

设计模式实践课程报告

郝 妍 2021229073

都泓含 2021229075

曹文颖 2021229088

一、系统的开发环境和框架

1. 项目构建: Maven 3.6
2. 集成框架: SpringBoot 2.6.3
3. 数据库: MySQL 5
4. Jdk 版本: jdk1.8
5. 开发工具: IDEA 2020.3
6. 持久层框架: MybatisPlus
7. 开发环境: Windows10

二、架构设计

整个项目基于 MVC 的设计模式, model 用来表征数据实体, mapper 用来访问数据, service 用来执行具体业务逻辑, controller 用来接收请求并发送响应。util 用来设计工具类, config 用来设计配置类。

三、开发流程

1. 创建工程

创建 SpringBoot 工程, 导入相关依赖

2. 设计并创建数据库

设计数据表, 使用 Navicat Premium 执行 sql 文件创建数据库

3. 创建数据库实体类

使用 lombok 插件, 利用@Data 注解帮助快速生成相应方法

4. 配置实体类和数据库映射

使用 lombok 插件, 利用@TableField 等注解匹配数据表

5. 创建数据访问层

利用 MyBatisPlus 的 BaseMapper 基类和 mapper.xml 映射文件, 创建实体类的
的数据访问层

6. 设计并创建业务层

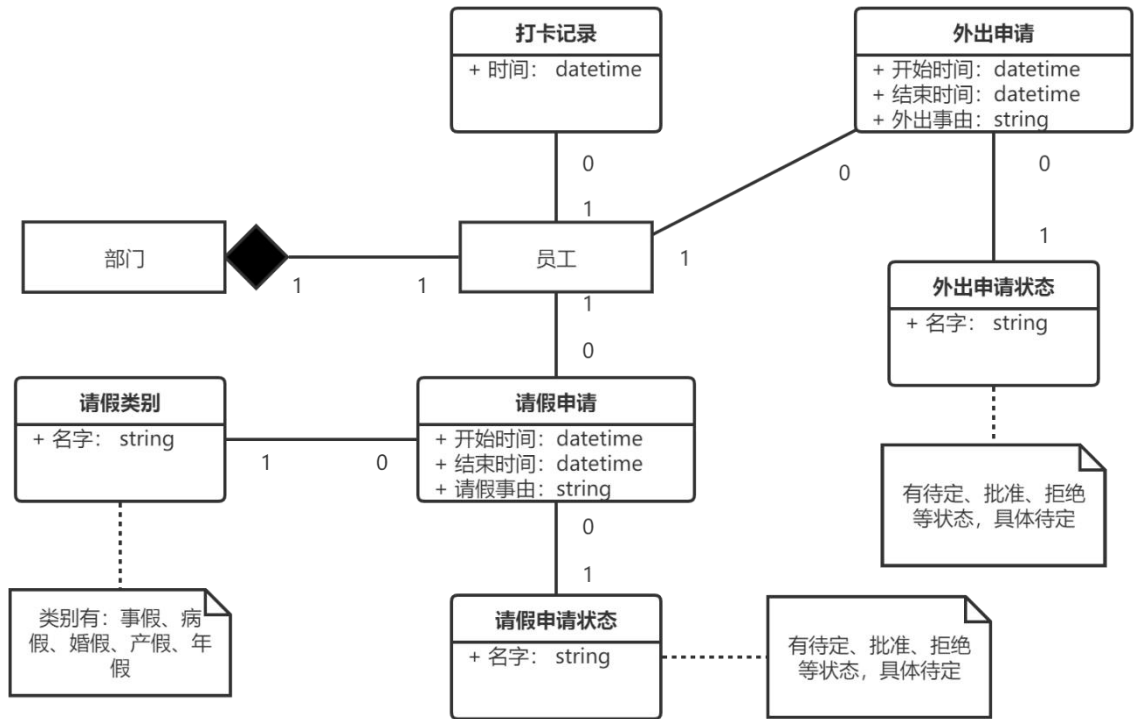
使用数据访问层编写业务层代码, 供控制器层调用

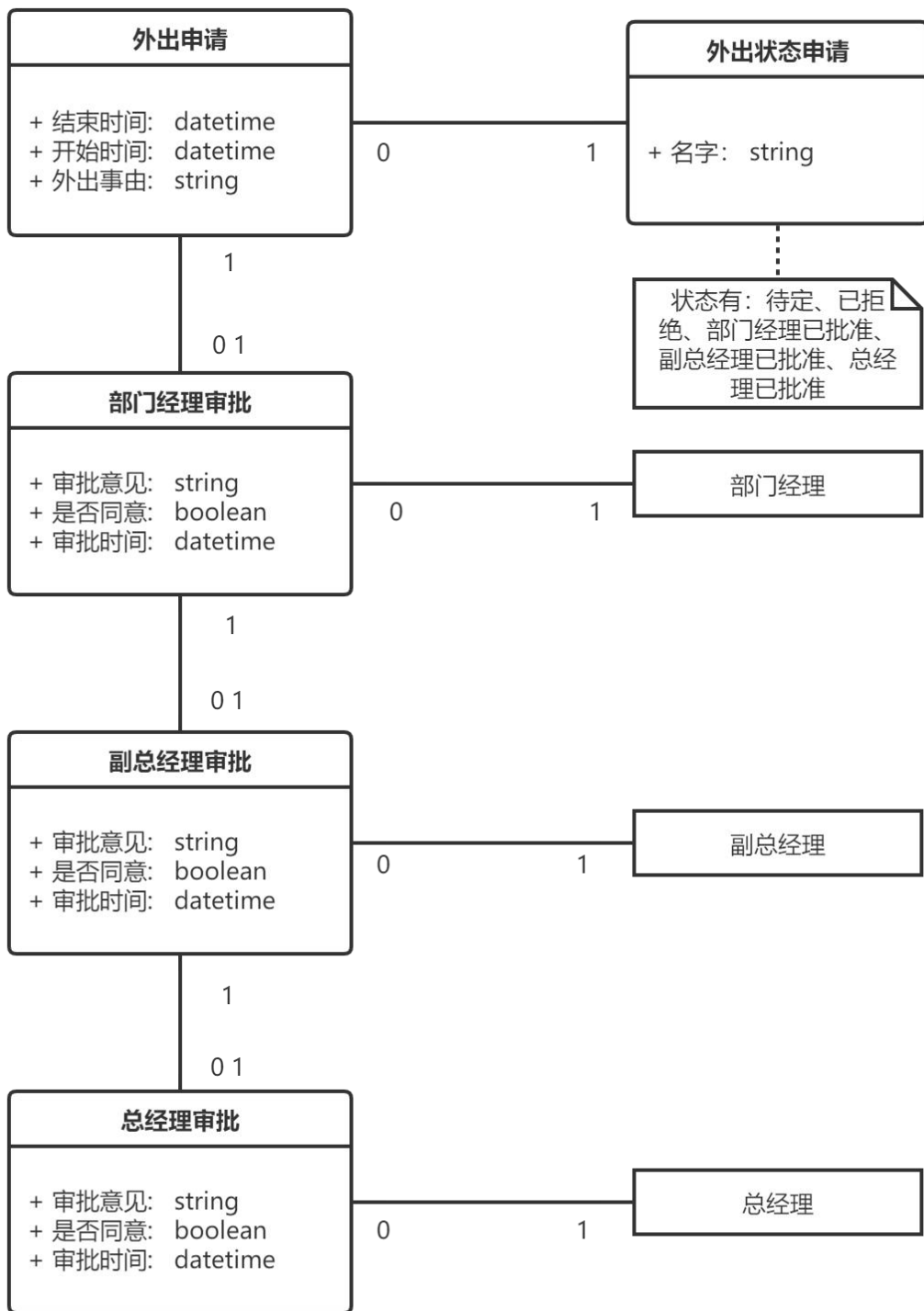
7. 设计并创建控制器层

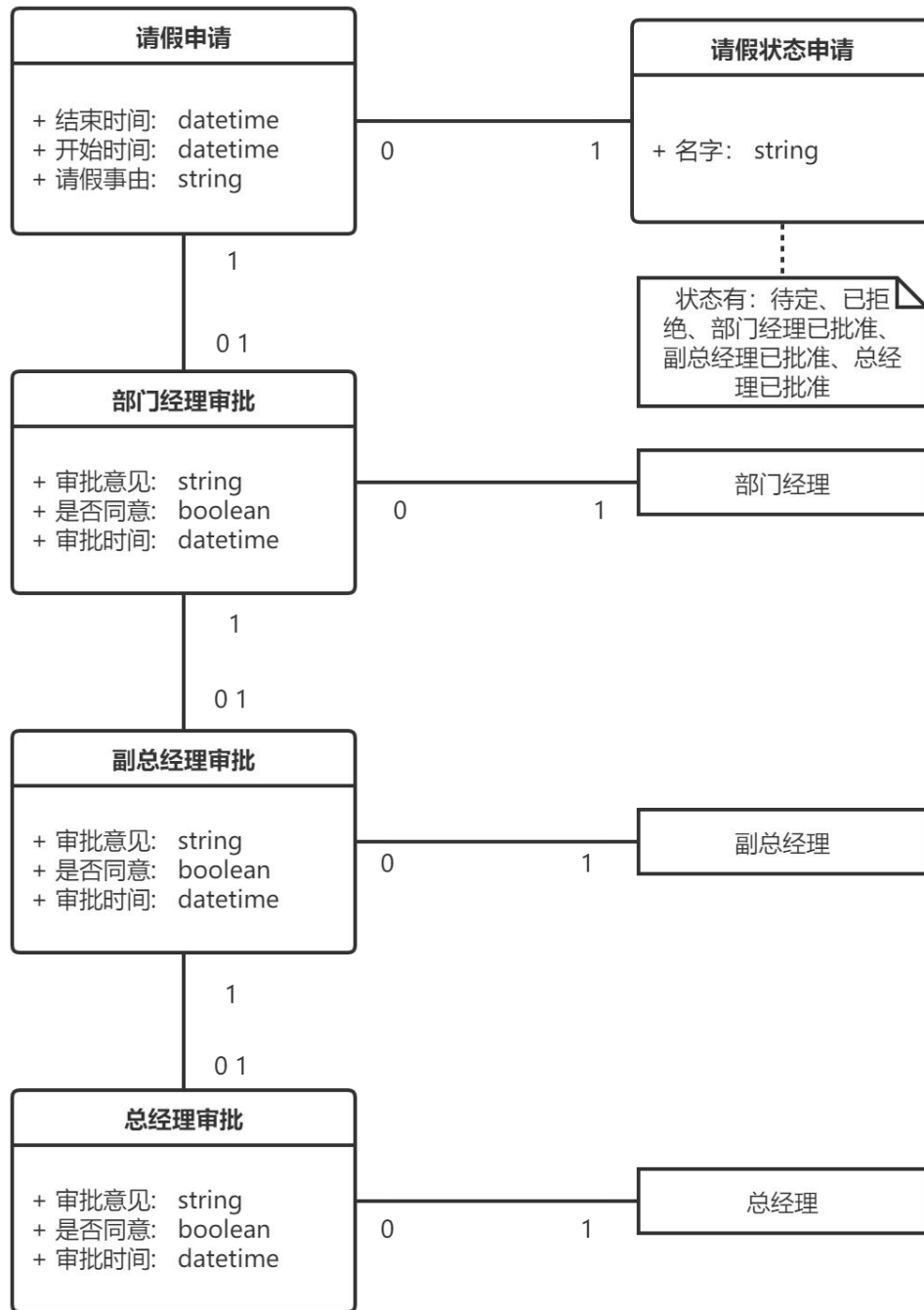
控制器层编写前端接口, 接收前端参数, 调用业务层代码并返回结果

四、需求分析

1. 业务概念图

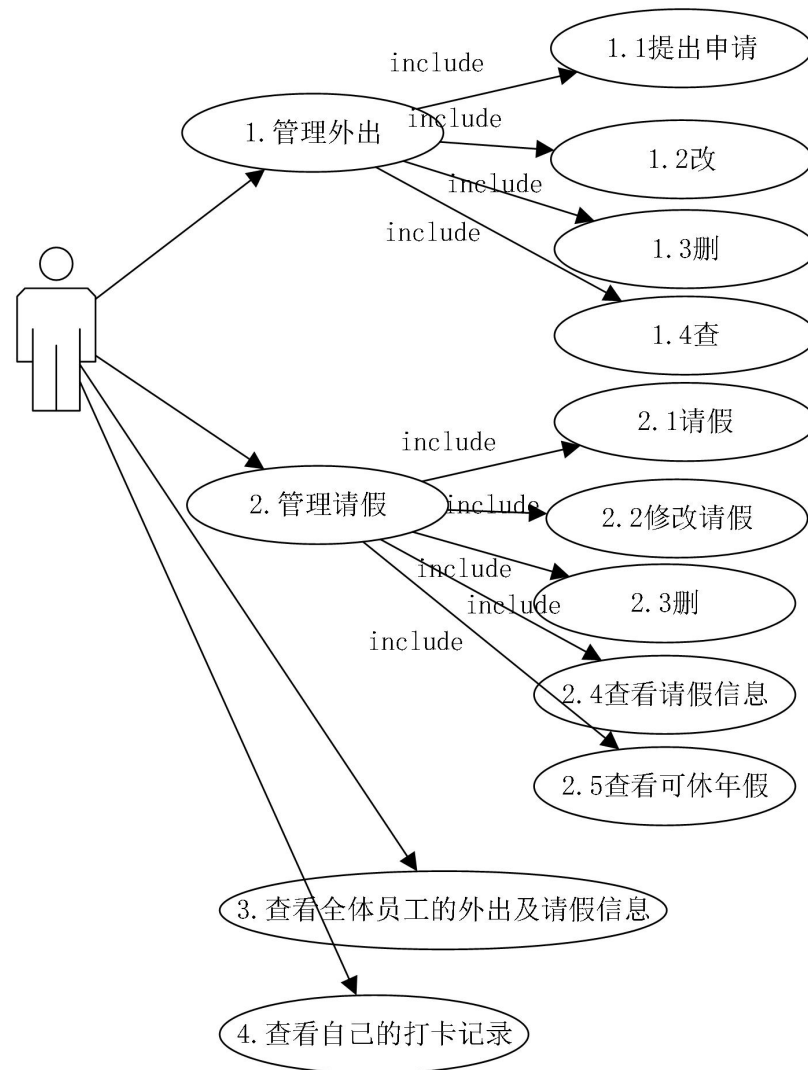


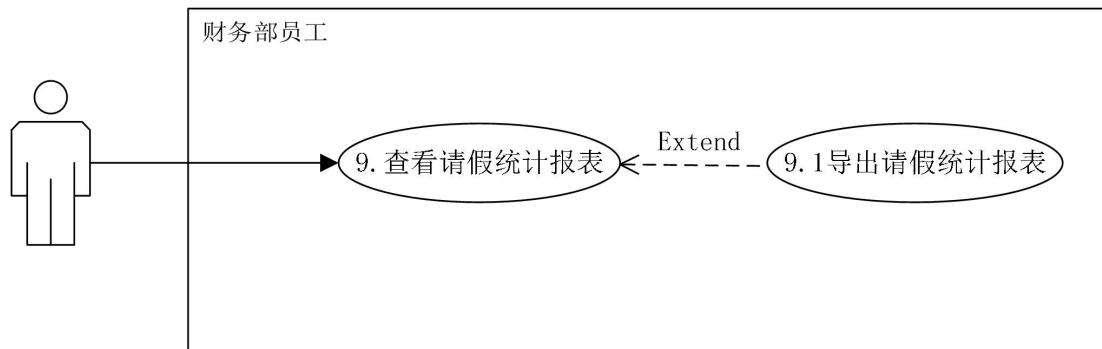
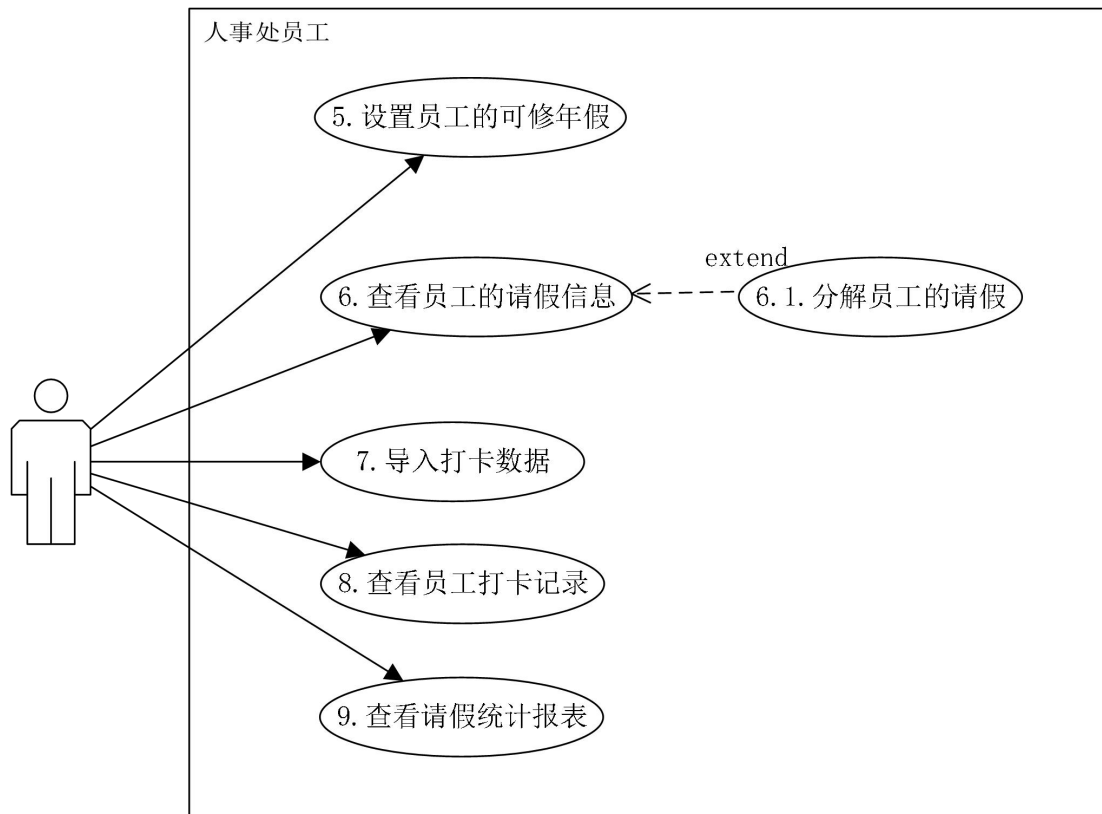




