

天津大学



考勤系统 设计文档

学院	<u>智能与计算学部</u>
专业	<u>软件工程</u>
组长姓名	<u>黄泓</u>
组员姓名	<u>王迪 叶紫钰 周陈铮</u>

1. 前言

系统设计是根据考勤与请假系统《软件需求规格说明书》进一步分析、归纳，设计实现需求的技术策略。并进一步细化技术框架和内容，对下一阶段的工作提供指导。为明确软件体系结构、实现技术、部署方案，与客户对系统功能需求及实现方式达成一致的认识，以便于安排项目进度、组织软件开发与测试，特撰写本文档。

2. 概要设计

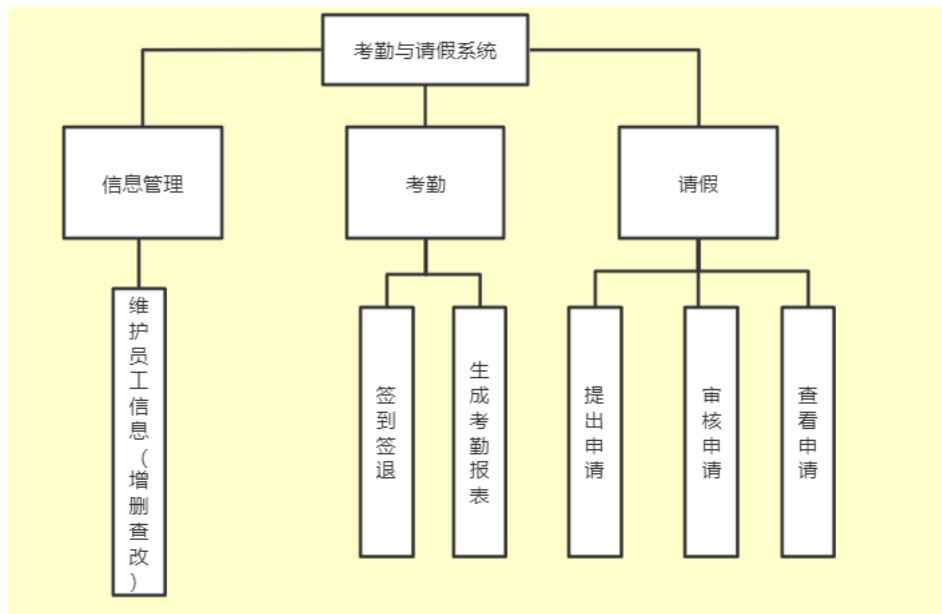
2.1 设计目的

- 规范员工的上下班、请假、外出工作等行为
- 方便准确计算员工的薪金
- 方便管理各种带薪假期
- 方便统计员工的加班信息
- 共享员工的请假及外出工作的信息

2.2 系统规格概述

考勤与请假系统采用 B/S 架构，前后端分离的实现方案，前端使用 vue+elementUI 的技术架构，后端使用 SpringBoot 架构实现，数据库采用 MySQL。

2.3 总体架构



考勤与请假系统，包括考勤、请假和信息管理三个模块，统筹解决部门员工的考勤和请假问题。

3. 详细设计

3.1 员工信息管理

3.1.1 功能概述

人事部员工能够根据信息在员工数据库中，增加新员工、删除旧员工、查询员工信息、更改员工信息，其中信息包括姓名、性别、电话、邮箱、部门等等；普通员工只能查询自己的信息，但无法修改。

3.1.2 录入新员工信息

编号	1.1	名称	录入新员工信息
执行者	人事部员工	优先级	高■ 低□
描述	人事部员工录入新员工信息		
基本流程	1.确认新员工的 id、姓名、性别、年龄、工龄、邮箱、部门编号 2.插入新员工信息		
结束状况	系统保存新员工信息数据，并提示成功增加的信息		

