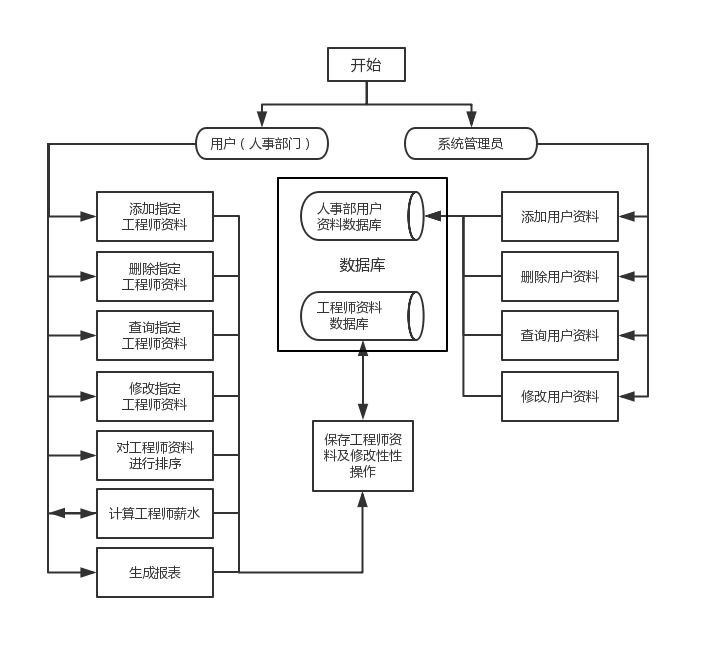
总体设计文档

1. 软件结构图



1. 模块功能介绍

2.1、登录模块

（1）选择相应的管理权限并输入账号和密码

（2）用户验证进去相应的个人主页，管理员则进入后台管理员模块

提供的接口：

请求数据：userName, code

返回数据：是否有次用户

2.2、管理员模块

2.2.1、输入工程师资料

工程师的资料主要包括编号、姓名、性别、生日、籍贯、学历、地址、电话、工龄、基本薪水。需要对这些输入的信息进行合法性检查。保证系统接收合法的输入。用户输入错误时具有提示功能和重新输入功能。

提供的接口：

请求数据：编号、姓名、性别、生日、籍贯、学历、地址、电话、工龄、基本薪水

返回数据：是否存储成功

2.2.2、删除指定工程师资料

可根据两种方式删除指定工程师资料，一是工程师的编号，二是工程师的姓名。删除资料后，该工程师的信息则完全在系统中删除了。其它工程师编号不变。

提供的接口：

请求数据：工程师的编号(id)或者姓名(userName)

返回数据：是否删除成功

调用的接口：查询工程师

2.2.3、修改指定工程师资料

可根据两种方式找到要修改的工程师，一是工程师的编号，二是工程师的姓名。打印原来该工程师的信息，同时提示用户进行修改。

提供的接口：

请求数据：编号(userID)或者姓名(userName)

返回数据：json格式的该工程师信息（后台返回），前台处理后，予以显示

调用的接口：查询工程师

2.3、用户模块

2.3.1、查询指定工程师资料

可根据两种方式查询指定工程师资料，一是工程师的编号，二是工程师的姓名。查询后打印该工程师的信息，如果没有该工程师资料则给用户提示。

提供的接口：

请求数据: 编号(userID)或者姓名(userName)

返回数据：json格式的该工程师信息（后台返回）或返回无该工程师的状态

2.3.2、计算工程师薪水

根据当月的月效益，计算工程师的当月工资。在计算机工资时要扣除当月的保险金。程师的月应发的薪水如下计算方法：

薪水＝（基本工资＋10╳月有效工作日天数＋月效益╳工作年限÷100）╳0.9－月保险金。

提供的接口：

请求数据：计算命令（字符串格式）

返回数据：计算结果（字符串格式）

调用的接口：查询工程师

2.3.3、保存工程师资料

当用户输入工程师资料、修改、排序后需要用户决定是否保存工程师资料。如果用户不保存，则默认输入和修改等不做保存。

提供的接口：

请求数据：工程师信息(编号、姓名、性别、生日、籍贯、学历、地址、电话、工龄、基本薪水)

