****

**考勤系统**



**学 院 智能与计算学部**

**专 业 软件工程**

**年 级 2018**

**组 长 张启洲**

**组 员 张春胜**

**组 员 谢梦鑫**

**指导教师 李罡**

**2020年11月 1 日**

# 1 需求分析

## 1.1 引言

### 1.1.1 编写目的

该需求规格说明书首先给出项目的整体结构和功能结构的概貌，从总体架构上给出我们考勤系统的整体轮廓。让读者在看过我们这篇文档后能对考勤系统有一个初步但生动的认识，能够对我们的系统有基本的了解。

其次，对功能需求和性能需求进行了详细的描述。便于用户、开发人员进行理解和交流，反应出用户问题的结构，可以作为考勤系统开发工作的基础和依据，以及可以作为后期确认测试和验收的依据。该需求规格说明书编制目的是明确本系统的详细需求，供用户确认系统的功能和性能，和用户形成一致 的理解和确认，作为进一步详细设计软件的基础。

### 1.1.2 项目背景

随着我国市场经济的快速发展，职工考勤管理系统在企业的日常管理中发挥着越来越重要的作用。职工考勤系统可以进行员工信息管理、出勤管理请假申请等，方便处理企业内部员工的相关考勤信息。打破了时间和地域的界限，使信息交流变得快捷、准确，为建立现代企业管理信息系统提供了充足的条件。

考勤与请假系统是在适应市场需求的客观前提下，为了满足中小型公司或事业单位管理自己的员工而开发的。

### 1.1.3 作用范围

作用范围所涉及的读者为

（1）一般员工：一般员工，请假要给部门经理审批；

（2）部门经理：高级员工，审批一般员工三天内的假期申请；

（3）总经理：总经理，审批三日以上的假期申请；

（4）人事科人员：维护全体员工的基本信息。

### 1.1.4 定义

#### 1.1.4.1 数据字典中的符号含义

|  |  |  |
| --- | --- | --- |
| **符号** | **含义** | **样例及说明** |
| = | 被定义为 | X=a 表示X由a组成 |
| + | 与 | X=a+b 表示X由a和b组成 |
| [... | ...] | 或 | X=[a|b] 表示X由a或b组成 |
| { ... } | 重复 | X={a} 表示X由0个或多个a组成 |
| m{ ... }n | 重复 | X=3{a}8 表示X中a至少出现3次，最多出现8次 |
| ( ... ) | 可选 | X=(a) 表示a可以在X中出现，也可以不出现 |
| “...” | 基本数据元素 | X="a" 表示X是取值为字符“a”的数据元素 |
| ·· | 连接符 | X=1··8 表示X可取1到8中的任意一个值 |

#### 1.1.4.2 专有名词定义

|  |  |
| --- | --- |
| **名词** | **所指对象或含义** |
| JavaScript | 一种高级的，解释执行的编程语言。主要在web浏览器(客户端)解释执行 |
| PHP | PHP是开源的通用计算机脚本语言，尤其适用于网络开发并可嵌入 HTML 中使用。可用于连接前台与数据库的语言，处理表单 |
| XML | 可扩展标记语言, 是一种简单的数据存储语言，使用一系列简单的标记描述数据 |
| MySQL | 数据库系统 |
| HTML | 超文本标记语言，是一种用于创建网页的标准标记语言 |
| CSS | 层叠样式表，用来为结构化文档（如 HTML 文档或 XML应用）添加样式的计算机语言 |
| JQuery | jQuery 是一套跨浏览器的 JavaScript 库，简化 HTML 与JavaScript 之间的操作 |
| Angular | AngularJS 是一款由 Google 维护的开源 JavaScript 库，用来协助单一页面应用程序运行 |
| SPA | 单页 Web 应用，就是只有一张 Web 页面的应用，是加载单个 HTML 页面并在用户与应用程序交互时动态更新该页面的 Web 应用程序。 |
| Browser/Server体系 | B/S 结构的软件是相对传统的C/S结构而言的。C/S结构应该具有这样的特征:客户端、服务器端都安装相应的软件，客户端、服务器端各完成相应的计算工作，服务器端保存数据库。 B/S 结构应该具有如下特征：服务器端都安装相应的软件，客户端不安装任何软件(不需要维护)。客户端运行程序是靠浏览器软件(如 IE ,Netscape 等)登陆服务器进行的。客户端在浏览器里完成一定的计算任务。 |

## 1.2 任务概述

### 1.2.1 目标

考勤与请假系统为企业提供出勤、请假管理。支持请假申请、出勤统计；人事科人员能够对系统进行全面管理，包括计算工龄年龄、核对请假材料等等。

### 1.2.2 运行环境

1）硬件环境

服务器

操作系统： Linux 操作系统

处理器： Intel E5 及以上处理器

内存： 16GB 以上

硬盘： 1T 以上

2）软件环境

数据库： MySQL 2017

服务器： Tomcat

开发工具： Microsoft Visual Studio Code

项目管理工具： Teamviewer

协同开发工具： TIM

代码版本管理工具： GIT

绘图工具： ProcessOn

### 1.2.3 条件与限制

1. 项目开发过程中应保证团队的软件及硬件资源需求不间断，必须存在可使用的编程以及测试软硬件，并且保护数据的保存可靠性。
2. 项目完成到正式投入使用前必须保证项目公布的必要条件，如：运行服务器，网站域名，管理人员等。
3. 项目正式投入使用后保证服务器的正常运转，管理者与程序员的不间断维护与管理。
4. 开发限期：2020年12月6日前完成

#### 1.2.3.1 编程语言限制

网页端采用Vue前端设计+后端Springboot进行程序设计。

#### 1.2.3.2 工具约束

采用IntelliJ IDEA、VS code开发

#### 1.2.3.3 性能约束

保证网络连接的稳定性，降低断网情况发生的概率。

对外界平台提供的信息能够保证按照要求及时更新。

能在用户提出登陆/退出请求后快速响应，保证3s内实现信息读写完成。

对用户使用的平台功能保证查询信息的完整记录不缺失，查询保证在2s内完成响应。

保证用户隐私信息的安全。

### 1.2.4 涉众分析

表1.1 涉众分析表

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **涉众** | **代表人物** | **待解决的问题/对系统的期望** |
| 1 | 一般员工 | 张豪 | 能够在系统中注册登录、能发起请假申请、查看请假是否被批准、查看自己的剩余假期。 |
| 2 | 部门经理 | 李四 | 能够在系统中注册登录、能发起请假申请、查看请假是否被批准、查看自己的剩余假期、**审批一般员工的请假申请、生成任意时间范围内某人或本部门全体成员的请假记录。** |
| 3 | 总经理 | 春春 | 能够在系统中注册登录、查看自己的剩余假期、**审批部门经理审批通过的请假申请、生成任意时间范围内某人或全体成员的请假记录** |
| 4 | 人事科人员 | 陈哥 | 能够在系统中注册登录、能发起请假申请、查看请假是否被批准、查看自己的剩余假期、**维护全体员工的基本信息、核对材料、每月底得到报表**。 |

### 1.2.5 假设与前提条件

数据输入过程不会出错，数据正确传输。使用人员掌握基本电脑操作。

## 1.3 数据需求

### 1.3.1 静态数据

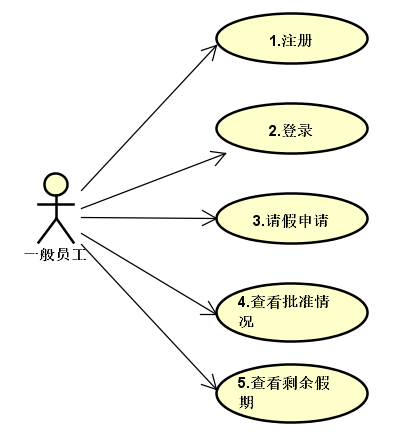
|  |  |  |
| --- | --- | --- |
| 名称 | 简述 | 有关数据流 |
| 一般员工 | 系统基本用户，无特殊权利 | 登录系统请求、请假申请请求、查看假期批准情况请求、查看剩余假期请求 |
| 部门经理 | 一个部门的经理 | 登录系统请求、请假申请请求、查看假期批准情况请求、查看剩余假期请求、审核一般员工的请假申请需求、生成请假记录请求 |
| 总经理 | 企业的总经理 | 登录系统请求、查看剩余假期请求、审批请假请求、生成请假记录请求 |
| 人事科人员 | 负责人事科的员工 | 登录系统请求、请假申请请求、查看假期批准情况请求、查看剩余假期请求、修改员工信息请求、报表生成请求、材料审批请求 |

