# 引言

## 1.1标识

标题：《总体设计说明书》

简称：SDS

版本号：1.0.0

说明：本文档支持Office、WPS等word文档阅读软件

## 1.2系统概述

国内的中小微企业，在招聘环节，他们都面临着招工难、踩坑多的困扰。招到一个缺乏责任感、能力不足、稳定性差的员工，对企业和团队来说，都将造成巨大损失。因此，需要建立一套跨组织人才管理系统，可以帮助新雇主客观全面真实地了解面试者曾经的工作能力与工作状态，以便更好地匹配岗位，并规避招错人的风险。

## 1.3文档概述

本文档是对该开发系统的一个总体的把握，旨在推动该项目软件工程的规范化，使设计人员遵循统一的详细设计书写规范，节省制作文档的时间，降低系统实现的风险，做到系统设计资料的规范性与全面性，以利于系统的实现、测试、维护、版本升级等。

## 1.4参考资料

《Software Engineering》 出版社：机械工业出版社 作者：Roger S.Pressman

《SQL编程基础》 出版社: 机械工业出版社作者：Patrick.J.J.

《深入浅出——MySQL开发、管理与应用实例》出版社：清华大学出版社 作者：邹建

《数据库系统概论》 出版社：高等教育出版社 作者：王珊，萨师煊

# 总体设计

## 2.1需求规定

#### 功能：

**用户方：**

1. 求职
2. 离职
3. 签到
4. 查看任务
5. 查看个人简历
6. 查看个人信息
7. 修改个人信息
8. 登录

**HR方：**

1. 求职

2. 离职

3. 签到

4. 查看任务

5. 查看个人简历

6. 查看个人信息

7. 修改个人信息

8. 登录

9. 评价

10. 同意用户求职

11. 同意用户离职

**主管方：**

1. 发布任务
2. 评价
3. 同意离职
4. 任命HR
5. 登录
6. 修改公司信息
7. 修改个人信息

**管理员方：**

1. 新增公司
2. 新增公司主管
3. 修改公司信息
4. 修改主管信息

#### 性能：

该系统在日常处理中的响应速度为秒级，达到实时要求，以及时反馈信息。在进行统计分析时，根据所需数据量的不同而从秒级到分钟级，原则是保证操作人员不会因为速度问题而影响工作效率。

#### 数据来源：

用户手动输入的数据

#### 输入要求：

1. 注册账号时不能与其他用户重复
2. 用户登录账号时，账号需详实准确
3. 用户登录账号时，账号与密码应匹配
4. 用户搜索时输入的关键字不得包含敏感词汇
5. 用户编写简历或评价时不得包含敏感词汇