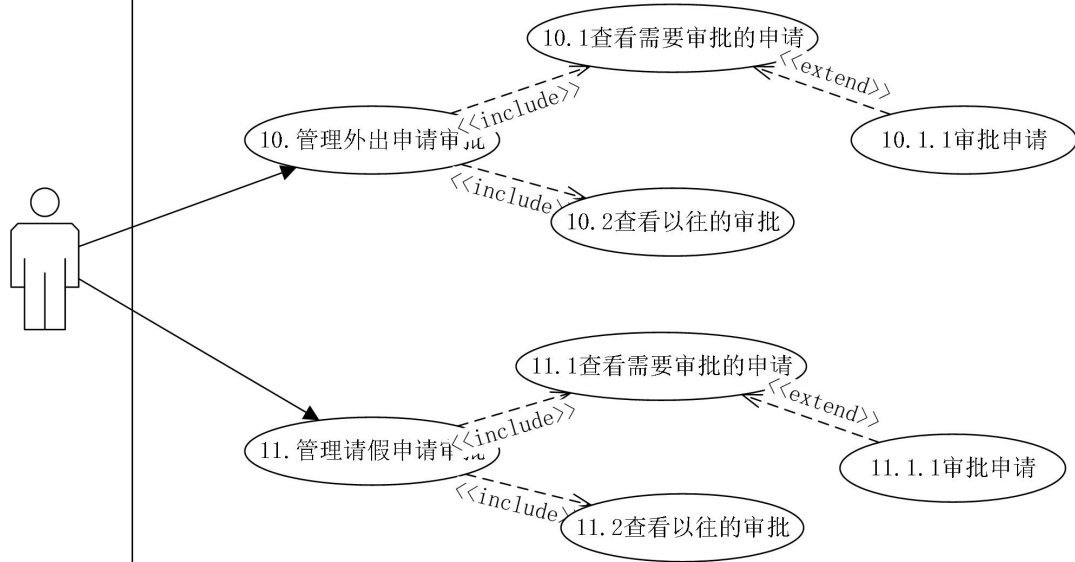
2. 用例图

员工用例图

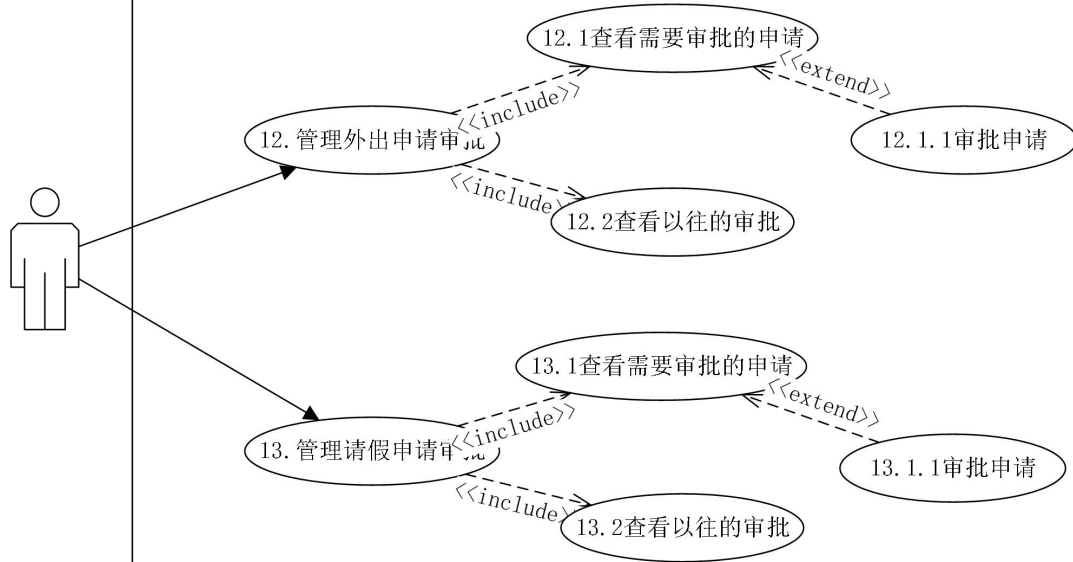


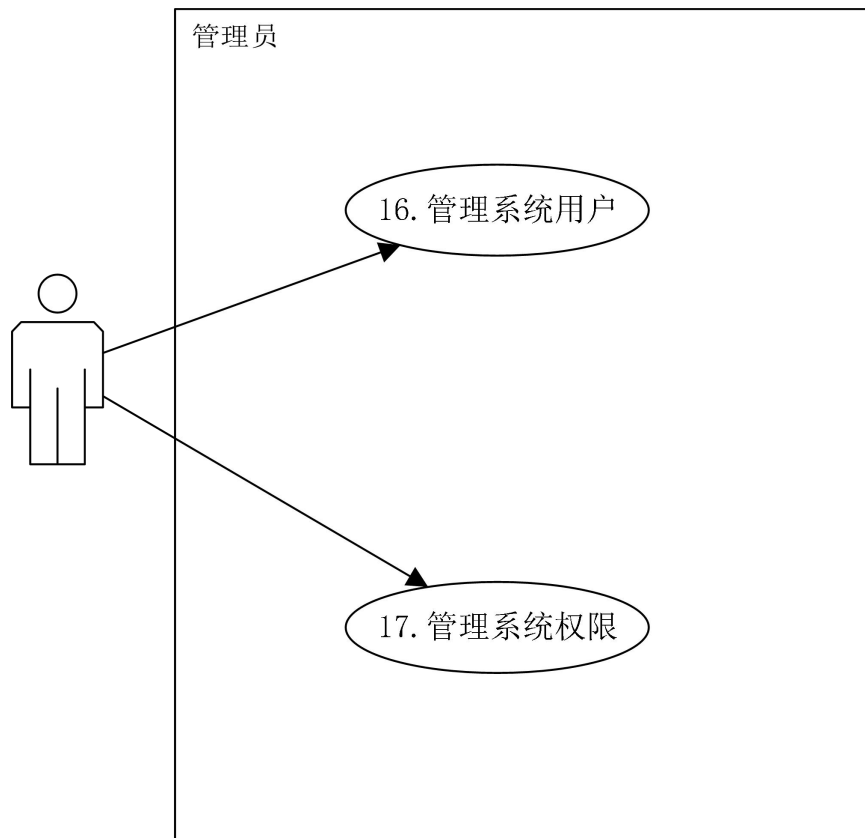
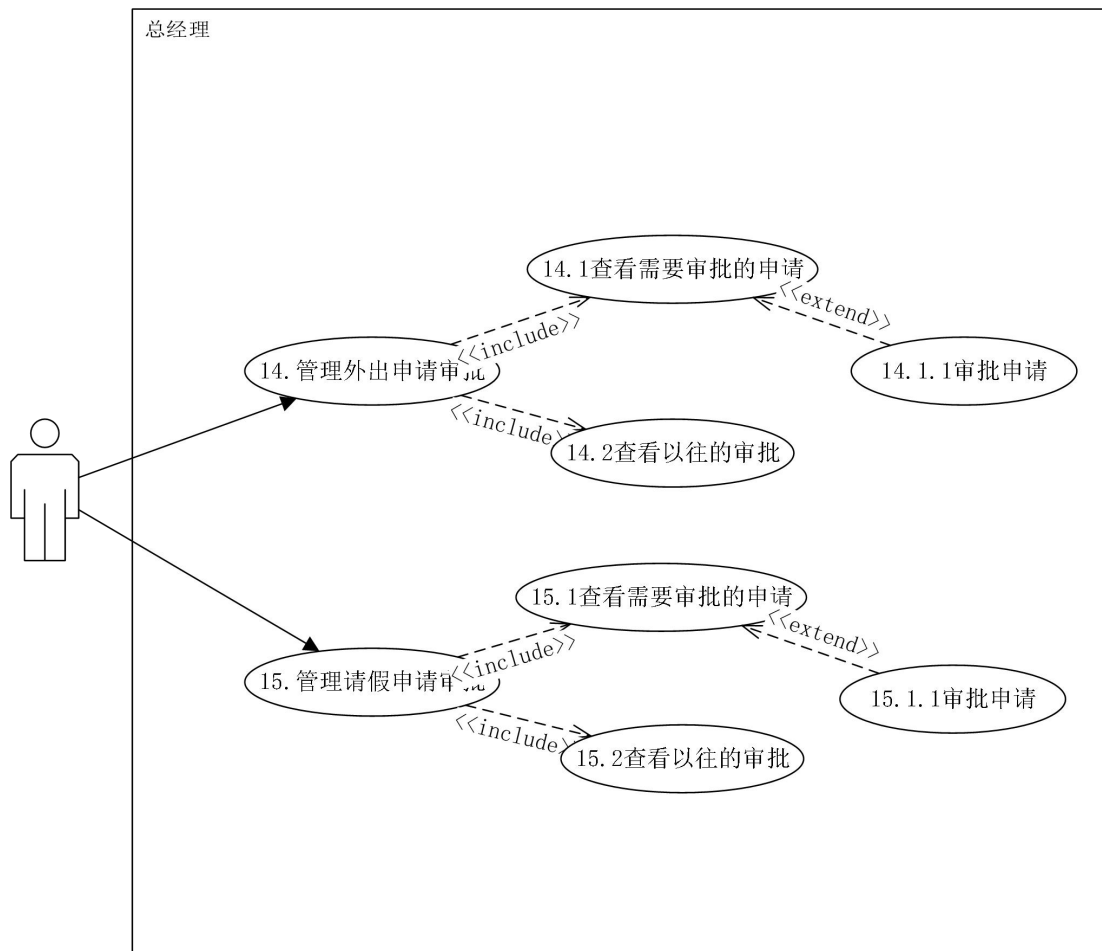


部门经理



副总经理





3. 用例描述

编号	1.1
名称	提出外出申请
执行者	普通员工
优先级	高
描述	普通员工录入外出信息，能成功提出外出申请
前置条件	无
基本流程	指示提出外出申请 显示外出申请表单 填写申请单，选择外出类别 指示提交申请 显示成功提交申请的信息
结束状况	系统保存外出申请数据，并提示成功提交申请的信息
可选流程 1	指示取消申请 5. 显示申请被取消的信息
异常流程	6. 修改外出申请，或取消外出申请
说明	外出申请单内容：申请者，开始时间，结束时间，外出事由，外出类别。 申请者默认当前用户，不可修改

编号	1.2
名称	修改外出申请
执行者	普通员工
优先级	高
描述	外出申请提出后，还没有任何审批前，申请者可修改外出申请 外出申请被拒绝后，申请者可修改外出申请，重新提交
前置条件	需存在已经提出的外出申请
结束状况	外出申请的状态变为“待定”，该申请需重新审批
说明	外出申请的状态为“已批准”时，申请者如果对该申请进行任何修改，其状态一律变为“待定”，需重新审批。 修改外出申请时，程序应进行判断，如果出现冲突，应拒绝本次修改，并给出相应提示。

编号	1.3
名称	删除外出申请
执行者	普通员工
优先级	高
描述	普通员工删除外出信息
前置条件	需存在已经提出的外出申请
基本流程	删除外出信息
结束状况	系统清除外出申请数据

编号	2.4
名称	查看外出申请
执行者	普通员工
优先级	高
描述	可方便的查看自己的外出申请的审批情况，可查看历史申请 系统默认按时间倒序显示当前用户的外出申请列表 在外出申请列表的基础上，用户可查看或修改其中一个具体的申请，或提出请假申请 用户在查看一个具体的申请时，才能删除该申请
前置条件	无
结束状况	系统数据不会发生任何变化

编号	2.1
名称	提出请假申请
执行者	普通员工
优先级	高
描述	普通员工录入请假信息，能成功提出请假申请
前置条件	无
基本流程	指示提出请假申请 显示请假申请表单 填写申请单，选择请假类别 指示提交申请 显示成功提交申请的信息
结束状况	系统保存请假申请数据，并提示成功提交申请的信息
可选流程 1	指示取消申请 5. 显示申请被取消的信息
异常流程	3. 填写请假单 请假类别为“年假” 5. 发现可休年假不足，显示提醒 6. 修改请假申请，或取消请假申请
说明	请假申请单内容：申请者，开始时间，结束时间，请假事由，请假类别。 申请者默认当前用户，不可修改 类别：事假，病假，婚假，产假，年假，只能而且必须二选一

编号	2.2
名称	修改请假申请
执行者	普通员工
优先级	高
描述	请假申请提出后，还没有任何审批前，申请者可修改请假申请 请假申请被拒绝后，申请者可修改请假申请，重新提交
前置条件	需存在已经提出的请假申请