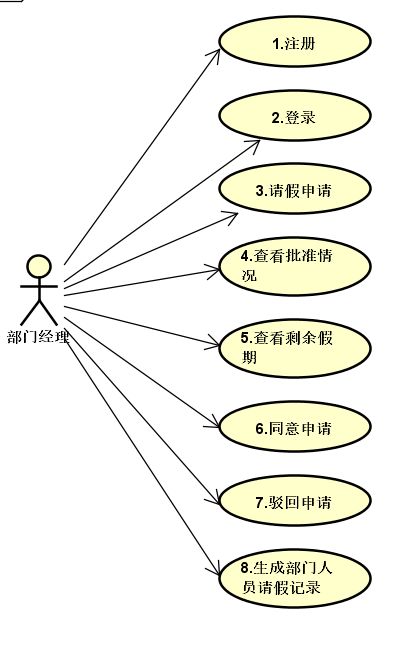
### 1.3.2 动态数据

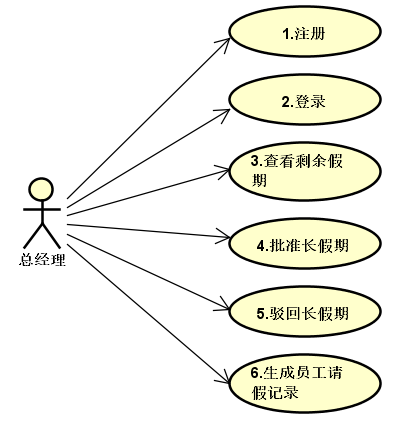
|  |  |  |  |
| --- | --- | --- | --- |
| 数据流名称 | 简述 | 来源 | 去向 |
| 登录系统请求 | 用户使用账号和密码进行系统登录 | 用户 | 用户、服务器 |
| 个人信息 | 系统相应用户登录后的结果 | 用户 | 用户信息数据、服务器 |
| 用户注册 | 新员工注册系统 | 用户 | 服务器 |
| 请假申请 | 员工提出请假申请 | 用户 | 服务器 |
| 假期审批 | 部门经理审批一般员工的假期申请 | 部门经理 | 服务器 |
| 长假审批 | 总经理审批部门经理通过的长假期 | 总经理 | 服务器 |
| 查看剩余假期 | 员工查看自己剩余假期 | 查询指令 | 用户 |
| 查看批准情况 | 员工查看已经申请的假期的审核情况 | 查询指令 | 用户 |
| 修改员工信息 | 人事科人员维护员工基本信息 | 人事科人员 | 用户 |

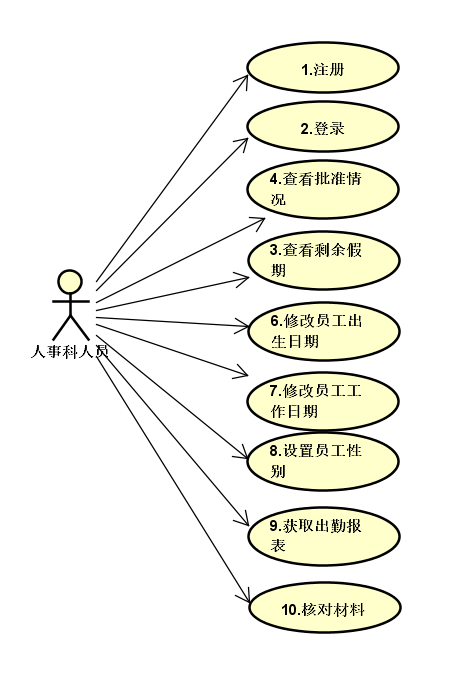
## 1.4 功能需求

### 1.4.1用例图



****

****

****

### 用例规约

1.4.2.1.一般员工

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 1 | 名称 | 注册 |
| 执行者 | 全体用户 | 优先级 | 高■ 低□ |
| 描述 | 新员工可以进行注册  具体要求：用户信息成功录入到数据库 | | |
| 前置条件 | 用户本身没进行注册 | | |
| 结束状况 | 注册成功 | | |
| 异常流程 | 如果已经注册，则注册失败 | | |
| 说明 | 新员工用来注册公司考勤系统的功能 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 2 | 名称 | 登录 |
| 执行者 | 全体用户 | 优先级 | 高■ 低□ |
| 描述 | 目标：已经完成注册的用户可以登录系统  具体要求：   1. 用户已经注册 2. 账号密码正确 | | |
| 前置条件 | 用户已经注册 | | |
| 基本流程 | 输入账号密码，点击登录 | | |
| 结束状况 | 登录成功后，跳转到系统初始界面 | | |
| 异常流程 | 账号密码错误、用户未注册，则登录失败 | | |
| 说明 | 基本登录操作 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 3 | 名称 | 请假申请 |
| 执行者 | 一般员工、部门经理、人事科人员 | 优先级 | 高■ 低□ |
| 描述 | 员工请假的请求  具体要求：  1. 选择请假天数、类型  2. 用户可以看到假期分类列表 | | |
| 前置条件 | 无 | | |
| 结束状况 | 等待审批 | | |
| 异常流程 | 请假天数不满足要求，系统自动驳回 | | |
| 说明 | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 4 | 名称 | 查看假期批准情况 |
| 执行者 | 用户 | 优先级 | 高■ 低□ |
| 描述 | 用户可以查看已经申请的假期的批准情况 | | |
| 前置条件 | 该用户已经提交了假期申请 | | |
| 结束状况 | 显示该请求的审批情况 | | |
| 异常流程 | 如不存在任何申请的假期，则提示 | | |
| 说明 | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 5 | 名称 | 查看剩余假期 |
| 执行者 | 用户 | 优先级 | 高■ 低□ |
| 描述 | 用户可以查看自己剩余的假期  显示各种假期的分类、剩余天数 | | |
| 前置条件 | 无 | | |
| 结束状况 | 显示该员工剩余假期的种类和天数 | | |
| 异常流程 | 无 | | |
| 说明 | 无 | | |

**1.4.2.2.部门经理**

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 6 | 名称 | 审核假期 |
| 执行者 | 部门经理 | 优先级 | 高□ 低■ |
| 描述 | 部门经理可以对员工已经申请的假期进行审核   1. 可以同意可以不同意 2. 不同意要说明理由 | | |
| 前置条件 | 此部门已经有员工提交了假期申请，材料审查合格 | | |
| 结束状况 | 将该员工的假期申请情况改变 | | |
| 异常流程 | 无 | | |
| 说明 | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 7 | 名称 | 生成部门人员请假记录 |
| 执行者 | 部门经理 | 优先级 | 高□ 低■ |
| 描述 | 部门经理可以生成任意时间范围内某人或本部门全体人员的请假记录   1. 选择生成人选 2. 选择生成时间段 | | |
| 前置条件 | 无 | | |
| 结束状况 | 返回想要生成的请假记录 | | |
| 异常流程 | 无 | | |
| 说明 | 无 | | |

**1.4.2.3.总经理**

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 8 | 名称 | 批准长假期 |
| 执行者 | 总经理 | 优先级 | 高□ 低■ |
| 描述 | 总经理可以对部门经理已经申请通过的长假期进行审批  1.可以同意可以不同意  2.不同意要说明理由 | | |
| 前置条件 | 已存在部门经理审批通过的长假期申请 | | |
| 结束状况 | 1. 长假期申请成功 2. 长假期申请失败 | | |
| 异常流程 | 无 | | |
| 说明 | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 9 | 名称 | 生成员工请假记录 |
| 执行者 | 总经理 | 优先级 | 高□ 低■ |
| 描述 | 总经理可以生成任意时间范围内某人或全体成员的请假记录   1. 选择生成人选 2. 选择生成时间段 | | |
| 前置条件 | 无 | | |
| 结束状况 | 返回想要生成的请假记录 | | |
| 异常流程 | 无 | | |
| 说明 | 无 | | |

**1.4.2.4.人事科人员**

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 10 | 名称 | 修改员工出生日期 |
| 执行者 | 人事科人员 | 优先级 | 高■低□ |
| 描述 | 人事科人员可以维护全体员工的出生日期，来计算年龄 | | |
| 前置条件 | 无 | | |
| 结束状况 | 修改员工年龄 | | |
| 异常流程 | 无 | | |
| 说明 | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 11 | 名称 | 修改员工工作日期 |
| 执行者 | 人事科人员 | 优先级 | 高■低□ |
| 描述 | 人事科人员可以维护全体员工的工作日期，来计算工龄 | | |
| 前置条件 | 无 | | |
| 结束状况 | 修改员工工龄 | | |
| 异常流程 | 无 | | |
| 说明 | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 12 | 名称 | 设置员工性别 |
| 执行者 | 人事科人员 | 优先级 | 高■低□ |
| 描述 | 人事科人员可以核对员工的性别 | | |
| 前置条件 | 无 | | |
| 结束状况 | 可以修改员工性别 | | |
| 异常流程 | 无 | | |
| 说明 | 无 | | |

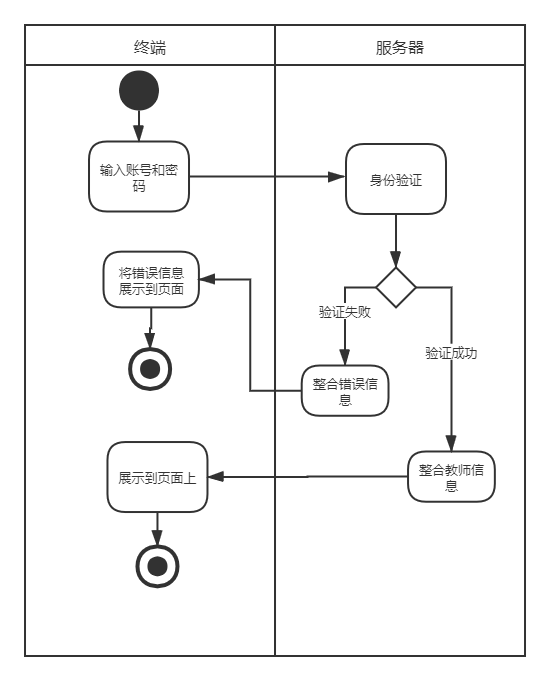
|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 13 | 名称 | 获取出勤报表 |
| 执行者 | 人事科人员 | 优先级 | 高■低□ |
| 描述 | 人事科人员月底可以在每月底得到报表 | | |
| 前置条件 | 无 | | |
| 结束状况 | 得到报表 | | |
| 异常流程 | 无 | | |
| 说明 | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 14 | 名称 | 核对材料 |
| 执行者 | 人事科人员 | 优先级 | 高■低□ |
| 描述 | 人事科人员负责核对请假申请所提交的材料 | | |
| 前置条件 | 员工在申请的假期需要提交材料 | | |
| 结束状况 | 审核完成，选择通过或不通过 | | |
| 异常流程 | 无 | | |
| 说明 | 无 | | |

