考勤管理系统

软件设计规格说明书

日期：2018-12-15

目录

1. 引言..................................................1
   1. 编写目的....................................................1
   2. 读者对象....................................................1
   3. 定义........................................................1
   4. 参考资料....................................................1
2. 软件设计约束..........................................1

2.1 设计目标和原则..............................................1

2.2 设计约束....................................................1

1. 软件设计描述..........................................2

3.1 结构设计....................................................2

3.2 数据设计....................................................3

3.3 详细设计....................................................4

1. 引言

1.1 编写目的

本手册作为用户与系统开发维护人员共同遵守的软件需求规范说明。

1.2 读者对象

用户代表和软件开发维护人员

1.3 定义

基于Angular的考勤系统开发

1.4 参考资料

• 《UML大战需求分析》

• 《基于Angular的Web开发》

1. 软件设计约束

2.1 设计目标和原则

2.1.1 设计目标

• 规范员工的上下班、外出工作的打卡行为。

• 方便员工与领导间的请假审批流程。

• 方便进行考勤时间的统计，包括正常班、休假、带薪假期的管理。

• 方便根据考勤时间表计算员工的薪资。

• 实现员工对自己信息的查询，包括剩余假期额度，打卡信息，外出工作信息等。当员工对自己的信息有异议时，可以向行政部发起申诉。

• 实现员工信息的共享，高层领导有手下员工的假期信息的查询与管理审批权，打卡信息和外出工作信息的查询权，普通员工享有员工的请假和外出工作信息查询权。

2.1.2 设计原则

• 系统符合实际生活需求

• 用户界面友好，操作简便

• 具有良好的性能

2.2 设计约束

• 操作系统：Windows 10

• 浏览器： IE5 以上浏览器

• 开发语言：Angular 2, HTML, CSS, typeScript

• 开发环境：node.js, npm, Angular Cli, Visual Studio Code