#### 输出要求：

1. 显示的信息中不能包含错别字
2. 显示的信息中不能包含敏感词汇

## 2.2运行环境

设备数量：三台小组成员pc

处理器：无特殊需求，现在市面上的硬件设备性能完全可以支撑使用。

存储器：8G及以上运行内存

输入/输出设备：鼠标、键盘、显示器、USB接口

辅助存储器：128G以上内存

通信/网络设备：网络驱动器，能正常连网即可

## 2.3供选择的方案

**2.3.1 供选择的方案1**

方案：

采用jsp框架+ tomcat服务器部署，数据库采用mysql

缺点：

1. jsp耦合度太高、前后端不分离，属于过时的技术。
2. 如今大部分项目同时使用多种数据库，完成访问速度。
3. Jsp需要部署tomcat服务器。

优点：

不需要前后端沟通

存在问题：

环境搭建较难。

经济可行性：

支出：

1. 云服务器租金
2. Office正版费用
3. Windows 10 正版费用

收入：

1. 广告收入
2. 按照网站人数增加带来的一系列收益，包括但不限于：会员收益等。

**2.3.2 供选择的方案2**

方案：

用html+css制作前端，用java web实现后端，用MySQL做数据库。

优点：

基本能实现预期效果。

能将小组的学习成本降到最低

缺点：

界面美观程度欠缺，用户体验可能不够好

局限性及存在的问题：

用户体验问题

经济可行性：

支出：

1. 云服务器租金
2. Office 正版费用
3. Windows 10 正版费用

收入：

1. 广告收入
2. 按照网站人数增加带来的一系列收益，包括但不限于：会员收益等。

**2.3.3 供选择的方案3**

方案：

要用Vue制作前端，用Spring boot框架实现后端，用MySQL、Redis做数据库。

优点：

基本能实现预期效果。

Spring boot可以简化搭建环境，不需要部署tomcat，直接运行jar包

Redis数据库访问速度快

缺点：

组员所需自学的科目过多，时间上可能来不及；

局限性及存在的问题：

大部分内容涉及到的技术都没有学过，需要自学；

经济可行性：

支出：

1. 云服务器租金
2. Office 正版费用
3. Windows 10正版费用

收入：

1. 广告收入
2. 按照网站人数增加带来的一系列收益，包括但不限于：会员收益等。

**最终经过小组商讨与权衡，决定采用方案3。**原因如下：

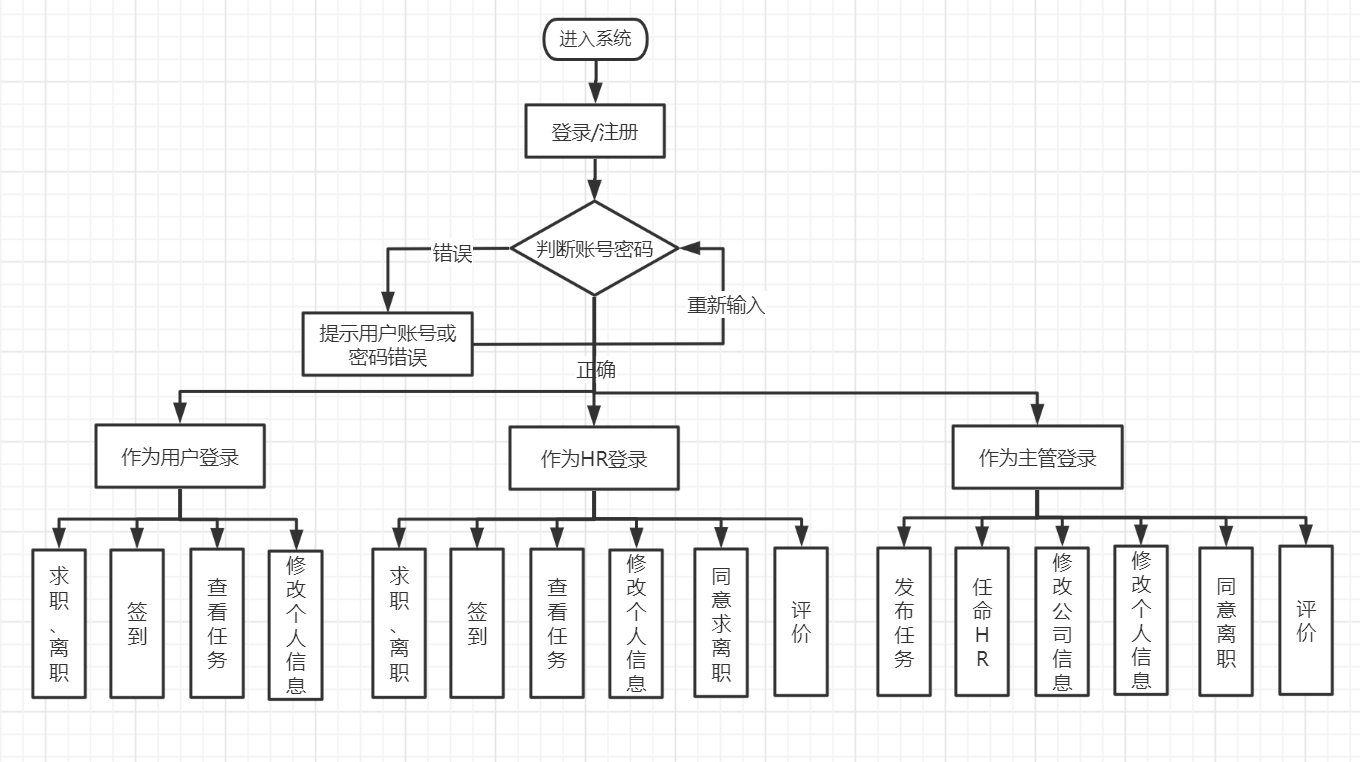
方案1的技术有些过时，前后端不分离不需要沟通，但是耦合度太高，环境的搭建也较为困难，可能会影响到后续的实现和测试