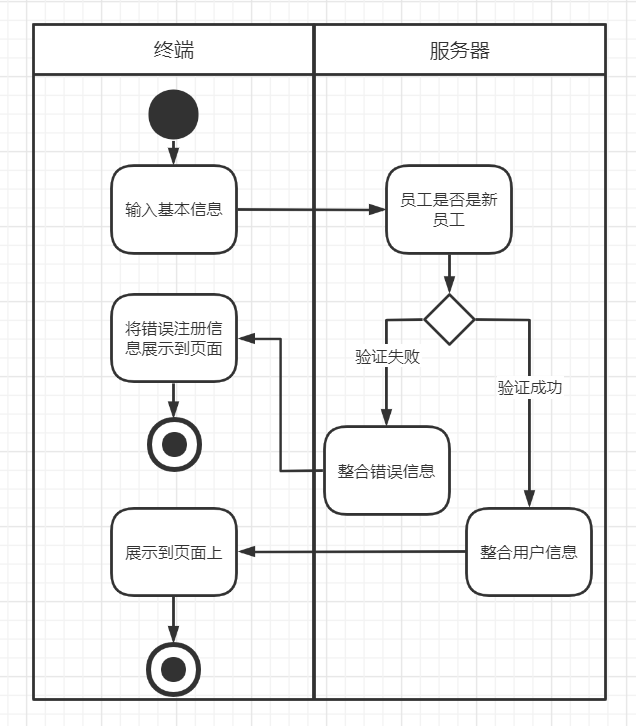
### 1.4.3 活动图

#### 1.4.3.1 一般员工

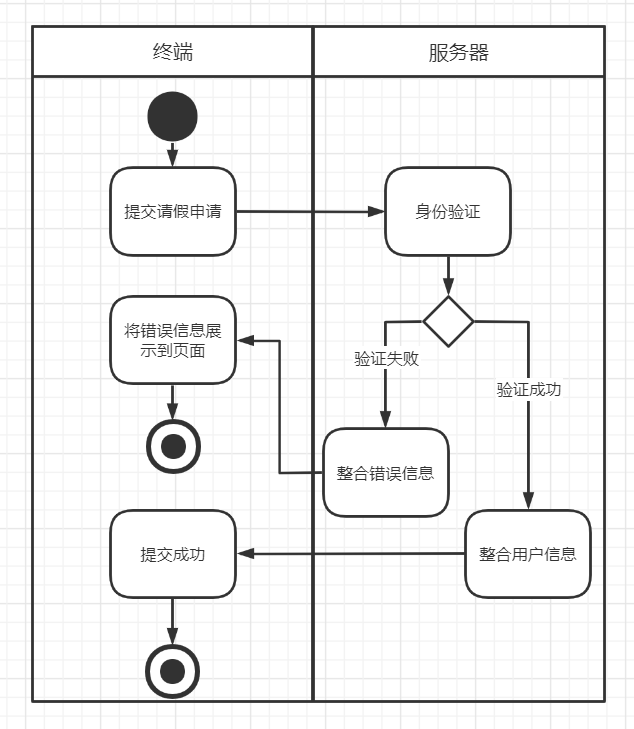
1.1 登录



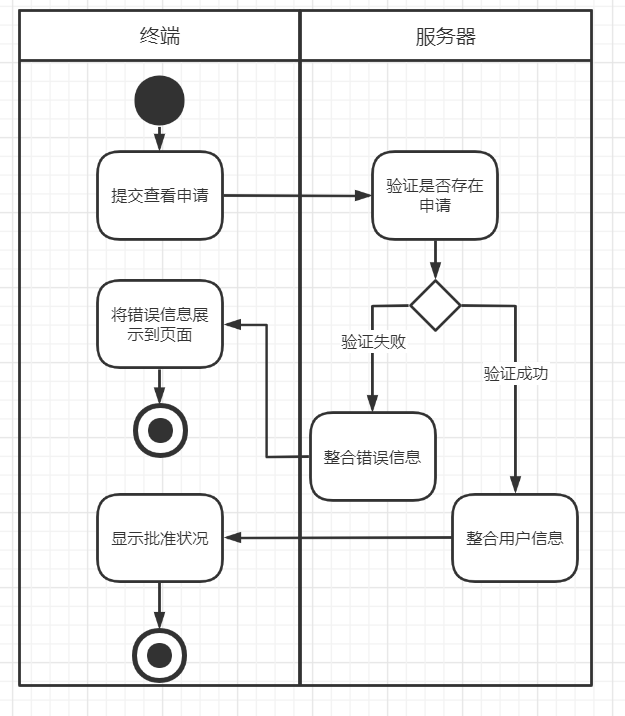
1.2 注册



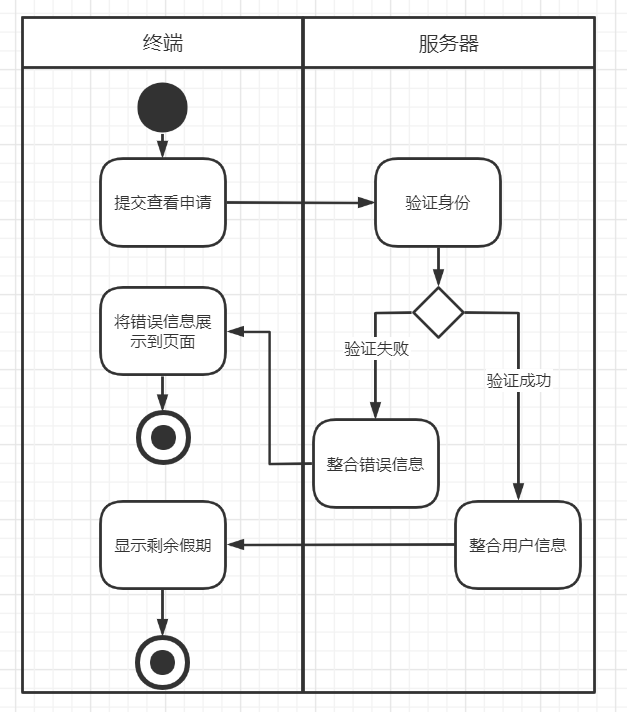
1.3 请假申请



1.4查看批准情况

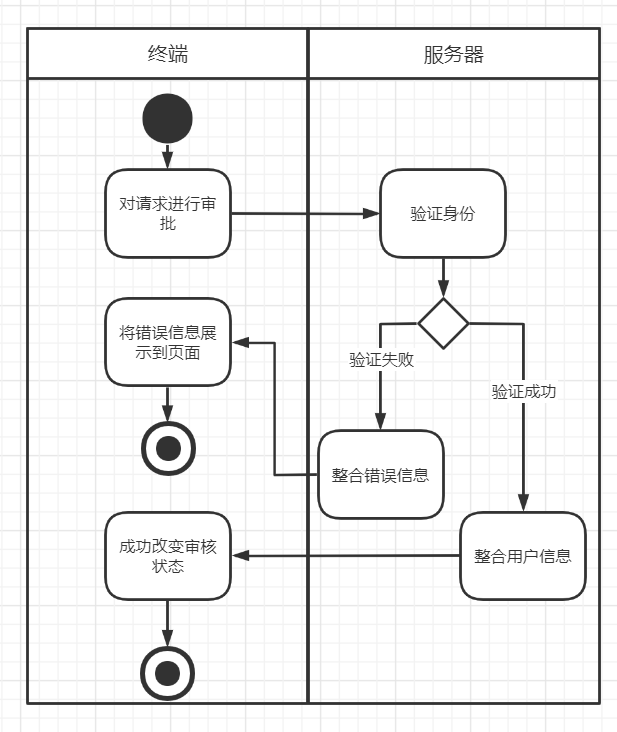


1.5查看剩余假期

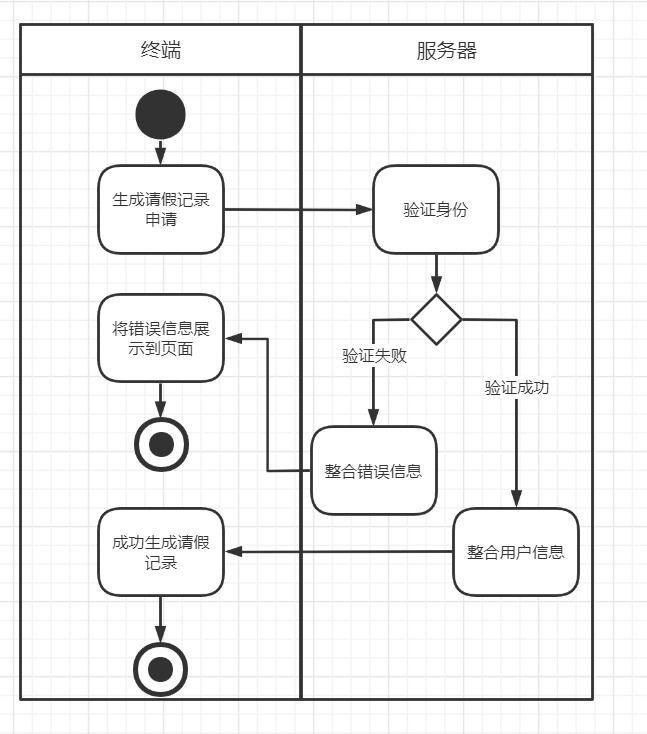


#### 1.4.3.2 部门经理

2.1 审批假期



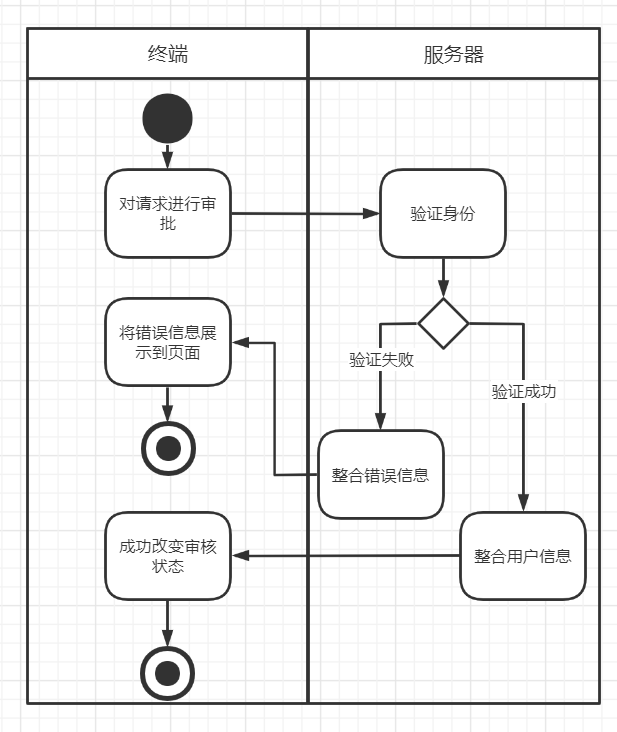
* 1. 生成请假记录



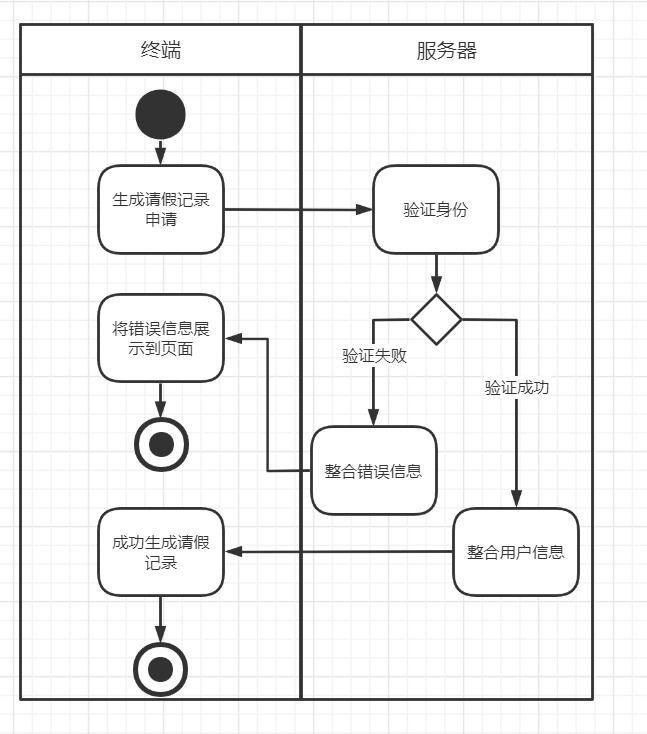
3.1

#### 1.4.3.3 总经理

3.1 审核长假期

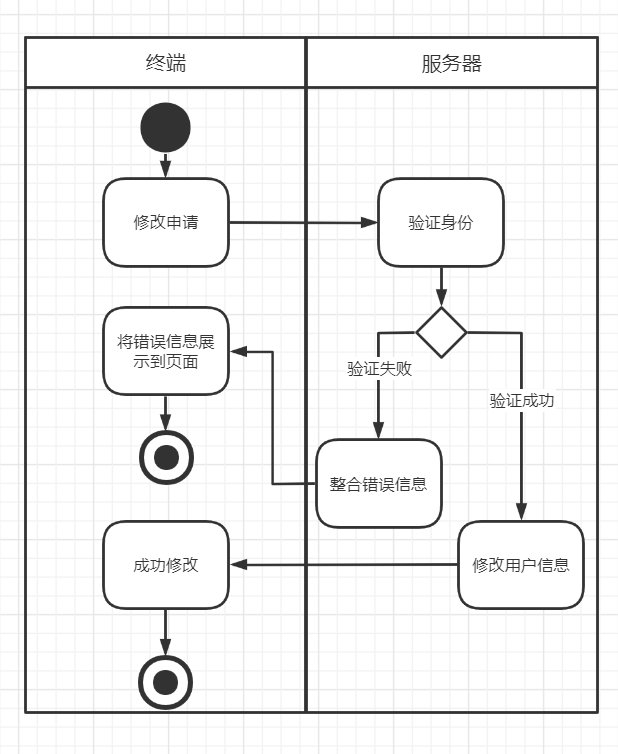


3.2生成员工请假记录

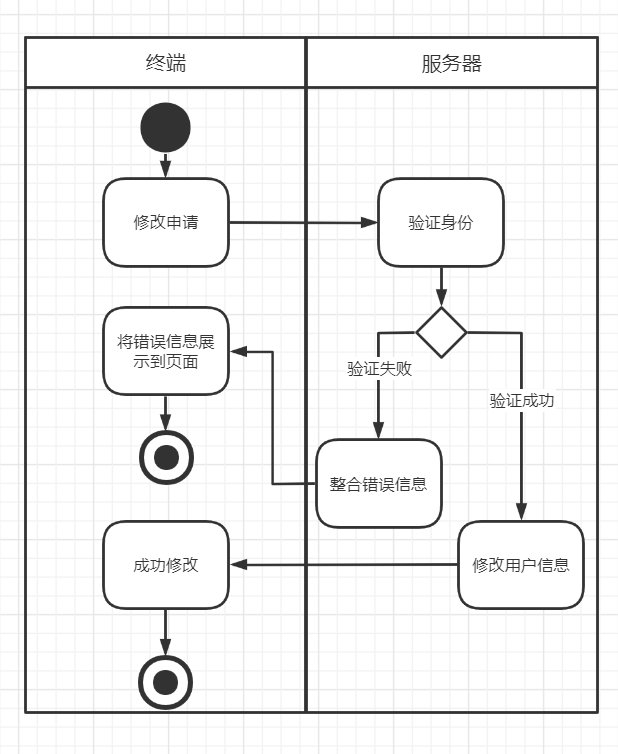


#### 1.4.3.4 人事科人员

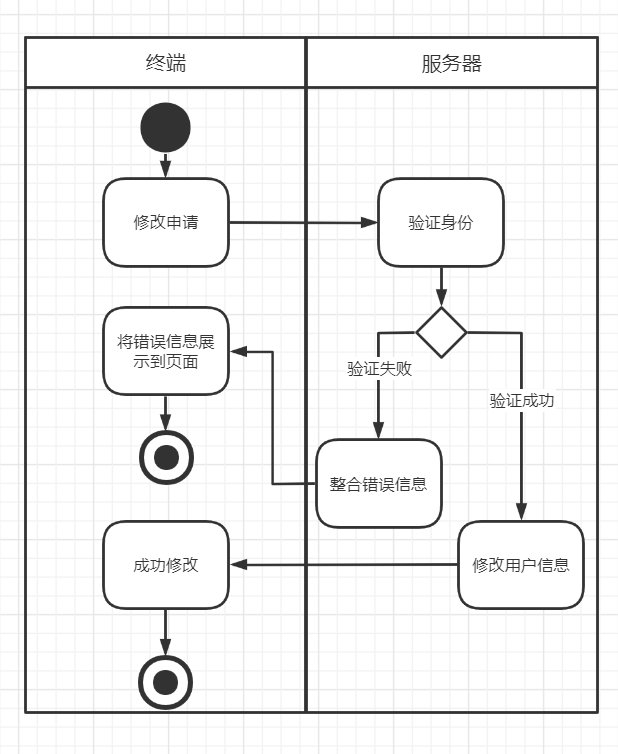
4.1 修改员工出生日期



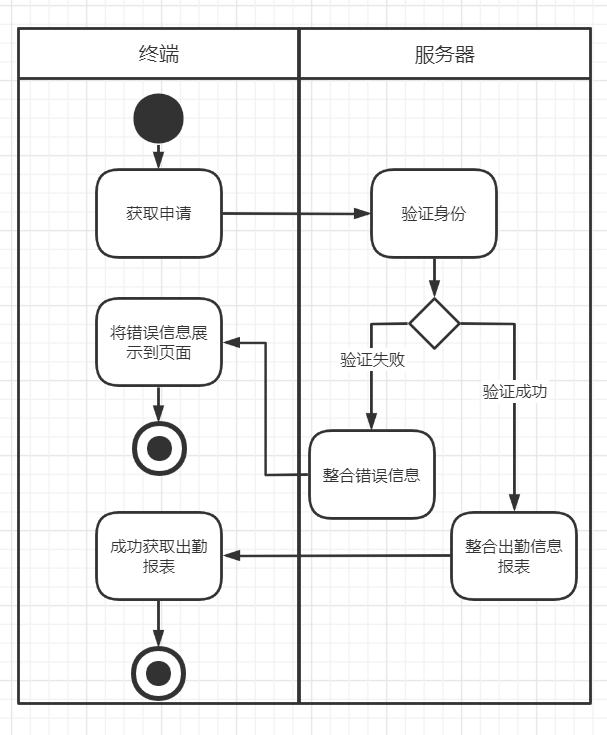
4.2 修改员工工作日期



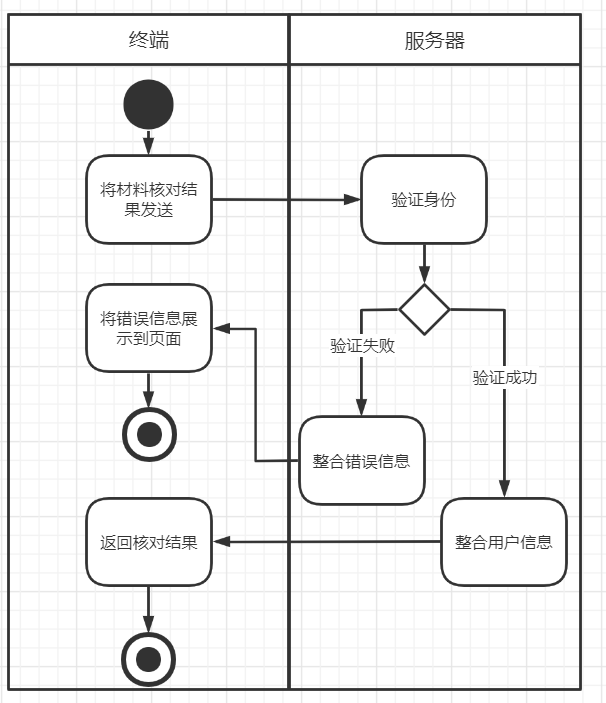
4.3 设置员工性别



4.4 获取出勤报表



4.5 核对材料



## 1.5 性能需求

### 1.5.1 数据精确度

该软件对数据输入输出的精度要求是小数点后3位。

该软件对于数据的传输精度要求是小数点后3位。

### 1.5.2 时间特性

对用户操作的各项功能上，响应时间应保持在1 s内完成。

对于其他的更新处理时间、数据转换与传输时间、运行时间等的要求都要在大多数人能够接受的范围之内。

### 1.5.3 适应性

1. 操作方式上的变化

通过无线网络的网页端登录和使用网线的客户端登录在交互上应具有基本的一致性

1. 运行环境上的变化

保证在windows、macOs等使用相同的交互界面保证跨平台的可用性

1. 同其他软件的接口变化 无
2. 精度和有效时限的变化 有
3. 计划的变化或改进 有

### 1.5.4 系统安全性要求

1. 数据库连接采用 MySQL 方法进行连接，从而提高数据库的可操作性和安全性。
2. 对某些页面进行强制用户登陆后方能进行相关操作，避免用户直接改变地址栏 URL 进行查找。

### 1.5.5容错要求

为了保障系统站能正常运行，尽量使得服务器数据库的数据合法化，以及前台能有效操作和浏览，该系统应做到：

1. 对数据输入或条件输入进行有效检验，禁止非合法数据阻碍客户端与服务器的正常运行，使得数据库也得到进一步安全性的保障。