3.1.3 查询员工信息

编号	1.2	名称	查询员工信息
执行者	人事部员工	优先级	高■ 低□
描述	人事部员工查询员工信息		

基本流程	1.输入待查询员工 id 2.返回员工数据记录
结束状况	查询成功，返回员工信息

3.1.3 删除员工信息

编号	1.3	名称	删除员工信息
执行者	人事部员工	优先级	高■ 低□
描述	人事部员工删除员工信息		
基本流程	1.输入待删除员工 id 2.删除员工数据记录		
结束状况	删除成功，对应员工信息消失		

3.1.3 更改员工信息

编号	1.3	名称	更改员工信息
执行者	人事部员工	优先级	高■ 低□
描述	人事部员工更改员工信息		
基本流程	1.输入待更改员工 id 2.输入待修改属性的新值 3.返回修改成功后的员工信息		
结束状况	修改成功，对应员工信息的属性值更新		

3.2 请假管理

3.2.1 功能概述

普通员工可以提出请假申请和删除自己的申请，查看申请的审核状态。部门经理可以查看、审核自己部门员工申请，总经理可以查看、审核所有员工请假申请。

3.2.2 提出请假申请

编号	2.1	名称	提出请假申请
执行者	普通员工	优先级	高■ 低□
描述	普通员工提出请假申请		
基本流程	1.填写请假申请表中的请假理由、请假时间等信息 2.提交申请		
结束状况	生成新的未审核请假申请		

3.2.3 删除请假申请

编号	2.2	名称	删除请假申请
执行者	人事部员工、部门经理、总经理、普通员工	优先级	高■ 低□

	工		
描述	员工自己以及处于管理层人员能够删除员工请假申请		
基本流程	1.输入员工 id 2.删除对应员工的请假申请		
结束状况	删除成功，员工请假申请消失		

3.2.3 查询本人请假申请

编号	2.3	名称	查询本人请假申请
执行者	普通员工	优先级	高■ 低□
描述	普通员工查询自己的请假申请信息（包括审核状态）		
基本流程	1.输入员工自己的 id 2.返回该员工所有的请假申请信息		
结束状况	查询成功，得到对应员工请假申请信息		

3.2.4 查询所有待部门经理审核的请假申请

编号	2.4	名称	查询待部门经理审核的请假信息
执行者	部门经理	优先级	高■ 低□
描述	部门经理查询他需要审核的请假申请		
基本流程	1.输入部门经理自己的 id 2.返回待审核的员工请假申请列表		
结束状况	查询成功，返回待审核的员工请假申请列表		

3.3 考勤管理

3.3.1 功能概述

所有员工都需要上班签到，下班签退，人事部员工可以获取每月员工打卡报表。

3.3.2 签到

编号	3.1	名称	签到
执行者	所有员工	优先级	高■ 低□
描述	所有员工上班签到		
基本流程	1.点击签到按钮 2.系统判断位置与时间是否符合上班要求 3.完成签到		
结束状况	签到成功，月度报表增加一条记录		

3.3.3 签退

编号	3.2	名称	签退
执行者	人事部员工、部门经	优先级	高■ 低□

	理、总经理、普通员工		
描述	所有员工下班签退		
基本流程	1.点击签退按钮 2.系统判断位置与时间是否符合下班要求 3.完成签退		
结束状况	签退成功，月度报表增加一条记录		

3.3.4 导出考勤报表

编号	3.3	名称	导出考勤报表
执行者	人事部员工	优先级	高■ 低□
描述	人事部员工能够导出本月所有员工的考勤报表		
基本流程	1.人事部员工选择导出本月报表 2.返回本月所有员工签到签退纪录 1		
结束状况	得到本月考勤记录表		

3.4 请假审核管理

3.4.1 功能概述

管理层人员（部门经理、总经理、人事部员工）可以查看、审核相应范围内员工的请假申请。

3.4.2 查询待审核请假申请

编号	4.1	名称	查询待审核请假申请
执行者	总经理	优先级	高■ 低□
描述	总经理查询所有员工的请假申请		
基本流程	1.查询 2.返回所有员工的待审核请假申请列表		
结束状况	显示所有员工的待审核请假申请列表		

