考勤系统
--软件设计说明书

学院：　　智能与计算学部
年级：　　2016 级
项目成员：　万子航、刘岳森、肖业凡、窦淑洁

2018 年 11 月 23 日

目录

1. 引言

1.1 编写目的

1.编写本说明书的目的书阐述用户要求，描述出系统模型、功能、性能要求以及其他约定，为后期的如软件设计等工作提供依据。

2.本设计说明书的与其读者为用户、系统设计人员以及其他开发人员和相关审核检测人员

1.2 项目背景

某 CMMI5 级的软件公司，员工人数 100 人左右，大部分员工是软件研发人员，包括项 目经理、软件设计师，程序员，测试工程师等，除此之外还包括行政人员、财务人员。公司 在软件研发和日常管理上有一套成熟的管理方法，在没有考勤系统之前，与考勤相关的管理 工作是这样的

- 每位员工需要上午上班时打一次卡，下午下班是打一次卡，中午休息不需要打卡
- 期间如果需要外出工作，从公司出发时需要打一次卡，回到公司需要再打一次卡
- 员工请假需要填写请假条，请假分为事假、病假、年假等多种情况，请假需要直接领导审批，甚至还需要高层领导的审批。
- 行政部每天统计考勤信息，包括打卡信息、外出信息、请假信息，每月将考勤汇总信息提交给财务部。
- 财务部根据考勤汇总信息，调整员工的薪金。 但这样的管理方式，出现了一些意外事件:
- 某员工想请年假，但行政部告知该员工的当年度年休假已经休完了。年休假的管理出现了问题，很可能会影响员工的工作积极性。
- 某员工投诉当月薪金多扣了钱，原因时考勤信息统计有误。于是财务部将责任推导行政部，行政部推诿财务部要求不明确。
- 某天出现了紧急情况，高层领导想找员工 A 来处理，但员工 A 当天请了假，高层 领导并不知情。 公司高层期望通过考勤系统提高考勤工作的效率和准确性，避免因为考勤问题影响正常 工作。

1.3 定义

术语	解释
年假	统一规定为 10 天
婚假	统一规定为 3 天
产假	统一规定为女士 100 天，男士陪产假 10 天
vue	前端框架，Vue 系列产品为 3D 自然环境的动画制作和渲染提供了一系列的解决方案。Vue 系列有很多不同的产品，这是为了满足不同阶层的用户的需要：可以满足专业的制作工作室，同样也能满足 3D 自由艺术家
spring	pring 是一个开放源代码的设计层面框架，他解决的是业务逻辑层和其他各层的松耦合问题，因此它将面向接口的编程思想贯穿整个系统应用。Spring 是于 2003 年兴起的一个轻量级的 Java 开发框架，由 Rod Johnson 创建。简单来说，Spring 是一个分层的 JavaSE/EE full-stack(一站式) 轻量级开源框架
tomcat	Tomcat 是 Apache 软件基金会（Apache Software Foundation）的 Jakarta 项目中的一个核心项目，由 Apache、Sun 和其他一些公司及个人共同开发而成。由于有了 Sun 的参与和支持，最新的 Servlet 和 JSP 规范总是能在 Tomcat 中得到体现，Tomcat 5 支持最新的 Servlet 2.4 和 JSP 2.0 规范。因为 Tomcat 技术先进、性能稳定，而且免费，因而深受 Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的 Web 应用服务器
哈希	一般翻译做“散列”，也有直接音译为“哈希”的，就是把任意长度的输入（又叫做预映射 pre-image）通过散列算法变换成固定长度的输出，该输出就是散列值。这种转换是一种压缩映射，也就是，散列值的空间通常远小于输入的空间，不同的输入可能会散列成相同的输出，所以不可能从散列值来确定唯一的输入值。简单的说就是一种将任意长度的消息压缩到某一固定长度的消息摘要的函数
mysql	MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL 是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件。

1.4 参考资料

资料名称	版本/日期	说明
《uml 大战需求分析》	第一版/2012.02	张传波著
RESTful API 设计指南	2014 年 5 月 22 日	网络日志

【列出有关资料的作者、标题、编号、发表日期、出版单位或资料来源，可包括：
a. 项目经核准的计划任务书、合同或上级机关的批文；
b. 项目开发计划；
c. 需求规格说明书；
d. 测试计划（初稿）；
e. 用户操作手册（初稿）；
f. 文档所引用的资料、采用的标准或规范。】

2．任务概述

2.1 目标

- 规范员工的上下班、请假、外出工作等行为
- 方便计算员工薪金
- 方便管理各种带薪假期
- 共享员工的请假及外出工作信息

2.2 运行环境

主流浏览器：根据支持 vue 框架的浏览器可知，该项目适用于除 IE 浏览器以外的主流浏览器

2.3 需求概述

设计一个基于 web 的考勤系统/平台：各个职员（包括员工、部门经理、副总经理、总经理）可登录并实现各自的考勤功能：员工可以打卡、请假、查询并修改自己的请假信息、查询自己的考勤信息与年假情况、以及同事的请假情况；项目经理、副总经理、总经理可以进行请假审批、查询结果、外出申请等功能。

2.4 条件与限制

1）开发技术限制
前端采用 vue 技术进行设计+后端使用 spring bootr 进行设计
2）工具约束
采用 IntelliJ IDEA、VS code，tomcat 开发
3）性能约束
使用尽量少的数据、页面，对方法、ajax 等进行优化，保证流畅度