2. 对连接数据库的错误操作或用户错误操作进行一定化的提示和警告，控制非正常浏览网站

### 1.5.6 其他性能要求

1. 保密性： 系统数据要做好保密设施，以免重要数据的丢失或者盗取。
2. 简洁性： 用户的操作界面必须做到简单明了，不能有太复杂的设置或者操作。
3. 可更新性：对于系统可以根据客户需要进行更新，维护等操作，扩展其功能，并使系统稳定。
4. 可转换性：只要系统能保证环境部署成功，并且网络正常，就能保证系统的正常运行。

## 1.6 运行需求

### 1.6.1 用户界面

界面设计应该简洁易懂，该部分需求应该不断优化，直至符合用户习惯。

### 1.6.2 硬件接口

无

### 1.6.3 软件接口

1）网络接口

软件通过与教务处系统接口进行数据交互，

2)数据接口

系统与后台的考勤信息，分类信息等数据库进行数据交互的接口

### 1.6.4 内部接口

内部接口方面，各模块之间采用函数调用、参数传递、返回值的方式进行信息传

递。具体参数的结构将在下面数据结构设计的内容中说明。接口传递的信息将是

以数据结构封装了的数据，以参数传递或返回值的形式在各模块间传输。

### 1.6.5 故障处理

1）软件在使用过程中所使用的计算机突然断电或者计算机系统突然崩溃

2）软件出现不可预测的BUG

基于上述两点的故障要求软件能够实时保护数据并存储，做到数据不因故障而丢失

3）软件使用过程网络连接突然断开

对于以上问题要做到软件能检测到异常的连接断开并能尝试重连或给出错误信息

## 1.7 其他需求

可使用性：保证界面简洁明确，可操作性强不存在有异议的交互功能。

安全保密：对账户信息进行经过加密后传输与存储，保证用户信息的安全性，在系统层面对用户的账号丢失找回机制进行保证安全性的设计。

可维护性：软件开发过程需编写详细的文档与代码注释，对使用的算法方法进行详细说明。

可移植性：尽量使用标准库里面所提供的函数，提高软件的跨系统可移植性。

# 

# 2. 概要设计

考勤系统采用了MVC模式，为一般员工提供注册，登陆，请假，查看假期批准情况，查看剩余假期功能。为部门经理提供审核假期，生成部门人员请假记录功能。为总经理提供生成员工请假记录功能。为人事科人员提供修改员工出生日期，修改员工工作日期，设置员工性别，获取出勤报表，核对材料功能。

系统采用框架设计，系统的各个子模块之间功能独立，各个子模块没有直接耦合，通过数据库之间的联系由框架进行整合，子模块的修改只是模块内的局部修改，不会导致修改的蔓延，尽量做到高内聚，低耦合的设计理念，从而使系统的抗修改能力大大提高，降低了系统开发的风险。

系统的架构设计主要考虑了一下几个方面

1. 性能：主要考虑访问频率，每个用户每天的访问次数，考虑到后期可能会访问量暴增，所以引入分布式部署，引入缓存
2. 可扩展性，因为考虑到系统的扩展性的需求
3. 高可用，系统一旦宕机，就会给用户带来了不可挽回的损失，因此必须做负载均衡。
4. 安全性：系统的信息有一定的隐私性，例如用户的个人信息，不包含一些隐私的信息，因此需要使用账号密码管理、数据库访问权限控制
5. 成本：主要成本在于服务器成本，流量成本，存储成本，

系统采用的技术栈：

1. 前端：Vue+elementUI