3.4.3 通过待审核请假申请

编号	4.2	名称	通过待审核请假申请
执行者	总经理	优先级	高■ 低□
描述	总经理通过所有员工提出的请假申请		
基本流程	1.输入提出请假申请员工 id 2.查看该员工提出的请假申请信息 3.通过该申请		
结束状况	请假申请状态变更为通过		

3.5 文件管理

3.5.1 功能概述

普通员工可以在提出请假申请时上传附件材料，管理层人员（部门经理、总经理、人事部员工）能够下载指定员工上传的附件材料，普通员工只能下载自己上传的附件材料。

3.5.2 上传图片

编号	5.1	名称	上传图片
执行者	所有员工	优先级	高■ 低□
描述	所有员工在提出请假申请时上传附件图片		
基本流程	1.员工提出情况申请 2.上传图片		
结束状况	请假申请信息中显示已上传图片		

3.5.3 获取图片

编号	5.2	名称	获取图片
执行者	所有员工	优先级	高■ 低□
描述	获取请假申请中上传的图片		
基本流程	1.普通员工查看自己的请假申请时，能获取自己上传的图片 2.管理层人员选择任意请假申请，获取图片		
结束状况	请假申请中图片被下载到本地		

4. 接口设计

4.1 信息管理

接口	说明
add(@RequestBody StaffVO staff)	增加一个员工
getByPersonnel	人事部查看员工信息
delete(@RequestParam("userId") String userId)	根据给定员工 id 删除该员工
update(@RequestBody StaffVO staff)	更新员工信息
getSelf(@RequestParam("userId"))	根据员工 id 查询部门、经理、人

String userId)	事部等信息
getMenu(@RequestParam("userId") String userId)	获取员工 id 为给定 userId 的员工的菜单栏信息

4.2 请假管理

接口	说明
apply(@RequestBody VacationVO vac)	员工提出请假申请
getSelf(@RequestParam("userId") String userId)	查询本人请假信息
delete(@RequestParam("userId") String userId)	根据给定员工 id 删除对应员工的请假信息
getByManager	查询对应部门经理需负责的请假申请
getByGeneral	查询总经理需负责的请假申请

4.3 考勤管理

接口	说明
getReport	人事部获取员工考勤报表
signIn	员工签到
signOut	员工签退

4.4 请假审核管理

接口	说明
getByPersonnel	人事部获取员工请假信息

getByManager	获取某个部门经理需审批的请假信息
getByGeneral	获取总经理需审批的请假信息

4.5 文件管理

接口	说明
upload	保存用户名为 userId 的用户在填写请假单时上传的附件图片
download	获取用户名为 userId 的用户上传的附件图片

5. 数据库设计

5.1 表结构

1) 员工表 staff

```
create table staff (
    user_id      char(5),      # 员工号,如'u1235'
    user_name    varchar(50),  # 姓名,如'刘亦菲'
    sex          varchar(10),  # 性别,{'男','女'}
    email        varchar(50),  # 电子邮箱
    age          int,          # 年龄
    employed     int,          # 工龄
    depart_id    int,          # 部门编号

    primary key (user_id) using btree
);
```

2) 签到表

```

create table sign_in (
    user_id      char(5),      # 员工号
    time_year    int,          # 签到时间-年
    time_month   int,          # 签到时间-月,1-12
    time_day     int,          # 签到时间-日,1-31
    time_hour    int,          # 签到时间-时,0-23
    time_minute  int           # 签到时间-分,0-59
);

```

3) 签退表

```

create table sign_out (
    user_id      char(5),      # 员工号
    time_year    int,          # 签退时间-年
    time_month   int,          # 签退时间-月,1-12
    time_day     int,          # 签退时间-日,1-31
    time_hour    int,          # 签退时间-时,0-23
    time_minute  int           # 签退时间-分,0-59
);