结束状况	请假申请的状态变为“待定”，该申请需重新审批
说明	<p>请假申请的状态为“已批准”时，申请者如果对该申请进行任何修改，其状态一律变为“待定”，需重新审批。</p> <p>修改请假申请时，程序应进行判断，如果出现冲突，应拒绝本次修改，并给出相应提示。</p>

编号	2.3
名称	删除请假申请
执行者	普通员工
优先级	高
描述	普通员工删除请假信息
前置条件	需存在已经提出的请假申请
基本流程	删除请假信息
结束状况	系统清除请假申请数据

编号	2.4
名称	查看请假申请
执行者	普通员工
优先级	高
描述	<p>可方便的查看自己的请假申请的审批情况，可查看历史申请</p> <p>系统默认按时间倒序显示当前用户的请假申请列表</p> <p>在请假申请列表的基础上，用户可查看或修改其中一个具体的申请，或提出请假申请</p> <p>用户在查看一个具体的申请时，才能删除该申请</p>
前置条件	无
结束状况	系统数据不会发生任何变化

编号	2.5
名称	查看可休年假情况
执行者	普通员工
优先级	低
描述	<p>用户能看到按时间倒序排列的自己的年假申请，并能看到自己的当年年假总天数，及剩余可休的年假天数。</p> <p>用户可在此基础上，查看或修改其中一个具体的申请，或提出请假申请</p>
前置条件	行政部已设置该员工当年可休年假
结束状况	系统数据不会发生任何变化

编号	3
名称	查看全体员工的外出及请假信息

执行者	普通员工
优先级	高
描述	<p>目标:</p> <p>能方便地查看全体员工的外出及请假情况具体要求:</p> <p>1.用户可方便地查看当天、当周、当月所有的外出及请假情况，系统缺省显示当周的情况。用户可方便地在当天、当周、当月之间切换。</p> <p>2 系统显示当天情况时，用户可方便地切换到前一天或后一天;类似地，系统显示当周、当月情况时，用户也可以方便地切换到前一周、后一周或前一个月、后一个月。</p> <p>3.还没有通过审批的外出或请假申请，均应显示出来。</p> <p>4.用户可查看具体的一条外出或请假申请。</p> <p>5 除了该请假申请的审批者能查看请假申请的“请假事由”其他人不能查看“请假事由”，但可查看谁在什么时间请了什么类别的假</p>
前置条件	无
结束状况	系统的数据不会发生任何变化
说明	

编号	4
名称	查看自己的打开记录
执行者	普通员工
优先级	高
描述	系统默认按照时间倒序显示该用户的打开记录，用户可选择一个日期范围查询相应打卡记录
前置条件	相应的打卡记录数据应先导入系统
结束状况	系统的数据不会发生任何变化
说明	<p>打卡信息包括:员工 ID 打卡日期、打卡时间</p> <p>该用例员工只能查看自己的打卡记录，故只需要显示打卡日期、打卡时间即可</p>

编号	5	名称	设置员工的可休年假
执行者	行政部员工	优先级	低
描述	<p>目标:</p> <p>行政部可根据公司的年休假制度，设置每位员工每年的可休年假数量。</p> <p>具体要求:</p> <p>可查看全体员工可休年假列表，列表显示员工姓名、部门、当年可休年假总天数，当年已休年假天数。</p> <p>在查看可休年假列表的基础上，可设置每个员工的可休年假总数，可查看每个员工当年的请假类别为年假的请假申请。</p>		
前置条件	无		
结束状况	系统保存了更新后的该员工的可休年假总天数		
说明	<p>通常情况下，行政部设置员工可休年假的时间为:</p> <p>在每个自然年的第一个工作日，重新设置每个员工的可休年假数量。</p>		

	在新员工转正的第一天，设置该员工的可休年假数量。 但系统不需要限制修改时间。
--	---

编号	6	名称	查看员工的请假信息
执行者	行政部员工	优先级	低
描述	<p>目标： 行政部根据公司的相关制度，审核员工的请假申请。</p> <p>具体要求： 系统默认按时间倒序，显示通过了最终审批、但未通过行政部审核的员工请假申请列表。</p> <p>可再选择查看具体的一条请假申请。</p> <p>不符合相关制度的请假申请，可按以下两种方式之一处理： 执行用例“6.1 分解员工的请假”，具体参见用例 6.1。</p> <p>该申请不通过审核，通知申请者修改申请。系统不支持这种处理方式，行政部可通过电话、Email、口头等方式，通知申请者修改请假申请。</p>		
前置条件	无		
结束状况	系统不保存任何信息		
说明	<p>参见请假审批流程活动图，通过副总经理审批的 3 天或以内的请假，通过总经理审批的超过三天的请假都需要行政部进行审核。</p> <p>实际上行政部不需要对全部请假进行审核，一般只需要对婚假、产假等涉及到比较复杂的国家政策的申请进行审核，行政部的审核也不需要立刻进行，有时候每月统一审查一次就可以了。本系统不支持行政部的审核功能，只支持查看功能，但行政部可以在查看的基础上，不通过本系统完成审核工作。</p>		

编号	6.1	名称	分解员工的请假
执行者	行政部员工	优先级	低
描述	<p>目标： 行政部可分解不符合要求的请假申请，使分解后的请假符合要求，分解后的请假总天数不变、起止时间不变。</p> <p>例：某员工申请了 10 天的婚假，但行政部审核时发现该员工不符合晚婚政策，只能享受 3 天婚假，于是与该员工协商，将该请假分解为 3 天婚假、5 天年假、2 天事假。</p> <p>具体要求： 在查看员工具体一条请假信息的基础上，可分解该请假。</p> <p>分解请假时，需输入请假类别、时长。</p> <p>分解后的总时长等于原来申请的时长，总起止时间不变，系统按照分解后申请的先后顺序自动生成各申请的起止时间。</p> <p>分解后的请假无需再次审批，自动为已批准状态</p>		
前置条件	无		
结束状况	系统保存了解析后的请假申请，原请假申请不再保留		
说明	<p>参见业务概念图。</p> <p>行政部与申请者的协商过程，是系统范围外的工作。</p>		

编号	7
名称	导入打卡数据
执行者	行政部员工
优先级	低
描述	<p>目标:</p> <p>将打卡记录导入到系统中,以便用户通过本系统查询打卡记录。具体要求:</p> <p>1.系统可导入保存有打卡记录的 Excel 文件。</p> <p>2 导入的数据以“增加”的方式保存到系统中,系统不判断新导入的数据是否与之前的数据有冲突</p>
前置条件	无
结束状况	打卡记录保存到系统中
说明	<p>打卡数据记录在打卡机中行政部需要每天用电脑连接打卡机来读取数据读取的数据是 Excel 格式,读取数据的软件是打卡机配套提供的。打卡记录包含:员工 ID 打卡日期、打卡时间</p>

编号	8
名称	查看员工的打卡记录
执行者	行政部员工
优先级	低
描述	<p>目的:</p> <p>掌握各员工的打卡情况,方便与员工的请假申请、外出申请进行比较,以核实各员工的考勤信息。具体要求:</p> <p>1.系统默认按照时间的倒序列出各员工的打卡记录,需要显示的内容有:员工姓名、所属部门、打卡日期、打卡时间。</p> <p>2.用户可按时间范围、所属部门、员工姓名来筛选显示打卡记录</p>
前置条件	系统需已存在导入的打卡记录
结束状况	系统数据不会发生变化
说明	<p>行政部是可以查看全体员工的打卡记录的,其目的是通过打卡记录、请假申请、外出申请的比较,来核实各员工的考勤情况,判断员工有没有迟到、早退、旷工等情况,制作相应的考勤报表提交给财务部,财务部根据该报表来计算员工当月的薪金。</p> <p>考勤报表是这样的一张报表:记录了当月影响员工薪金的所有考勤情况,影响员工薪金的考勤情况有:迟到、早退、旷工、非带薪假期。该报表由行政部制作,交由财务部作为员工薪金计算及调整的依据。</p>

编号	9
名称	查看请假统计报表
执行者	行政部员工,财务部员工
优先级	高

描述	<p>目标:</p> <p>行政部的目标有:根据请假统计报表, 检查各员工的请假情况, 特别是带薪年假, 是否符合公司的相关制度要求。</p> <p>核实各员工的请假情况, 作为制作考勤报表的依据。财务部的目标有:作为当月员工薪金计算的参考依据。具体要求:</p> <p>1 报表首先根据员工分组, 然后根据请假类别分组, 列出分组后汇总的请假天数。</p> <p>2 可按日期范围、所属部门、员工姓名、请假类别来筛选统计数据范围。</p> <p>3.可在查看报表的基础上执行用例“91 导出请假统计报表”</p>
前置条件	无
结束状况	系统数据不会发生变化
说明	<p>考勤报表参见用例 “8 查看员工的打卡记录”的用列表中的说明。</p> <p>财务部计算当月员工薪金的直接依据是行政部提交的“考勤报表”, 该请假统计报表只是参考。行政部每月需要根据请假统计报表, 同时还需要查看员工打卡记录、外出申请记录、请假申请记录等, 经过综合判断后制作考勤报表</p>

编号	9.1
名称	导出请假统计报表
执行者	行政部员工, 财务部员工
优先级	高
描述	<p>目标:</p> <p>本用例主要有行政部执行, 导出 Excel 报表后, 行政部可在该 Excel 文件的基础上制作“考勤报表”。具体要求:</p> <p>1.用户可在查看请假统计报表的基础上, 指示导出到 Excel 文件中。</p> <p>2 系统将当前统计报表中的数据导出到 Excel 文件中, 并且该文件输出到用户所在的电脑上</p>
前置条件	无
结束状况	系统数据不会发生变化
说明	Excel 文件的内容及格式, 与用例 “9.查看请假统计报表” 的一致

编号	11.1
名称	查看需要审批的申请
执行者	部门经理
优先级	高
描述	<p>目标:</p> <p>部门经理可方便地查看需要他审批的申请, 并可以在此基础上方便地审批申请。具体要求:</p> <p>1.系统默认按照请假申请提出时间的顺序, 列出状态为“待定”的请假申请列表。</p> <p>2.该请假申请列表需显示:申请者姓名、所属部门、请假类别、请假起止时间、请假事由、请假申请的状态。</p>