2. 后端：

用户层：token

开发层：开发框架：Spring Boot + MyBatisPlus + Shiro + Redis

服务器：Tomcat

存储层：MySQL

## 2.1 系统的主要功能

系统使用者分为管理员（总经理）与用户（工作成员）两部分。

普通用户可以登录系统、浏览相关内容

系统管理一般由总经理担任。

程序采用B/S框架，采用SSM技术，（spring + spring MVC + MyBatisPlus），采用面对对象的设计方法，PnP的设计理念，在框架的组织下，程序员无需直接了解项目的流程等细节，只需要关心起负责的功能模块，对程序员的要求大大降低，也避免了程序员个人能力不足对项目造成的影响。大大提高了系统的可维护性，降低了维护的风险。1.3 主要的设计约束和限制

## 2.2 编程语言限制

客户端使用Java语言进行程序编写，网页端采用elementUI前端设计+后端Java进行程序设计。

## 2.3 工具约束

采用IntelliJ IDEA、VS code开发。

## 2.4 性能约束

保证网络连接的稳定性，降低断网情况发生的概率

能在用户提出登陆/退出请求后快速响应，保证3s内实现网络连接和断开

对用户使用的平台其他功能完整记录不缺失，查询保证在3s内完成响应

保证用户隐私信息的安全

## 2.5架构设计

### 2.5.1 数据流复审

#### 2.5.1.1 静态数据

|  |  |  |
| --- | --- | --- |
| 名称 | 简述 | 有关数据流 |
| 用户 | 使用本系统的用户 | 请假，查看剩余假期。 |
| 管理员 | 主要由开发人员担任，负责对进行全面管理。 |  |

#### 2.5.1.2 动态数据

|  |  |  |  |
| --- | --- | --- | --- |
| 数据流名称 | 简述 | 来源 | 去向 |
| 注册用户请求  登录系统请求 | 用户使用username，mobile, password来注册成为用户  用户使用账号和密码进行系统登录 | 员工，部门经理  员工，部门经理 | 服务器  服务器 |
| 个人信息 | 系统响应管理登录后的结果 | 员工，部门经理 | 用户流量数据、服务器 |
| 请假需求 | 用户通过发送请求数据以及自身的职位 | 员工，部门经理 | 服务器 |
| 查看批准情况 | 员工登陆账号 | 员工 | 服务器 |
| 查看剩余假期 | 员工登陆账号 | 员工，部门经理， | 服务器 |
| 批准长假期 | 总经理登陆账号 | 总经理 | 服务器数据存储 |
| 驳回长假期 | 总经理登陆账号 | 总经理 | 服务器 |
| 生成员工请假记录 | 总经理 | 总经理 | 服务器 |
| 修改员工出生日期 | 人事科人员 | 员工，部门经理，总经理，人事科人员 | 服务器 |
| 修改员工工作日期 | 人事科人员 | 员工，部门经理，总经理，人事科人员 | 服务器 |
| 设置员工性别 | 人事科人员 | 员工，部门经理，总经理，人事科人员 | 数据库 |
| 获取出勤报表 | 人事科人员 | 员工，部门经理，总经理，人事科人员 | 数据库 |