```

4) 部门表

```

# 部门表
create table department (
    id            int auto_increment,      # 部门编号,from 1
    depart       varchar(50),             # 部门名

    primary key (id) using btree
);

```

5) 考勤表

```

create table attendance (
    user_id      char(5),      # 员工号
    work_hour    int,          # 出勤(工作)时长(时)
    paid         int,          # 带薪假期时长(时)
    unpaid       int,          # 不带薪假期时长(时)
    valid        int,          # 有效工作时长,用于计算工资(时)

    primary key (user_id) using btree
);

```

6) 假期余额表

```

# 假期余额表
create table remaining (
    user_id      char(5),      # 员工号
    annual       int,          # 剩余年假(天)
    sick         int,          # 剩余带薪病假(天)
    marriage     int,          # 婚假
    check_up     int,          # 产检假
    maternity    int,          # 产假
    feeding      int,          # 哺乳假
    paternity    int,          # 陪产假
    affair       int,          # 事假

    primary key (user_id) using btree
);

```

7) 假期类型表

```

# 假期类型表
create table vac_type (
    id           int auto_increment, # 假期类型编号,from 1
    type        varchar(50),        # 假期类型名

    primary key (id) using btree
);

```

8) 请假单状态表

```

create table vac_status (
    id           int auto_increment, # 请假单状态编号,from 1
    status      varchar(50),        # 请假单状态名

    primary key (id) using btree
);

```

9) 请假单表

```

create table vacation (
    user_id      char(5),           # 员工号
    user_name    varchar(50),       # 姓名
    type_id      int,               # 假期类型编号
    status_id    int,               # 请假单状态编号
    depart_id    int,               # 员工所属部门编号
    start        timestamp,         # 请假开始时间
    end          timestamp,         # 请假结束时间
    vac_long     int,               # 请假时长(时)
    reason       varchar(200),      # 请假理由

    primary key (user_id) using btree
);

```

10) 权限表

```

create table sys_menu (
    menu_id      bigint(20) not null auto_increment, # id
    name         varchar(50) not null,              # 权限名称
    permission   varchar(200) default null,         # 权限标识

    primary key (menu_id) using btree
) engine=innodb auto_increment=1 default charset=utf8 row_format=dynamic;

```

11) 角色与权限关系表

```

# 角色与权限关系表
create table sys_role_menu (
    id          bigint(11) not null auto_increment, # id,
    role_id     bigint(11) default null,            # 角色id
    menu_id     bigint(11) default null,            # 权限id

    primary key (id) using btree
) engine=innodb auto_increment=1 default charset=utf8 row_format=dynamic;

```

12) 系统用户表

```

create table sys_user (
    user_id      bigint(11) not null auto_increment, # 用户id
    username     char(5) not null,                   # 用户名(以员工号作为 用户名)
    password     varchar(100) default null,          # 密码
    status       varchar(10) default null,           # 状态 prohibit: 禁用    normal: 正常

    primary key (user_id) using btree,
    unique key username (username) using btree
) engine=innodb auto_increment=1 default charset=utf8 row_format=dynamic;

```

13) 用户与角色关系表

```
create table sys_user_role (
  id      bigint(11) not null auto_increment,      # id
  user_id bigint(11) default null,                  # 用户id
  role_id bigint(11) default null,                  # 角色id

  primary key (id) using btree
) engine=innodb auto_increment=1 default charset=utf8 row_format=dynamic;
```

14) 角色表

```
create table sys_role (
  role_id      bigint(11) not null auto_increment,  # 角色id
  role_name     varchar(50) not null,               # 角色名称

  primary key (role_id) using btree
) engine=innodb auto_increment=1 default charset=utf8 row_format=dynamic;
```

6. 环境配置

6.1 开发环境

前端——*Visual Studio Code*

后端——*IntelliJ IDEA*

数据库——*Navicat Premium*

6.2 运行环境

前端——*Google Chrome*

后端——*JVM*

数据库——*MySQL*

6.3 测试环境

前端——*Google Chorme*

后端——*Postman*