方案2技术难度相对较低，但是存在用户体验较差的问题。

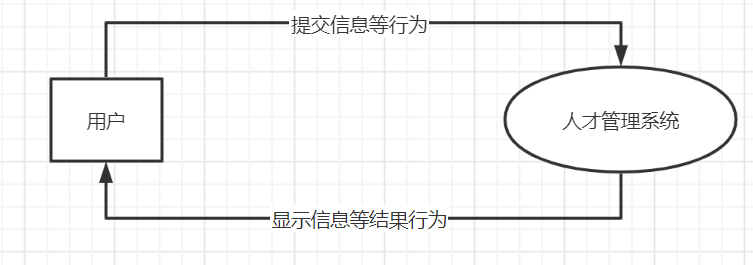
方案3前后端分离，耦合度较低，采用框架也能优化用户体验。

## 2.4基本设计概念和处理流程

### 2.4.1系统流程图



### 2.4.2数据流图



# 3.接口设计

## 3.1用户接口

1.将要采用的图形用户界面（GUI）标准。

2.屏幕布局自适应。

3.每个屏幕的标准按钮（参考UI界面图）；

4.错误信息显示标准——提示框跳出错误信息。

## 3.2外部接口

1. 将要采用的图形用户界面（GUI）标准

2. 屏幕布局自适应

3. 每个屏幕的标准按钮（参考UI界面图）

4. 错误信息显示标准——提示框跳出错误信息

## 3.3内部接口

|  |  |  |
| --- | --- | --- |
| **接口** | **模块** | **描述** |
| /admin/allCompany | 公司管理 | 管理员获取系统中所有公司数据 |
| /admin/allUser | 用户管理 | 管理员获取系统中所有用户数据 |
| /admin/setCompany | 公司管理 | 新录入公司 |
| /admin/setMaster | 公司管理 | 设置公司主管 |
| /hr/UserSignIn | HR | 获取总体考勤 |
| /hr/checkSetSginIn | HR | 判断今日是否发布签到 |
| /hr/companyHr | HR | 查看公司HR |
| /hr/companyUser | HR | 查看公司员工 |
| /hr/demission | HR | 处理用户离职 |
| /hr/dimissionList | HR | 获取用户离职申请列表 |
| /hr/downloadUserResume | HR | 下载他人简历 |
| /hr/getUserInfo | HR | 获取用户信息 |
| /hr/getUserWork | HR | 获取他人职业经历 |
| /hr/getWorkAchieve | HR | 获取绩效 |
| /hr/getWorkEvent | HR | 获取重大事件 |
| /hr/getWorkSign | HR | 获取出勤记录 |
| /hr/job | HR | 处理用户离职 |
| /hr/jobList | HR | 获取用户求职申请列表 |
| /hr/notSignIn | HR | 查看今日未签到 |
| /hr/setEstimate | HR | 修改评价 |
| /hr/setEvent | HR | 添加重大事件 |
| /hr/setSignIn | HR | 发布签到 |
| /master/demission | Master | 处理离职 |
| /master/demissionList | Master | 获取HR离职申请列表 |
| /master/estimateAchieve | Master | 评价绩效 |
| /master/job | Master | 处理求职 |
| /master/jobList | Master | 获取HR求职申请列表 |
| /master/scoreTask | Master | 评分任务 |
| /master/setAchieve | Master | 发布绩效 |
| /master/setCompany | Master | 修改公司信息 |
| /master/setHr | Master | 设置HR |
| /master/setTask | Master | 发布任务 |
| /master/uploadCompanyPhoto | Master | 上传公司图片 |
| /user/UserSignIn | User | 获取自己总体考勤 |
| /user/applyDimission | User | 申请离职 |
| /user/applyJob | User | 申请求职 |
| /user/checkEmail | User | 判断邮箱合法 |
| /user/checkIdentity | User | 判断身份证合法 |
| /user/checkPhone | User | 判断手机号合法 |
| /user/checkSignIn | User | 判断是否签到 |
| /user/checkUserName | User | 判断用户名合法 |
| /user/dimissionList | User | 获取离职申请列表 |
| /user/downloadSelfResume | User | 下载自己简历 |
| /user/getAllCompany | User | 查看公司列表 |
| /user/getCompanyInfo | User | 查看公司详情 |
| /user/getSelfWork | User | 获取个人职业经历 |
| /user/getTask | User | 查看任务 |
| /user/getUserInfo | User | 获取用户信息 |
| /user/getWorkAchieve | User | 获取绩效 |
| /user/getWorkEvent | User | 获取重大事件 |
| /user/jobList | User | 获取求职申请列表 |
| /user/login | User | 用户登录 |
| /user/register | User | 注册 |
| /user/searchCompany | User | 搜索公司 |
| /user/signIn | User | 签到 |
| /user/subTask | User | 确认提交任务 |
| /user/update | User | 更新个人信息 |
| /user/uploadUserPhoto | User | 上传自己照片 |
| /user/uploadUserResume | User | 上传自己简历 |
| /user/workSignIn | User | 获取自己某一时期的考勤 |