# 3.数据库设计

为了缓解服务器压力、提高效率，决定采用分库分表的形式，以便于将数据分别挂载到多个服务器上、分担服务器压力。

## 3.1

### 3.1.1 员工表 worker

数据项说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 说明 | 类型 | 字段约束 | 字段类型 |
| woker\_id | 员工ID | int | auto\_increament | PK |
| name | 用户昵称 | varchar(6) | Not null |  |
| sex | 性别 | varchar(2) | Not null |  |
| age | 年龄 | int | Not null |  |
| worker\_degree | 员工等级 | varchar(6) | Not null |  |

### 3.1.2 系统用户表 work\_user

数据项说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 说明 | 类型 | 字段约束 | 字段类型 |
| worker\_id | 自增主键 | bigint | not null | PK |
| username | 假期类型 | varchar(50) | not null |  |
| password | 假期描述 | varchar(100) | not null |  |
| salt | 具体假期 | varchar(20) | not null |  |
| email | 邮箱 | varchar(100) | not null |  |
| mobile | 手机号 | varchar(100) | not null |  |
| status | 状态 | tinyint | not null |  |
| create\_user\_id | 创建者Id | bigint | not null |  |
| create\_time | 创建时间 | datetime | not null |  |

### 3.1.3 请假记录表 off\_note

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 说明 | 类型 | 字段约束 | 字段类型 |
| id | id | int | Not null |  |
| worker\_id | 员工id | int | Not null |  |
| off\_num | 请假编号 | int | auto\_increment | PK |
| off\_time | 开始时间 | date |  |  |
| off\_end | 结束时间 | date |  |  |
| off\_days | 请假天数 | int |  |  |

### 3.1.4 请假记录表 worker\_user\_token

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 说明 | 类型 | 字段约束 | 字段类型 |
| worker\_id | id | bigint | Not null |  |
| token | 员工id | varchar(100) | Not null |  |
| expire\_time | 请假编号 | datetime | auto\_increment | PK |
| update\_time | 开始时间 | datetime |  |  |

### 3.1.5 请假批准表 manager\_approve

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 说明 | 类型 | 字段约束 | 字段类型 |
| id | id | int | auto\_increment | PK |
| manager\_id | 总经理id | int |  |  |
| off\_id | 请假编号 | int | not null |  |
| ratified | 是否同意 | int |  |  |
| off\_time | 请假开始时间 | date | not null |  |
| end\_time | 请假结束时间 | date | not null |  |

### 3.1.4 出差记录表 manager\_approve

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 说明 | 类型 | 字段约束 | 字段类型 |
| id | id | int | Not null |  |
| worker\_id | 员工id | int | Not null |  |
| out\_num | 出差编号 | int | auto\_increment | PK |
| out\_time | 开始时间 | date |  |  |
| back\_time | 结束时间 | date |  |  |
| out\_days | 出差天数 | int |  |  |

# 4 详细设计

## 4.1接口设计

### 4.1.1 管理员模块

|  |  |
| --- | --- |
| URL | /attendsys/sys/login |
| Method | POST |
| 入参 | SysLoginForm |
| 说明：管理员登陆 | |

|  |  |
| --- | --- |
| URL | /attendsys/logout |
| Method | POST |
| 入参 | Null |
| 说明：管理员登出 | |

|  |  |
| --- | --- |
| URL | /attendsys/sys/user/list |
| Method | GET |
| 入参 | params |
| 说明：返回用户列表 | |

|  |  |
| --- | --- |
| URL | /attendsys/sys/user/info |
| Method | GET |
| 入参 | Null |
| 说明：返回用户登陆信息 | |

|  |  |
| --- | --- |
| URL | /attendsys/sys/user/password |
| Method | POST |
| 入参 | PasswordForm |
| 说明：修改密码 | |

|  |  |
| --- | --- |
| URL | /attendsys/sys/user/info/{id} |
| Method | POST |
| 入参 | userId |
| 说明：更具Id查询 | |

|  |  |
| --- | --- |
| URL | /attendsys/sys/user/save |
| Method | Post |
| 入参 | user |
| 说明：新增管理员 | |

|  |  |
| --- | --- |
| URL | /attendsys/sys/user/update |
| Method | POST |
| 入参 | user |
| 说明：更新管理员信息 | |

|  |  |
| --- | --- |
| URL | /attendsys/sys/user/delete |
| Method | POST |
| 入参 | userId |
| 说明:删除管理员 | |

### 4.1.2 用户模块

|  |  |
| --- | --- |
| URL | /attendsys/worker/worker/list |
| Method | GET |
| 入参 | params |
| 说明: 获取worker信息 | |

|  |  |
| --- | --- |
| URL | /attendsys/worker/worker/info/{workerId} |
| Method | GET |
| 入参 | workerId |
| 说明：根据workerId查询用户 | |

|  |  |
| --- | --- |
| URL | /attendsys/worker/workuser/list |
| Method | GET |
| 入参 | params |
| 说明：获得workuser信息 | |

|  |  |
| --- | --- |
| URL | /attendsys/worker/workuser/login |
| Method | POST |
| 入参 | WorkerLoginForm |
| 说明：worker登陆 | |

|  |  |
| --- | --- |
| URL | /attendsys/worker/workuser/logout |
| Method | POST |
| 入参 | WorkLogoutForm |
| 说明：worker登出 | |

|  |  |
| --- | --- |
| URL | /attendsys/worker/offnote/save |
| Method | POST |
| 入参 | offNote |
| 说明：存入请假记录 | |

|  |  |
| --- | --- |
| URL | / attendsys/worker/offnote/off |
| Method | POST |
| 入参 | offNote |
| 说明：请假 | |

|  |  |
| --- | --- |
| URL | / attendsys/worker/mangerapprove/list |
| Method | GET |
| 入参 | params |
| 说明：获取所有批假条 | |

|  |  |
| --- | --- |
| URL | / attendsys/worker/mangerapprove/ratify |
| Method | POST |
| 入参 | params |
| 说明：批请假条 | |

# 5 软件测试

## 5.1 引言

### 5.1.1 编写目的

本部门的编写是向公司员工、管理员，以及更多的项目阅读者介绍本项目从开发到完成期间计划的测试流程、阐明针对于项目中的核心功能模块的测试思路，分享本项目在测试过程中的重难点，供与老师和读者进行评价。

### 5.1.2 术语和缩略语

无

### 5.1.3 参考资料

[1] 质量管理体系国家标准理解与实施(2008 版)

