

ACME工资管理系统开发人员工作总结

53140206 吴志伟

[ACME工资管理系统开发人员工作总结](#)

[项目概述](#)

[个人工作内容](#)

[员工打卡功能的构建](#)

[委托员工订单功能](#)

项目概述

ACME需要一款新的工资管理系统，按照工时和销售额自动生成。

系统外观将会设计成windows风格，允许员工增/改以下内容：

操作	内容
输入	打卡信息
输入	付款订单
修改	员工偏好（例如开支途径）
创建	多种报告

ACME不想更换现有的 **项目管理数据库**，该数据库运行在IBM小型机上，包含了全部与项目和金额相关的信息。薪资系统将可以访问 **项目管理数据库**，但不能修改其中的内容。并且薪资系统必须及时将准确的金额，按照员工指定的收款方式发放给员工。

时薪制员工每周五发工资，他们的打卡记录上会显示每天的工时。超过每天8小时的工作时间将有50%的加班费。

固定工资制员工和前者一样打卡，但他们将会在每月最后一个工作日开支。

临时工 ~~固定工资+提成~~，不同的工资由其销售额决定，他们打卡时将会上报日期和当天的销售额。每个员工的提成比例不同，例如10%,25%或35%

该系统最重要的功能将会是员工工作报告，员工将能查询以下信息：

查询的信息
总工作时长
员工为指定项目的贡献的工时，亦即： 收费号
年初至今收到的总工资
剩余假期

等等。

员工可选的开支途径有：

开支途径：
打到员工指定的银行账号
邮到员工指定的地址
在办公室现场开支

薪资系统的管理员负责维护员工信息，他需要

操作	内容
添加	新员工
删除	员工
修改	员工信息：姓名、薪酬制度（时薪/固定工资/提成）
生成	管理报告

系统将会在每周五以及每月的最后一个工作日自动运行，由于系统存有每名员工的开支日期，所以它将会从每名员工两次开支之间的记录中计算工资，并将其发给员工。系统将全自动运行，不再需要任何人工操作。