# 4.运行设计

## 4.1运行模块组合

### 4.1.1用户功能模块组合

|  |  |
| --- | --- |
| 功能模块 | 主要功能 |
| 登录 | 输入账号密码进行登录 |
| 注册 | 输入用户信息进行注册 |
| 修改信息 | 输入新的用户信息并保存 |
| 重置密码 | 输入与旧密码不同的新密码进行重置密码 |
| 注销 | 用户退出登录 |
| 签到 | 点击签到按钮完成当日打卡 |
| 求职、离职 | 向HR发送求职、离职报告 |
| 查看 | 获取工作任务列表 |

### 4.1.2HR功能模块组合

|  |  |
| --- | --- |
| 功能模块 | 主要功能 |
| 登录 | 输入账号密码进行登录 |
| 注册 | 输入用户信息进行注册 |
| 修改信息 | 输入新的用户信息并保存 |
| 重置密码 | 输入与旧密码不同的新密码进行重置密码 |
| 注销 | 用户退出登录 |
| 签到 | 点击签到按钮完成当日打卡 |
| 求职、离职 | 向主管发送求职、离职报告 |
| 同意求职、离职 | 通过确认用户的求职、离职请求完成 |
| 评价 | 输入对在职员工的评价 |

### 4.1.3主管功能模块组合

|  |  |
| --- | --- |
| 功能模块 | 主要功能 |
| 登录 | 输入账号密码进行登录 |
| 修改信息 | 输入新的用户信息并保存 |
| 重置密码 | 输入与旧密码不同的新密码进行重置密码 |
| 注销 | 用户退出登录 |
| 签到 | 点击签到按钮完成当日打卡 |
| 同意HR求职、离职 | 通过确认用户的求职、离职请求完成 |
| 评价 | 输入对在职员工的评价 |
| 发布任务 | 输入需要完成的工作，并分配 |

### 4.1.4管理员功能模块组合

|  |  |
| --- | --- |
| 功能模块 | 主要功能 |
| 登录 | 输入账号密码进行登录 |
| 注销 | 退出登录 |
| 新增公司 | 点击新增公司按钮完成新增 |
| 新增公司主管 | 点击新增主管按钮完成新增主管 |
| 修改主管信息 | 输入新的主管信息并保存 |

## 4.2运行控制

### 4.2.1用户功能模块运行控制

|  |  |
| --- | --- |
| 功能模块 | 控制方法 |
| 登录 | 验证用户名口令，系统核对后才能进行登录 |
| 注册 | 验证用户名是否已存在，系统核对后才能进行注册 |
| 修改信息 | 验证修改信息是否存在敏感词汇，核对后才能修改 |
| 重置密码 | 验证新密码与旧密码是否相同，系统核对后才能重置 |
| 注销 | 询问用户是否确认退出后，再运行下一步 |
| 签到 | 点击签到按钮完成当日打卡 |
| 求职、离职 | 验证信息中是否存在敏感词汇，核对后才能发送 |
| 查看 | 获取工作任务列表 |

### 4.2.2HR功能模块运行控制

|  |  |
| --- | --- |
| 功能模块 | 控制方法 |
| 登录 | 验证用户名口令，系统核对后才能进行登录 |
| 注册 | 验证用户名是否已存在，系统核对后才能进行注册 |
| 修改信息 | 验证修改信息是否存在敏感词汇，核对后才能修改 |
| 重置密码 | 验证新密码与旧密码是否相同，系统核对后才能重置 |
| 注销 | 询问用户是否确认退出后，再运行下一步 |
| 签到 | 点击签到按钮完成当日打卡 |
| 求职、离职 | 验证信息中是否存在敏感词汇，核对后才能发送 |
| 同意求职、离职 | 保存操作记录，防止恶意操作 |
| 评价 | 验证信息中是否存在敏感词汇，核对后才能发表 |