[2] ISO 9001 质量体系——设计、开发、生产、安装和服务的质量保证模式

[3] 软件工程（原书第 9 版）. 机械工业出版社:2011.5

## 5.2. 测试需求分析

### 5.2.1 测试策略与目标

**测试重点：**

1. 添加人员、请假、审批的逻辑过程测试

2. 个人信息模块的逻辑测试，模块内对个人信息数据库的相关操作正确性的测试

3. 动物识别有关模块的正确性测试，测试模块内对于信息的整合发送是否准确

**测试难点：**

1. 添加人员、请假、审批的逻辑过程测试

2. 并发下的系统测试

3. 安全性系统测试

**目标：**定义项目在发布时候的质量等级为量化管理级。

### 5.2.2 测试范围

**测试广度：**测试相关类型的了解。具体包括如下：需求分析、测试流程、测试管理、开发流 程、开发技术、测试模型、基础测试方法和测试技术、功能测试、性能测试、可靠性测试、易用性测试、稳定性测试，兼容性测试。

**测试深度：**在同一个测试步骤阶段运用多种手段全方位进行测试。

### 5.2.3 测试环境

**硬件环境：**PC

**操作系统环境：**windows

**应用服务器环境：**Apache、tomcat、Nginx 集群环境：负载均衡

### 5.2.4测试输出

测试样例说明书

分类别的阶段性测试文档

问题反馈报告

## 5.3 计划测试过程

### 5.3.1 初步静态测试

对代码进行走读和程序分析，以便分析程序和文档之间的一致性，规范程序结构。

采用程序数据流监控法。

### 5.3.2 单元测试

具体包括各模块的独立测试。

**1.登录功能模块：**进行程序分析确认逻辑正确，使用动态测试法和黑盒测试法不断检测，使用等价分类法和边界分析法，把程序的输入数据集合按输入条件划分为若干个等价类，判断是否准确实现了指令的正确整合发送和实现。

测试用例**：**用户名1-密码1（系统未导入）结果：登陆失败，返回错误原因

用户名2-密码2（已注册已登录）结果：登陆失败，返回错误原因

用户名3-密码3（已注册未登录）结果：登陆成功

用户名1-密码3（密码不匹配用户名）结果：登陆失败，返回错误原因

**2.查看个人信息模块：**进行程序分析确认逻辑正确，使用黑盒测试法不断检测，判断是否准确实现了正确分发请求，正确整合反馈的功能。

测试用例：无

**3.** **人员检索功能模块：**进行程序分析确认逻辑正确，使用动态测试法和黑盒测试法不断检测，使用等价分类法和边界分析法，把程序的输入数据集合按输入条件划分为若干个等价类，判断是否准确实现了指令的正确整合发送和实现。

测试用例：个人信息（正确）加筛选信息（正确）加请求（正确） 结果：查询成功

个人信息（正确）加筛选信息（正确）加请求（错误） 结果：查询失败

个人信息（正确）加筛选信息（错误）加请求 结果：查询失败

个人信息（错误）加请求 结果：查询失败

**4. 浏览功能模块：**进行程序分析确认逻辑正确，使用动态测试法和黑盒测试法不断检测，使用等价分类法和边界分析法，把程序的输入数据集合按输入条件划分为若干 个等价类，判断是否准确实现了指令的正确整合发送和实现。

测试用例：个人信息（正确）加请求（正确） 结果：查看成功

个人信息（正确）加请求（错误） 结果：查看失败

个人信息（错误）加请求 结果：查看失败

**5. 修改信息功能模块：**进行程序分析确认逻辑正确，使用动态测试法和黑盒测试法不断检测，使用等价分类法和边界分析法，把程序的输入数据集合按输入条件划分为若干个等价类，判断是否准确实现了指令的正确整合发送和实现。

测试用例：个人信息（正确）加请求（正确） 结果：查看成功

个人信息（正确）加请求（错误） 结果：查看失败

个人信息（错误）加请求 结果：查看失败

**6. 审批假期功能模块：**进行程序分析确认逻辑正确，使用动态测试法和黑盒测试法不断检测，使用等价分类法和边界分析法，把程序的输入数据集合按输入条件划分为若干个等价类，判断是否准确实现了指令的正确整合发送和实现。

测试用例：个人信息加假期信息（不规范）加指令 结果：指令失败

个人信息加假期信息（过长）加指令 结果：指令失败

个人信息加假期信息（长度为最大值）加指令 结果：指令失败

个人信息加假期信息（规范）加指令 结果：指令成功

**7. 申请假期功能模块：**进行程序分析确认逻辑正确，使用动态测试法和黑盒测试法不断检测，使用等价分类法和边界分析法，把程序的输入数据集合按输入条件划分为若干个等价类，判断是否准确实现了指令的正确整合发送和实现。

测试用例：个人信息（正确）加请假信息（正确）加请求（正确） 结果：查询成功

个人信息（正确）加请假信息（正确）加请求（错误） 结果：查询失败

个人信息（正确）加信息（错误）加请求 结果：查询失败

个人信息（错误）加请求 结果：查询失败

### 5.3.3 组装测试

采取自顶向下的集成方式，从主控模块开始,按照软件的控制层次结构,以深度优先策略,逐步把各个模块集成在一起。

按设计要求把通过单元测试的各个模块组装在一起，测试各模块之间的连接组合功能是否按照需求完成，以便发现与接口有有关的各种错误。

#### 5.3.3.1 步骤

（1) 以主控模块作为测试驱动模块,把对主控模块进行单元测试时引入的所有桩模块用实际模块替代;

（2) 依据深度优先的集成策略,每次只替代一个或多个桩模块;

（3)每集成一次立即测试一遍;

（4)每组测试完成后,着手下一次桩模块替换;

（5)为避免引入新错误,须不断进行回归测试(即全部或部分地重复已做过的测试)。

（6)从第二步开始,循环执行上述步骤,直至整个程序结构构造完毕。

### 5.3.4 确认测试

确认测试检查该系统能否按合同要求进行工作,即是否满足软件需求说明书中的确认标准。测试过程则定义了一些特殊的测试用例,旨在说明软件与需求是否一致。

#### 5.3.4.1 功能需求测试

**1.进行有效性测试（黑盒测试）**

有效性测试是在模拟的环境（可能是就是开发的环境）下，运用黑盒测试的方法，验证所测试件是否满足需求规格说明书列出的需求。首先制定测试计划，规定要做测试的种类，还需要制定一组测试步骤，描述具体的测试用例。通过实施预定的测试计划和测试步骤，确定软件的特性是否与需求相符，确保所有的软件功能需求都能得到满足，所有的软件性能需求能达到，所有的文档都是正确且便于使用。

选课管理系统的主要功能为用户信息管理，课程信息管理，学生成绩管理等功能，具体的功能大致分为用户登陆、查看个人信息、查看课程信息、查看通知、录入成绩、查看选课信息、发布通知、修改选课时间、学生查看可选课程以及相关课程、查看成绩、选课、退课等。

**2. 软件配置审查**

在确认测试的过程，应当严格遵守用户手册和操作手册中规定的使用步骤，细记录发现的遗漏和错误，并且适当地补充和改正。

#### 5.3.4.2 性能需求测试

**1、数据精确度：**确认逻辑正确，使用黑盒测试法不断检测，确保输入和输出的精度。

**2、时间特性：**确认逻辑正确，使用黑盒测试法不断检测，确保响应时间可以在用户可接受的范围内。

测试方法如下：

分别使用不同数量的输入数据，测试系统计算这些数据所需要的时间，看对用户是否友好，争取达到需求分析要求的1s时间。

**3、适应性：**

测试方法如下：

（1）分别使用无线网页端和带网线的客户端进行交互，看是否具有基本一致性

（2）分别使用windows,mac,linux等系统进行交互，查看该系统的跨平台性。

#### 5.3.4.3运行需求测试

**1. 用户界面：**确认逻辑正确，界面完整、简洁、无歧义，可用性强

测试用例：无

**2. 网络与数据库接口：**确认逻辑正确，使用黑盒测试法不断检测，确保可以通过服务器进行数据交互、通过数据库进行数据交互。

测试用例如下：

|  |  |
| --- | --- |
| 测试用例 | 结果 |
| 向服务器请求登录 | 正确返回列表 |
| 向服务器请求用户列表 | 正确返回列表 |
| 向服务器请求请假列表 | 正确返回列表 |
| 个人信息（正确）加修改数据库请求 | 请求成功 |
| 个人信息（错误）加修改数据库请求 | 请求失败 |

### 5.3.5 系统测试

#### 5.3.5.1恢复测试

采用各种办法使系统发生以下故障：

（1）软件在使用过程中所使用的计算机突然断电或者计算机系统突然崩溃

（2）软件出现不可预测的BUG

（3）软件使用过程网络连接突然断开

基于（1）（2）两点的故障，检测软件是否能够实时保护数据并存储，做到数据不因故障而丢失

对于故障（3），检测软件是否能检测到异常的连接断开并能尝试重连或给出错误信息。

#### 5.3.5.2安全测试

检查系统对非法侵入的防范能力。测试人员假扮非法入侵者，采用各种办法试图突破防线。验证是否对账户信息进行经过加密后传输与存储，保证安全性的设计，使非法侵入的代价超过信息价值。