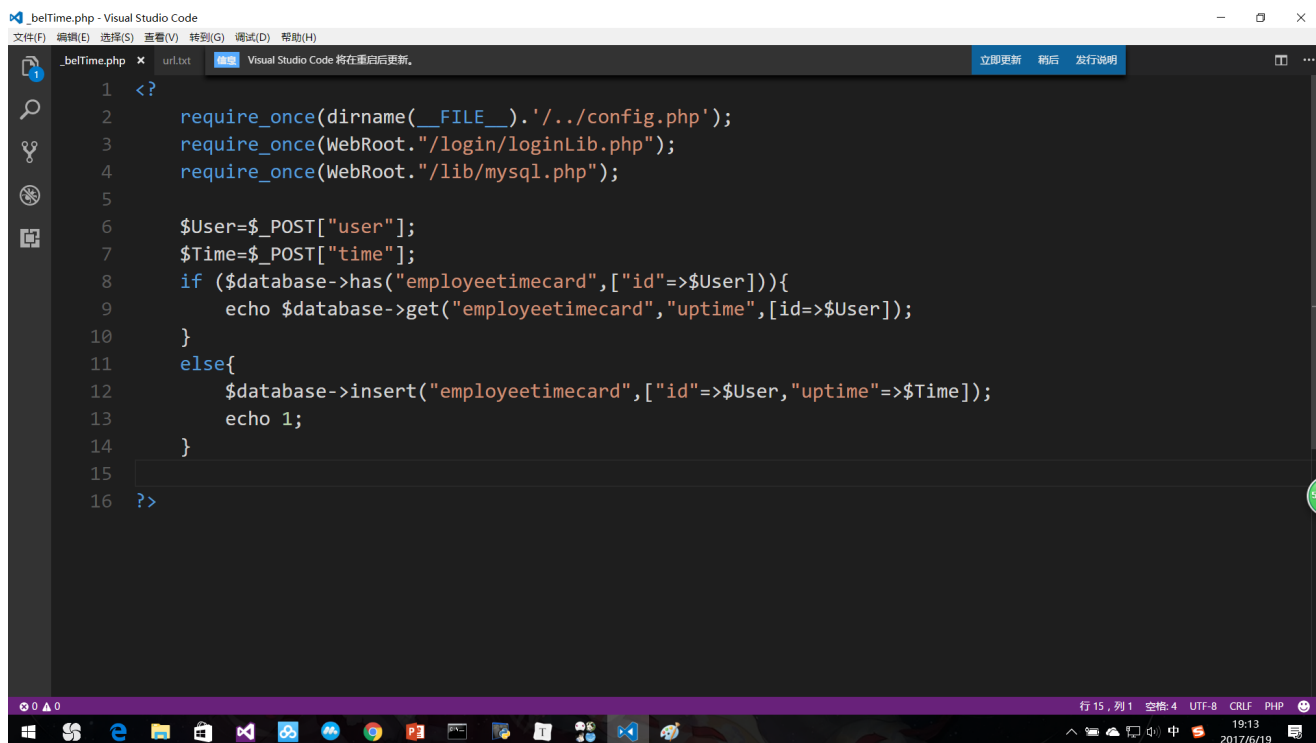
个人工作内容

在上述的项目中我主要负责开发员工打卡功能和委托员工订单功能。我们组共同使用了Symfony框架结构，结合html完成了一个能实现上述目的的网页。

员工打卡功能的构建

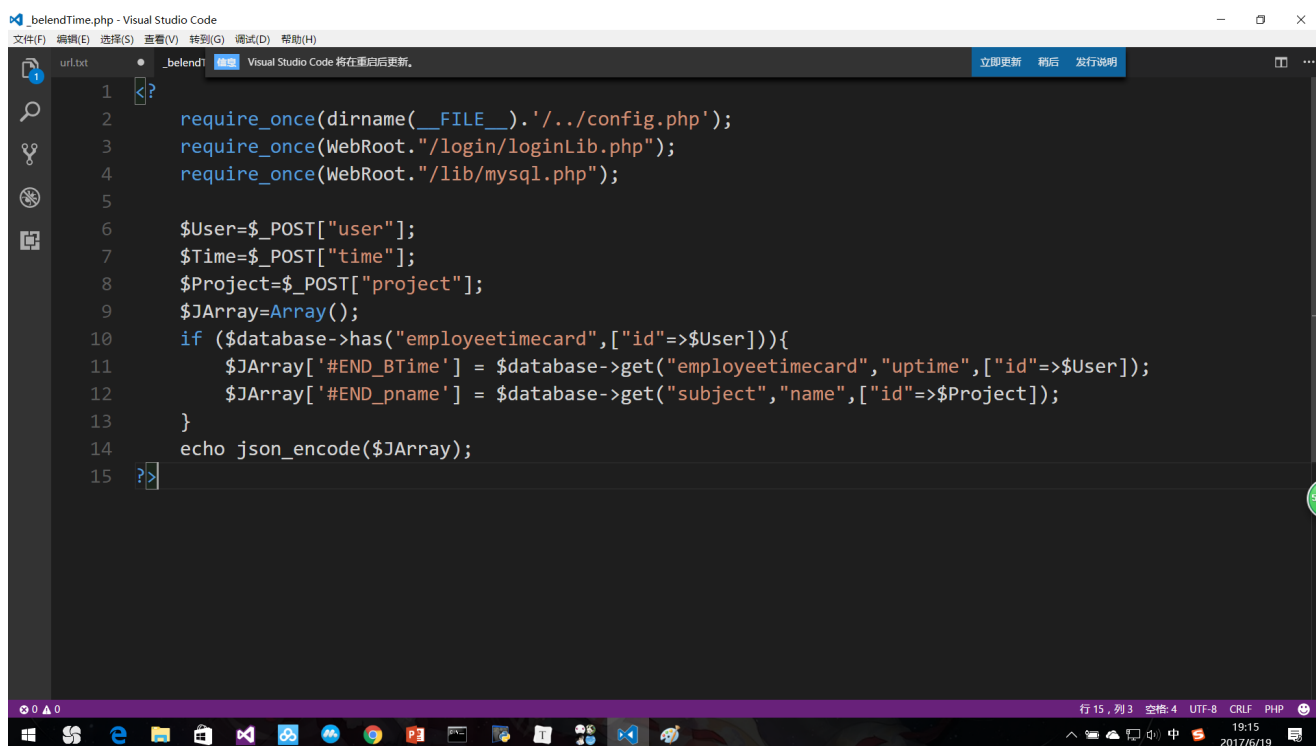
在构建员工打卡功能上，我将整个打卡过程分为3部分：开始打卡，选择项目，结束打卡。其中选择项目一项可以多次选择并以最后一次（选择结束打卡时间之前）项目为最终结果，并记录在时间表数据库上。因此在完成这项工作的时候，我们有许多想法，主要关于的就是项目的选择上，由于项目的可能出现的反复选择，如果将每一次选择都并入总数据库，会产生很多不必要的浪费。因此最终我们选择了新开一个临时数据库用于存储员工选择的项目，并在结束打卡后将其对应项删除，这样一来就大大节省了查询大数据所用的时间。