### 4.2.3主管功能模块运行控制

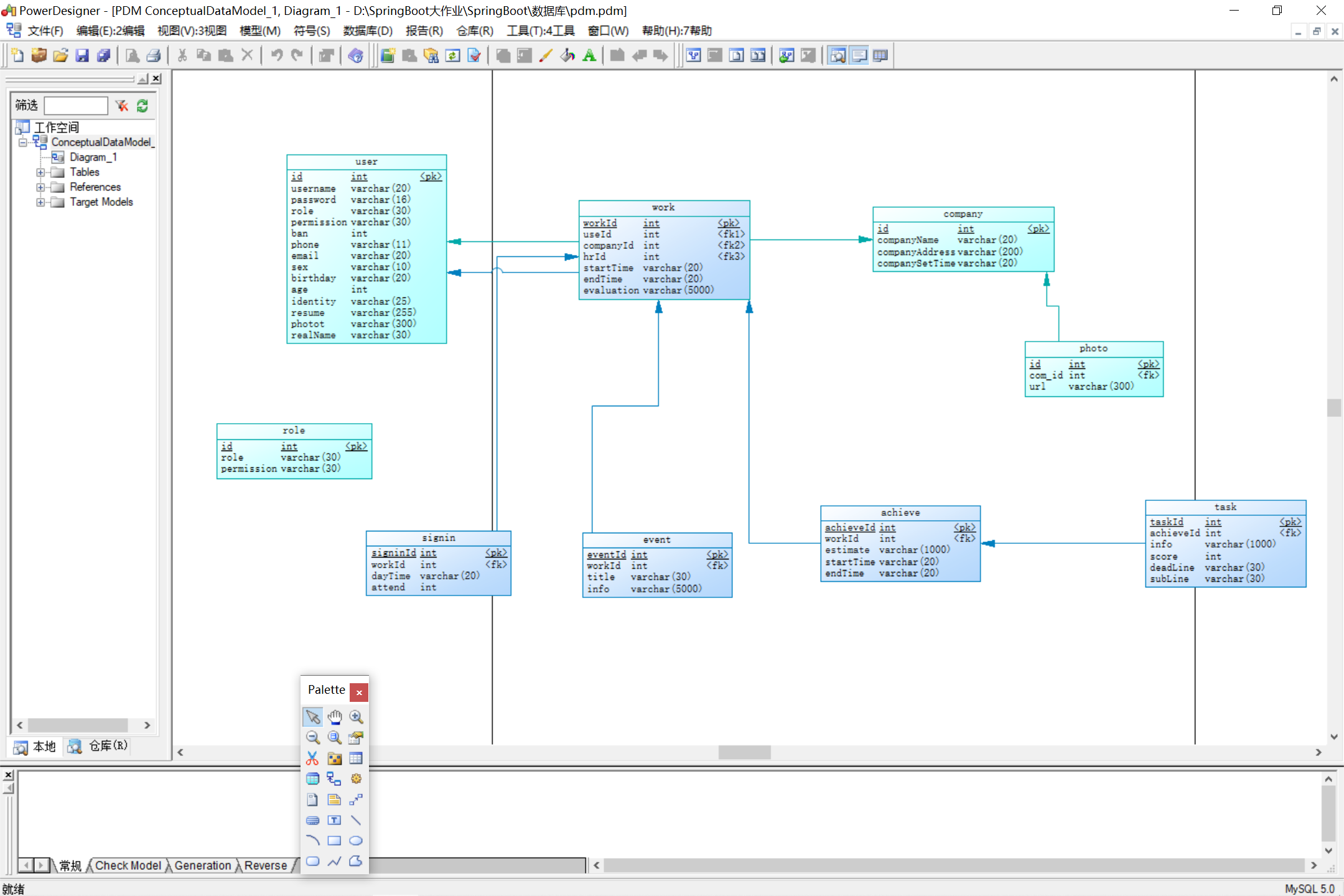
|  |  |
| --- | --- |
| 功能模块 | 控制方法 |
| 登录 | 验证用户名口令，系统核对后才能进行登录 |
| 修改信息 | 验证修改信息是否存在敏感词汇，核对后才能修改 |
| 重置密码 | 验证新密码与旧密码是否相同，系统核对后才能重置 |
| 注销 | 询问用户是否确认退出后，再运行下一步 |
| 签到 | 点击签到按钮完成当日打卡 |
| 同意HR求职、离职 | 保存操作记录，防止恶意操作 |
| 评价 | 验证信息中是否存在敏感词汇，核对后才能发表 |
| 发布任务 | 验证信息中是否存在敏感词汇，核对后才能发表 |