4）使用场景

在企业内接受考勤管理的员工和考勤制度的管理人员

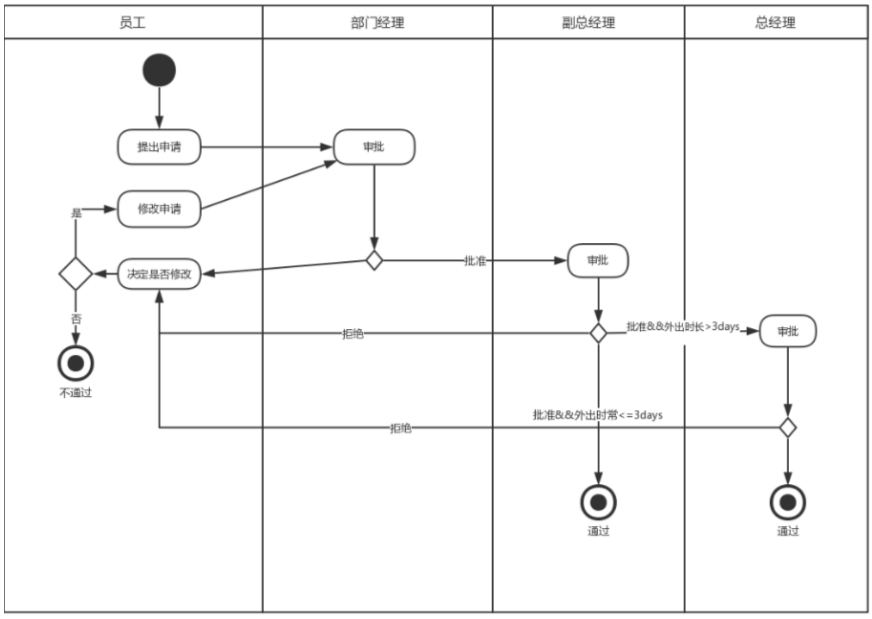
5）功能设计具体要求

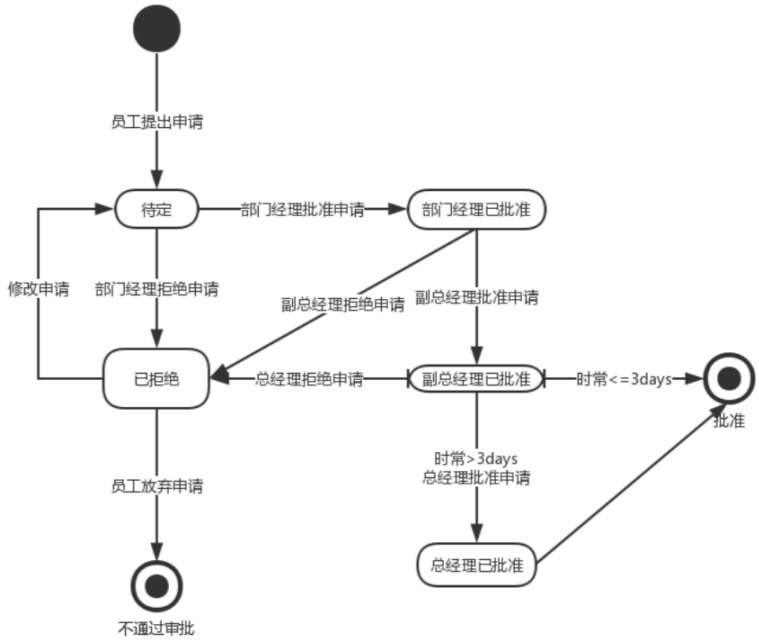
各个职员功能不同、界面不同，功能齐全的同时保证程序的简洁、易用

3. 总体设计

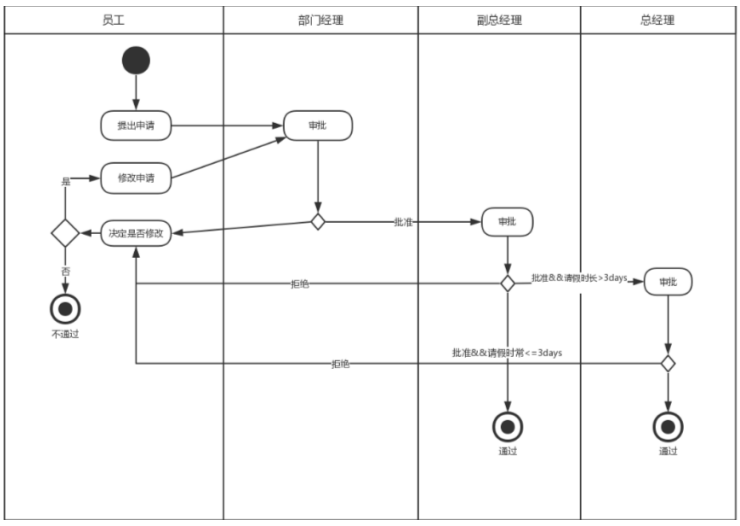
3.1 处理流程

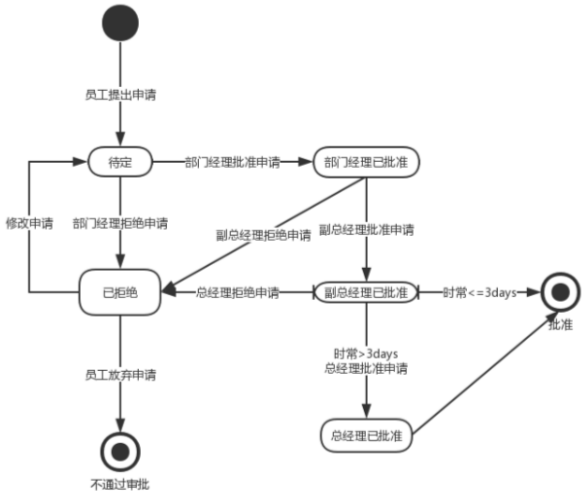
外出申请审批流程

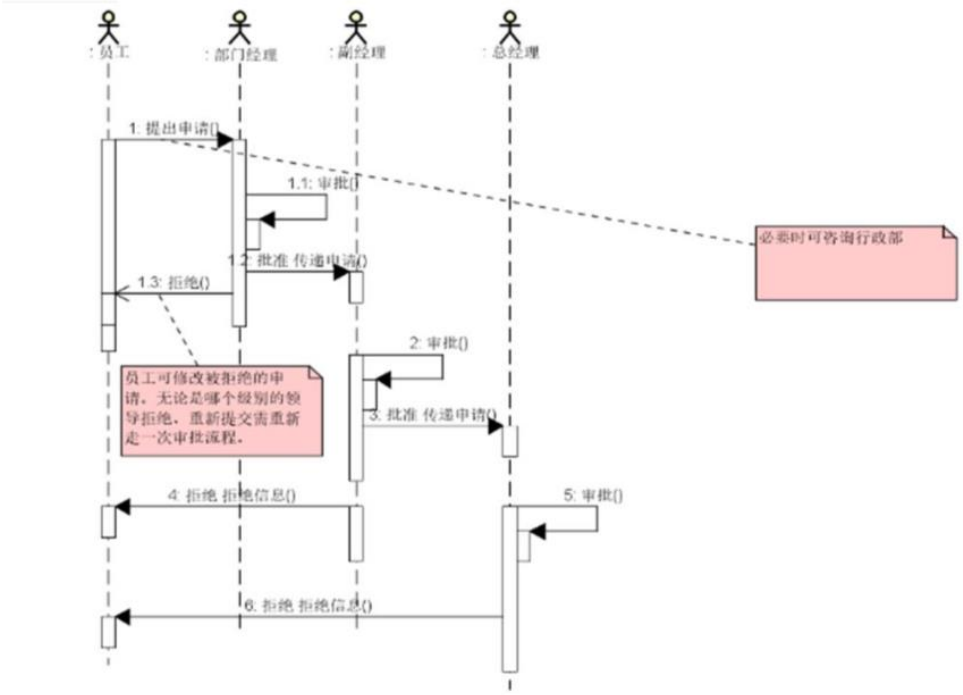



员工提出申请
待定
部门经理批准申请
部门经理已批准
修改申请
部门经理拒绝申请
副总经理拒绝申请
副总经理批准申请
已拒绝
总经理拒绝申请
副总经理已批准
时常<=3days
批准
时常>3days
总经理批准申请
员工放弃申请
总经理已批准
不通过审批

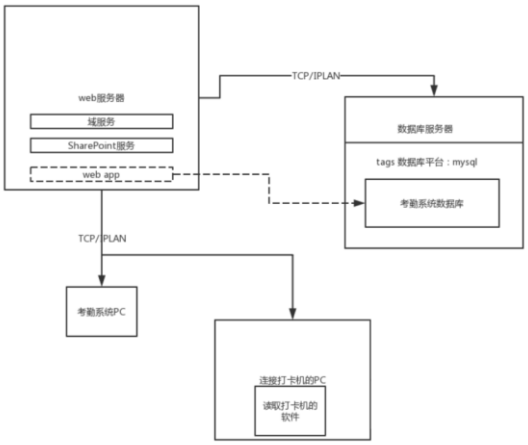
请假审批流程

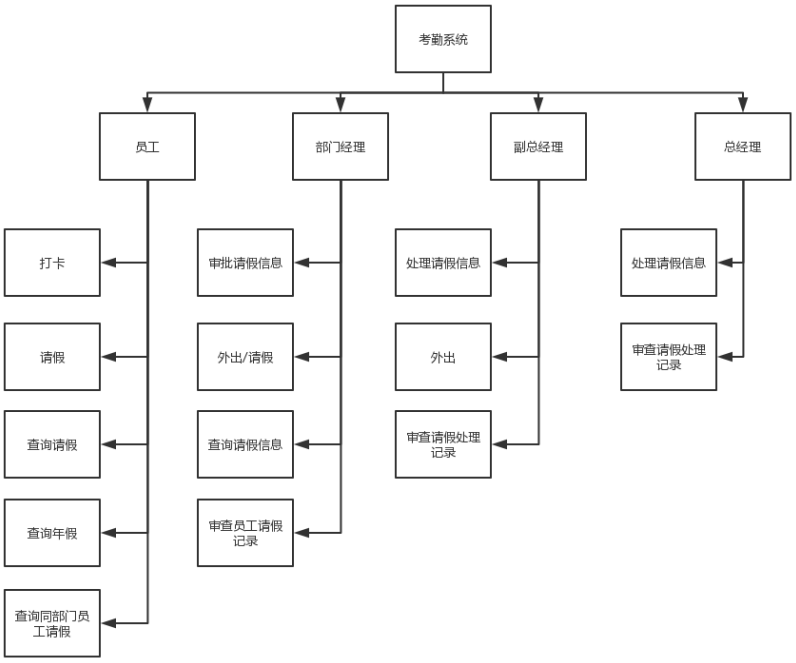

员工
部门经理
副总经理
总经理
提出申请
审批
修改申请
是
决定是否修改
批准
审批
否
不通过
拒绝
批准&&请假时长>3days
审批
批准&&请假时常<=3days
拒绝
通过
通过





3.2 总体结构和模块外部设计





3.3 功能分配

1) 打卡

a. 功能描述

实现员工上下班打卡功能，打卡记录分上下班分别存储入数据库内，行政人员可查询管理全部员工的打卡信息，打卡记录包括员工 id、日期、上班打卡时间、下班打卡时

间

b. 性能描述

精度：时间记录格式为 年-月-日 时分秒，打卡及查询操作响应时间在用户可接受范围内

2）请假申请查询

a. 功能描述

用户提出请假申请，完善请假申请的各项信息（id、原因、详细原因、起止时间、天数），提交申请，前端将数据传输给后端存入数据库中等待审批，请假查询界面可显示所有请假信息（包括未通过、已通过、未审批的请假信息），并标注每条请假信息请假人、时间、是否通过，有按钮，可以对未处理的请假信息进行取消或修改。