• 容量要求：能够支持100台客户端PC的访问

• 性能要求：

- 实用性：本系统应能够系统、便捷、快速地实现对员工工作时间的管理，从而保障企业的高效运行。

- 操作简单：本系统应适用于不同计算机水平的使用者，系统的操作尽可能简单易行。

- 技术先进：产品的系统设计和开发应紧跟计算机的发展潮流，采用目前较先进的设计思想，利用最先进的开发技术和开发工具。

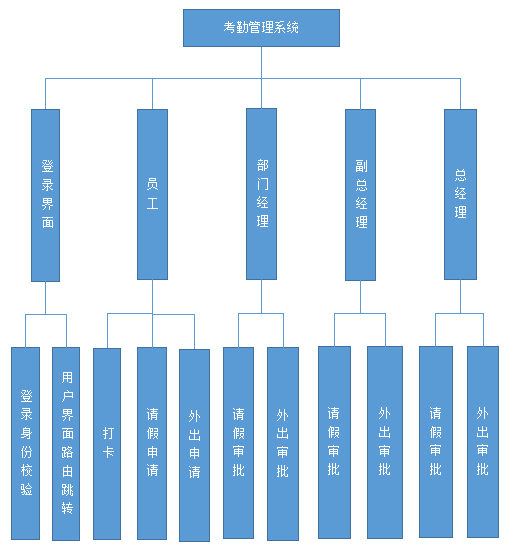
- 安装使用简便：服务器端安装简洁明了，客户机无需再装任何软件，可直接使用。

- 适应性和可扩充性：应能适应公司的具体环境，开发过程中应充分考虑可扩充性问题。

1. 软件设计描述

3.1 总体结构设计

3.1.1系统结构



3.1.2 软件架构

本系统有三个服务器：Web服务器，邮件服务器，数据库服务器。

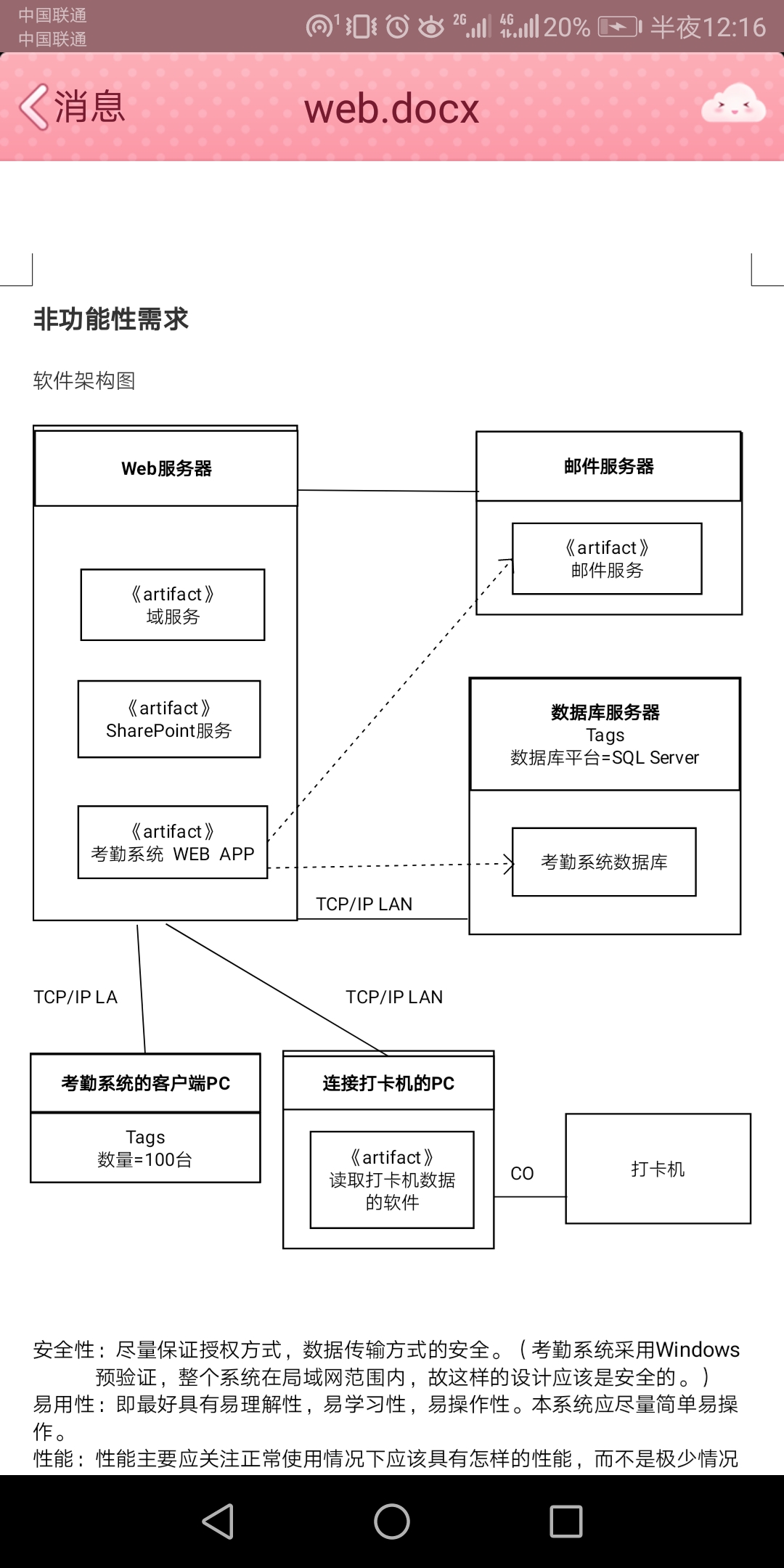
涉及的构件有：考勤系统客户端PC，连接打卡机的PC，打卡机。

其中

Web服务器需要：域服务，SharePoint服务，考勤系统WEB APP。

邮件服务器需要：邮件服务。

数据库服务器需要：考勤数据库系统。



3.2 数据设计

静态数据：员工信息、部门信息

动态数据：考勤信息、请假信息、出差信息

表3.2-1 用户信息表

|  |  |
| --- | --- |
| 字段 | 解释 |
| userID | 用户编号 |
| userName | 用户名 |
| password | 用户密码 |

表3.2-2 员工信息表

|  |  |
| --- | --- |
| 字段 | 解释 |
| userID | 员工编号 |
| userName | 员工姓名 |

表3.2-3 打卡信息表

|  |  |
| --- | --- |
| 字段 | 解释 |
| userID | 员工编号 |
| userName | 员工姓名 |
| clockDate | 打卡时间 |
| type | 打卡类型 |

表3.2-4 假期申请表

|  |  |
| --- | --- |
| 字段 | 解释 |
| userID | 员工编号 |
| leaveCategoryID | 请假类型 |
| leaveTime | 请假时间 |
| leaveReason | 请假原因 |
| askTime | 申请时间 |

表3.2-5 外出申请表

|  |  |
| --- | --- |
| 字段 | 解释 |
| userID | 员工编号 |
| outTime | 外出时间 |
| outReason | 外出原因 |
| askTime | 申请时间 |

3.3 详细设计

3.3.1 登录界面设计

设计界面，通过表单获取用户输入的id和password，通过服务获取json中的员工信息，两者比对，一致则传入员工id并跳转到相应的界面，其中id在1-10中为总经理，11-100中为副总经理，101-1000为部门经理，1000+为员工，在员工界面设立路由守卫，如果未登录则强制跳转登录界面。