下面是开始打卡的代码实现：通过向临时数据库插入开始打卡的记录：



```
1 <?
2 require_once(dirname(__FILE__).'../config.php');
3 require_once(WebRoot."/login/loginLib.php");
4 require_once(WebRoot."/lib/mysql.php");
5
6 $User=$_POST["user"];
7 $Time=$_POST["time"];
8 if ($database->has("employeetimecard",["id"=>$User])){
9     echo $database->get("employeetimecard","uptime",["id"=>$User]);
10 }
11 else{
12     $database->insert("employeetimecard",["id"=>$User,"uptime"=>$Time]);
13     echo 1;
14 }
15
16 ?>
```

以下是选择项目的数据库部分操作，将项目放入同一项中，由于临时数据库一般不大所以便于随时修改项目：



```
1 <?
2 require_once(dirname(__FILE__).'../config.php');
3 require_once(WebRoot."/login/loginLib.php");
4 require_once(WebRoot."/lib/mysql.php");
5
6 $User=$_POST["user"];
7 $Time=$_POST["time"];
8 $Project=$_POST["project"];
9 $JArray=array();
10 if ($database->has("employeetimecard",["id"=>$User])){
11     $JArray['#END_BTime'] = $database->get("employeetimecard","uptime",["id"=>$User]);
12     $JArray['#END_pname'] = $database->get("subject","name",["id"=>$Project]);
13 }
14 echo json_encode($JArray);
15 ?>
```

以下是最终提交打卡数据的数据库部分，在这个部分我们将临时数据库中的那一项删除，而后将打卡记录添加到最终打卡记录数据库中，并计算总工作时间分别添加到项目管理表和工资管理表中。

```
.belfinalTime.php - Visual Studio Code
文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 调试(D) 帮助(H)

.belfinalTime.php x
1  <?
2  require_once(dirname(__FILE__).'../config.php');
3  require_once(WebRoot."/login/loginLib.php");
4  require_once(WebRoot."/lib/mysql.php");
5  require_once(dirname(__FILE__).'../recordpay.php');
6  $User=$_POST["user"];
7  $BTime=$_POST["begintime"];
8  $Time=$_POST["endtime"];
9  $Name=$_POST["name"];
10 $Project=$_POST["project"];
11 if ($database->has("employeetimecard",["id"=>$User]))
12 {
13     $database->delete("employeetimecard",["id"=>$User]);
14     $finaltime = (int)((strtotime($Time) - strtotime($BTime))/3600);
15     if ($database->has("grade",["AND"=>["user"=>$User,"subject"=>$Project]]))
16     {
17         $score = $database->get("grade","score",["user"=>$User]);
18         $finaltime = $score + $finaltime;
19         $database->update("grade",["score"=>$finaltime],["user"=>$User]);
20     }
21     else
22     {
23         $database->insert("grade",["user"=>$User,"subject"=>$Project,"score"=>$finaltime]);
24     }
25 }
```

```
.belfinalTime.php - Visual Studio Code
文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 调试(D) 帮助(H)

.belfinalTime.php x
19     $database->update("grade",["score"=>$finaltime],["user"=>$User]);
20 }
21 else
22 {
23     $database->insert("grade",["user"=>$User,"subject"=>$Project,"score"=>$finaltime]);
24 }
25 if ($finaltime > 8)
26 {
27     $normaltime = 8;
28     $extratime = $finaltime - 8;
29 }
30 else
31 {
32     $normaltime = $finaltime;
33     $extratime = 0;
34 }
35 updateTimeard($database,$User,$normaltime,$extratime);
36 $datetime = new DateTime($BTime);
37 $ID = $datetime->format('Y-m-d H:i:s').$User;
38 echo $database->insert("AllTimecard",["id"=>$ID,"user"=>$User,"name"=>$Name,"uptime"=>$BTime,"dow
39 }
40 }
```