b. 性能描述

保证多个请假信息显示的查询速度，操作响应时间在用户可接受范围内

3）外出申请查询

a. 功能描述

用户提出外出申请，完善外出申请的各项信息，提交申请，前端将数据传输给后端存入数据库中等待审批，外出查询界面可现实所有外出信息，并标注每条外出信息、外出人、时间、是否通过。

b. 性能描述

保证多个外出信息显示的查询速度操作响应时间在用户可接受范围内

4）请假信息审批

a. 功能描述

经理层对员工提交的请假信息进行审批，系统从数据库中调出请假信息，分层级进行审批并给出意见，将审批建议存入数据库，审批后修改请假信息中的审批状态，进入下一领导审批，直到全部批准，将请假信息数据库更新。上层管理者可以审查下层管理者处理请假信息的情况。

b. 性能描述

操作响应时间在用户可接受范围内

4. 接口设计

4.1 外部接口

可运行的 pc 机、浏览器
需求的话可以设置打卡机，其他暂无

4.2 内部接口

接口名称	输入参数	返回参数

员工		
user_signup （新职员任职，将个人信息传到数据库中）	姓名：name（string） 身份/职位：status（string） 个人卡号：id（string） 密码：passwords（string）	输入正确成功注册 Response.data Message：true（bool） 未成功注册： Message：false（bool）
user_login （公司职员登录）	卡号：id（string） 密码：passwords（string） 身份：state（string）	信息正确登录成功： Response.data Message：true（bool） 未成功登录： Message：false（bool）
apply_leave （申请请假）	请假员工卡号：id（string） 职业：status（string） 请假信息/原因：reason（string） 请假时间：days：（int） 请假开始：start：string 请假结束：end：string	请假信息提交成功： Response.data state：true（bool） 未提交成功： state：false（bool）