	<p>3.用户可直接在此请假申请列表的基础上。直接审批某个申请，参见用例“11.11 审批申请”</p> <p>4、用户可在此请假申请列表的基础上。选择查看具体的某个申请，并进行审批，参见用例“11.11 审批申请”</p>
前置条件	无
结束状况	系统的数据不会发生变化
说明	<p>需要部门经理审批的请假申请是状态为“待定”的申请:申请者提出请假申请后，申请的状态为“待定”。</p> <p>申请者修改被拒绝的申请，申请的状态变为“待定”</p>

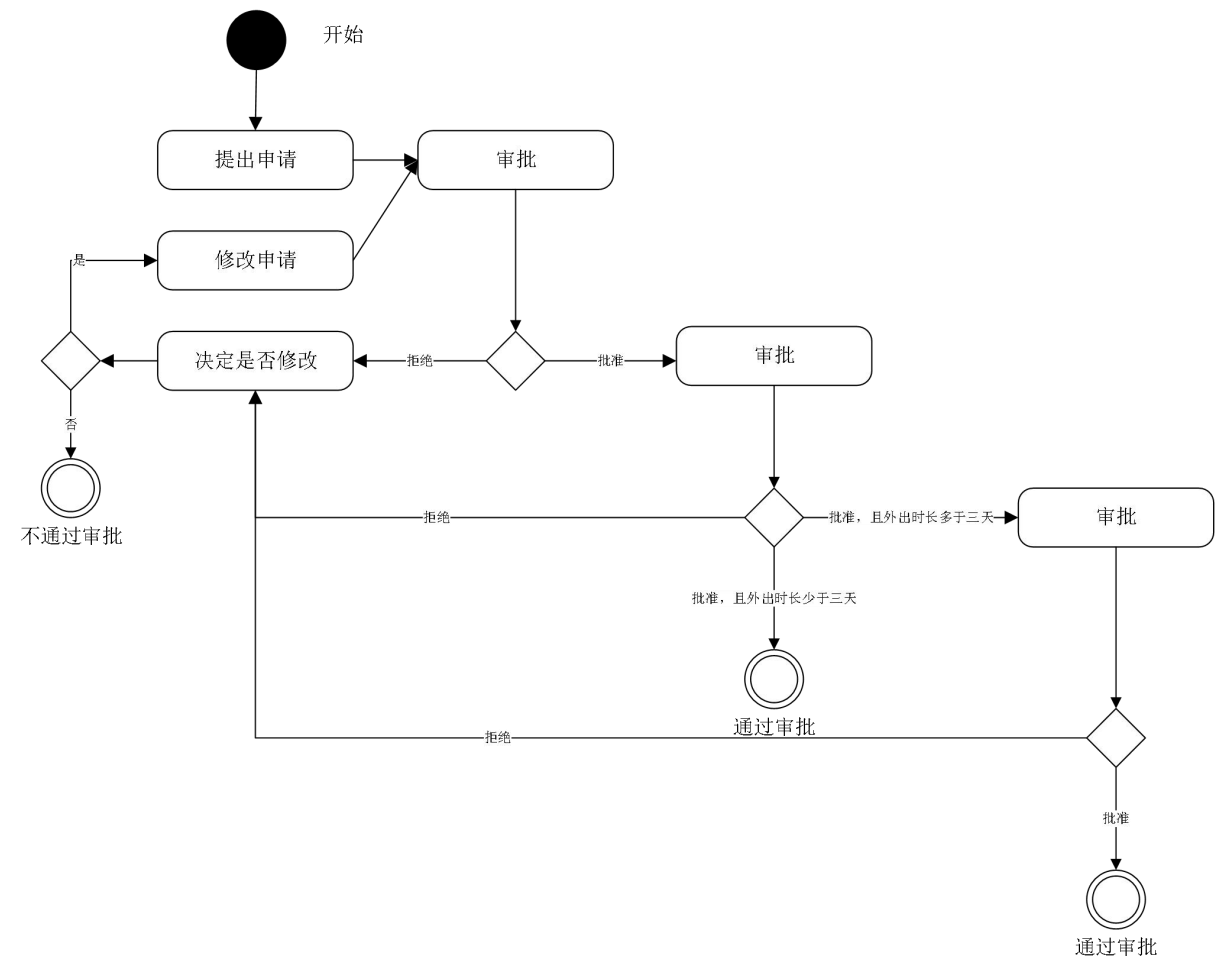
编号	11.1.1
名称	审批申请
执行者	部门经理
优先级	高
描述	<p>目标:</p> <p>用户能根据请假申请的信息，审批该请假申请。具体要求:</p> <p>1.参见用例“11.1 查看需要审批的申请”。用户可在请假申请列表上直接审批其中一条申请。或在查看某一个具体的申请时，审批该申请。</p> <p>2.审批时需选择批准或拒绝，同时可填入审批意见。</p> <p>3.审批时间不需要用户输入，由系统自动确定</p>
前置条件	无
结束状况	系统保存了该申请的审批信息，如果请假申请被批准，则该申请变为“部门经理已审批”，如果是拒绝，则状态为“已拒绝”

编号	11.2
名称	查看以往的审批
执行者	部门经理
优先级	低
描述	<p>目标:</p> <p>用户可方便地查看他曾经审批过的请假申请，了解请假申请的后续审批情况具体要求:</p> <p>1.系统按照请假申请提出时间的倒序，列出用户曾经审批过的请假申请列表。</p> <p>2.请假申请列表需显示:申请者姓名、所属部门、请假类别、请假起止时间、请假事由、请假申请的状态</p>
前置条件	无
结束状况	系统的数据不会发生变化

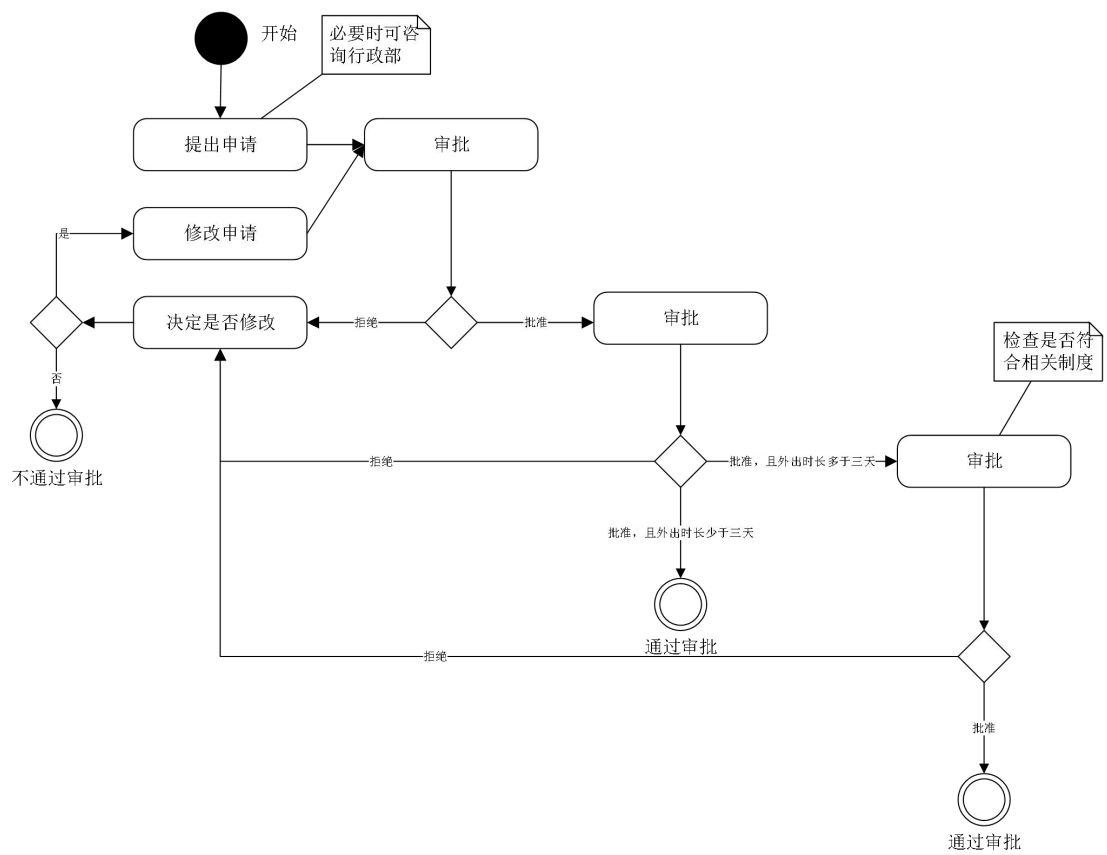
编号	13.1
名称	查看需要审批的申请
执行者	副总经理
优先级	高
描述	1.需副总经理审批的是状态为“部门经理已审批”的请假申请。 2.请假申请列表还需要显示部门经理的审批意见。 3.查看某个具体的申请时，还需显示部门经理的审批意见
前置条件	无
结束状况	系统的数据不会发生变化