返回数据：是否保存成功的状态

2.3.4、对工程师的资料进行排序

对工程师资料进行排序，排序使用三种方式：编号排序（升序）、姓名排序（升序）和工龄排序（降序）。采用哪种排序方式，由用户选择。

提供的接口：

请求数据：编号(userID)或者姓名（userName）或者年龄（userAge）

返回数据：json格式的经排序后的工程师信息（后台返回），前台处理后，予以显示

调用的接口：查询工程师

2.3.5、输出工程师资料

打印所有的工程师资料（不包括当月薪水）。

提供的接口：

请求数据：打印所有指令(以字符串方式传入)

返回数据：json格式的所有工程师信息（后台返回），前台处理后予以显示

调用的接口：查询工程师

2.3.6、清空工程师所有资料

把所有的工程师资料删除，可以重新输入工程师资料。

提供的接口：

请求数据：清除所有指令(以字符串方式传入)

返回数据：是否清除成功

2.3.7、退出系统

当用户不再使用该系统后，可退出该系统。如果用户进行了影响工程师资料信息的操作，提示用户是否进行保存。

提供的接口：

请求数据：用户信息

返回数据：成功退出

1. 数据库设计

#创建用户数据库

CREATE DATABASE db\_engineer #创建数据库，名字暂定

USE db\_engineer;

#用户数据库部分

#所有用户信息存在一张表中

DROP TABLE IF EXISTS engineers;

DROP TABLE IF EXISTS users;

CREATE TABLE users(

user\_id int primary key auto\_increment, #自动编号

user\_name varchar(20) not null, #姓名

user\_type tinyint not null, #账号类型,一般账号为0

sex tinyint(1) not null, #性别，只能为0或1，未做检查

username varchar(20) not null, #用户名

password varchar(20) not null #密码

);

#简单的语句

INSERT INTO users (user\_name, user\_type, sex, username, password)

VALUES (“小明“,1,1,”这是用户名”,”12345678901234567890”);

INSERT INTO users (user\_name, user\_type, sex, username, password)

VALUES (“小白“,1, 0,”这是用户名”,”12345678901234567890”);

INSERT INTO users (user\_name, user\_type, sex, username, password)

VALUES (“小黑“,1,1,”这是用户名”,”12345678901234567890”);

SELECT \* FROM users;

#用户数据库部分

DROP TABLE IF EXISTS engineers;

CREATE TABLE engineers (

engineer\_id int primary key auto\_increment, #自动编号

engineer\_name varchar(20) not null, #姓名

sex tinyint(1) not null, #性别，只能为0或1

birthday date not null, #生日，格式为年月日

native\_place varchar(10) not null, #籍贯，长度为10

education tinyint(1) not null, #学历

address varchar(30) not null, #地址，长度为30

mobile\_phone varchar(15) not null, #手机号码，长度为15

working\_years tinyint not null, #工龄

salary float not null, #基本薪水

belong\_to int not null, #该字段决定此记录是归属于哪个用户

foreign key (belong\_to) references users(user\_id) #

);

#简单的语句

Insert into engineers (engineer\_name, sex, birthday, native\_place, education, address, mobile\_phone, working\_years, salary, belong\_to)VALUES

(“李明”,0,”1999-01-01”,”浙江”,

1, ”这是地址”,”15566778899”,15,4444.44,1);

select \* from engineers where belong\_to = 1; #查询1号用户所创建的数据表

1. 实现计划

4.1、人员任务分配

需求分析文档：李依茹

总体设计文档：陈厚载、周建霖、顾博盛、陈荣、李依茹

数据库设计：顾博盛

后台设计：陈荣

前台设计：陈厚载、周建霖、顾博盛、陈荣、李依茹

软件测试：陈厚载、周建霖、顾博盛、陈荣、李依茹

4.2时间进度安排

第1周：分组

第2周：在github.com上建立仓库

第3-5周：设计并完善需求分析文档

第6-7周：设计并完善总体设计文档，设计数据库及软件后台

第7-12周：完成软件功能及页面，并完善所有内容

第13-14周：软件测试