Check_meaasge_colleage
（查询自己同事信息）
	员工卡号：id（string）	请假信息提交成功：
		Response.data
		Message{
		state：true（bool）
		reason：string
		Days：int
		start：string
		end：string
		}
		未提交成功：
		state：false（bool）
		Message：null

Check_meaasge_leave
（查询自己请假信息）
	请假员工卡号：id（string）	请假信息提交成功：
		Response.data
		Message{
		state：true（bool）
		reason：string
		Days：int
		start：string
		end：string
		}
		未提交成功：
		state：false（bool）
		Message：null

Check_message_annual_leave	员工卡号：id（string）	提交成功：

（查询年假情况和工资）		Response.data
		Message{
		state：true（bool）
		Days：int
		Salary：string
		}
		未提交成功：
		Message{
		state：false（bool）
		}

change_message_leave （修改请假信息）	请假员工卡号：id（string） 请假信息/原因：reason（string） 请假时间：days：（int） 请假开始：start：string 请假结束：end：string 职业：status（string）	请假信息修改成功： Response.data state：true（bool） 未修改成功： state：false（bool）
部门经理		

Get_leaves_manager_project
（获得部门以及相关部门请假情况）
	Status:string	Message{
		state：true（bool）
		reason：string
		Days：int
		start：string
		end：string
		}
		未提交成功：
		state：false（bool）
		Message：null

check_manager_vice （副总经理审核）	请假员工卡号：id（string） 职业：status（string） 请假信息/原因：reason（string） 请假时间：days：（int）	请假审核通过： Response.data state：true（bool） 未通过审核： state：false（bool）
Deal_with_bussiness （部门经理请假）	Status:string Reason：string Start：string End：string Days：string	申请成功 State:true 申请/提交失败 State：false
总经理		
Getleaves （得到请假信息）		Message{ state：true（bool） reason：string Days：int start：string

		end：string } 未提交成功： state：false（bool） Message：null

Get_forleaves
（查看部门处理请假信息）
		Message{
		state：true（bool）
		reason：string
		Days：int
		start：string
		end：string
		}
		未提交成功：
		state：false（bool）
		Message：null

5. 数据设计

5.1 数据表设计：

表 1：用户表

User:

属性名	类型
user_id（用户编号）	char（20）
passwords（用户密码）	char（20）
name（用户姓名）	char（20）
age（年龄）	int
sex（性别）	char（20）
department(员工部门)	char（20）
position（职位）	char（20）
annual_leave_length （年假长度）	int
remain_annual_leave （剩余年假）	int

表 2：打卡记录表

Punch_record

属性名	类型
sign_id（打卡记录编号）	Char(20)

user_id（用户编号）	Char(20)
day(今天日期)	date
sign_in（上班打卡时间）	datetime
sign_out（下班打卡时间）	datetime

表 3：外出申请表

Out_resords

属性名	类型
out_id（外出申请编号）	Char(20)
user_id（用户编号）	char(20)
start_time（开始时间）	datetime
end_time（结束时间）	datetime
reason（外出原因）	char(100)
state（审批状态）	int
days(天数）	int

表 4：请假申请表

Leave_records

属性名	类型
leave_id（请假申请编号）	Char(20)
user_id（用户编号）	Char(20)
start_time（开始时间）	datetime
end_time（结束时间）	datetime
reason（请假原因）	char(100)
state（审批状态）	int
days（总天数）	int
type（请假类型）	char（20）

表 5：部门经理处理事务表

Department_deals

属性名	类型
deal_id（处理事务编号）	Char(20)
department_manager_id（部门经理编号）	Char(20)
request_id(请求编号)	Char(20)
request_type（请求类型）	Char(20)
approval_time（审批时间）	date
reason(审批原因)	char(200)
state(处理状态)	int

表 6：副经理处理事务表

Deputy_deals

属性名	类型
deal_id（处理事务编号）	Char(20)
deputy _manager_id（副总经理编号）	Char(20)
request_id(请求编号)	Char(20)
request_type（请求类型）	Char（20）
approval_time（审批时间）	date
reason(审批原因)	char(200)
state(处理状态)	int

表 7：总经理处理事务表

Manager_deals

属性名	类型
deal_id（处理事务编号）	char（20）
manager_id	Char（20）
request_id(请求编号)	Char(20)
request_type（请求类型）	Char(20)
approval_time（审批时间）	date
Reason(审批原因）	char(200)
state(处理状态)	int

6. 程序描述

6.1 功能

2) 打卡

实现员工上下班打卡功能，打卡记录分上下班分别存储入数据库内，行政人员可查询管理全部员工的打卡信息，打卡记录包括员工 id、日期、上班打卡时间、下班打卡时间

5）请假申请查询

用户提出请假申请，完善请假申请的各项信息（id、原因、详细原因、起止时间、天数），提交申请，前端将数据传输给后端存入数据库中等待审批，请假查询界面可显示所有请假信息（包括未通过、已通过、未审批的请假信息），并标注每条请假信息请假人、时间、是否通过，有按钮，可以对未处理的请假信息进行取消或修改。

6）外出申请查询

用户提出外出申请，完善外出申请的各项信息，提交申请，前端将数据传输给后端存入数据库中等待审批，外出查询界面可现实所有外出信息，并标注每条外出信息、外出人、时间、是否通过。