### 4.2.4管理员功能模块运行控制

|  |  |
| --- | --- |
| 功能模块 | 控制方法 |
| 登录 | 验证用户名口令，系统核对后才能进行登录 |
| 注销 | 询问用户是否确认退出后，再运行下一步 |
| 新增公司 | 点击新增公司按钮完成新增 |
| 新增公司主管 | 点击新增主管按钮完成新增主管 |
| 修改主管信息 | 输入新的主管信息并保存 |

# 5系统数据结构设计

## 5.1逻辑结构设计要点



1. user

Id，username，password，role，permission，ban，phone，email，sex，birthday，age，identity，resume，photo，companyid，realName

1. company

Id，companyName，companyAddress，companySetTime

1. Achieve

achieveId，workId，estimate，startTime，endTime，score，info

1. Apply

applyId，userId，companyId，subTime，info，acceptor，isAcceptor，applyType

1. Event

eventId，workId，title，info

1. Photo

Id，com\_id，url

1. Role

Id， role，permission

1. Signin

SigninId，workId，dayTime，attend

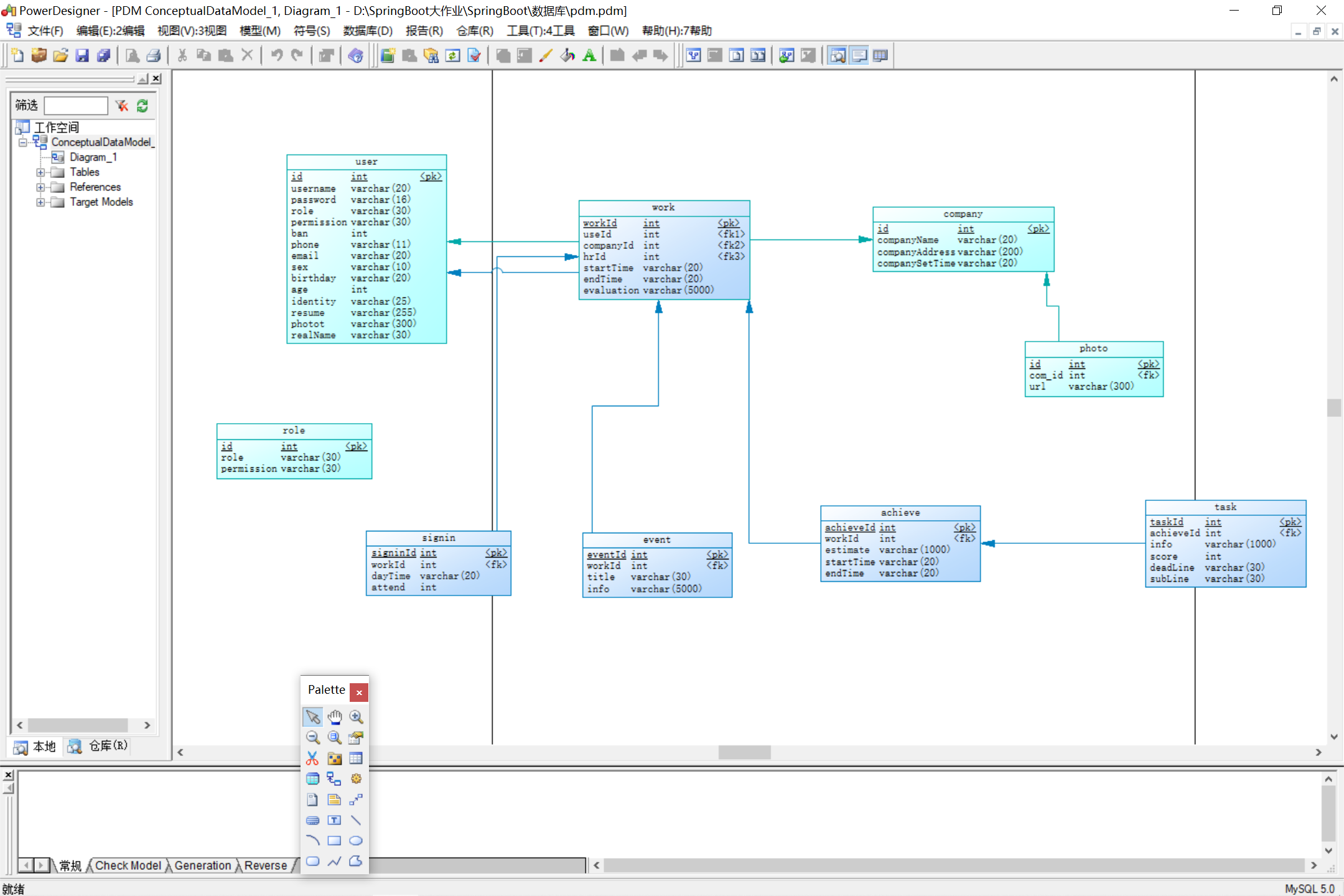
1. Task

taskId，achieveId，info，score，deadLine，subLine

1. Work

workId，useId，companyId，hrld，startTime，endTime，evaluation，isEnd

## 5.2物理结构设计要点



1. User

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | Id | int | 11 |  | 否 | 是 | 自动递增 |
| 2 | Password | vachar | 16 |  | 否 |  |  |
| 3 | Name | vachar | 20 |  | 否 |  |  |
| 4 | Role | vachar | 30 |  | 否 |  |  |
| 5 | Permission | vachar | 30 |  |  |  |  |
| 6 | Ban | Int | 11 |  |  |  |  |
| 7 | Phone | vachar | 11 |  |  |  |  |
| 8 | Email | vachar | 20 |  |  |  |  |
| 9 | Sex | vachar | 10 |  |  |  |  |
| 10 | Age | Int | 11 |  |  |  |  |
| 11 | Identity | vachar | 20 |  |  |  |  |