委托员工订单功能

对于订单功能我的做法是在添加订单时根据订单生产的当前时间和员工ID唯一定义一个订单号，便于员工查询，并且当订单状态被修改为Close时将计算订单信息中的收益，添加到该员工的工资数据库中。

以下为建立订单时对数据库的操作：

```
_employeepurchase.php - Visual Studio Code
文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 调试(D) 帮助(H)

_employeepurchase.php x
1 <?
2 $Name=$_POST["name"];
3 $telephone=$_POST["telephone"];
4 $address=$_POST["address"];
5 $date=$_POST["date"];
6 $pid=$_POST["pid"];
7 $user=$_POST["user"];
8 $datetime=$_POST["datetime"];
9 require_once(dirname(__FILE__).'../config.php');
10 require_once(WebRoot."/login/loginLib.php");
11 require_once(WebRoot."/lib/mysql.php");
12 $ID = $user.$datetime;
13 if ($database->has("product",["id"=>$pid]))
14 {
15     echo $database->insert("purchase_order",["client_name"=>$Name,
16         "telephone"=>$telephone,
17         "address"=>$address,
18         "date"=>$date,
19         "pid"=>$pid,
20         "oid"=>$ID,
21         "user"=>$user,
22         "state"=>"open"]);
23 }
```

以下为通过订单号寻找订单信息的数据库交互，通过JSON将数据从php传输到JavaScript中，并且当订单状态为Close或者不是自己的订单时无法查询到订单：

```
_editpurchase.php - Visual Studio Code
文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 调试(D) 帮助(H)

_editpurchase.php x Cannot validate since no PHP executable is set. Use the setting 'php.validate.executablePath' to configure the PHP executable. 关闭
3 require_once(WebRoot."/login/loginLib.php");
4 require_once(WebRoot."/lib/mysql.php");
5 $User=$_POST["user"];
6 $Purchase_id=$_POST["purchase_id"];
7 $JArray=Array();
8 if ($database->has("purchase_order",["AND"=>["oid"=>$Purchase_id,"user"=>$User,"state"=>"open"]]))
9 {
10     $_purchase=$database->select("purchase_order","*",["oid"=>$Purchase_id]);
11     foreach($_purchase as $purchase)
12     {
13         $JArray['#PClient_name'] = $purchase["client_name"];
14         $JArray['#PTelephone'] = $purchase["telephone"];
15         $JArray['#PAddress'] = $purchase["address"];
16         $JArray['#PPid'] = $purchase["pid"];
17         $JArray['#PDate'] = $purchase["date"];
18         $JArray['#PUser'] = $purchase["user"];
19         $JArray['#POid'] = $purchase["oid"];
20         $JArray['#PState'] = $purchase["state"];
21     }
22     echo json_encode($JArray);
23 }
24 }
25 else{
```

以下为修改订单时与数据库的交互，当订单状态为Close时，调用员工工资数据库，增加员工在该订单的获利：

```
changepurchase.php - Visual Studio Code
文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 调试(D) 帮助(H)

_changepurchase.php x
7  $Name=$_POST["client_name"];
8  $telephone=$_POST["telephone"];
9  $address=$_POST["address"];
10 $date=$_POST["date"];
11 $pid=$_POST["pid"];
12 $state=$_POST["state"];
13 $oid=$_POST["oid"];
14 if ($database->has("purchase_order",["oid"=>$oid])){
15     $user = $database->get("purchase_order","user",["oid"=>$oid]);
16     if ($state == "close"){
17         $cost = $database->get("product","product_cost",["id"=>$pid]);
18         updateSales($database,$user,$cost);
19     }
20     echo $database->update("purchase_order",["client_name"=>$Name,
21         "telephone"=>$telephone,
22         "address"=>$address,
23         "pid"=>$pid,
24         "date"=>$date,
25         "state"=>$state],["oid"=>$oid]);
26 }
27 else{
28     echo -1;
29 }
```

行 15, 列 71 空格 4 UTF-8 CRLF PHP 19:26 2017/6/19