7）请假信息审批

经理层对员工提交的请假信息进行审批，系统从数据库中调出请假信息，分层级进行审批并给出意见，将审批建议存入数据库，审批后修改请假信息中的审批状态，进入下一领导审批，直到全部批准，将请假信息数据库更新。上层管理者可以审查下层管理者处理请假信息的情况。

6.2 性能

1.精度：时间记录格式为 年-月-日 时分秒，打卡及查询操作响应时间在用户可接受范围内

2.保证多个请假信息显示的查询速度，操作响应时间在用户可接受范围内

2.保证多个外出信息显示的查询速度操作响应时间在用户可接受范围

6.3 输入项目

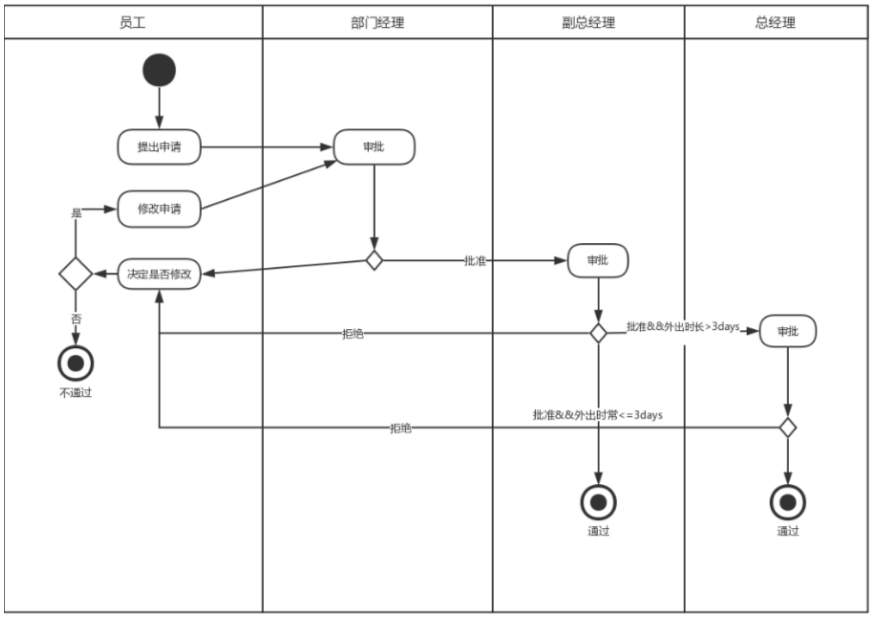
web 程序界面，输入框进行输入，按钮等切换界面进行交互

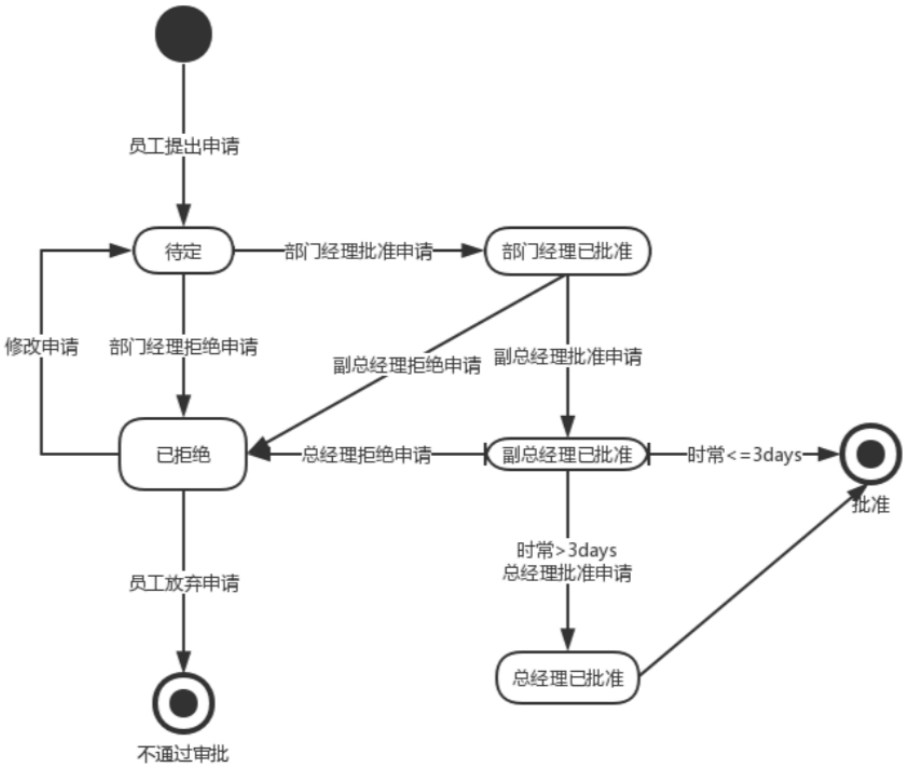
6.4 输出项目