| 12 | Resume | vachar | 255 |  |  |  |  |
| 13 | Photo | vachar | 300 |  |  |  |  |
| 14 | companyId | vachar | 255 |  |  |  |  |
| 15 | realName | vachar | 255 |  |  |  |  |

1. Achieve

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | achieveId | int | 11 |  | 否 | 是 | 自动递增 |
| 2 | workId | Int | 11 |  | 否 |  |  |
| 3 | Estimate | vachar | 1000 |  | 否 |  |  |
| 4 | startTime | vachar | 30 |  | 否 |  |  |
| 5 | endTime | vachar | 30 |  |  |  |  |
| 6 | Score | Int | 7 |  |  |  |  |
| 7 | Info | vachar | 255 |  |  |  |  |

1. Apply

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | applyId | int | 7 |  | 否 | 是 | 自动递增 |
| 2 | UserId | Int | 7 |  | 否 |  |  |
| 3 | companyId | Int | 7 |  | 否 |  |  |
| 4 | SubTime | vachar | 30 |  | 否 |  |  |
| 5 | Info | vachar | 5000 |  |  |  |  |
| 6 | Acceptor | Int | 7 |  |  |  |  |
| 7 | isAccept | Int | 7 |  |  |  |  |
| 8 | ApplyType | Int | 7 |  |  |  |  |

1. Comapny

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | Id | int | 7 |  | 否 | 是 | 自动递增 |
| 2 | companyName | vachar | 20 |  | 否 |  |  |
| 3 | companyAddress | vachar | 200 |  | 否 |  |  |
| 4 | companySetTime | vachar | 20 |  | 否 |  |  |

1. Event

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | eventId | int | 7 |  | 否 | 是 | 自动递增 |
| 2 | workId | Int | 7 |  | 否 |  |  |
| 3 | Title | vachar | 200 |  | 否 |  |  |
| 4 | Info | vachar | 20 |  | 否 |  |  |

