveApplication(@RequestBody ModifyLeaveInfo modifyLeaveInfo) {

    }

*/\**

*\* 修改外出申请*

*\*/*

    @PreAuthorize("hasRole('普通员工')")

    @PostMapping("/api/application/mod/out")

    void updateOutApplication(@RequestBody ModifyOutInfo modifyOutInfo) {

    }

*/\**

*\* 查看个人考勤记录*

*\*/*

    @PreAuthorize("hasRole('普通员工')")

    @GetMapping("/api/record")

    PersonRecordVO getRecords(@RequestParam int id) {

    }

*/\**

*\* 查看个人状态*

*\*/*

    @PreAuthorize("hasRole('普通员工')")

    @GetMapping("/api/user/status")

    CurrentStatusVO getStatus(@RequestParam int id) {

    }

*/\**

*\* 查看个人假期剩余情况*

*\*/*

    @PreAuthorize("hasRole('普通员工')")

    @GetMapping("/api/user/Holidaybalance")

    HolidayBalanceVO getHolidayBalance(@RequestParam int id) {

    }

*/\**

*\* 查看全体员工列表*

*\*/*

    @PreAuthorize("hasRole('普通员工')")

    @GetMapping("/api/user/employees")

    UserListVO getMembersList() {

    }

*/\**

*\* 查看其他员工状态*

*\*/*

    @PreAuthorize("hasRole('普通员工')")

    @GetMapping("/api/user/employees/{otherUserId}")

    OtherMemberStatusVO getOthersStatus(@PathVariable int otherUserId) {

    }

*/\**

*\* 获取系统处理后的数据*

*\* 某个人的全部数据*

*\*/*

    @GetMapping("/api/administration-department/after-process/data")

    UserDataVO getUserDataById(@RequestParam int id) {

    }

*/\**

*\* 获取自己的申请信息*

*\*/*

    @PreAuthorize("hasRole('普通员工')")

    @GetMapping("/api/user/messages/application")

    MessagesVO getApplications(@RequestParam(name = "type") int type, @RequestParam int id) {

    }

*/\**

*\* 获取自己要处理的信息*

*\*/*

    @PreAuthorize("hasRole('项目经理')")

    @GetMapping("/api/user/messages/process-application")

    MessagesVO getProcess(@RequestParam(name = "type") int type, @RequestParam int id) {

    }

*/\**

*\* 处理请假申请*

*\*/*

    @PreAuthorize("hasRole('项目经理')")

    @PostMapping("/api/application/process-command")

    void processApplication(@RequestBody ProcessInfo processInfo) {

    }

*/\**

*\* 查看其他管理者的审批情况*

*\* done1*

*\*/*

    @PreAuthorize("hasRole('副总经理')")

    @GetMapping("/api/manager/approval-results")

    ApplicationForBossVO getApplicationResults(@RequestParam int id) {

    }

*/\**

*\* 查看全体请假员工*

*\*/*

    @PreAuthorize("hasRole('副总经理')")

    @GetMapping("/api/manager/all/leaving-members")

    LeaveMembersVO getLeaveMembers() {

    }

*/\**

*\* 查看全体外出员工*

*\*/*

    @PreAuthorize("hasRole('副总经理')")

    @GetMapping("/api/manager/all/outing-members")

    OutMembersVO getOutMembers() {

    }

*/\**

*\* 查看全体加班员工*

*\*/*

    @PreAuthorize("hasRole('副总经理')")

    @GetMapping("/api/manager/all/overtime-members")

    OverTimeMembersVO getOvertimeMembers() {

    }

}

## 5.数据库设计(E/R图)