后端传给前端数据动态渲染显示给用户

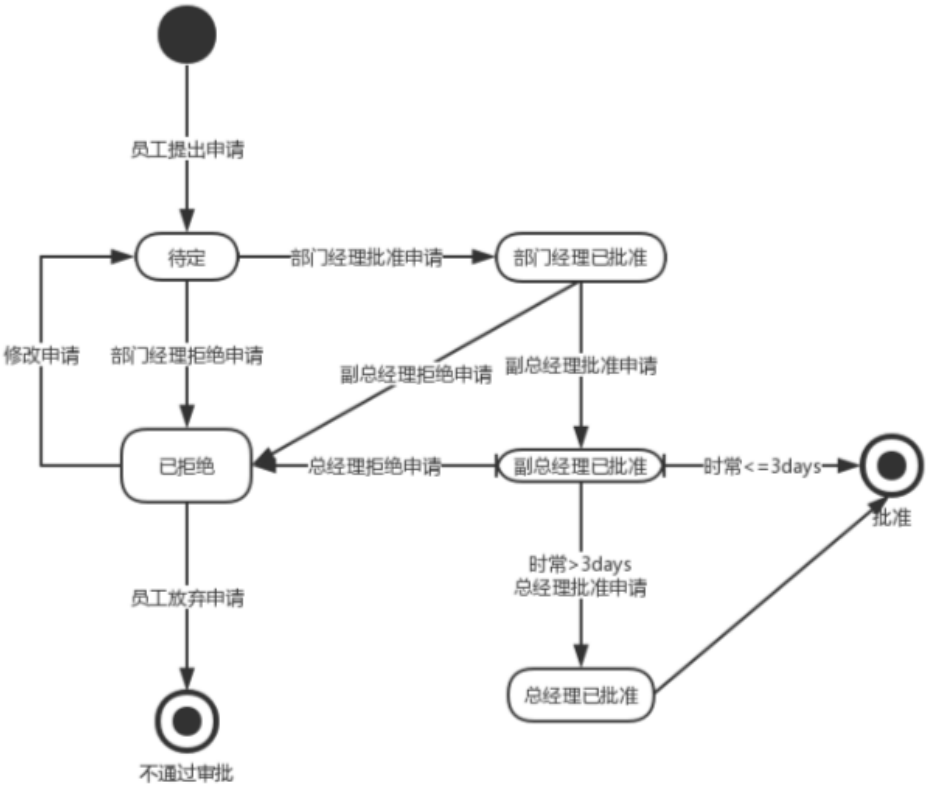
6.5 程序逻辑

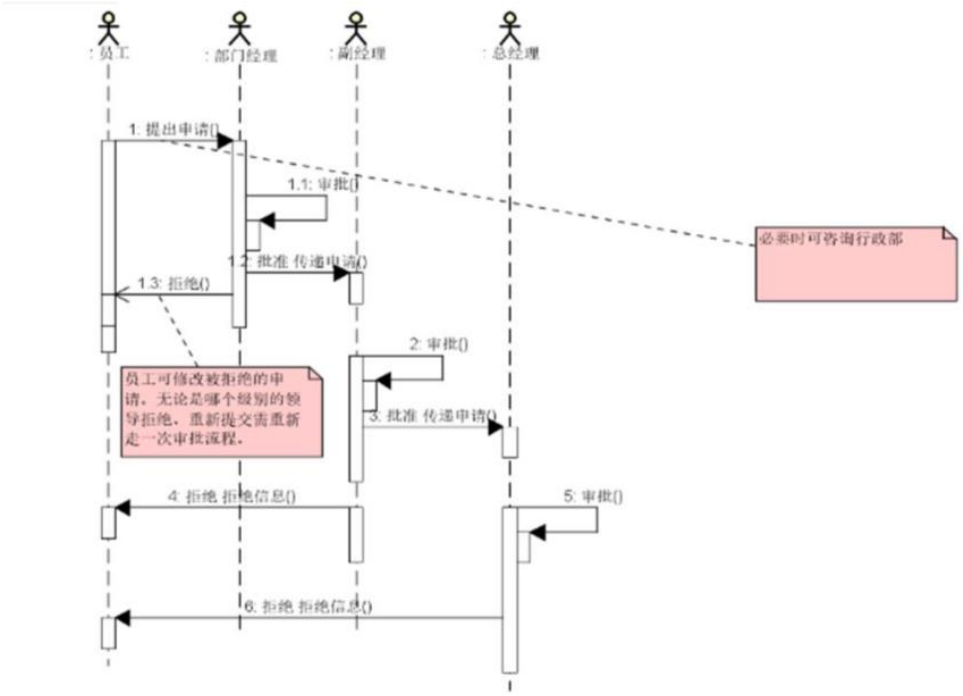
程序流程


员工
部门经理
副总经理
总经理
提出申请
审批
修改申请
是
决定是否修改
否
不通过
批准
审批
拒绝
批准&&外出时长>3days
审批
批准&&外出时常<=3days
拒绝
通过
通过

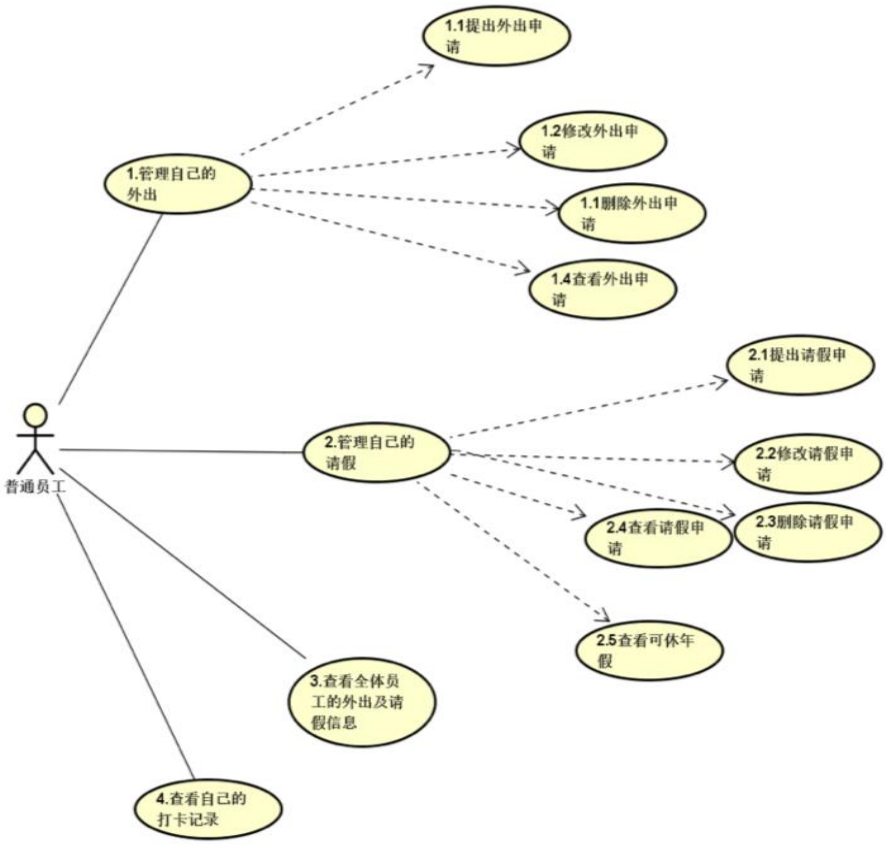

员工提出申请
待定
部门经理批准申请
部门经理已批准
修改申请
部门经理拒绝申请
副总经理拒绝申请
副总经理批准申请
已拒绝
总经理拒绝申请
副总经理已批准
时常<=3days
批准
时常>3days
总经理批准申请
员工放弃申请
总经理已批准
不通过审批