1. Photo

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | Id | int | 7 |  | 否 | 是 | 自动递增 |
| 2 | com\_id | Int | 7 |  | 否 |  |  |
| 3 | Url | vachar | 200 |  | 否 |  |  |

1. Role

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | Id | int | 7 |  | 否 | 是 | 自动递增 |
| 2 | Role | vachar | 30 |  | 否 |  |  |
| 3 | Permission | vachar | 30 |  | 否 |  |  |

1. Signin

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | signinId | int | 7 |  | 否 | 是 | 自动递增 |
| 2 | workId | Int | 7 |  | 否 |  |  |
| 3 | dayTime | vachar | 30 |  | 否 |  |  |
| 4 | Attend | Int | 11 |  |  |  |  |

1. Task

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | taskId | int | 7 |  | 否 | 是 | 自动递增 |
| 2 | achieveId | Int | 7 |  | 否 |  |  |
| 3 | info | vachar | 1000 |  | 否 |  |  |
| 4 | score | Int | 11 |  |  |  |  |
| 5 | deadLine | vachar | 30 |  |  |  |  |
| 6 | subLine | vachar | 30 |  |  |  |  |

1. Work

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | workId | int | 7 |  | 否 | 是 | 自动递增 |
| 2 | useId | Int | 7 |  | 否 |  |  |
| 3 | companyId | Int | 7 |  | 否 |  |  |
| 4 | hrId | Int | 7 |  |  |  |  |
| 5 | startTime | vachar | 30 |  |  |  |  |
| 6 | endTime | vachar | 30 |  |  |  |  |
| 7 | Evaluation | vachar | 1000 |  |  |  |  |
| 8 | isEnd | Int | 7 |  |  |  |  |

# 6.系统出错设计

## 6.1出错信息

|  |  |  |
| --- | --- | --- |
| **出错情况** | **出错提示** | **处理方法** |
| 输入信息无法满足要求 | “密码输入错误，请重新输入！”  “格式不符合要求，请重新输入!” | 须对每个模块的数据进行数据分析，判断错误类型，再生成相应的错误提示语句，送到输出模块中 |
| 网络传输超时 | “网络错误”  “请求失败” | 在出错的相应模块中输出简单的出错语句，并将程序重置，返回输入阶段 |

## 6.2补救措施

1. 所有服务器都必须安装不间断电源以防止停电或电压不稳造成的数据丢 失的损失。若真断电时，客户机上将不会有太大的影响，主要是服务器上：在断电后恢复过程可采用MYSQL的日志文件，对其进行ROLLBACK 处理，对数据进行恢复。

2. 在网络传输方面，可考虑建立一条成本较低的后备网络，以保证当主网 络断路时数据的通信。

3. 对于软错误，须对每个模块的数据进行数据分析，判断错误类型，再生 成相应的错误提示语句，送到输出模块中。

4. 对于硬错误，可在出错的相应模块中输出简单的出错语句，并将程序重 置。返回输入阶段。

5. 在硬件方面要选择较可靠、稳定的服务器机种，保证系统运行时的可靠 性。

6. 用户忘记密码时可以选择忘记密码，验证密保问题后可重置密码。

## 6.3系统维护设计

维护方面主要为对服务器上的数据库数据进行维护。可使用 MYSQL的数据库维护功能机制。例如，定期为数据库进行Backup，维护管理数据库死锁问题和维护数据库内数据的一致性等。

# 7.安全保密设计

**采用三层客户服务器数据访问结构**

在以往的开发设计过程中，我们发现对数据的访问可以采用许多第三方工具。为了防止一些别有用心的用户绕过有较好安全保护措施的客户端应用程序，直接访问数据库中的数据，我们采用了三层客户服务器数据访问结构。将软件功能分成表示层、功能层、数据层。中间功能层起到代理作用，用户对数据库的访问必须由表示层向中间功能层申请，中间功能层受理申请后，记录访问要求并核实用户对管理系统的访问权限，然后向数据层提出访问申请，这样用户的访问权限只针对管理信息系统，数据层不必对用户提供直接访问权限。

这种访问结构，可以把对数据层访问的识别名称和密码设得足够长、足够复杂，且随时变化，未授权的用户就难以绕过功能层而利用数据库工具或黑客手段非法访问数据层。三层客户服务器数据访问结构为严格的安全管理打下坚实的基础。