3.3.2 员工部分设计

3.3.2.1员工部分申请请假和外出设计：

新建hero4组件

通过在service内定义变量，在不同组件之间通过变量可以实现数据的共享，在html页面使用ngmodel实现输入框内数据的变化，通过按钮的点击实现两个页面进行数据的存储和分析：

请假申请：

//html内：

<p style = "text-decoration:line-through">请假申请：</p>

<input type="text" class="a-upload" [(ngModel)]="savedata\_qingjia" placeholder="请假时间，申请请假时间，请假原因" >

<button (click)="saveDate\_qingjia()" >提交</button>

//ts内：

saveDate\_qingia(){

this.heroService.setInputValue\_qingjia(this.savedata\_qingjia);

}

外出申请：

//html内：

<p style = "text-decoration:line-through">外出申请：</p>

<input type="text" class="a-upload" [(ngModel)]="savedata\_waichu" placeholder="外出时间，申请外出时间，外出原因" >

<button (click)="saveDate\_waichu()" >提交</button>

//ts内：

saveDate\_waichu(){

this.heroService.setInputValue\_waichu(this.savedata\_waichu);

}

3.3.2.2 员工部分的获取打卡信息和查看请假是否批准设计：

首先通过服务，获取本地json数据的api，在通过url获取数据将数据保存到数组内，在html内建立button，通过点击button获取信息；在service内编写函数获取url内数据并存入数组，在ts内编写函数将信息传到html数据里，在html内显示小窗口。

查看打卡信息和请假是否批准信息和经理设计的查看请假反馈相似，具体的html部分是这样的：

<button (click)="getClockIn()">今日是否打卡</button>

<button (click)="getleaveResult()">请假是否批准</button>

<button (click)="getoutResult()">外出是否批准</button>

3.3.3 经理部分设计

3.3.3.1经理部分的获取反馈信息和反馈信息设计：

通过在service内定义变量，在不同组件之间通过变量可以实现数据的共享，在html页面使用ngmodel实现输入框内数据的变化，通过按钮的点击实现两个页面进行数据的分析和获取：

反馈信息：

//html内：

<div class = "fankui">

<p style = "text-decoration:line-through">反馈信息：</p>

<input type="text" class = "a-upload" [(ngModel)]="savestaff" >

<button (click)="saveStaffDate()" >反馈给员工</button>

</div>

//ts内：

saveStaffDate() {

this.heroService.setInputStaff(this.savestaff);

}

获取反馈信息：

//html内：

<div class = "fankui">

<p style = "text-decoration:line-through">获取反馈信息：</p>

<input type="text" class = "a-upload" [(ngModel)]="getdata" >

<button (click)="getDate()" >获取副经理反馈</button>

</div>

//ts内：

getDate() {

this.getdata = this.heroService.getInputValue1();

}

3.3.3.2 经理部分的获取请假信息和外出信息设计：

首先通过服务，获取本地json数据的api，在通过url获取数据将数据保存到数组内，在html内建立一个表格，建立button，通过点击按钮将数据显示在表内；在service内编写函数获取url内数据并存入数组，在ts内编写函数通过按钮将数组数据放入表格，在html内将表格显示

外出和请假获取信息相似，下面代码举一个例子：

//html内:

<button (click)="getOutDate()" >显示外出申请</button>

<div class="user-item-container">

<table class ="table table-bordered">

<thead>//内容省略</thead>

<tbody>//内容省略</tbody>

</table>

</div>

//ts内

getOutDate(){

this.outdata = this.heroService. getOutThree()

.then(result =>{

console.log(result);

this.out = result;

});

}

//service内：

getOutThree():Promise<Hero[]>{

const url = `${this.api\_url +'/lastOutAnswer'}`;

return this.http.get(url, {headers: this.headers})

.toPromise()

.then(res => res.json() as Hero[])

.catch(this.handleError);

}

3.3.3.3 经理部分的进行批准/不批准申请活动的设计：

在3.3.3.2内已经将数据可以获取并显示，但当经理进行同意或拒绝后，可以将数据进行删除；同样建立一个button，并通过按钮可以将数据删除，并弹出警告框提示操作者将信息进行反馈：

在ts内编写函数可将数组数据删除，在html内编写将button添加删除函数：

//在html内：

<button (click)="hintout()">批准外出</button>

<button (click)="deleteOutDate()">不批准外出</button>

//在ts内：

deleteOutDate(){

alert("反馈信息给员工");

this.out = [];

}

hintout(){

alert("反馈信息给员工");

this.out =[];

}