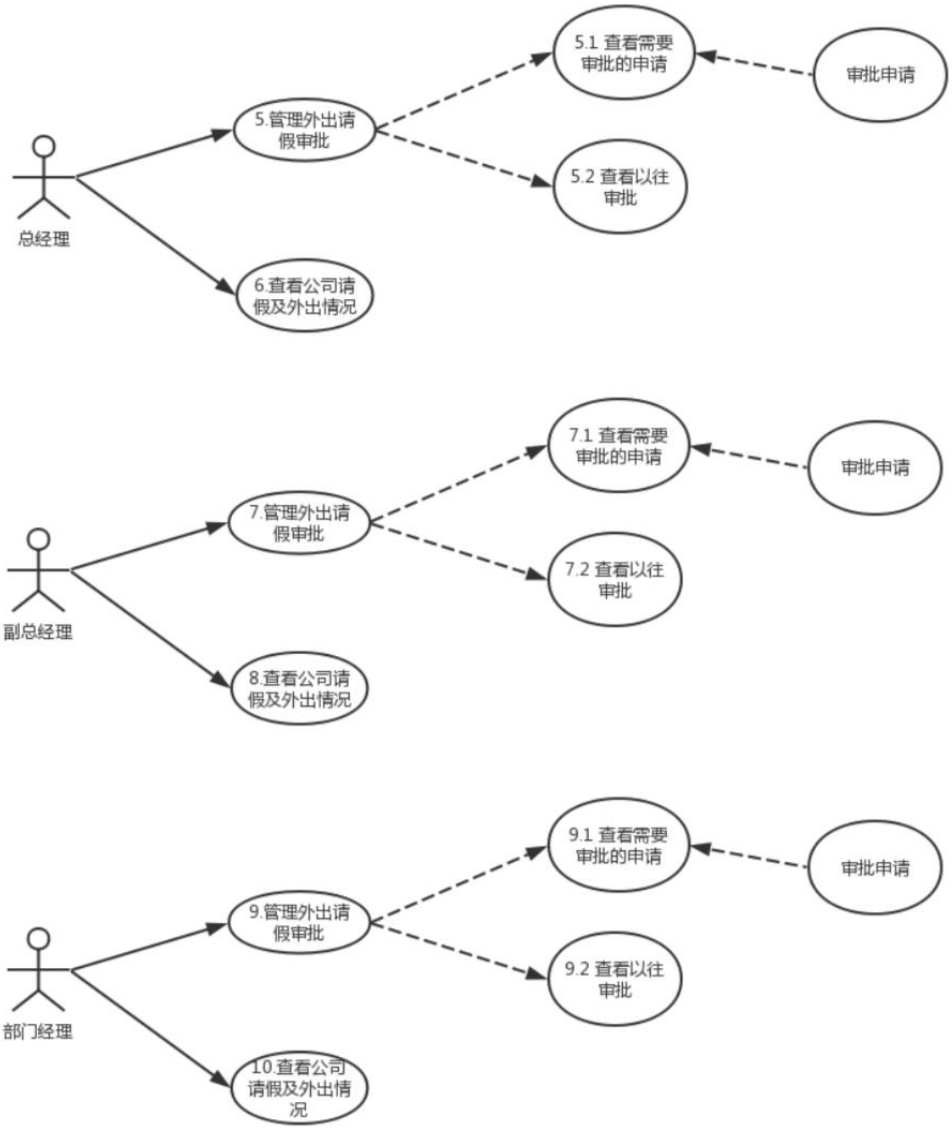
请假审批流程


员工提出申请
待定
部门经理批准申请
部门经理已批准
修改申请
部门经理拒绝申请
副总经理拒绝申请
副总经理批准申请
已拒绝
总经理拒绝申请
副总经理已批准
时常<=3days
批准
时常>3days
总经理批准申请
员工放弃申请
总经理已批准
不通过审批


:员工
:部门经理
:副经理
:总经理
1: 提出申请()
1.1: 审批()
1.2: 批准 传递申请()
1.3: 拒绝()
必要时可咨询行政部
2: 审批()
员工可修改被拒绝的申请。无论是哪个级别的领导拒绝，重新提交需重新走一次审批流程。
3: 批准 传递申请()
4: 拒绝 拒绝信息()
5: 审批()
6: 拒绝 拒绝信息()

用例图如下


1.1提出外出申请
1.2修改外出申请
1.管理自己的外出
1.1删除外出申请
1.4查看外出申请
2.1提出请假申请
普通员工
2.管理自己的请假
2.2修改请假申请
2.3删除请假申请
2.4查看请假申请
2.5查看可休年假
3.查看全体员工的外出及请假信息
4.查看自己的打卡记录


总经理
5.管理外出请假审批
5.1 查看需要审批的申请
5.2 查看以往审批
审批申请
6.查看公司请假及外出情况
副总经理
7.管理外出请假审批
7.1 查看需要审批的申请
7.2 查看以往审批
审批申请
8.查看公司请假及外出情况
部门经理
9.管理外出请假审批
9.1 查看需要审批的申请
9.2 查看以往审批
审批申请
10.查看公司请假及外出情况

6.6 接口

接口名称	输入参数	返回参数
员工		
user_signup （新职员任职，将个人信息传到数据库中）	姓名：name（string） 身份/职位：status（string） 个人卡号：id（string） 密码：passwords（string）	输入正确成功注册 Response.data Message：true（bool） 未成功注册： Message：false（bool）
user_login （公司职员登录）	卡号：id（string） 密码：passwords（string） 身份：state（string）	信息正确登录成功： Response.data Message：true（bool） 未成功登录： Message：false（bool）
apply_leave （申请请假）	请假员工卡号：id（string） 职业：status（string） 请假信息/原因：reason（string） 请假时间：days：（int） 请假开始：start：string 请假结束：end：string	请假信息提交成功： Response.data state：true（bool） 未提交成功： state：false（bool）
Check_meaasge_colleage （查询自己同事信息）	员工卡号：id（string）	请假信息提交成功： Response.data Message{ state：true（bool）

		reason：string Days：int start：string end：string } 未提交成功： state：false（bool） Message：null