#### 5.3.5.3强度与性能测试

迫使系统在异常的资源配置下运行，检查系统对异常情况的抵抗能力。对高并发等系统压力较大的情况，保证系统能够正常运行。

# 6 风险评估和控制

## 6.1 主要风险综述

### 6.1.1人员风险

是否有足够人员可用

开发人员技术是否过关

开发人员是否对工作有正确的期望

开发人员是否经过必要的培训

开发人员的流动是否能保证项目连续性

### 6.1.2 过程风险

开发组织是否已经拟定了一份已经成文的、用于本项目开发的软件过程的说明

开发人员是否同意按照文档所写的软件过程进行开发工作，并自愿使用它

管理者和开发人员是否接受过一系列的软件工程培训

是否定期对需求规约、设计和编码进行正式的技术评审

是否定期对测试过程和测试情况进行评审

是否使用一个机制来控制用户需求的变化及其对软件的影响

### 6.1.3 技术风险

使用的技术是否不过时

当前可用的技术框架是否合理

是否没有特别苛刻的技术要求

客户是否能确定所要求的功能是可行的

需求中是否不需要采用新的分析、设计、测试方法

需求中是否要求使用传统的软件开发方法

### 6.1.4开发环境风险

是否有可用的软件项目管理工具

是否有可用的软件过程管理工具

是否有可用的分析及设计工具

是否有可用的测试工具

是否有可用的软件配置管理工具

是否有专门的开发工具管理

### 6.1.5 软件规模风险

是否以LOC（代码行）或FP（功能点）估算产品的规模

是否以程序、文件或事务处理的数目来估算产品规模

产品创建或使用的数据库大小如何

产品的用户数是否庞大

### 6.1.6客户风险

客户的需求是否清晰

客户是否不经常改动需求

客户合作态度是否友善

客户信誉是否良好

## 6.2项目风险表

描述预计项目中可能发生的风险。风险等级是指该风险对项目进度、质量和成本影响的严重程度，可分为四个等级，等级越高影响越严重。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 风险 | 类别 | 概率 | 影响 |
| 1 | 是否使用一个机制来控制用户需求的变化及其对软件的影响 | 过程风险 | 2 | 2 |
| 2 | 产品创建或使用的数据库大小如何 | 软件规模风险 | 2 | 2 |
| 3 | 产品的用户数是否庞大 | 软件规模风险 | 2 | 2 |
| 4 | 开发人员的流动是否能保证项目连续性 | 人员风险 | 3 | 1 |
| 5 | 是否有可用的测试工具 | 开发环境风险 | 1 | 2 |
| 6 | 是否有可用的软件配置管理工具 | 开发环境风险 | 1 | 2 |
| 7 | 是否有可用的软件项目管理工具 | 开发环境风险 | 1 | 2 |
| 8 | 开发人员是否经过必要的培训 | 人员风险 | 2 | 3 |
| 9 | 是否有可用的软件过程管理工具 | 开发环境风险 | 3 | 2 |
| 10 | 是否定期对需求规约、设计和编码进行正式的技术评审 | 过程风险 | 3 | 3 |
| 11 | 是否定期对测试过程和测试情况进行评审 | 过程风险 | 3 | 3 |
| 12 | 是否有可用的分析及设计工具 | 开发环境风险 | 3 | 3 |
| 13 | 是否以程序、文件或事务处理的数目来估算产品规模 | 软件规模风险 | 3 | 2 |
| 14 | 当前可用的技术框架是否合理 | 技术风险 | 3 | 2 |
| 15 | 开发人员技术是否过关 | 人员风险 | 3 | 3 |
| 16 | 开发组织是否已经拟定了一份已经成文的、用于本项目开发的软件过程的说明 | 过程风险 | 3 | 3 |
| 17 | 需求中是否不需要采用新的分析、设计、测试方法 | 技术风险 | 3 | 3 |
| 18 | 是否没有特别苛刻的技术要求 | 技术风险 | 4 | 3 |
| 19 | 客户的需求是否清晰 | 客户风险 | 3 | 3 |
| 20 | 客户是否不经常改动需求 | 客户风险 | 3 | 3 |
| 21 | 是否以LOC（代码行）或FP（功能点）估算产品的规模 | 软件规模风险 | 3 | 3 |
| 22 | 是否有足够人员可用 | 人员风险 | 4 | 3 |
| 23 | 开发人员是否同意按照文档所写的软件过程进行开发工作，并自愿使用它 | 过程风险 | 4 | 3 |
| 24 | 管理者和开发人员是否接受过一系列的软件工程培训 | 过程风险 | 4 | 3 |
| 25 | 开发人员是否对工作有正确的期望 | 人员风险 | 4 | 3 |
| 26 | 客户是否能确定所要求的功能是可行的 | 技术风险 | 4 | 4 |
| 27 | 是否有专门的开发工具管理 | 开发环境风险 | 4 | 4 |
| 28 | 客户合作态度是否友善 | 客户风险 | 4 | 4 |
| 29 | 需求中是否要求使用传统的软件开发方法 | 技术风险 | 4 | 4 |
| 30 | 使用的技术是否不过时 | 技术风险 | 4 | 4 |
| 31 | 客户信誉是否良好 | 客户风险 | 4 | 4 |

### 6.2.1 中止线以上风险描述

**人员风险：**在开发过程中，由于开发团队里的人员可能存在某方面人员缺乏、人员技术开发技术不过关、工作态度不端正，以及人员波动等因素，给开发该系统带来风险。

**过程风险：**在开发过程中，由于可能存在设计软件开发文档不完善、不合理，在客户需求发生变化时未能及时进行调整开发计划，软件编码完成后测试阶段不完善等情况，导致最终软件系统未能符合客户需求或者软件性能低下。

**技术风险：**在进行需求分析时，未能准确分析出需求的可行性，当前团队缺乏开发该系统部分模块的开发人员或当前开发技术过时，进行盲目开发，造成最终软件系统过于庞杂，灵活度不高，实用性极差。

**开发环境风险：**在软件开发各个阶段，缺乏需要的各种开发、管理、测试工具，造成软件开发过程较慢，产生风险。

**软件规模风险：**定量估算软件规模的方法不准确,从而导致对软件规模估计产生偏差;在做需求分析时,未能准确把握使用该系统的用户数量,导致用户使用数量超出系统最大承载数。

**客户风险：**客户可能存在需求描述不清晰，经常发生需求变动，合作态度不友善，信誉不好等情况，给软件开发过程带来阻力。

### 6.2.2 影响概率及影响因素

(1)需制定的规范和标准较多，而同时需完成其他工作，使得可使用的时间和资源有限。

(2)由于设备未到位导致延误开发。

(3)由于都是全新的课题，对于技术的掌握程度和经验仍很欠缺。

(4)由于学习曲线过长延误时间。

(5)测评结果对功能规格和系统设计影响较大。

## 6.3风险缓解

### 6.3.1 人员风险

**a 缓解**

（1）一般策略

进行人员调整，在软件需求分析、设计分析阶段确定好所需技术人员。定义清晰的沟通计划和沟通制度

（2）缓解风险的特定步骤

按照以往开发经验对那些不能稳定开发的技术人员进行更换。制订项目资源管理办法，建立有效的激励机制，为关键人员准备后备人员。

**b 监控**

（1）被监控的因素

开发人员的技术，职业素养

（2）监控方法

按软件开发技术需求对技术人员进行筛查和合理的分工，按以往的开发经验评估技术人员的开发素养。

**c管理**

（1）意外事件计划

对于意外事件，由项目负责人重新规划软件的开发计划，并且调整软件开的人员。加强人员培训和考核，不得已时替换相关人员。

（2）特殊考虑

对于一些至关重要的开发人员，可进行酌情处理。

### 6.3.2过程风险

**a.缓解**

（1）一般策略

对软件文档开发阶段进行细致审查，并及时按照客户的需要更改软件开发文档。

（2）缓解风险的特定步骤

在软件测试阶段，用各种方法进行细致全面地测试。在系统实施时，首先要对用户现状及用户需求做详尽的描述。通常由于用户对业务理解还在不断深化，往往在系统实施时用户对需求的描述会随着实施的不断深入而有所改变，造成系统需求的不稳定。

**b.监控**

（1）被监控的因素

软件开发文档的合理性，用户需求的变化，以及测试阶段的准确性。

（2）监控方法

定时检查软件开发文档，及时与客户沟通

**c.管理**

（1）意外事件计划

若发现在编码过程中难以进行，或者客户需求变化过大，可以经项目负责人批准重新进行软件文档设计。

（2）特殊考虑

如在测试阶段出现难以解决的问题，返回前面阶段重新设计分析。

### 6.3.3技术风险

**a缓解**

（1）一般策略

技术人员按照开发该软件所需技术进行学习或更换相关技术人员。

（2）缓解风险的特定步骤

特别要注意学习某些重要技术的更新。

（3）系统故障

1.与用户维护人员一起分析，确定故障成因；

2.如果是软件运行平台硬件故障，立即向用户方维护部门领导汇报，争取尽早联系相关硬件供应商解决问题；

3.如果因应用软件漏洞造成系统故障，应尽快修补漏洞，使系统恢复正常运行；

4.对于用户误操作造成的故障，应在软件编制时应该充分考虑纠错措施，减轻或避免误操作带来的问题；同时应重视用户培训；

**b.监控**

（1）被监控的因素

现有技术人员所掌握的技术和开发该软件系统所需技术。

（2）监控方法

在编写软件文档阶段安排技术学习，在编码阶段通过多种渠道检测新技术。

**c.管理**

（1）意外事件计划

对于意外事件，由项目负责人安排人员变动和与客户进行沟通，采用别的方案代替。

（2）特殊考虑

对于无法解决的技术问题以替代方案解决时，变动不宜太大。

### 6.3.4开发环境风险

**a缓解**

（1）一般策略

及时配备所需开发工具。

（2）缓解风险的特定步骤

在上一阶段开发时进行下一阶段的工具准备。

**b.监控**

（1）被监控的因素

各个阶段所需开发工具。

（2）监控方法

实时根据需要的工具进行配备。

**c.管理**

（1）意外事件计划

若软件开发工具成本过高，经项目负责人处理，以较为合理的方式代替。

（2）特殊考虑

若客户不同意代替计划，可与客户重新协商项目价格。

### 6.3.5软件规模风险

**a缓解**

1. 一般策略   
   使用合理的定量估算软件规模的方法，并用多种估算方法进行计算，以避免出错；在需求分析阶段与客户充分沟通，并对涉及的用户数量作足够的估计。

1.建立项目实施领导小组，明确项目的目标和各自的权限；

2.配备经验丰富的项目经理；

3.定期向项目的高层管理人员和用户报告项目实施的进展及存在的问题；

4.控制实施范围的变化，形成书面文档、陈述更改原因，待管理部门批准后方可实施更改；

5.建立当项目实施出现问题时进行汇报和解决的工作流程。

（2）缓解风险的特定步骤

用其他估算方法重新计算软件的规模，调整项目的实施计划；在项目实施阶段，提高软件所能承受的最大用户数量。

**b.监控**

（1）被监控的因素

软件规模估算的值；客户提出的用户数量和软件的最大承载数。

（2）监控方法

进行估算的软件规模大小是否与实际相符合；经常与客户交流，及时了解软件所要达到的最大用户数。

**c.管理**

（1）意外事件计划

对于意外事件，由项目负责人重新进行估算，并修改项目计划；与客户交流后发现用户数量大于软件所能承受的最大数量，由项目负责人与客户重新协商，重新做需求分析或者是修改交付产品的日期。

（2）特殊考虑

在项目到中后期，对客户提出的新需求，进行临时程序修改，如不可修改，告知用户在下个版本解决。

# 7. 个人工作

## 7.1 任务分工

前端：张春胜

后端：张启洲：系统功能模块

后端：谢梦鑫：用户功能模块