4. 活动图

4.1 员工外出活动图

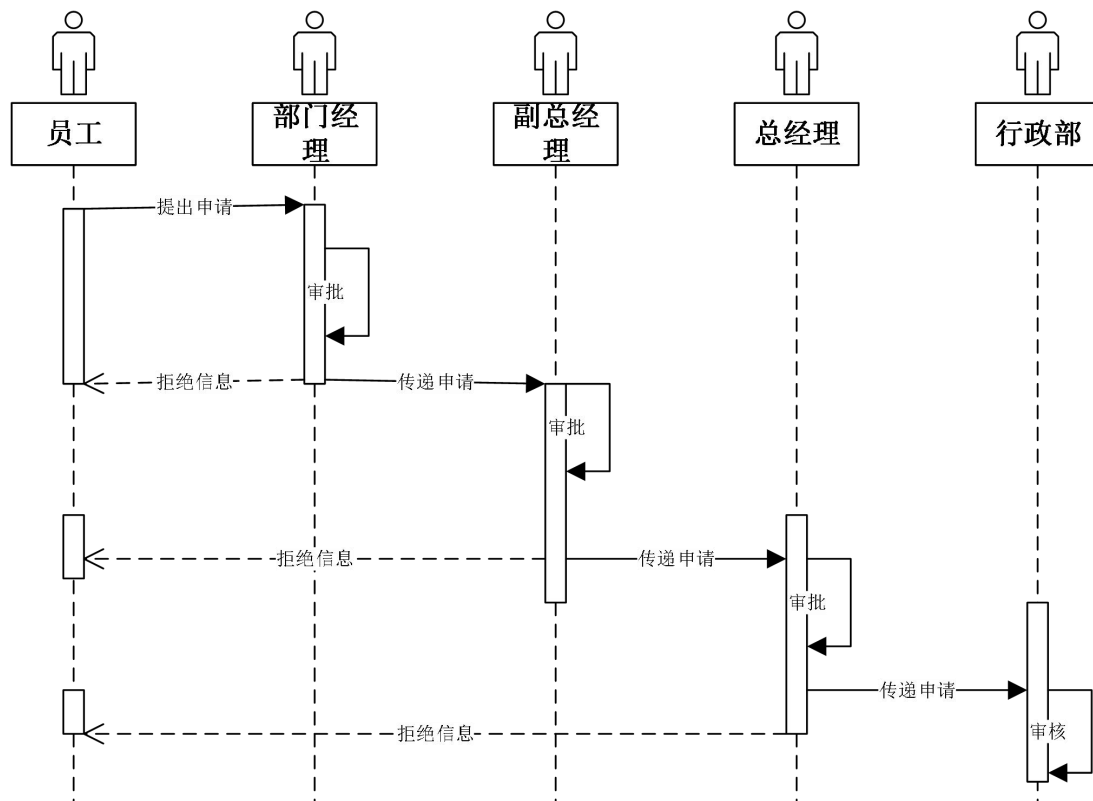


4.2 员工请假活动图

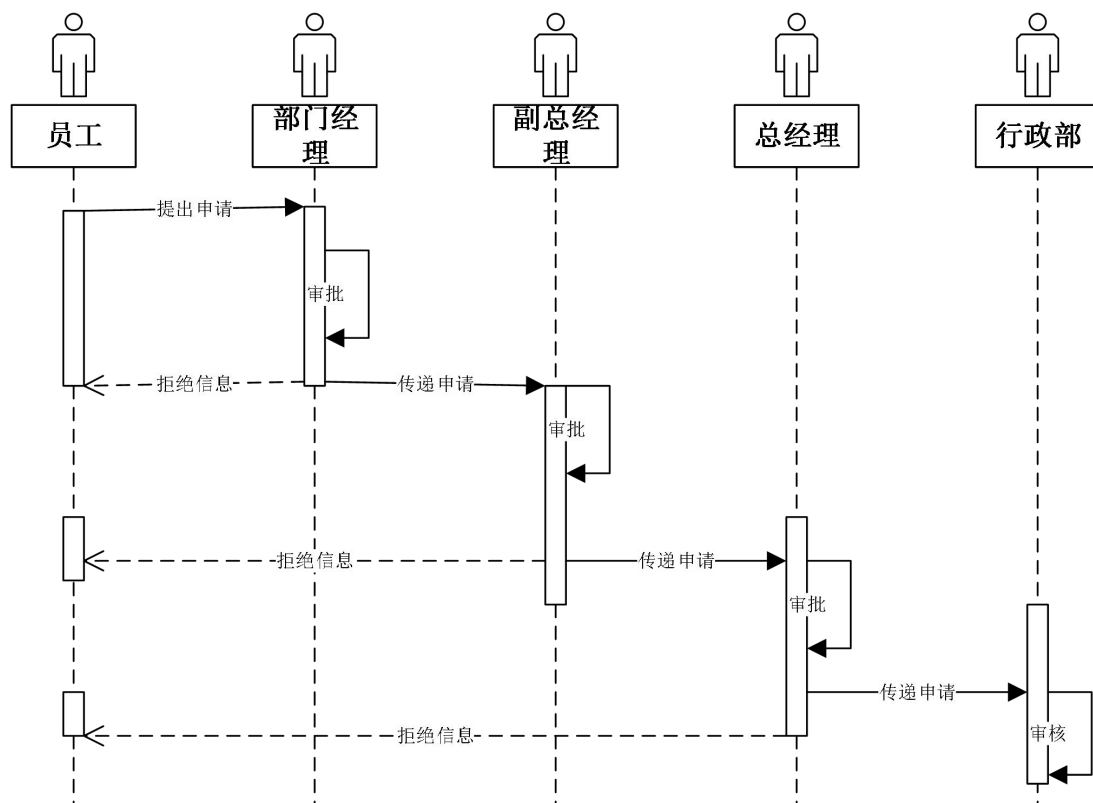


5. 时序图

5.1 员工外出时序图

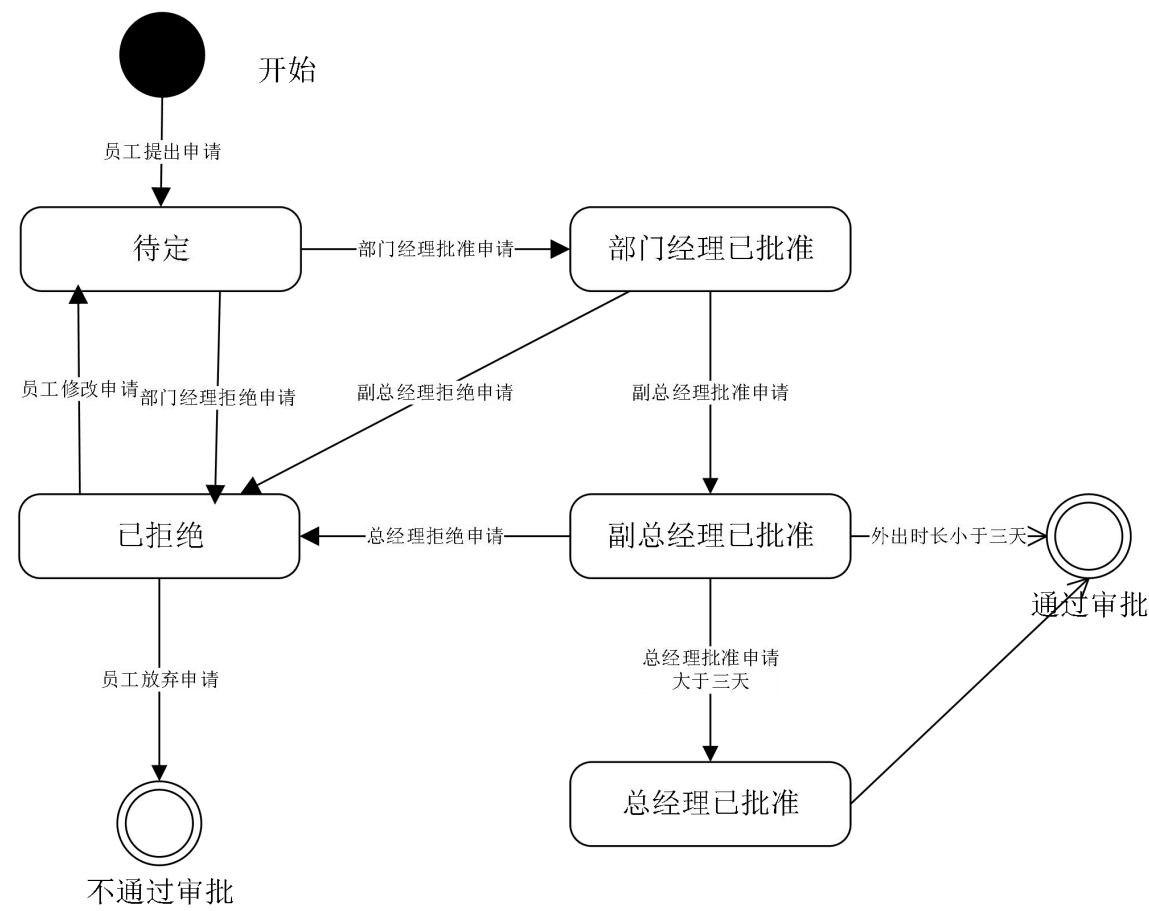


5.2 员工请假时序图

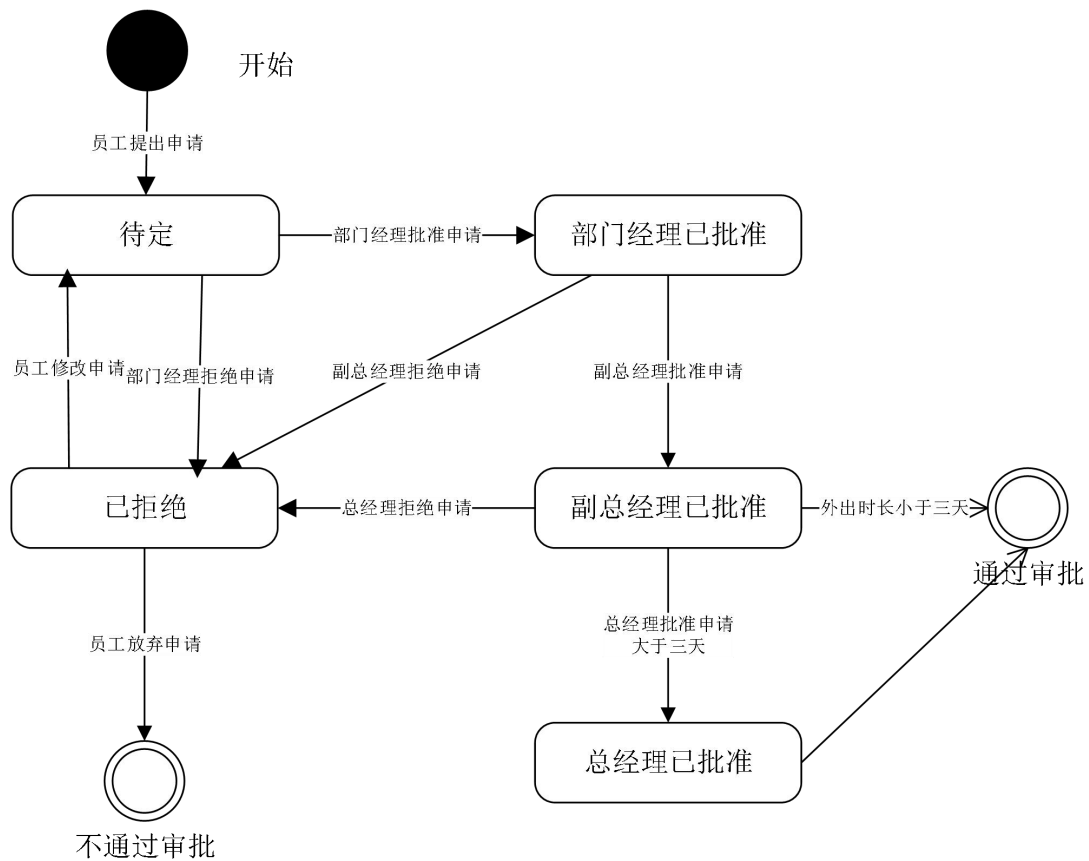


6. 状态机图

6.1 员工外出状态机图



6.2 员工请假状态机图



五、数据库设计

1. 数据库的需求分析

员工考勤管理系统需要提供数据信息的查询、添加、修改以及删除等功能。从而要求数据库能完全的满足各类信息实体的相关数据要求,通过对上述系统功能的分析,根据员工考勤管理系统的功能需求总结出如下数据项:

(1) 用户信息: 主键、用户名、密码、员工编号、姓名、性别、部门、用户类型。

(2) 部门信息: 主键、部门名称、部门描述

(3) 员工签到信息: 主键、员工、签到时间、签到日期、上午/下午、签到类型

(4) 员工补签信息: 主键、员工、补签日期、上午/下午、补签原因、审核结果

(5) 员工请假信息: 主键、员工、请假日期、起始日期、起始上午/下午、截止日期、截止上午/下午、请假类型、请假原因、审核结果

(6) 员工离岗信息: 主键、员工、离岗日期、起始日期、起始上午/下午、截

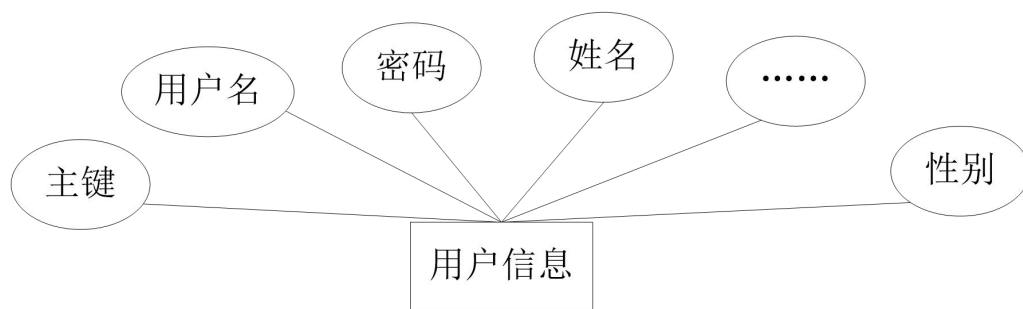
止日期、截止上午/下午、离岗事由、审核结果

(7) 考勤时间信息：主键、上午上班时间、下午上班时间

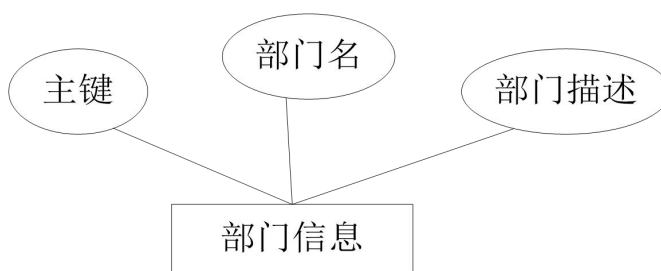
(8) 节假日信息：主键、节假日、节假日说明

2.数据库结构设计

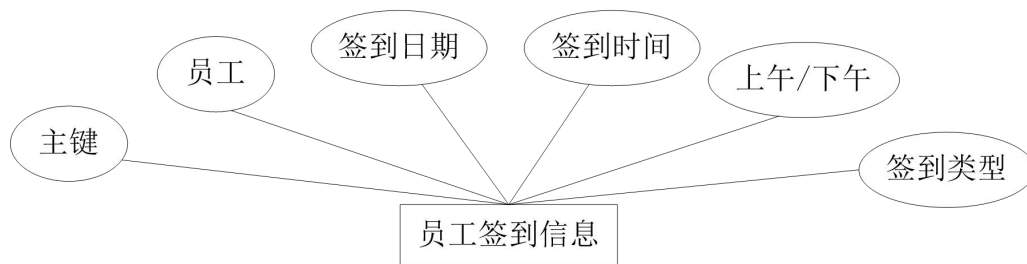
数据库结构设计一般采用实体-联系方法,即 E-R 图。根据数据库需求分析规划出的需要的实体结构,并且确立实体之间的关系。本系统的实体有:用户信息实体、部门信息实体、员工签到信息实体、员工补签信息实体、员工请假信息实体、员工离岗信息实体、考勤时间信息实体和节假日信息实体等八个实体信息。各个实体具体的实体属性图如下:



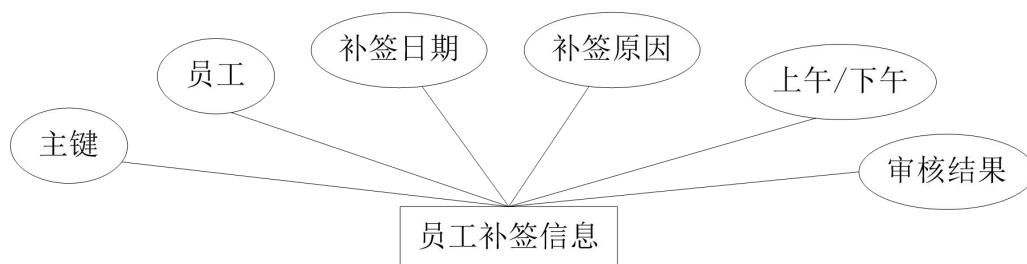
用户信息实体属性



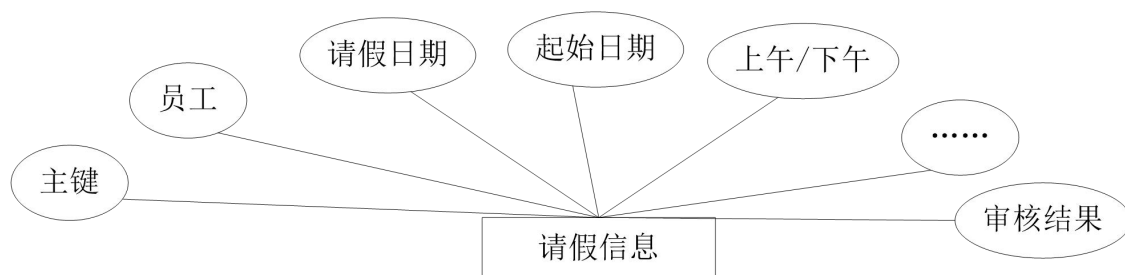
部门信息实体属性



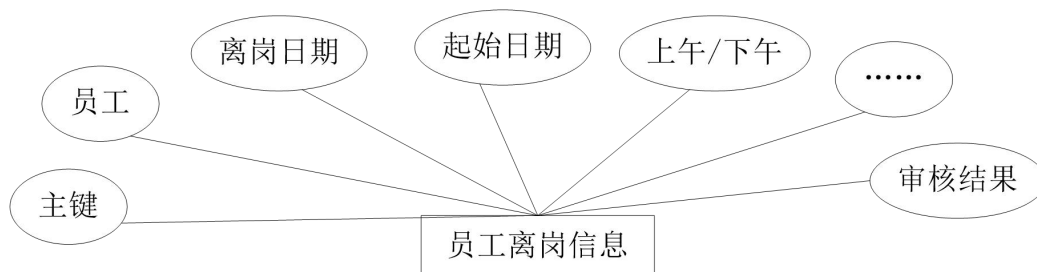
员工签到信息实体属性



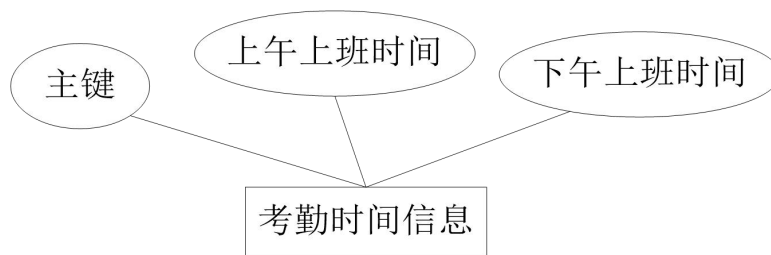
员工补签信息实体属性



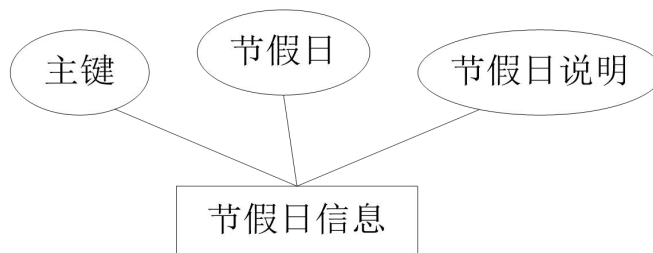
请假信息实体属性



员工离岗信息实体属性

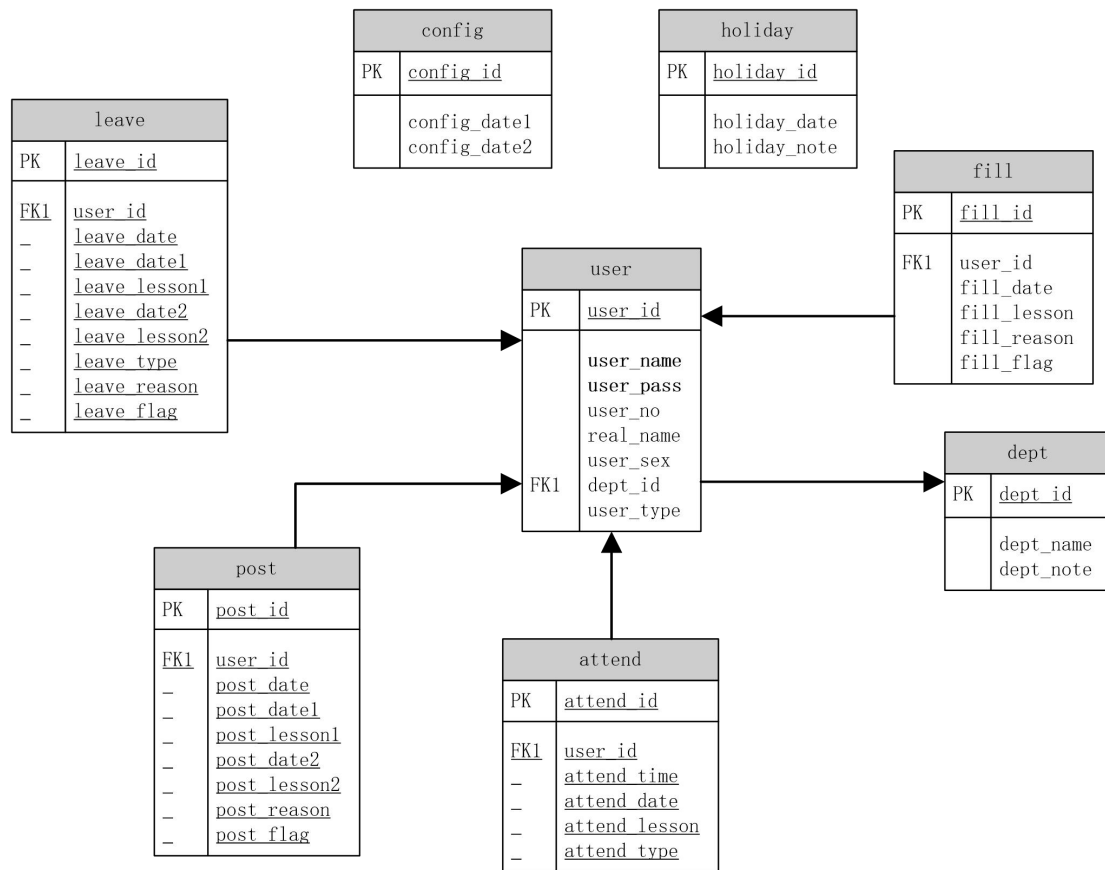


考勤时间信息实体属性



节假日信息实体属性

根据各个实体间所包含的字段以及实体间的关联画出 E-R 图如下：



3.数据库逻辑结构设计

根据上述对数据库的需求分析和结构分析，我们可以列出数据库的各个表的详细逻辑结构设计。员工考勤管理系统网站中共设计了 8 个数据库表，每个表的所包含的字段类型、含义等信息如下面各表所示：

3.1 用户表（user）

字段名称	字段表示	字段类型(长度)	主外键	约束	备注
主键	id	int(11)	PK	NOT NULL	
用户名	name	varchar(50)		NOT NULL	
密码	pass	varchar(200)		NOT NULL	
员工编号	no	varchar(50)		NOT NULL	
姓名	real_name	varchar(50)			
性别	sex	int			1: 男 2: 女
部门	dept_id	int			
用户类型	type	int		NOT NULL	1:员工 2:管理员

3.2 部门表（depart）

字段名称	字段表示	字段类型(长度)	主外键	约束	备注
主键	depart_id	int(11)	PK	NOT NULL	
部门名称	depart_name	varchar(50)		NOT NULL	
部门描述	depart_note	varchar(300)			

3.3 员工签到表（attend）

字段名称	字段表示	字段类型(长度)	主外键	约束	备注
主键	attend_id	int(11)	PK	NOT NULL	
员工	user_id	int(11)		NOT NULL	
签到时间	attend_time	datetime			
签到日期	attend_date	date		NOT NULL	
上午/下午	attend_lesson	int		NOT NULL	1:上午 2:下午
签到类型	attend_type	int		NOT NULL	1:未签到 2:已签到 3:迟签到 4:请假 5:离岗 6:早退

3.4 员工补签表（fill）

字段名称	字段表示	字段类型(长度)	主外键	约束	备注
主键	fill_id	int(11)	PK	NOT NULL	
员工	user_id	int(11)		NOT NULL	
补签日期	fill_date	date		NOT NULL	
上午/下午	fill_lesson	int		NOT NULL	1:上午 2:下午
补签原因	fill_reason	varchar(200)		NOT NULL	
审核结果	fill_flag	int		NOT NULL	1:待审核 2:审核通过 3:审核未通过

3.5 员工请假表 (leave)

字段名称	字段表示	字段类型(长度)	主外键	约束	备注
主键	leave_id	int(11)	PK	NOT NULL	
员工	user_id	int(11)		NOT NULL	
请假日期	leave_date	date		NOT NULL	
起始日期	leave_date1	date		NOT NULL	
起始上午/下午	leave_lesson1	int		NOT NULL	1:上午 2:下午
截止日期	leave_date2	date		NOT NULL	
截止上午/下午	leave_lesson2	int		NOT NULL	1:上午 2:下午
请假类型	leave_type	int		NOT NULL	1:年假 2:病假 3:事假
请假原因	leave_reason	varchar(200)			
审核结果	leave_flag	int		NOT NULL	1:待审核 2:审核通过 3:审核未通过

3.6 员工离岗表 (post)

字段名称	字段表示	字段类型(长度)	主外键	约束	备注
主键	leave_id	int(11)	PK	NOT NULL	
员工	user_id	int(11)		NOT NULL	
离岗日期	post_date	date		NOT NULL	
起始日期	post_date1	date		NOT NULL	
起始上午/下午	post_lesson1	int		NOT NULL	1:上午 2:下午
截止日期	post_date2	date		NOT NULL	
截止上午/下午	post_lesson2	int		NOT NULL	1:上午 2:下午

离岗事由	post_reason	varchar(200)			
审核结果	post_flag	int		NOT NULL	1:待审核 2:审核通过 3:审核未通过

3.7 考勤时间表 (config)

字段名称	字段表示	字段类型(长度)	主外键	约束	备注
主键	config_id	int(11)	PK	NOT NULL	
上午上班时间	config_date1	varchar(50)		NOT NULL	
下午上班时间	config_date2	varchar(50)		NOT NULL	

3.8 节假日表 (holiday)

字段名称	字段表示	字段类型(长度)	主外键	约束	备注
主键	holiday_id	int(11)	PK	NOT NULL	
节假日	holiday_date	varchar(50)		NOT NULL	
节假日说明	holiday_note	varchar(300)			

六、设计模式分析

1. MyBatis-Plus

简称 MP，是一个 MyBatis 的增强工具，在 MyBatis 的基础上只做增强不做修改，为简化开发、提高效率而生。

2. Spring 框架

Spring 框架是 Java 平台上的一种开源应用框架，提供具有控制反转特性的容器。Spring 框架为开发提供了一系列的解决方案，比如利用控制反转的核心特性，并通过依赖注入实现控制反转来实现管理对象生命周期容器化，利用面向切面编程进行声明式的事务管理，整合多种持久化技术管理数据访问，提供大量优秀的 Web 框架方便开发等。

3. 设计模式分析

3.1 构建者模式

Mybatis 的初始化工作非常复杂，不是只用一个构造函数就能搞定的。所以使用了建造者模式，使用了大量的 Builder，进行分层构造，核心对象 Configuration 使用了 XmlConfigBuilder 来进行构造。

在 Mybatis 环境的初始化过程中，SqlSessionFactoryBuilde 会调用 XMLConfigBuilder 读取所有的 MybatisMapConfig.xml 和所有的 *Mapper.xml 文件，构建 Mybatis 运行的核心对象 Configuration 对象，然后将该 Configuration 对象作为参数构建一个 SqlSessionFactory 对象。

其中 XMLConfigBuilder 在构建 Configuration 对象时，也会调用 XMLMapperBuilder 用于读取 *Mapper 文件，而 XMLMapperBuilder 会使用 XMLStatementBuilder 来读取和 build 所有的 SQL 语句。

在这个过程中，有一个相似的特点，就是这些 Builder 会读取文件或者配置，然后做大量的 XPathParser 解析、配置或语法的解析、反射生成对象、存入结果缓存等步骤，这么多的工作都不是一个构造函数所能包括的，因此大量采用了 Builder 模式来解决。

3.2 工厂模式

Mybatis 中执行 Sql 语句、获取 Mappers、管理事务的核心接口 SqlSession 的创建过程使用到了工厂模式。有一个 SqlSessionFactory 来负责 SqlSession 的创建。该 Factory 的 openSession() 方法重载了很多个，分别支持 autoCommit、Executor、Transaction 等参数的输入，来构建核心的 SqlSession 对象。

3.3 代理模式

代理模式可以认为是 Mybatis 的核心使用的模式，正是由于这个模式，我们只需要编写 Mapper.java 接口，不需要实现，由 Mybatis 后台帮我们完成具体 SQL 的执行。当我们使用 Configuration 的 getMapper 方法，会调用 mapperRegistry.getMapper 方法而该方法又会调用 mapperProxyFactory.newInstance(sqlSession) 来生成一个具体的代理：非常典型的，该 MapperProxy 类实现了 InvocationHandler 接口，并且实现了该接口的 invoke 方法。通过这种方式，我们只需要编写 Mapper.java 接口类，当真正执行一个 Mapper 接口的时候，就会转发给 MapperProxy.invoke 方法，而该方法则会调用后续的 sqlSession.cud>executor.execute>prepareStatement 等一系列方法，完成 SQL 的执行和返回。

3.4 单例模式

Spring 中依赖注入的 Bean 实例默认是单例的。Spring 的依赖注入（包括 lazy-init 方式）都是发生在 AbstractBeanFactory 的 getBean 里。getBean 的 doGetBean 方法调用 getSingleton 进行 bean 的创建。spring 中的单例模式完成了后半句话，即提供了全局的访问点 BeanFactory。但没有从构造器级别去控制单例，这是因为 spring 管理的是任意的 java 对象。

3.5 适配器模式

SpringMVC 中的适配器 HandlerAdapter 根据 Handler 规则执行不同的 Handler。DispatcherServlet 根据 HandlerMapping 返回的 handler，向 HandlerAdapter 发起请求，处理 Handler。HandlerAdapter 根据规则找到对应的 Handler 并让其执行，执行完毕后 Handler 会向 HandlerAdapter 返回一个 ModelAndView，最后由 HandlerAdapter 向 DispatcherServlet 返回一个 ModelAndView。

3.6 装饰器模式

Spring 中用到的包装器模式在类名上有两种表现：一种是类名中含有 Wrapper，另一种是类名中含有 Decorator。动态地给一个对象添加一些额外的职责。就增加功能来说，Decorator 模式比生成子类更为灵活。

3.7 观察者模式

spring 的事件驱动模型使用的是观察者模式，Spring 中 Observer 模式常用的地方是 listener 的实现。事件机制的实现需要三个部分：事件源、事件、事件监听器。继承自 jdk 的 EventObject，所有的事件都需要继承 ApplicationEvent，并且通过构造器参数 source 得到事件源。该类的实现类 ApplicationContextEvent

表示 `ApplicaitonContext` 的容器事件。