Check_meaasge_leave
（查询自己请假信息）
	请假员工卡号：id（string）	请假信息提交成功：
		Response.data
		Message{
		state：true（bool）
		reason：string
		Days：int
		start：string
		end：string
		}
		未提交成功：
		state：false（bool）
		Message：null

Check_message_annual_leave （查询年假情况和工资）	员工卡号：id（string）	提交成功： Response.data Message{ state：true（bool）

		Days：int
		Salary：string
		}
		未提交成功：
		Message{
		state：false（bool）
		}

change_message_leave （修改请假信息）	请假员工卡号：id（string） 请假信息/原因：reason（string） 请假时间：days：（int） 请假开始：start：string 请假结束：end：string 职业：status（string）	请假信息修改成功： Response.data state：true（bool） 未修改成功： state：false（bool）
部门经理		

Get_leaves_manager_project（获得部门以及相关部门请假情况）	Status:string	Message{
		state：true（bool）
		reason：string
		Days：int
		start：string
		end：string
		}
		未提交成功：
		state：false（bool）
		Message：null

check_manager_vice （副总经理审核）	请假员工卡号：id（string） 职业：status（string） 请假信息/原因：reason（string） 请假时间：days：（int）	请假审核通过： Response.data state：true（bool） 未通过审核： state：false（bool）
Deal_with_bussiness （部门经理请假）	Status:string Reason：string Start：string End：string Days：string	申请成功 State:true 申请/提交失败 State：false
总经理		

Getleaves
 	
（得到请假信息）
		Message{
		
		state：true（bool）
		
		reason：string
		
		Days：int
		
		start：string
		
		end：string
		
		}
		
		未提交成功：
		
		state：false（bool）
		
		Message：null

Get_forleaves （查看部门处理请假信息）		Message{ state：true（bool） reason：string Days：int

		start：string end：string } 未提交成功： state：false（bool） Message：null

6.7 存储分配

正常采用：

存储分配策略包括：静态存储分配、栈和堆式存储分配；

存储分配算法包括：最佳适应算法、最先适应算法、循环最先适应算法。

6.8 限制条件

1）开发技术限制

前端采用 vue 技术进行设计+后端使用 spring bootr 进行设计

2）工具约束

采用 IntelliJ IDEA、VS code，tomcat 开发

4）性能约束

使用尽量少的数据、页面，对方法、ajax 等进行优化，保证流畅度

4）使用场景

在企业内接受考勤管理的员工和考勤制度的管理人员

5）功能设计具体要求

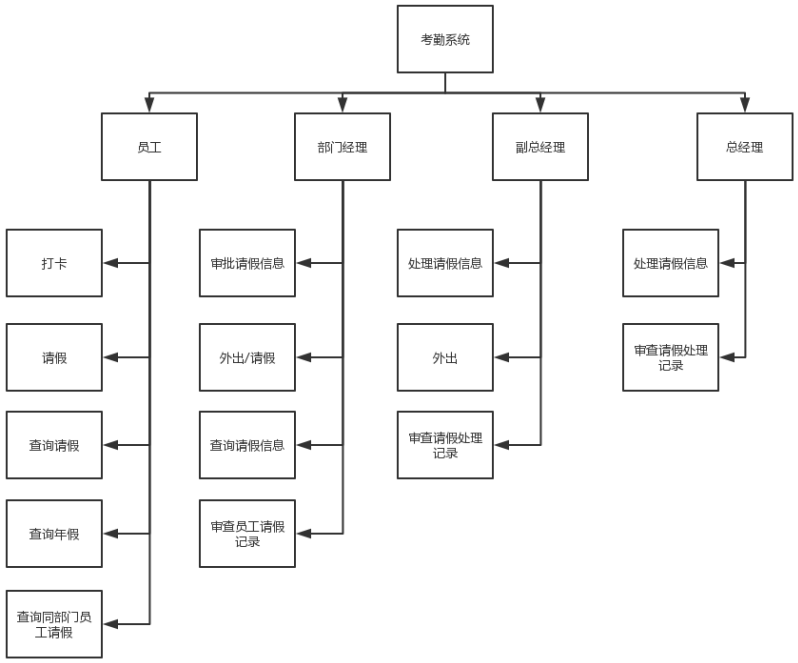
各个职员功能不同、界面不同，功能齐全的同时保证程序的简洁、易用

6.9 测试要点

1. 各个界面正常运行、切换
2. 数据库运行正常
3. 前后端正常联通
4. 各个接口运行正常、功能可以正常运行
5. 网络连接不顺时各种异常处理
6. 输入不符合要求时的健壮性测试

7. 运行设计

7.1 运行模块的组合



1.前后端部署到服务器，通过浏览器运行 web 程序
2.前端界面分出各个功能模块，分别对应后端接口

7.2 运行控制

1.Npm run dev 来测试前端页面
2.Npm run build 打包后部署到服务器测试
3.postman 来测试接口
4.后端部署到 tomcat 进行前后端链接测试
5.最后放到浏览器上运行

7.3 运行时间

用户使用期间的时间即为在浏览器的运行时间，在公司开始工作的时候开启服务，持续到员工下班

8. 出错处理设计

8.1 出错输出信息

1.由于前后端通信出的错误：显示提交、申请失败，稍后重试
2.由于网络错误：显示网络错误，重新连接
3.由于其他异常错误：通过异常处理机制进行分类输出

8.2 出错处理对策

1.遇到网络连接等第三方问题，采用恢复、再启动等方式
2.遇到小设计问题则通过性能降级的方式，暂时关闭一些功能，开启基本功能
3.遇到重大运行、设计问题错误的情况，则关闭程序进行修复或再设计，期间人工实现功能

9. 安全保密设计

1.职员信息权限仅限管理员访问、对数据库可读写等权限加以管理和限制
2.对密码进行散列化（哈希处理）
3.对网络安全加以考虑